massachusetts institute of technology — artificial intelligence laboratory

A Biological Model of Object Recognition with Feature Learning

Jennifer Louie

AI Technical Report 2003-009 June 2003
CBCL Memo 227

A Biological Model of Object Recognition with Feature Learning

by

Jennifer Louie

Submitted to the Department of Electrical Engineering and Computer Science in partial fulfillment of the requirements for the degree of

Master of Engineering in Computer Science and Engineering

at the

MASSACHUSETTS INSTITUTE OF TECHNOLOGY

May 2003

Certified by: Tomaso Poggio
Eugene McDermott Professor
Thesis Supervisor

Accepted by: Arthur C. Smith
Chairman, Department Committee on Graduate Students

A Biological Model of Object Recognition with Feature Learning

by

Jennifer Louie

Submitted to the Department of Electrical Engineering and Computer Science on May 21, 2003, in partial fulfillment of the requirements for the degree of Master of Engineering in Computer Science and Engineering

Abstract

Previous biological models of object recognition in cortex have been evaluated using idealized scenes and have hard-coded features, such as the HMAX model by Riesenhuber and Poggio [10]. Because HMAX uses the same set of features for all object classes, it does not perform well in the task of detecting a target object in clutter. This thesis presents a new model that integrates learning of object-specific features with the HMAX. The new model performs better than the standard HMAX and comparably to a computer vision system on face detection. Results from experimenting with unsupervised learning of features and the use of a biologically-plausible classifier are presented.

Thesis Supervisor: Tomaso Poggio
Title: Eugene McDermott Professor

Acknowledgments

I'd like to thank Max for his guidance and words of wisdom, Thomas for his infusion of idea and patience, and Tommy for being my thesis supervisor. To my fellow MEngers (Amy, Ed, Rob, and Ezra), thanks for the support and keeping tabs on me. Lastly, to my family for always being there.

This research was sponsored by grants from: Office of Naval Research (DARPA) Contract No. N00014-00-1-0907, Office of Naval Research (DARPA) Contract No. N00014-02-1-0915, National Science Foundation (ITR/IM) Contract No. IIS-0085836, National Science Foundation (ITR/SYS) Contract No. IIS-0112991, National Science Foundation (ITR) Contract No. IIS-0209289, National Science Foundation-NIH (CRCNS) Contract No. EIA-0218693, and National Science Foundation-NIH (CRCNS) Contract No. EIA-0218506.

Additional support was provided by: AT&T, Central Research Institute of Electric Power Industry, Center for e-Business (MIT), DaimlerChrysler AG, Compaq/Digital Equipment Corporation, Eastman Kodak Company, Honda R&D Co., Ltd., ITRI, Komatsu Ltd., The Eugene McDermott Foundation, Merrill-Lynch, Mitsubishi Corporation, NEC Fund, Nippon Telegraph & Telephone, Oxygen, Siemens Corporate Research, Inc., Sony MOU, Sumitomo Metal Industries, Toyota Motor Corporation, and WatchVision Co., Ltd.

Contents

1 Introduction **11**
- 1.1 Related Work . . . 12
  - 1.1.1 Computer Vision . . . 12
  - 1.1.2 Biological Vision . . . 13
- 1.2 Motivation . . . 16
- 1.3 Roadmap . . . 17

2 Basic Face Detection **19**
- 2.1 Face Detection Task . . . 19
- 2.2 Methods . . . 20
  - 2.2.1 Feature Learning . . . 20
  - 2.2.2 Classification . . . 23
- 2.3 Results . . . 23
  - 2.3.1 Comparison to Standard HMAX and Machine Vision System . . . 23
  - 2.3.2 Parameter Dependence . . . 24

3 Invariance in HMAX with Feature Learning **28**
- 3.1 Scale Invariance . . . 28
- 3.2 Translation Invariance . . . 32

4 Exploring Features **34**
- 4.1 Different Feature Sets . . . 34
- 4.2 Feature Selection . . . 41
- 4.3 Conclusions . . . 44

5 Biologically Plausible Classifier **46**
- 5.1 Methods . . . 46
- 5.2 Results . . . 47
  - 5.2.1 Face Prototype Number Dependence . . . 47
  - 5.2.2 Using Face Prototypes on Previous Experiments 51

5.3 Conclusions 55

6 Discussion **56**

List of Figures

1.1 The HMAX model. The first layer, S1, consists of filters tuned to different areas of the visual field, orientations (oriented bars at 0, 45, 90, and 135 degrees) and scales. These filters are analogous to the simple cell receptive fields found in the V1 area of the brain. The C1 layer responses are obtained by performing a max pooling operations over S1 filters that are tuned to the same orientation, but different scales and positions over some neighborhood. In the S2 layer, the simple features from the C1 layer (the 4 bar orientations) are combined into 2 by 2 arrangements to form 256 intermediate feature detectors. Each C2 layer unit takes the max over all S2 units differing in position and scale for a specific feature and feeds its output into the view-tuned units. In our new model, we replace the hard-coded 256 intermediate features at the S2 level with features the system learns. 15

2.1 Typical stimuli used in our experiments. From left to right: Training faces and non-faces, "cluttered (test) faces", "difficult (test) faces" and test non-faces. 20

2.2 Typical stimuli and associated responses of the C1 complex cells (4 orientations). Top: Sample synthetic face , cluttered face, real face, non-faces. Bottom: The corresponding C1 activations to those images. Each of the four subfigures in the C1 activation figures maps to the four bar orientations (clockwise from top left: 0, 45, 135, 90 degrees). For simplicity, only the response at one scale is displayed. Note that an individual C1 cell is not particularly selective either to face or to non-face stimuli. 21

2.3 Sketch of the HMAX model with feature learning: Patterns on the model "retina" are first filtered through a continuous layer S1 (simplified on the sketch) of overlapping simple cell-like receptive fields (first derivative of gaussians) at different scales and orientations. Neighboring S1 cells in turn are pooled by C1 cells through a MAX operation. The next S2 layer contains the RBF-like units that are tuned to object-parts and compute a function of the distance between the input units and the stored prototypes ($p = 4$ in the example). On top of the system, C2 cells perform a MAX operation over the whole visual field and provide the final encoding of the stimulus, constituting the input to the classifier. The difference to standard HMAX lies in the connectivity from C1→S2 layer: While in standard HMAX, these connections are hardwired to produce 256 2×2 combinations of C1 inputs, they are now learned from the data. (Figure adapted from [12]) 22
2.4 Comparison between the new model using object-specific learned features and the standard HMAX by test set. For synthetic and cluttered face test sets, the best set of features had parameters:$p = 5$, $n = 480$, $m = 120$. For real face test set, the best set of features were $p = 2$, $n = 500$, $m = 125$. The new model generalizes well on all sets and outperforms standard HMAX. 25
2.5 Average C2 activation of synthetic test face and test non-face set. Left: using standard HMAX features. Right: using features learning from synthetic faces. 26
2.6 Performance (ROC area) of features learned from synthetic faces with respect to number of learned features n and p (fixed $m = 100$). Performance increases with the number of learned features to a certain level and levels off. Top left: system performance on synthetic test set. Top right: system performance on cluttered test set. Bottom: performance on real test set. 26
2.7 Performance (ROC area) with respect to % face area covered and p. Intermediate size features performed best on synthetic and cluttered sets, small features performed best on real faces. Top left: system performance on synthetic test set. Top right: system performance on cluttered test set. Bottom : performance on real test set. 27

3.1 C1 activations of face and non-face at different scale bands. Top (from left to right): Sample synthetic face, C1 activation of face at band 1, band 2, band 3, and band 4. Bottom: Sample non-faces, C1 activation of non-face at band 1, band 2, band 3, and band 4. Each of the four subfigures in the C1 activation figures maps to the four bar orientations (clockwise from top left: 0, 45, 135, 90 degrees). 29
3.2 Example images of rescaled faces. From left to right: training scale, test face rescaled -0.4 octave, test face rescaled +0.4 octave 29
3.3 ROC area vs. log of rescale factor. Trained on synthetic faces, tested on 900 rescaled synthetic test faces. Images size is 100x100 pixels 30
3.4 Average C2 activation vs. log of rescale factor. Trained on synthetic faces, tested on 900 rescaled synthetic test faces. Image size is 200x200 pixels 31
3.5 Examples of translated faces. From left to right: training position, test face shifted 20 pixels, test face shifted 50 pixels 32
3.6 ROC area vs. translation amount. Trained on 200 centered synthetic faces, tested on 900 translated synthetic test faces. 33

4.1 Performance of features extracted from synthetic, cluttered, and real training sets, tested on synthetic, cluttered, and real tests sets using SVM classifier. 36
4.2 Average C2 activation of training sets. Left: using face only features Right: using mixed features. 37
4.3 ROC distribution of feature sets when calculated over their respective training sets 38
4.4 ROC distribution of feature sets when calculated over synthetic face set 39
4.5 ROC distribution of feature sets when calculated over cluttered face set 39
4.6 ROC distribution of feature sets when calculated over real face set 40
4.7 Comparison of HMAX with feature learning, trained on real faces and tested on real faces, with computer vision systems. 40

4.8 Performance of feature selection on "mixed" features. Left: for cluttered face set. Right: for real face set. In each figure, ROC area of performance with (from left to right): face only features, all mixed features, highest and lowest ROC, only highest ROC, average C2 activation, mutual information, and randomly. ROC areas are given at the top of each bar. 42
4.9 Performance of feature selection on "mixed cluttered" features. Top left: for synthetic face set. Top right: for cluttered face set. Bottom: for real face set. In each figure, ROC area of performance with (from left to right): face only features, all mixed features, highest and lowest ROC, only highest ROC, average C2 activation, mutual information, and randomly. ROC areas are given at the top of each bar. 43
4.10 Feature ROC comparison between the "mixed" features training set and test sets. Left: Feature ROC taken over training set *vs.* cluttered faces and non-face test sets. Right: Feature ROC taken over training set *vs.* real faces and non-face test sets. 44
4.11 Feature ROC comparison between the "mixed cluttered" features training set and test sets. Top left: Feature ROC taken over training set *vs.* synthetic face and non-face test sets. Top right: Feature ROC taken over training set *vs.* cluttered face and non-face test sets. Bottom: Feature ROC taken over training set *vs.* real face and non-face test sets. 45

5.1 Varying number of face prototypes. Trained and tested on synthetic, cluttered sets using k-means classifier. . . . 49
5.2 Distribution of average C2 activations on training face set for different features types. 50
5.3 Comparing performance of SVM to k-means classifier on the four feature types. Number of face prototypes = 10. From top left going clockwise: on face only features, mixed features, mixed cluttered features, and cluttered features 51
5.4 Comparison of HMAX with feature learning (using SVM and k-means as classifier, trained on real faces and tested on real faces, with computer vision systems. The k-means system used 1 face prototype. 52

5.5 Performance of feature selection on "mixed" features using the k-means classifier. Left: for cluttered face set. Right: for real face set. Feature selection methods listed in the legend in the same notation used as Chapter 4. . 53
5.6 Performance of feature selection on "mixed cluttered" features using the k-means classifier. Top: for synthetic face set. Bottom left: for cluttered face set. Bottom right: for real face set. Feature selection methods listed in the legend in the same notation as in Chapter 4. 54

Chapter 1

Introduction

Detecting a pedestrian in your view while driving. Classifying an animal as a cat or a dog. Recognizing a familiar face in a crowd. These are all examples of object recognition at work. A system that performs object recognition is solving a difficult computational problem. There is high variability in appearance between objects within the same class and variability in viewing conditions for a specific object. The system must be able to detect the presence of an object–for example, a face–under different illuminations, scale, and views, while distinguishing it from background clutter and other classes.

