massachusetts institute of technology — artificial intelligence laboratory

Face Representation in Cortex: Studies Using a Simple and Not So Special Model

Ezra Rosen

AI Technical Report 2003-010 June 2003
CBCL Memo 228

Face Representation in Cortex: Studies using a simple and not so special model

by

Ezra Rosen

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

Master of Engineering in Electrical Engineering and Computer Science

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

Face Representation in Cortex: Studies using a simple and not so special model

by

Ezra Rosen

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 Electrical Engineering and Computer Science

Abstract

The face inversion effect has been widely documented as an effect of the uniqueness of face processing. Using a computational model, we show that the face inversion effect is a byproduct of expertise with respect to the face object class. In simulations using HMAX, a hierarchical, shape based model, we show that the magnitude of the inversion effect is a function of the specificity of the representation. Using many, sharply tuned units, an "expert" has a large inversion effect. On the other hand, if fewer, broadly tuned units are used, the expertise is lost, and this "novice" has a small inversion effect. As the size of the inversion effect is a product of the representation, not the object class, given the right training we can create experts and novices in any object class. Using the same representations as with faces, we create experts and novices for cars. We also measure the feasibility of a view-based model for recognition of rotated objects using HMAX. Using faces, we show that transfer of learning to novel views is possible. Given only one training view, the view-based model can recognize a face at a new orientation via interpolation from the views to which it had been tuned. Although the model can generalize well to upright faces, inverted faces yield poor performance because the features change differently under rotation.

Thesis Supervisor: Tomaso Poggio
Title: Eugene McDermott Professor

Acknowledgments

I would like to thank my parents for supporting me, my sister and brother for putting up with me, and Max and Tommy for teaching me.

Contents

List of Figures

Chapter 1

Introduction

Object recognition is of vital necessity to animals, and face recognition is one of the most important subsets of object recognition. It has been hypthesized that faces are special, and the notion of special can be divided into two parts. The first argues that a specific region of the visual cortex is devoted to faces, while the second posits a unique computational mechanism inherent in face processing. Prosopagnosia could be the result of the impairment of a specific region of the brain, giving credence that there exists a face region within the visual cortex. A large face inversion effect provides the evidence for a specific mechanism for processing faces. In comparison to recognition of other classes of visual stimuli, multiple studies have shown that face recognition is disproportionately impaired by vertical stimulus inversion[3, 4, 30]. Here, we confine ourselves to addressing the question of whether face processing is special, hence we investigate the face inversion effect in detail.

1.1 The effect of inversion on recognition memory for faces

In studying faces, inverted faces are an ideal stimulus for use in experiments. Inverted faces are equally as complex as their upright counterparts, but subjects have no experience with them. One of the first to exploit this to try to learn about face recognition was Yin[30]. In his experiment, he tested recognition of faces and other classes of objects that are usually only seen in one orientation, such as houses and airplanes. Both upright and inverted images were presented. For upright stimuli, faces had the highest performance, but the inverted faces were the most difficult stimuli to recognize. He found that inversion affects face recognition disproportionately relative to the other classes of objects that he tested. Because of the similar complexity of the different object classes involved, he thought that the large inversion effect was a sign that faces were different.

In a recognition task involving upright faces, inverted faces, or other objects such as houses, upright face recognition was found to be most sensitive to the disruption of the configuration of features[8]. This is known as holistic or configural processing. Configural features are relationships between features of an object; an example would be the distance between the eyes, if we make the assumption that the eyes are features. With face inversion, relationships like "on top of" becomes "underneath", and "left of" becomes "right of". Different from the idea of configural information is that of featural information. Feature based information does not use relationships between features, but rather analyzes each part of the image individually. No one is sure what constitutes a facial feature, but one can still think about featural information as the presence or absence of some of these features. Both configural and featural recognition systems work in parallel in any discrimination task. Although Yin could not trace the origins of the large inversion effect, he found a clue from verbal reports of subjects after their experiments. They reported using two different strategies to remember the images, the first was to look for a distinguishing feature, and the second to get a general impression of the whole picture. The former was used for all objects but faces, while the latter was used only in faces. Despite this, they reported being unable to use the second strategy for the inverted face. This suggests that faces are recognized as single complex wholes, without decomposition into separately represented parts[7].