The primate visual system seems to perform object recognition effortlessly while computer vision systems still lag behind in performance. How does the primate visual system manage to work both quickly and with high accuracy? Evidence from experiments with primates indicates that the ventral visual pathway, the neural pathway for initial object recognition processing, has a hierarchical, feed-forward architecture [11]. Several biological models have been proposed to interpret the findings from these experiments. One such computational model of object recognition in cortex is HMAX. HMAX models the ventral visual pathway, from the primary visual cortex (V1), the first visual area in the cortex, to the inferotemporal cortex, an area of the brain shown to be critical to object recognition [5]. The HMAX model architecture is based on experimental results on the primate visual cortex, and therefore can be used to make testable predictions about the visual system.

While HMAX performs well for paperclip-like objects [10], the hard-coded features do not generalize well to natural images and clutter (see Chapter 2). In this thesis we build upon HMAX by adding object-

specific features and apply the new model to the task of face detection. We evaluate the properties of the new model and compare its performance to the original HMAX model and machine vision systems. Further extensions were made to the architecture to explore unsupervised learning of features and the use of a biologically plausible classifier.

1.1 Related Work

Object recognition can be viewed as a learning problem. The system is first trained on example images of the target object class and other objects, learning to distinguish between them. Then, given new images, the system can detect the presence of the target object class.

In object recognition systems, there are two main variables in an approach that distinguish one system from another. The first variable is what features the system uses to represent object classes. These features can be generic, which can be used for any class, or class-specific. The second variable is the classifier, the module that determines whether an object is from the target class or not, after being trained on labeled examples. In this section, I will review previous computer vision and biologically motivated object recognition systems with different approaches to feature representation and classification.

1.1.1 Computer Vision

An example of a system that uses generic features is described in [8]. The system represents object classes in terms of local oriented multi-scale intensity differences between adjacent regions in the images and is trained using a support vector machine (SVM) classifier. A SVM is an algorithm that finds the optimal separating hyperplane between two classes [16]. SVM can be used for separable and non-separable data sets. For separable data, a linear SVM is used, and the best separating hyperplane is found in the feature space. For non-separable cases, a non-linear SVM is used. The feature space is first transformed by a kernel function into a high-dimensional space, where the optimal hyperplane is found.

In contrast, [2] describes a component-based face detection system that uses class-specific features. The system automatically learns components by growing image parts from initial seed regions until error in detection is minimized. From these image parts, components are chosen to represent faces. In this system, the image parts and their geometric arrangement are used to train a two-level SVM. The first level

of classification consists of component experts that detect the presence of the components. The second level classifies the image based on the components categorized in the first level and their positions in the image.

Another object recognition system that uses fragments from images as features is [14]. This system uses feature selection on the feature set, a technique we will explore in a later chapter. Ullman and Sali choose fragments from training images that maximize the mutual information between the fragment and the class it represents. During classification, first the system searches the test image at each location for the presence of the stored fragments. In the second stage, each location is associated with a magnitude M, a weighted sum of the fragments found at that location. For each candidate location, the system verifies that (1) the fragments are from a sufficient subset of the stored fragments and (2) positions of the fragments are consistent with each other (e.g. for detecting an upright face, the mouth fragment should be located below the nose). Based on the magnitude and the verification, the system decides whether or not the presence of the target class is in a candidate location.

1.1.2 Biological Vision

The primate visual system has a hierarchical structure, building up from simple to more complex units. Processing in the visual system starts in the primary visual cortex (V1), where simple cells respond optimally to an edge at a particular location and orientation. As one travels further along the visual pathway to higher order visual areas of the cortex, cells have increasing receptive field size as well as increasing complexity. The last purely visual area in the cortex is the inferotemporal cortex (IT). In results presented in [4], neurons were found in monkey IT that were tuned to specific views of training objects for an object recognition task. In addition, neurons were found that were scale, translation, and rotation invariant to some degree. These results motivated the following view-based object recognition systems.

SEEMORE

SEEMORE is a biologically inspired visual object recognition system [6]. SEEMORE uses a set of receptive-field like feature channels to encode objects. Each feature channel F_i is sensitive to color, angles, blobs, contours or texture. The activity of F_i can be estimated as the number of occurrences of that feature in the image. The sum of

occurrences is taken over various parameters such as position and scale depending on the feature type.

The training and test sets for SEEMORE are color video images of 3D rigid and non-rigid objects. The training set consists of several views of each object alone, varying in view angle and scale. For testing, the system has to recognize novel views of the objects presented alone on a blank background or degraded. Five possible degradations are applied to the test views: scrambling the image, adding occlusion, adding another object, changing the color, or adding noise. The system uses nearest-neighbor for classification. The distance between two views is calculated as the weighted city-block distance between their feature vectors. The training view that has the least distance from a test view is considered the best match.

Although SEEMORE has some qualities similar to biological visual systems, such as the use of receptive-field like features and its view-based approach, the goal of the system was not to be a descriptive model of an actual animal visual system [6] and therefore can not be used to make testable predictions about biological visual systems.

HMAX

HMAX models the ventral visual pathway, from the primary visual cortex (V1), the first visual area in the cortex, to the inferotemporal cortex, an area critical to object recognition [5]. HMAX's structure is made up of alternating levels of S units, which perform pattern matching, and C units, which take the max of the S level responses.

An overview of the model can be seen in Figure 1.1. The first layer, S1, consists of filters (first derivative of gaussians) tuned to different areas of the visual field, orientations (oriented bars at 0, 45, 90, and 135 degrees) and scales. These filters are analogous to the simple cell receptive fields found in the V1 area of the brain. The C1 layer responses are obtained by performing a max pooling operations over S1 filters that are tuned to the same orientation, but different scales and positions over some neighborhood. In the S2 layer, the simple features from the C1 layer (the 4 bar orientations) are combined into 2 by 2 arrangements to form 256 intermediate feature detectors. Each C2 layer unit takes the max over all S2 units differing in position and scale for a specific feature and feeds its output into the view-tuned units.

By having this alternating S and C level architecture, HMAX can increase specificity in feature detectors and increase invariance. The S levels increase specificity and maintain invariance. The increase in specificity stems from the combination of simpler features from lower

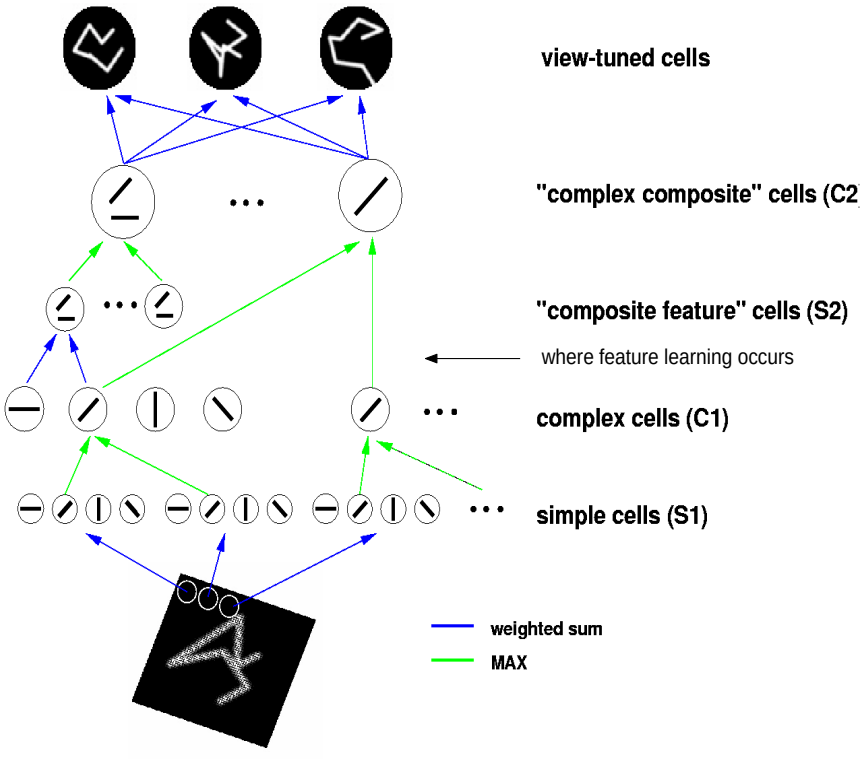


Figure 1.1: The HMAX model. The first layer, S1, consists of filters tuned to different areas of the visual field, orientations (oriented bars at 0, 45, 90, and 135 degrees) and scales. These filters are analogous to the simple cell receptive fields found in the V1 area of the brain. The C1 layer responses are obtained by performing a max pooling operations over S1 filters that are tuned to the same orientation, but different scales and positions over some neighborhood. In the S2 layer, the simple features from the C1 layer (the 4 bar orientations) are combined into 2 by 2 arrangements to form 256 intermediate feature detectors. Each C2 layer unit takes the max over all S2 units differing in position and scale for a specific feature and feeds its output into the view-tuned units. In our new model, we replace the hard-coded 256 intermediate features at the S2 level with features the system learns.

levels into more complex features.

HMAX manages to increase invariance due to the max pooling operation at the C levels. For example, suppose a horizontal bar at a certain position is presented to the system. Since each S1 filter template matches with one of four orientations at differing positions and scales, one S1 cell will respond most strongly to this bar. If the bar is translated, the S1 filter that responded most strongly to the horizontal bar at that position has a weaker response. The filter whose response is greatest to the horizontal bar at the new position will have a stronger response. When max is taken over the S1 cells in the two cases, the C1 cell that receives input from all S1 filters that prefer horizontal bars will receive the same level of input on both cases.

An alternative to taking the max is taking the sum of the responses. When taking the sum of the S1 outputs, the C1 cell would also receive the same input from the bar in the original position and the moved position. Since one input to C1 would have decreased, but the other would have increased, the total response remains the same. However, taking the sum does not maintain feature specificity when there are multiple bars in the visual field. If a C1 cell is presented with an image containing a horizontal and vertical bar, when summing the inputs, the response level does not indicate whether or not there is a horizontal bar in the field. Responses to the vertical and the horizontal bar are both included in the summation. On the other hand, if the max is taken, the response would be of the most strongly activated input cell. This response indicates what bar orientation is present in the image. Because max pooling preserves bar orientation information, it is robust to clutter [10].

The HMAX architecture is based on experimental findings on the ventral visual pathway and is consistent with results from physiological experiments on the primate visual system. As a result, it is a good biological model for making testable predictions.

1.2 Motivation

The motivation for my research is two-fold. On the computational neuroscience side, previous experiments with biological models have mostly been with single objects on a blank background, which do not simulate realistic viewing conditions. By using HMAX on face detection, we are testing out a biologically plausible model of object recognition to see how well it performs on a real world task.

In addition, in HMAX, the intermediate features are hard-coded

into the model and learning only occurs from the C2 level to the view-tuned units. The original HMAX model uses the same features for all object classes. Because these features are 2 by 2 combination of bar orientations, they may work well for paperclip like objects [10], but not for natural images like faces. When detecting faces in an image with background clutter, these generic features do not differentiate between the face and the background clutter. For a face on clutter, some features might respond strongly to the face while others respond strongly to the clutter, since the features are specific to neither. If the responses to clutter are stronger than the ones to faces, when taking the maximum activation over all these features, the resulting activation pattern will signal the presence of clutter, instead of a face. Therefore these features perform badly in face detection. The extension to HMAX would permit learning of features specific to the object class and explores learning at lower stages in the visual system. Since these features are specific to faces, even in the presence of clutter, these features will have a greater activation to faces than clutter parts of the images. When taking the maximum activation over these features, the activation pattern will be robust to clutter and still signal the presence of a face. Using class-specific features should improve performance in cluttered images.

For computer vision, this system can give some insight how to improve current object recognition algorithms . In general, computer vision algorithms use a centralized approach to account for translation and scale variation in images. To achieve translation invariance, a global window is scanned over the image to search for the target object. To normalize for scale, the image is replicated at different scales, and each of them are searched in turn. In contrast, the biological model uses distributed processing through local receptive fields, whose outputs are pooled together. The pooling builds up translation and scale invariance in the features themselves, allowing the system to detect objects in images of different scales and positions without having to preprocess the image.

1.3 Roadmap

Chapter 2 explains the basic face detection task, HMAX with feature learning architecture, and analyzes results from simulations varying system parameters. Performance from these experiment are then compared to the original HMAX. Chapter 3 presents results from testing the scale and translation invariance of HMAX with feature learning. Next, in Chapter 4, I investigate unsupervised learning of features. Chapter

5 presents results from using a biologically-plausible classifier with the system. Chapter 6 contains conclusions and discussion of future work.

Chapter 2

Basic Face Detection

In this chapter, we discuss the basic HMAX with feature learning architecture, compare its performance to standard (original) HMAX, and present results on parameter dependence experiments.

2.1 Face Detection Task

Each system (i.e. standard HMAX and HMAX with feature learning) is trained on a reduced data set similar to [2] consisting of 200 synthetic frontal face images generated from 3D head models [17] and 500 non-face images that are scenery pictures. The test sets consist of 900 "synthetic faces", 900 "cluttered faces", and 179 "real faces". The "synthetic faces" are generated from taking face images from 3D head models [17] that are different from training but are synthesized under similar illumination conditions. The "cluttered faces" are the "synthetic faces" set, but with the non-face image as background. The "real faces" are real frontal faces from the CMU PIE face database [13] presenting untrained extreme illumination conditions. The negative test set consists of 4,377 background images consider in [1] to be difficult non-face set. We decided to use a non-face set for testing different type from the training non-face set because we wanted to test using non-faces that could possibly be mistaken for faces. Examples for each set are given in Figure 2.1.

Figure 2.1: Typical stimuli used in our experiments. From left to right: Training faces and non-faces, "cluttered (test) faces", "difficult (test) faces" and test non-faces.

2.2 Methods

2.2.1 Feature Learning

To obtain class-specific features, the following steps are performed (the steps are shown in Figure 2.3): (1) Obtain C1 activations of training images using HMAX. Figure 2.2 shows example C1 activations from faces and non-faces. (2) Extract patches from training faces at the C1 layer level. The locations of the patches are randomized with each run. There are two parameters that can vary at this step: the *patch size* p and the *number of patches* m extracted from each face. Each patch is a $p \times p \times 4$ pattern of C1 activation $\mathbf{w}$, where the last 4 comes from the four different preferred orientations of C1 units. (3) Obtain the set of features $\mathbf{u}$ by performing k-means, a clustering method [3], on the patches. K-means groups the patches by similarity. The representative patches from each group are chosen as features, the number of which is determined by another parameter n. These features replace the intermediate S2 features in the original HMAX. The level in the HMAX hierarchy where feature learning takes place is indicated by the arrow in Figure 1.1. In all simulations, p varied between 2 and 20, n varied between 4 and 3,000, and m varied between 1 and 750. These S2 units behave like gaussian RBF-units and compute a function of the squared distance between an input pattern and the stored prototype: $f(x) = \exp - \frac{||x-\mathbf{u}||^2}{2\sigma^2}$, with σ chosen proportional to patch size.

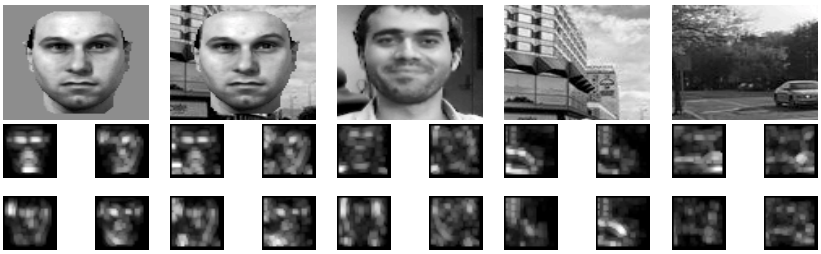


Figure 2.2: Typical stimuli and associated responses of the C1 complex cells (4 orientations). Top: Sample synthetic face , cluttered face, real face, non-faces. Bottom: The corresponding C1 activations to those images. Each of the four subfigures in the C1 activation figures maps to the four bar orientations (clockwise from top left: 0, 45, 135, 90 degrees). For simplicity, only the response at one scale is displayed. Note that an individual C1 cell is not particularly selective either to face or to non-face stimuli.

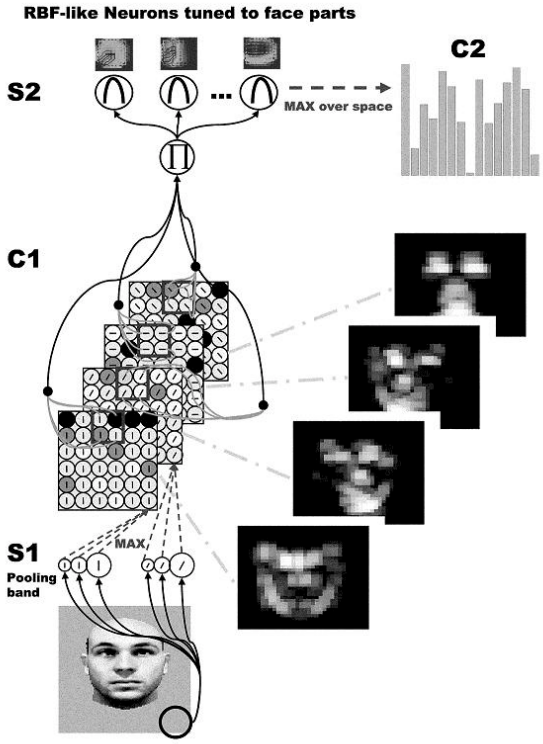


Figure 2.3: Sketch of the HMAX model with feature learning: Patterns on the model "retina" are first filtered through a continuous layer S1 (simplified on the sketch) of overlapping simple cell-like receptive fields (first derivative of gaussians) at different scales and orientations. Neighboring S1 cells in turn are pooled by C1 cells through a MAX operation. The next S2 layer contains the RBF-like units that are tuned to object-parts and compute a function of the distance between the input units and the stored prototypes ($p = 4$ in the example). On top of the system, C2 cells perform a MAX operation over the whole visual field and provide the final encoding of the stimulus, constituting the input to the classifier. The difference to standard HMAX lies in the connectivity from C1→S2 layer: While in standard HMAX, these connections are hardwired to produce 256 2×2 combinations of C1 inputs, they are now learned from the data. (Figure adapted from [12])

2.2.2 Classification

After HMAX encodes the images by a vector of C2 activations, this representation is used as input to the classifier. The system uses a Support Vector Machine [16] (SVM) classifier, a learning technique that has been used successfully in recent machine vision systems [2]. It is important to note that this classifier was not chosen for its biological plausibility, but rather as an established classification back-end that allows us to compare the quality of the different feature sets for the detection task independent of the classification technique.

2.3 Results

2.3.1 Comparison to Standard HMAX and Machine Vision System

As we can see from Fig. 2.4, the performance of standard HMAX system on the face detection task is pretty much at chance: The system does not generalize well to faces with similar illumination conditions but include background ("cluttered faces") or to faces in untrained illumination conditions ("real faces"). This indicates that the generic features in standard HMAX are insufficient to perform robust face detection. The 256 features cannot be expected to show any specificity for faces *vs.* background patterns. In particular, for an image containing a face on a background pattern, some S2 features will be most activated by image patches belonging to the face. But, for other S2 features, a part of the background might cause a stronger activation than any part of the face, thus interfering with the response that would have been caused by the face alone. This interference leads to poor generalization performances, as shown in Fig. 2.4.

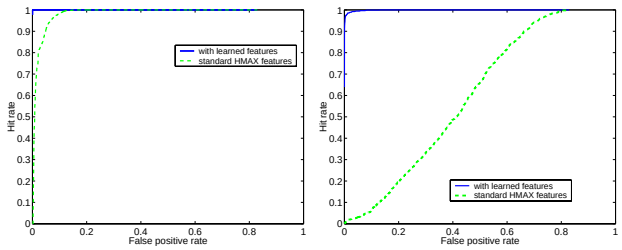
As an illustration of the feature quality of the new model *vs.* standard HMAX, we compared the average C2 activations on test images (synthetic faces and non-faces) using standard HMAX's hard-coded 256 features and 200 face-specific features. As shown in Fig. 2.5, using the learned features, the average activations are linearly separable, with the faces having higher activations than non-faces. In contrast, with the hard-coded features, the activation for faces fall in the same range as non-faces, making it difficult to separate the classes by activation.

2.3.2 Parameter Dependence

Fig. 2.7 shows the dependence of the model's performance on patch size p and the percentage of face area covered by the features (the area taken up by one feature (p^2) times the number of patches extracted per faces (m) divided by the area covered by one face). As the percentage of the face area covered by the features increases, the overlap between features should in principle increase. Features of intermediate sizes work best for "synthetic" and "cluttered" faces [1], while smaller features are better for "real" faces. Intermediate features work best for detecting faces that are similar to the training faces because first, compared with larger features, they probably have more flexibility in matching a greater number of faces. Secondly, compared to smaller features they are probably more selective to faces. Those results are in good agreement with [15] where gray-value features of intermediate sizes where shown to have higher mutual information. When the training and test sets contain different types of faces, such as synthetic faces *vs.* real faces, the larger the features, the less capable they are to generalize to real faces. Smaller feature work the best for real faces because they capture the least amount of detail specific to face type.

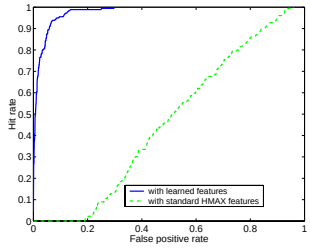
Performance as a function of the number of features n show first a rise with increasing numbers of features due to the increased discriminatory power of the feature dictionary. However, at some point performance levels off. With smaller features ($p = 2, 5$), the leveling off point occurs at a larger n than for larger features. Because small features are less specific to faces, when there is a low number of them, the activation pattern of face and non-faces are similar. With a more populated feature space for faces, the activation pattern will become more specific to faces. For large features, such as 20x20 features which almost cover an entire face, a feature set of one will already have a strong preferences to similar faces. Therefore, increasing the number of features has little effect. Fig. 2.6 shows performances for $p = 2, 5, 7, 10, 15, 20$, $m = 100$, and $n = 25, 50, 100, 200, 300$.

[1] 5×5 and 7×7 features for which performances are best correspond to cells' receptive field of about a third of a face.



(a) synthetic faces and non-faces

(b) cluttered faces and non-faces



(c) real faces and non-faces

Figure 2.4: Comparison between the new model using object-specific learned features and the standard HMAX by test set. For synthetic and cluttered face test sets, the best set of features had parameters:$p = 5$, $n = 480$, $m = 120$. For real face test set, the best set of features were $p = 2$, $n = 500$, $m = 125$. The new model generalizes well on all sets and outperforms standard HMAX.

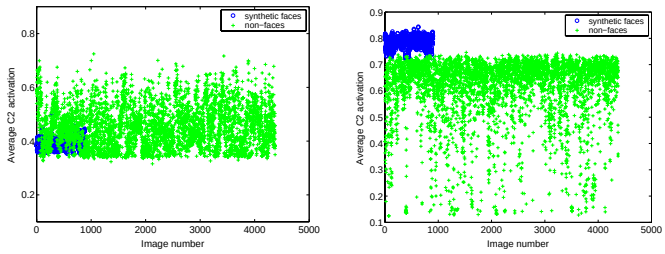


Figure 2.5: Average C2 activation of synthetic test face and test non-face set. Left: using standard HMAX features. Right: using features learning from synthetic faces.

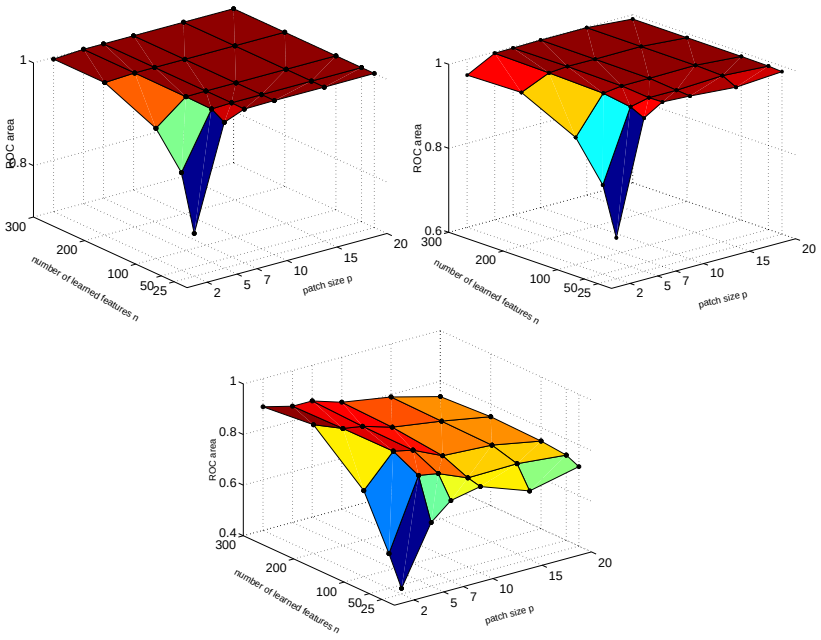


Figure 2.6: Performance (ROC area) of features learned from synthetic faces with respect to number of learned features n and p (fixed $m = 100$). Performance increases with the number of learned features to a certain level and levels off. Top left: system performance on synthetic test set. Top right: system performance on cluttered test set. Bottom: performance on real test set.

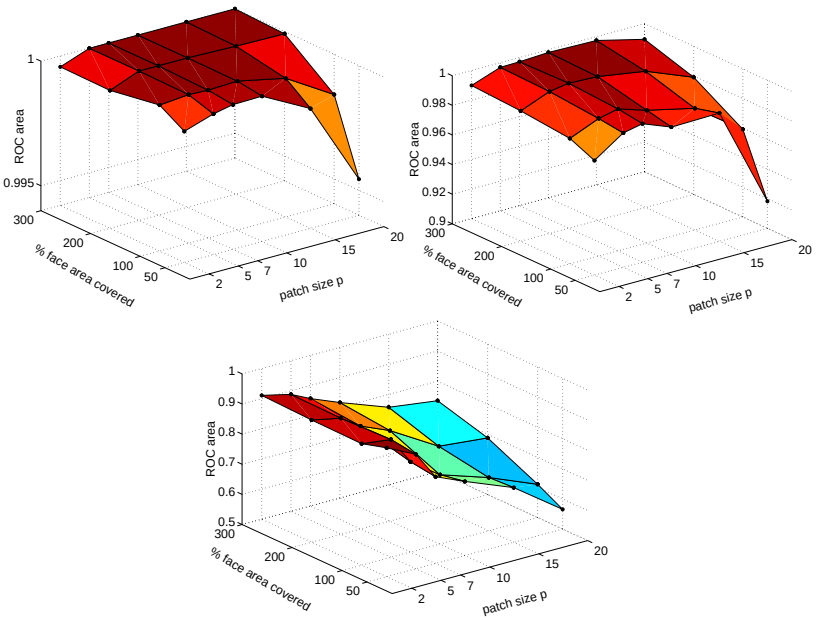


Figure 2.7: Performance (ROC area) with respect to % face area covered and p. Intermediate size features performed best on synthetic and cluttered sets, small features performed best on real faces. Top left: system performance on synthetic test set. Top right: system performance on cluttered test set. Bottom : performance on real test set.

Chapter 3

Invariance in HMAX with Feature Learning

In physiological experiments on monkeys, cells in the inferotemporal cortex demonstrated some degree of translation and scale invariance [4]. Simulation results have shown that the standard HMAX model exhibits scale and translation invariance [9], consistent with the physiological results. This chapter examines invariance in the performance of the new model, HMAX with feature learning.

3.1 Scale Invariance

Scale invariance is a result of the pooling at the C1 and C2 levels of HMAX. Pooling at the C1 level is performed in four scale bands. Band 1, 2, 3, 4 have filter standard deviation ranges of 1.75-2.25, 2.75-3.75, 4.25-5.25, and 5.75-7.25 pixels and spatial pooling ranges over neighborhoods of 4x4, 6x6, 9x9, 12x12 cells respectively. At the C2 level, the system pools over S2 activations of all bands to get the maximum response.

In the simulations discussed in the previous chapter, the features were extracted at band 2, and the C2 activations were a result of pooling over all bands. In this section, we wish to explore how each band contributes to the pooling at the C2 level. As band size increases, the area of the image which a receptive field covers increases. Example C1 activations at each band are shown in Fig. 3.1. Our hypothesis is that as face size changes, the band most tuned to that scale will "take over" and become the maximum responding band.

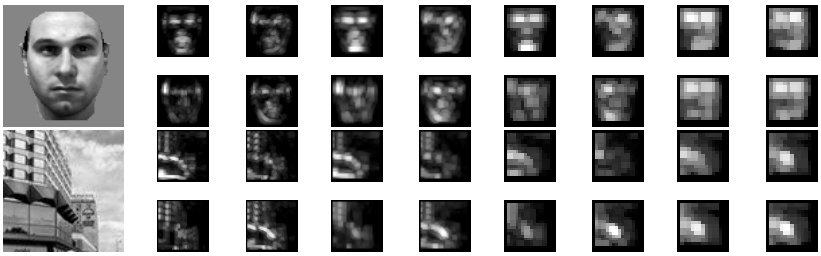


Figure 3.1: C1 activations of face and non-face at different scale bands. Top (from left to right): Sample synthetic face, C1 activation of face at band 1, band 2, band 3, and band 4. Bottom: Sample non-faces, C1 activation of non-face at band 1, band 2, band 3, and band 4. Each of the four subfigures in the C1 activation figures maps to the four bar orientations (clockwise from top left: 0, 45, 135, 90 degrees).

Figure 3.2: Example images of rescaled faces. From left to right: training scale, test face rescaled -0.4 octave, test face rescaled +0.4 octave

In the experiment, features are extracted from synthetic faces at band 2, then the system is trained using all bands. The system is then tested on synthetic faces on a uniform background, resized from 0.5-1.5 times the training size (Fig. 3.2) using bands 1-4 individually at the C2 level and also pooling over all bands. The test non-face sets are kept at normal size, but are pooled over the same bands as their respective face test sets. The rescale range of 0.5-1.5 was chosen to try to test bands a half-octave above and an octave below the training band.

As shown in Fig. 3.3, for small faces, the system at band 1 performs the best out of all the bands. As face size increases, performance at band 1 drops and band 2 take over to become the dominate band. At band 3, system performance also increase as face size increases. At

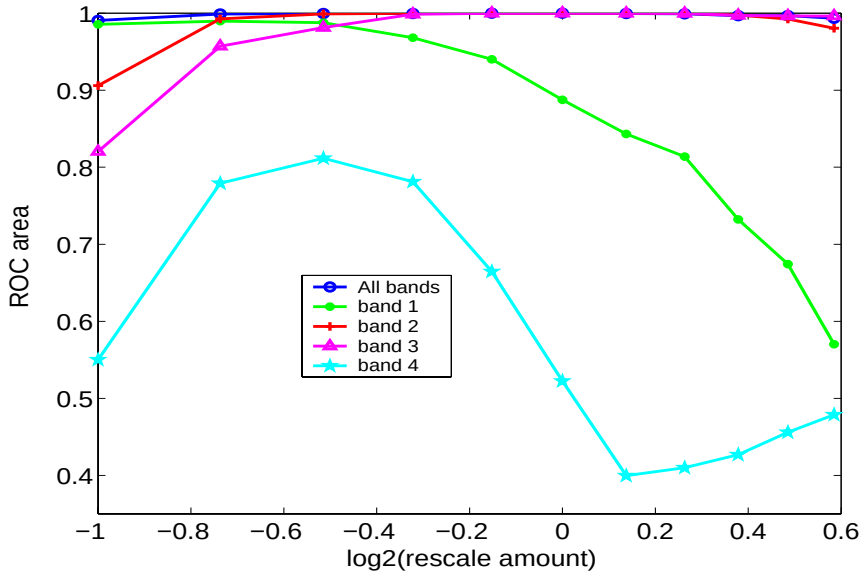


Figure 3.3: ROC area vs. log of rescale factor. Trained on synthetic faces, tested on 900 rescaled synthetic test faces. Images size is 100x100 pixels

large face sizes (1.5 times training size), band 3 becomes the dominate band while band 2 starts to decrease in performance. Band 4 has poor performance for all face sizes. Since its receptive fields are an octave above the training band's, to see if band 4 continues its upward trend in performance we re-ran the simulations with 200x200 images and a rescale range of 0.5-2 times the training size.

The average C2 activation to synthetic test faces *vs.* rescale amount is shown in Fig. 3.4. The behavior of the C2 activations as image size changes is consistent with the ROC area data above. At small sizes, band 1 has the greatest average C2 activations. As the size becomes closer to the training size, band 2 becomes the most activated band. At large face sizes, band 3 is the most activated. For band 4, as expected, the C2 activation increases as face size increases, however, its activation is consistently lower than any of the other bands. In this rescale range, band 4 is bad for detecting faces. Additional experiments to try is to increase the image size and rescale range furthers to see if band 4 follows this upward trend, or train with band 3 and since band 4 and

3 are closer in scale than band 2 and 4, performance should improve.

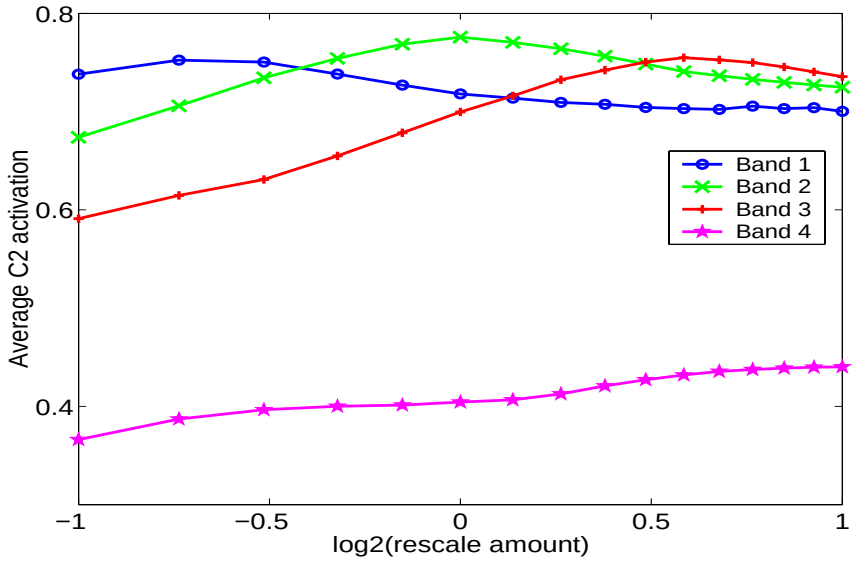


Figure 3.4: Average C2 activation vs. log of rescale factor. Trained on synthetic faces, tested on 900 rescaled synthetic test faces. Image size is 200x200 pixels

These results (from performance measured by ROC area and average C2 activations) agree with the "take over" effect we expected to see. As face size decreases and band scale is held constant, the area of the face a C1 cell covers increases. The C1 activations of the smaller face will match poorly with the features trained at band 2. However, when the C1 activations are taken using band 1, each C1 cell pools over a smaller area, thereby compensating for rescaling. Similarly as face size increases from the training size, the C1 cell covers less area. Going from band 2 to band 3, each C1 cell pools over a larger area.

When using all bands (Fig. 3.3), performance stays relatively constant for sizes around the training size, then starts to drop off slightly at the ends. The system has constant performance even though face size changes because the C2 responses are pooled from all bands. As the face size varies, we see from the performance of the system on individual bands that at least one band will be strongly activated and signal the presence of a face. Although face scale may change, by pooling over

all bands, the system can still detects the presence of the resized face.

3.2 Translation Invariance

Like scale invariance, translation invariance is the result of the HMAX pooling mechanism. From the S1 to the C1 level, each C1 cell pools over a local neighborhood of S1 cells, the range determined by the scale band. At the C2 level, after pooling over all scales, HMAX pools over all positions to get the maximum response to a feature.

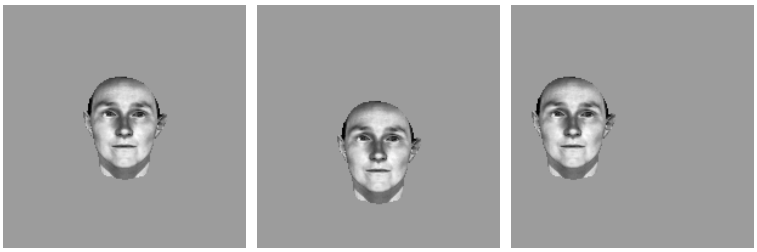


Figure 3.5: Examples of translated faces. From left to right: training position, test face shifted 20 pixels, test face shifted 50 pixels

To test translation invariance, we trained the system on 200x200 pixels faces and non-faces. The training faces are centered frontal faces. For the face test set, we translated the images 0, 10, 20, 30, 40, and 50 pixels either up, down, left, or right. Example training and test faces can be seen in Fig. 3.5.

From the results of this experiments (Fig. 3.6), we can see that performance stays relatively constant as face position changes, demonstrating the translation invariance property of HMAX.

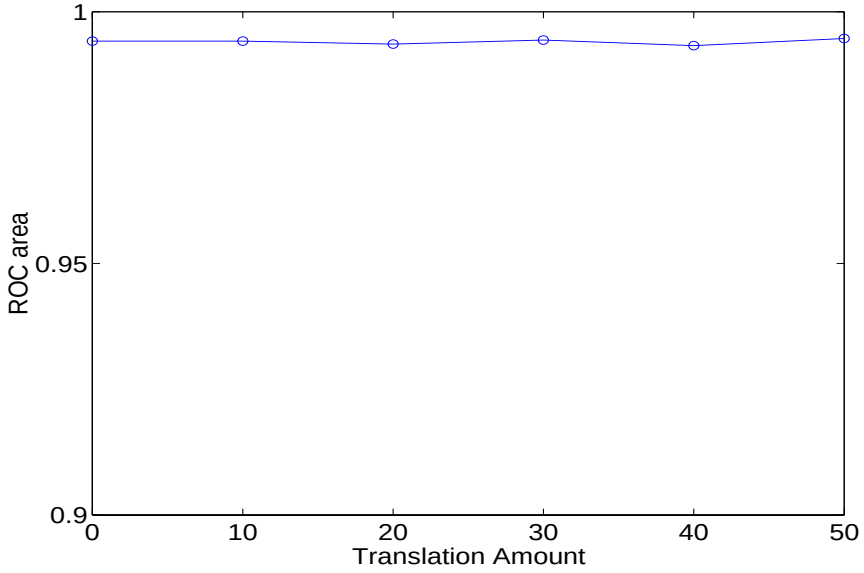


Figure 3.6: ROC area vs. translation amount. Trained on 200 centered synthetic faces, tested on 900 translated synthetic test faces.

Chapter 4

Exploring Features

In the previous experiments, the system has been trained using features extracted only from faces. However, training with features from synthetic faces on blank background does not reflect the real world learning situation where there are imperfect training stimuli consisting of both the target class and distractor objects. In this chapter, I explore (1) training with more realistic feature sets, and (2) selecting "good" features from these sets to improve performance.

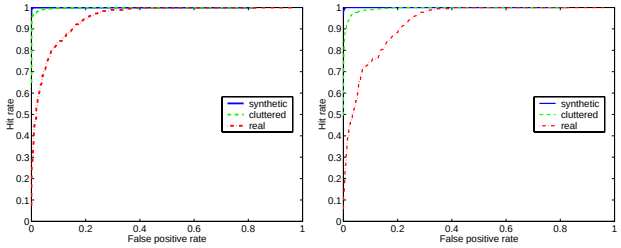
4.1 Different Feature Sets

The various feature sets used for training are:

1. "face only" features - from synthetic faces with blank background (the same set used in previous chapters, mentioned here for comparison)
2. "mixed" features - from synthetic faces with blank background and from non-faces (equal amount of face and non-face patches fed into k-means to get feature set)
3. "cluttered" features" - from cluttered synthetic faces (training set size of 900)
4. "mixed cluttered" features - from both cluttered synthetic faces and non-faces (equal amount of cluttered face and non-face patches fed into k-means to get feature set)
5. features from real faces (training set size of 42)

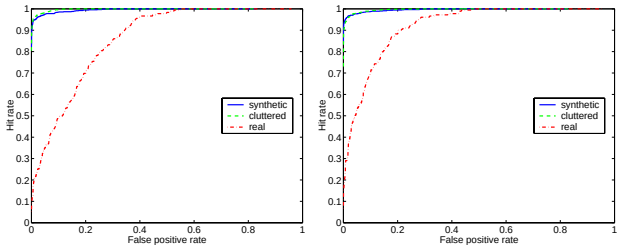
For each simulation, the training faces used correspond with the feature set used. For example, when training using "mixed cluttered" features, cluttered faces are used as the training face set for the classifier. The test sets used are the same as the system described in Chapter 2: 900 synthetic faces, 900 cluttered faces, 179 real faces, and 4,377 non-faces.

The performance of the feature sets are shown in Fig. 4.1. For all feature sets, the test face set most similar to the training set performed best. This result makes sense since the most similar test set would have the same distribution of C2 activations as the training set.



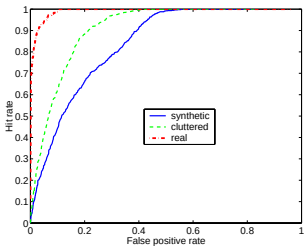
(a) face only features

(b) mixed features



(c) cluttered features

(d) mixed cluttered features



(e) real face features

Figure 4.1: Performance of features extracted from synthetic, cluttered, and real training sets, tested on synthetic, cluttered, and real tests sets using SVM classifier.

"Mixed" features perform worse than face only features. Since these features consist of face and non-face patches, these features are no longer as discriminatory for faces. Faces respond poorly to the non-face tuned features while non-faces are more activated. Looking at the training sets' C2 activations using "mixed" features (Fig. 4.2), we see that the average C2 activation of synthetic faces decreases as compared to the average C2 activation using face only features, while the average C2 activation of non-faces increases. As a result, the two classes are not as easily separable, accounting for the poor performance. To improve performance, feature selection is explored in the next section.

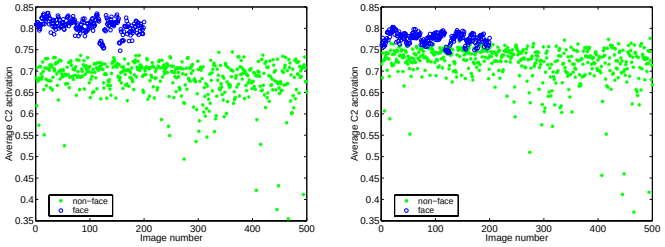


Figure 4.2: Average C2 activation of training sets. Left: using face only features Right: using mixed features.

"Mixed clutter" features also display poor performance for the cluttered face test set, although performance on real faces is better than when trained on cluttered features. To explore the reason behind these results, we have to examine the features themselves, what is the distribution of "good" features (ones that are better at distinguishing between faces and non-faces) and "bad" features. One technique to measure how "good" a feature is by calculating its ROC. Figures 4.3 to 4.6 show the distribution of features by ROC for feature sets 1-4.

Mixed features sets ("mixed", "mixed cluttered") have more features with low ROCs than pure face feature sets ("face only", "cluttered"), but less features with high ROCs. If we take low ROC to mean that these features are good non-face detectors, including non-face patches produces features tuned to non-faces. In Fig. 4.6, when using "cluttered" features *vs.* "mixed cluttered" features on real faces, both have very few good face detectors, as indicated by the absences of high ROC features. However, the "mixed cluttered" set has more features tuned to non-faces. Having more non-face features may be a reason why "mixed cluttered" performs better on real faces: these features can better distinguish non-faces from real faces.

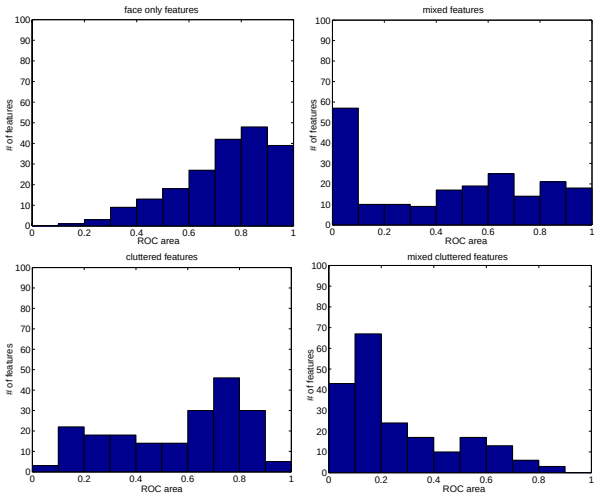


Figure 4.3: ROC distribution of feature sets when calculated over their respective training sets

We compare our system trained on real faces with other face detections systems: the component-based system described in [2], and a whole face classifier [7]. HMAX with feature learning performs better than machine vision systems (Fig. 4.7). Some possible reasons for the better performance: (1) our system uses real faces to train, while the component-based system uses synthetic faces, so our features are more tuned to real faces (2) our features are constructed from C1 units, while the component-based system's features are pixel values. Our features, along with HMAX's hierarchical structure, make the features more generalizable to images in different viewing conditions. (3) the component-based system uses an SVM classifier to learn features while our system uses k-means. The SVM requires a large number of training examples in order to find the best separating hyperplane. Since we only train with 42 faces, we should expect the computer vision system 's performance to improve if we increase training set size. The whole face classifier is trained on real faces and uses a whole face template to detect faces. From these results, it seems that face parts are more flexible to variations in faces than a face template.

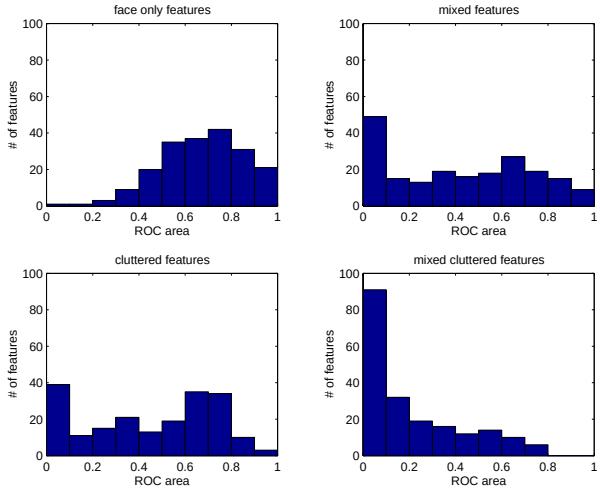


Figure 4.4: ROC distribution of feature sets when calculated over synthetic face set

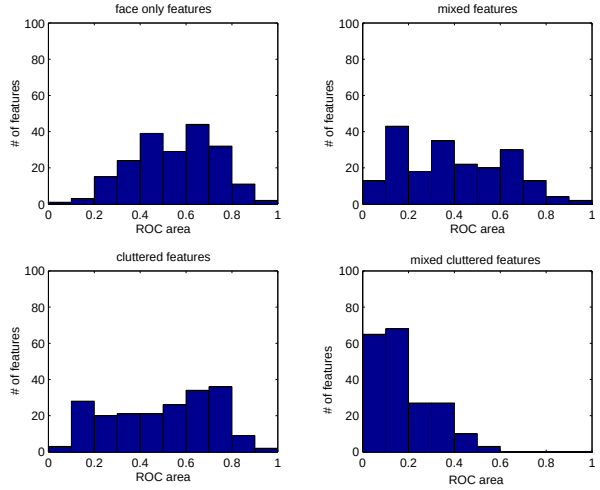


Figure 4.5: ROC distribution of feature sets when calculated over cluttered face set

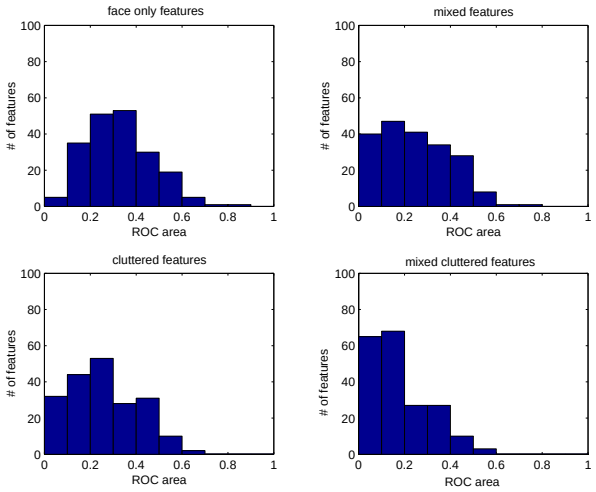


Figure 4.6: ROC distribution of feature sets when calculated over real face set

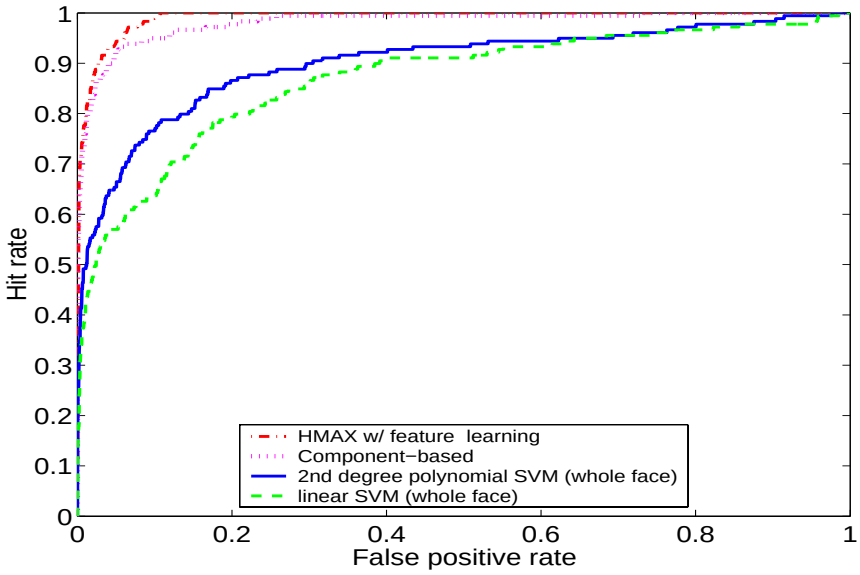


Figure 4.7: Comparison of HMAX with feature learning, trained on real faces and tested on real faces, with computer vision systems.

4.2 Feature Selection

In training using the "mixed" and "mixed cluttered" feature sets, we are taking a step toward unsupervised learning. Instead of training only on features from the target class, the system is given features possibly from faces and non-faces.

We try to improve performance of these mixed feature sets by selecting a subset of "good" features. We apply the following methods to select features, picking feature by :

1. highest ROC - pick features that have high hit rates for faces and low false alarm rates for non-faces

2. highest and lowest ROC - features that are are good face or non-face detectors. Chosen by taking the features with ROC farthest from 0.5

3. highest average C2 activation - high C2 activations on training faces maybe equivalent to good face detecting features [10]

4. mutual information - pick out features that contribute the most amount of information to deciding image class. Mutual information for a feature is calculated by:

   $MI(C, X) = \sum_C \sum_X p(c, x) \log(p(c, x)/(p(x)p(c))$

   where C is the class (face or non-face) and X is the feature (value ranges from 0-1). This feature selection method has been used in computer vision systems [15]. Note: In the algorithm, X takes on discrete values. Since the responses to a feature can take on a continuous value between 0-1, we discretized the responses, flooring the value to the nearest tenth.

5. random - baseline performance to compare with above methods (averaged over five iterations)

Results of applying the five feature selection techniques on "mixed" features and "mixed cluttered" features to get 100 best features are shown in Fig. 4.8 and Fig. 4.9 respectively.

In all the feature selection results, picking features by highest ROC alone (method 1) performed better than by highest and lowest ROC (method 2). From the better performance, we can conclude that picking by highest ROC, even though it may include features with ROC's around chance, the system performs better than including low ROC features but having fewer high ROC features. Although from the previous section, we saw that good non-face features did help performance

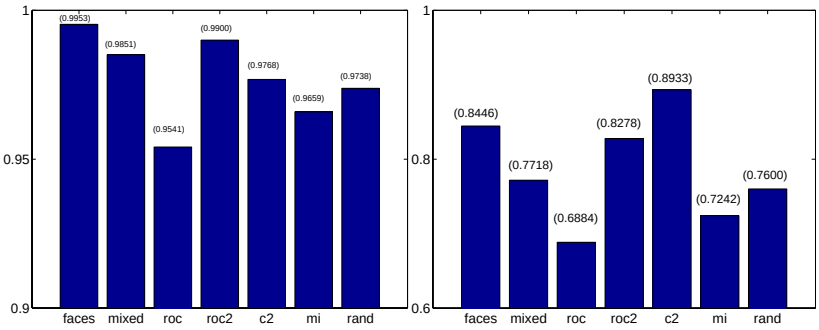


Figure 4.8: Performance of feature selection on "mixed" features. Left: for cluttered face set. Right: for real face set. In each figure, ROC area of performance with (from left to right): face only features, all mixed features, highest and lowest ROC, only highest ROC, average C2 activation, mutual information, and randomly. ROC areas are given at the top of each bar.

for the real face set, in that case there were very few face features so good non-faces features had more impact. In comparison to having more face features, non-face features seem not to be as important. There seems to be two possible reasons for this result that come from comparing the ROC of features on the training sets versus the test sets (Fig. 4.10 and Fig. 4.11). First, when picking features by method 1, we get some features that have ROCs around chance for the training set, but they have high ROCs for the test sets. If we use method 2, these features are not picked. Secondly, the training and test non-face sets are different types of non-faces. The first consists of scenery pictures, while the latter are hard non-faces as deemed by an LDA classifier [1]. The features tuned to the training non-face set may perform poorly on the test non-face set. In Fig. 4.11, we see that the training and test feature ROCs are less correlated for low ROCs than for high ROC , showing that non-face detectors do not generalize well.

In the "mixed" features set, for cluttered faces, selection by highest ROC value performed the best, almost as well as faces only. For real faces, feature selection by C2 activation performed the best. Also, in the "mixed cluttered" feature set, C2 average selection method performed the best out of all the methods for all test sets. Since the activations were averaged over only face responses, picking the features with the high response to faces might translate into good face detectors that are robust to clutter [10].

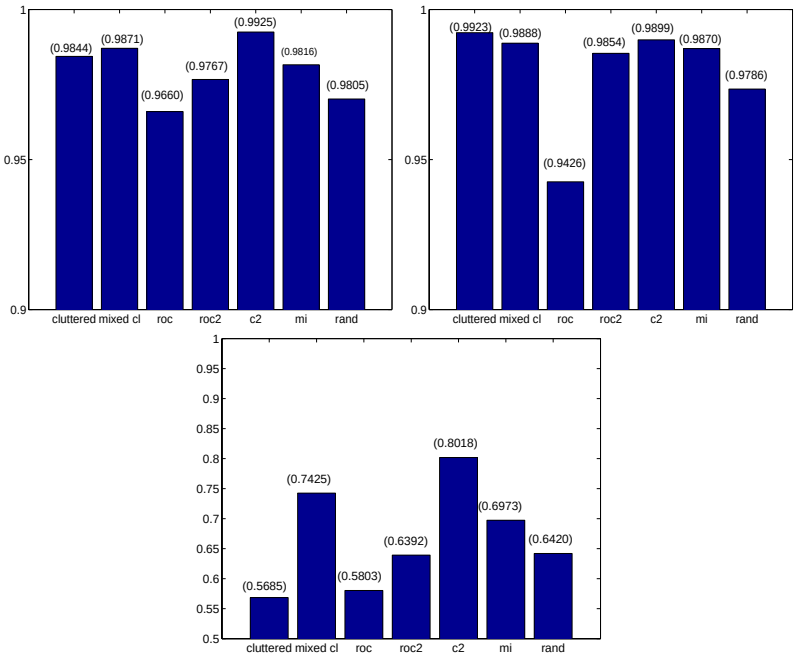


Figure 4.9: Performance of feature selection on "mixed cluttered"features. Top left: for synthetic face set. Top right: for cluttered face set. Bottom: for real face set. In each figure, ROC area of performance with (from left to right): face only features, all mixed features, highest and lowest ROC, only highest ROC, average C2 activation, mutual information, and randomly. ROC areas are given at the top of each bar.

Mutual information (MI) of "mixed" features calculated using the training set have low correlation (all less than 0.1) with ones calculated using the test sets. The features that have high MI for the training set may or may not have high MI for the test sets. Therefore we do not expect performance of feature selection by MI to be any better than random, which is what we see in Fig. 4.8. For the "mixed cluttered" feature set, the MI correlation between the training set and synthetic, cluttered, and real test sets are 0.15, 0.20, and 0.0550 respectively. The increased correlation for synthetic and cluttered sets may be why we see better performances for this set (Fig. 4.9), than for "mixed" features.

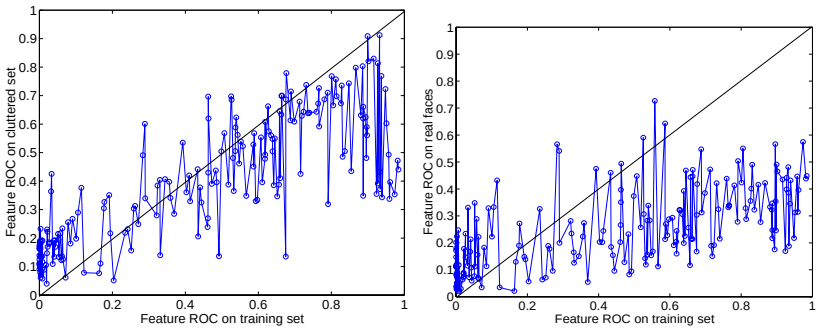


Figure 4.10: Feature ROC comparison between the "mixed" features training set and test sets. Left: Feature ROC taken over training set *vs.* cluttered faces and non-face test sets. Right: Feature ROC taken over training set *vs.* real faces and non-face test sets.

4.3 Conclusions

In this chapter, we explored using unsupervised learning to obtain features, then selecting "good" features to improve performance. The results have shown that only selecting good face features (from using methods such as highest ROC and average C2 activation) are more effective than selecting both good face and non-face features. Because faces have less variability in shape than non-faces (which can be images of buildings, cars, trees, etc.), a good non-face feature for one set of non-faces may generalize poorly to other types of non-faces, while face feature responses are more consistent across sets. Selecting features by average C2 activation gives us a simple, biologically-plausible method to find good features. A face tuned cell can be created by choosing C2 units as afferents that response highly to faces.

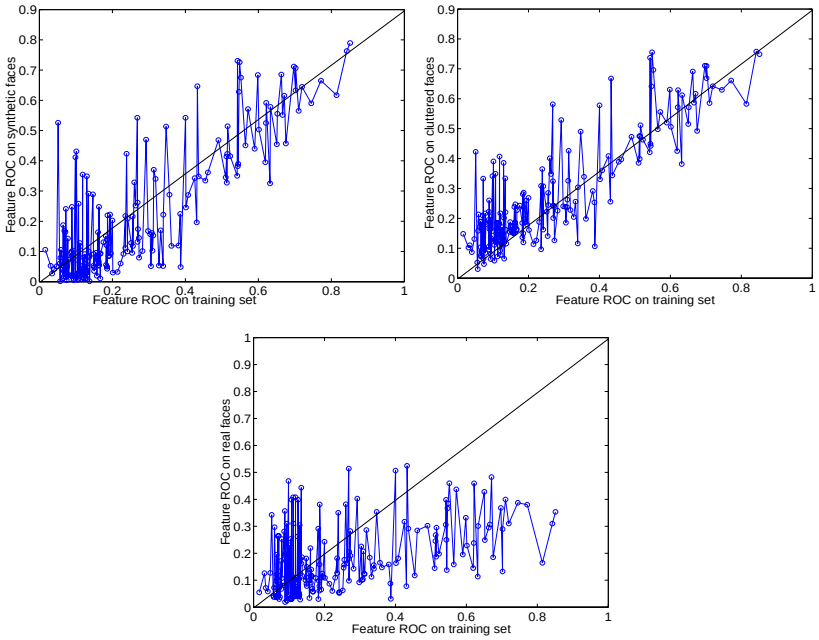


Figure 4.11: Feature ROC comparison between the "mixed cluttered" features training set and test sets. Top left: Feature ROC taken over training set *vs.* synthetic face and non-face test sets. Top right: Feature ROC taken over training set *vs.* cluttered face and non-face test sets. Bottom: Feature ROC taken over training set *vs.* real face and non-face test sets.

Chapter 5

Biologically Plausible Classifier

In all the experiments discussed so far, the classifier used is the SVM. As touched upon in Chapter 2, we decided to use an SVM so that we could compare the results to other representations. However, SVM is not a biologically-plausible classifier. In the training stage of an SVM, in order to find the optimal separating hyperplane, the classifier has to solve a quadratic programming problem. This problem can not easily be solved by neural computation. In the following set of experiments, we replace the SVM with a simpler classification procedure.

5.1 Methods

After obtaining the C2 activations of the training face set, we use k-means (same algorithm used for getting features from patches) on the C2 activation to get "face prototypes" (FP). Instead of creating representative face parts, now we are getting representative whole faces, encoded by an activation pattern over C2 units. The number of face prototypes is a parameter f, which varied from 1 to 30 in these simulations. To classify an image, the system takes the Euclidean distance between the image's C2 activation vector and the C2 activation of each face prototype. The minimum distance over all the face prototypes is recorded as that image's likeness to a face. The face prototypes can be thought of as RBF-like face units, so the minimum distance to the prototypes is equivalent to the maximum activation over all the face units. Our hypothesis is that face images will have similar C2 activa-

tion patterns as the face prototypes, so their maximum activation will be larger than non-face images'. Then to distinguish faces from non-faces, the system can set a maximum activation threshold value, where anything above the threshold is a face, anything below it is a non-face, creating a simple classifier. In these experiments, the classifier did not have a set threshold. To measure performance, we varied the threshold to produce an ROC curve.

Since k-means initializes its centers randomly, for all experiments in this chapter, we average the results over five runs.

5.2 Results

5.2.1 Face Prototype Number Dependence

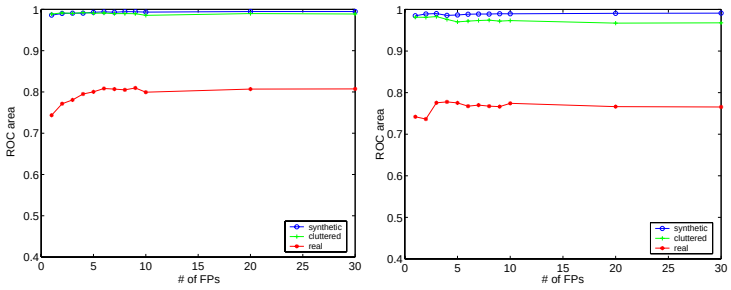
We varied the number of face prototypes from 1 to 30 on all five training sets to see how performance changes as the number of face prototypes increased. The results are shown in Fig. 5.1.

Performance does not vary greatly when training on cluttered and synthetic faces. Yet with real faces, prototype of one gives the best performance, then for increasing number of prototypes, the ROC area drops sharply then levels off.

The face prototypes cover the C2 unit space for faces. As the number of prototypes increases, the better the space that is covered. What the C2 unit space for face and non-faces looks like will determine what effect increasing coverage will have on the performance of using k-means as classifier. Looking at the distribution of the average C2 activation of the training sets might give us a clue (Fig. 5.2). For feature sets trained on synthetic and cluttered faces, most of the average face C2 activations for these features cluster around the mean activation, with distribution falling off as the activations are farther from the mean. Therefore, the first face prototypes capture the majority of the faces. As face prototype number increases, additional prototypes capture the outliers. These prototypes might increase performance by covering more of the C2 unit space, but they also might decrease performance if they also are closer to non-faces. However, since outliers are few, performance does not fluctuate greatly as a result.

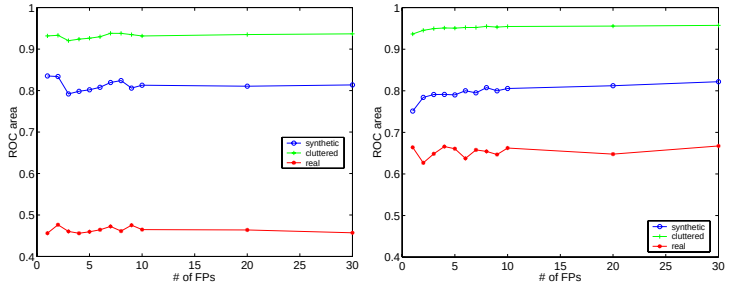
The distribution of the average C2 units of the training real faces is similar to the other training sets. Its standard deviation is higher than the other sets, which indicates that the outliers are further away. Additional face prototypes are then further away from the mean as well, potentially capturing more non-faces and decreasing performance. However, taking the average does not give us much insight into what is

happening in the feature space because it reduces the whole space into one number. One possible solution is to reduce the high-dimensional feature space into a small enough space so that the data can be visualized yet still maintain the space's structure. We can only speculate that the feature space is shaped such that the additional face prototypes capture the face outliers but also non-faces.



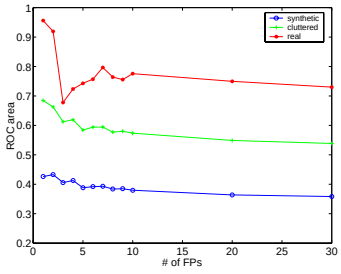
(a) face only features

(b) mixed features



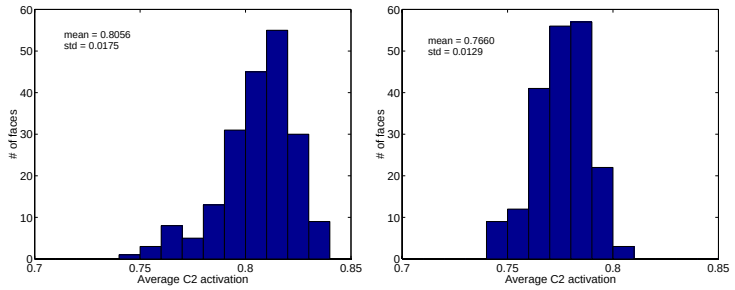
(c) cluttered features

(d) mixed cluttered features

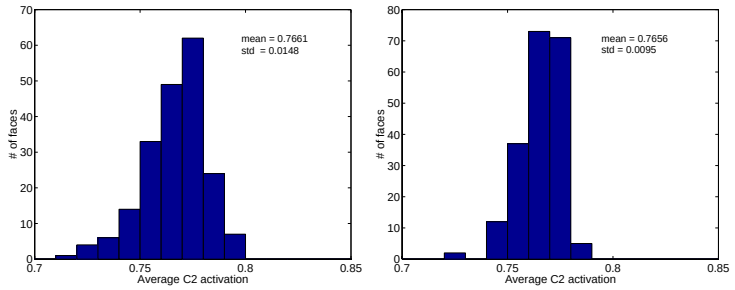


(e) real face features

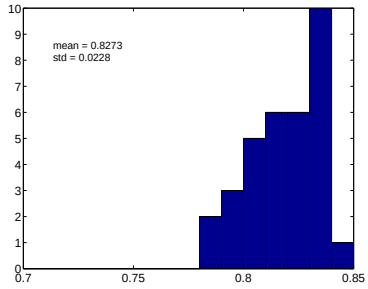
Figure 5.1: Varying number of face prototypes. Trained and tested on synthetic, cluttered sets using k-means classifier.



(a) face only features (b) mixed features



(c) cluttered features (d) mixed cluttered features



(e) real face features

Figure 5.2: Distribution of average C2 activations on training face set for different features types.

5.2.2 Using Face Prototypes on Previous Experiments

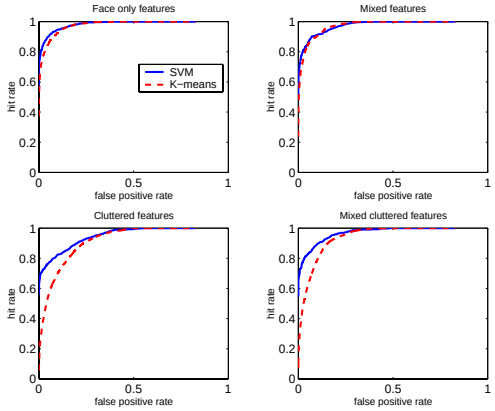


Figure 5.3: Comparing performance of SVM to k-means classifier on the four feature types. Number of face prototypes = 10. From top left going clockwise: on face only features, mixed features, mixed cluttered features, and cluttered features

We re-ran the same simulations presented in the last chapter, but replaced the SVM with the k-means classifier. Figure 5.3 compares the performance of using the SVM versus the k-means classifier. For face only, "mixed", and real faces feature sets (Fig. 5.4), k-means performance is comparable to the SVM. For cluttered and "mixed cluttered" feature sets, k-means performs worse then the SVM. A possible reason for the decreased performance of cluttered training sets is that k-means only uses the training face set to classify. Face prototypes of cluttered faces, might be similar to both faces and non-faces, making the two sets hard to separate based solely on distance to the nearest face. In contrast, the SVM uses both face and non-face information to find the best separating plane between the two.

The results for the feature selection simulations are shown in Figures 5.5 and 5.6. The relative performance of the various feature selection methods changes when we use k-means. For example, mutual information replaces C2 as the best method for "mixed cluttered" features. Because k-means and SVM are inherently different (one uses a separating plane, the other using minimum distance to a center), it is expected that their outcomes might differ. The two classifiers weigh the features differently. The SVM sets weights to the features that

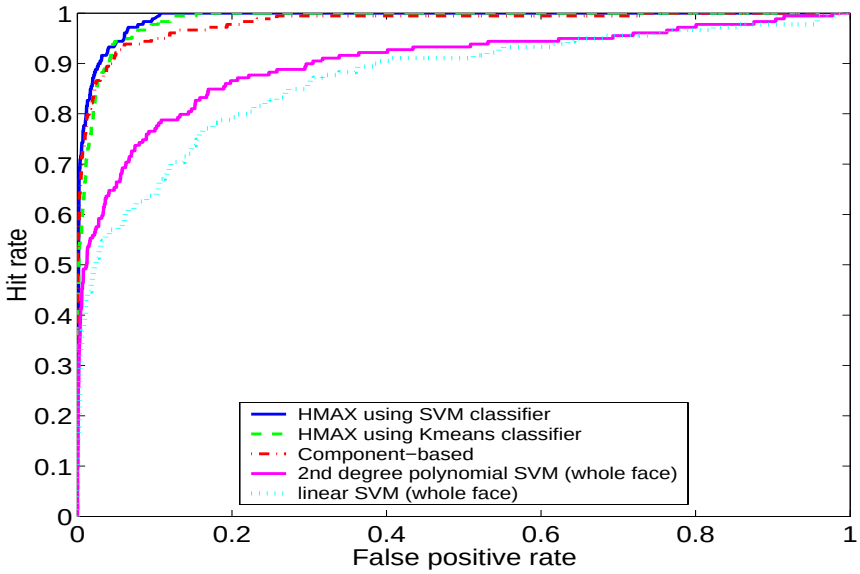


Figure 5.4: Comparison of HMAX with feature learning (using SVM and k-means as classifier, trained on real faces and tested on real faces, with computer vision systems. The k-means system used 1 face prototype.

maximizes the width of the separating hyperplane. In the k-means classifier, how much each feature contributes depends on where in the feature space the nearest face prototype is. Any of the feature selection methods could have selected both good and bad features in varying proportions. Performance then depends on which features the classifier weights more. Further exploration into the reasons for the different outputs is relegated to future work.

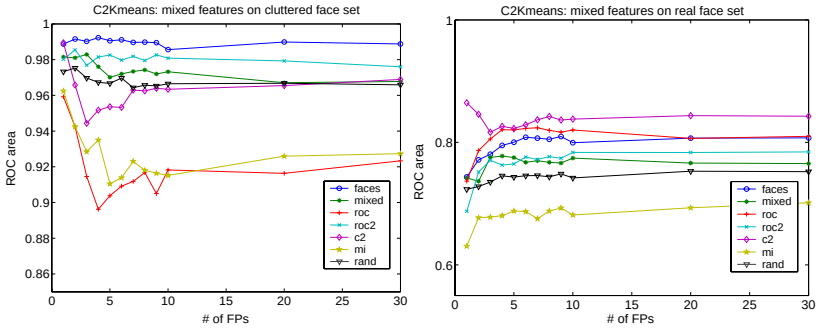


Figure 5.5: Performance of feature selection on "mixed"features using the k-means classifier. Left: for cluttered face set. Right: for real face set. Feature selection methods listed in the legend in the same notation used as Chapter 4.

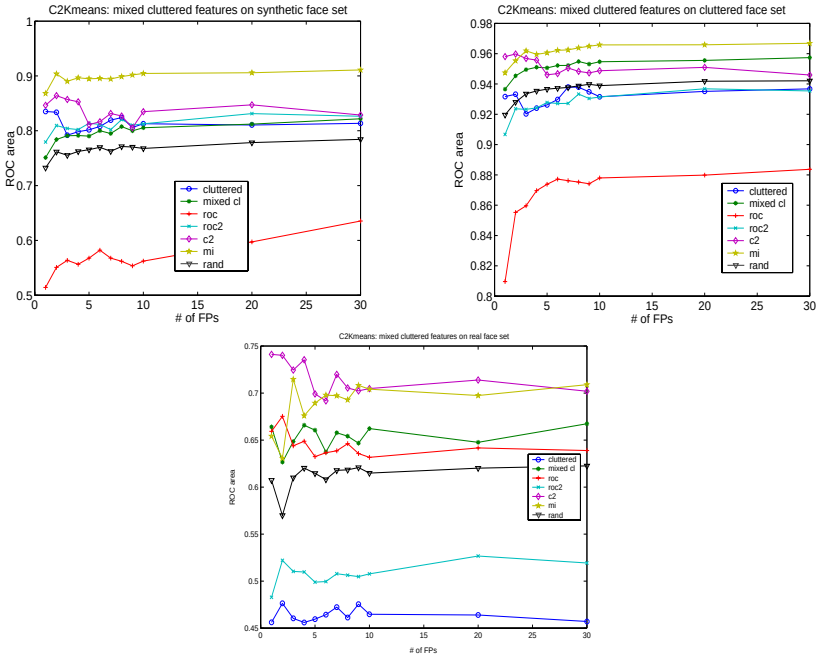


Figure 5.6: Performance of feature selection on "mixed cluttered"features using the k-means classifier. Top: for synthetic face set. Bottom left: for cluttered face set. Bottom right: for real face set. Feature selection methods listed in the legend in the same notation as in Chapter 4.

5.3 Conclusions

The SVM 's training stage is complex and not biologically plausible, but once the separating hyperplane is found, the classification stage is simple. Given the hyperplane, a data point is classified based on which side of the plane it is on. The k-means training stage is biologically feasible to implement using self-organizing cortical maps. For classification, face-tuned cells can set an activation threshold, and anything above that threshold is labeled a face. Both the SVM and k-means classifier have some usability issues. The SVM requires a substantial number of training points in order to perform well. In addition, there are parameters that one can vary such as kernel type and data chunk size that influence performance. For the k-means classifier, performance is poorer if non-face information is needed to classify images, for example images that contain clutter. Secondly, the issue of how many face prototypes to use to get the best performance is dependent on the shape of the feature space. What the optimal number for one training set may not apply to another set.

Chapter 6

Discussion

Computer vision system traditionally have simple features, such as wavelets and pixel value, along with a complex classification procedure (preprocessing to normalize for illumination, searching for object in different scaled images and position) [8, 2, 14]. In contrast, HMAX, a biological computer vision system, has complex features, but a simpler classification stage. The pooling performed in HMAX builds scale and translation invariance into the final C2 encoding. However, HMAX's hard-coded features do not perform well on the face detection task.

The goal of the new HMAX model was to replace the hard-coded features with object specific features –namely face parts, and see if performance improved. As expected, HMAX with feature learning performed better than the standard HMAX on the face detection task. The average C2 activations of the two type of features show that the object specific features are more tuned toward faces than non-faces, while HMAX's features have no such preference. By integrating object-specificity into the HMAX architecture, we were able to build a system whose performance is competitive with current computer vision systems.

Additional simulations found that the new model also exhibited scale and translation invariance. Explorations into unsupervised feature learning, feature selections, and the use of a simple classifier gave promising results. However, more investigation into the underlying mechanisms behind the results needs to be done in order to have a full understanding.

For future work on the HMAX model, further exploration on feature selection and alternative classification methods needs to be done to turn the new model into a fully biologically-plausible system. In our

experiments, we have only looked at a basic face detection task. Theoretically, the model should be easily extendable to other tasks just by replacing the features. It will be interesting to apply the model for car detection, and for even more specific recognition tasks, such as recognition between faces with the same features we currently use. Results would determine if the system works well regardless of the specific recognition task.

Bibliography

[1] B. Heisele, T. Poggio, and M. Pontil. Face detection in still gray images. CBCL *Paper 187/*AI *Memo 1687 (*MIT*)*, 2000.

[2] B. Heisele, T. Serre, M. Pontil, and T. Poggio. Component-based face detection. *Proc. IEEE Conf. on Computer Vision and Pattern Recognition*, 1:657–62, 2001.

[3] A. Jain, M. Murty, and P. Flynn. Data clustering: A review. *ACM Computing Surveys*, 3(31):264–323, 1999.

[4] N. Logothetis, J. Pauls, and T. Poggio. Shape representation in the inferior temporal cortex of monkeys. *Current Biology*, 5(5):552–563, 1995.

[5] N. Logothetis and D. Sheinberg. Visual object recognition. *Annu. Rev. Neurosci.*, 19:577–621, 1996.

[6] B. W. Mel. Seemore: Combining color, shape and texture histogramming in a neurally inspired approach to visual object recognition. *Neural Computation*, 9:777–804, 1997.

[7] E. Osuna. *Support Vector Machines: Training and Applications.* PhD thesis, MIT, EECS and OR, Cambridge, MA, 1998.

[8] C. Papageorgiou and T. Poggio. A trainable system for object detection. *International Journal of Computer Vision*, 38(1):15–33, 2000.

[9] M. Riesenhuber. *How a Part of the Brain Might or Might Not Work: A New Hierarchical Model of Object Recognition.* PhD thesis, MIT, Brain and Cognitive Sciences, Cambridge, MA, 2000.

[10] M. Riesenhuber and T. Poggio. Hierarchical models of object recognition in cortex. *Nature Neuroscience*, 2:1019–1025, 1999.

[11] M. Riesenhuber and T. Poggio. *How Visual Cortex Recognizes Objects: The Tale of the Standard Model*. MIT Press, 2003.

[12] T. Serre, M. Riesenhuber, J. Louie, and T. Poggio. On the role of object-specific features in real world object recognition in biological vision. *Biologically Motivated Computer Vision*, pages 387–397, 2002.

[13] T. Sim, S. Baker, and M. Bsat. The CMU pose, illumination, and expression (PIE) database. *International Conference on Automatic Face and Gesture Recognition*, pages 53–8, 2002.

[14] S. Ullman and E. Sali. Object classification using a fragment-base representation. *Biologically Motivated Computer Vision*, pages 73–87, 2000.

[15] S. Ullman, M. Vidal-Naquet, and E. Sali. Visual features of intermediate complexity and their use in classification. *Nat. Neurosci.*, 5(7):682–87, 2002.

[16] V. Vapnik. *The nature of statistical learning*. Springer Verlag, 1995.

[17] T. Vetter. Synthesis of novel views from a single face. *International Journal of Computer Vision*, 28(2):103–116, 1998.