Going against this, there is some evidence that a disproportionate effect of inversion is not found in a face-matching task. Valentine[26] did not find a differential effect of orientation upon matching faces and houses in a 'mental rotation' procedure using sequential presentation. When thinking about recognition experiments, the role of perception and the role of memory must be separated. Therefore, it is possible that the disproportionate effect of inversion only emerges when the task involves recognizing a face as one stored in memory. Also, Valentine and Bruce[27] examined the effect of inversion upon recognition of classes of faces which differ in their familiarity - own- and other-race faces. It was argued that two races of faces will be approximately equated on all properties except familiarity. Therefore, if the greater familiarity with recognizing faces causes the disproportionate effect of inversion, recognition of own race faces should show a larger effect of inversion than recognition of other race-faces. In fact the opposite interaction was obtained.

Although the disproportionate inversion effect inherent to faces is undeniable, its cause does not lie in the nature of the face stimulus. Rather, we think that the inversion effect observed with faces is a product of expertise. The magnitude of inversion effect is a function of the tightness of the tuning to a given object class. People are very familiar with faces, and we call a group that is familiar with an object class expert. On the other hand, people are novices with respect to other object classes, as they are less familiar with those stimuli. The HMAX model is used to synthetically create different representations for faces that have varying specificity. Using identical stimuli, we show how these different representations yield differing inversion effects with faces. To strengthen this point, we show that this result generalizes to other object classes as well.

1.2 Learning and Generalization

Apart from analyzing the way in which faces are processed and investigating whether this method is special, we can also use the HMAX model to study learning in the visual system. A recognition system needs not only recognize a given object in one form, but under all possible transformations. For example, rotation or facial expression can change some or all of the features of a face to varying degrees. It is unlikely that a separate "grandmother cell" for each of these manifestations of the same image exists because of the huge number of necessary cells this would require.

This problem can be solved by creating a network that is capable of learning, effectively reducing the number of units needed to recognize any one object. We hypothesize that this network is view-based, meaning that only a small amount of feature configurations for a given object need to be stored. The memory of any one face is represented implicitly by a population of faces rotated at different angles[22]. If such a network were tuned to a set of representative views of some prototype faces, it could interpolate to recognize test faces presented at a novel orientation. This model should be able to generalize from a single training view of a face to novel views. Also, this generalization should be possible for any object class whose members share a 3D structure. We create such a model using HMAX and show that these types of learning and generalization can occur for upright face stimuli.

Chapter 2

Methods

2.1 The HMAX Model

The HMAX model has been described elsewhere[21], and on the WWW[10], but a description of its structure and functionality will be briefly given here. The input to the model is a square array of grayscale pixels, and its output is a variable number of view tuned units(VTUs). The computation is arranged hierarchically, and the levels from bottom to top are known as S1, C1, S2, C2, and the VTU level.

The S1 units are the lowest level, and they receive the image in its raw form. The S1 units are dense(there is one at every pixel of image), and each is a Gaussian filter, in particular the second derivative with respect to the horizontal axis of a Gaussian. These Gaussian filters are square-normalized to one and sum-normalized to zero. These filters come in sizes varying from 7×7 to 29×29 squares, in increments of 2 pixels, and at orientations of 0, 45, 90, and 135 degrees. At each pixel, there exists every type of filter in terms of size and orientation. The output of each of these filters is the convolution(dot-product) of the filter patch with the fraction of stimulus with which it overlaps. Thus, these activations have values between -1 and 1, and the model takes the absolute value to force these responses between 0 and 1.

Moving up the hierarchy, the S1 cells feed into the C1 layer. C1 cells are divided into scale bands, and these scale bands reflect which S1 cells are fed into any given C1 cell. The four scale bands are filters of size 7-9, 11-15, 17-21, and 23-29, so a C1 cell in the first scale band only accepts input from S1 filters of size 7 or 9. One C1 cell pools over all S1 cells in its scale band that are in its vicinity, or pooling region. These pooling regions also change size with scale band, and are 4, 6,

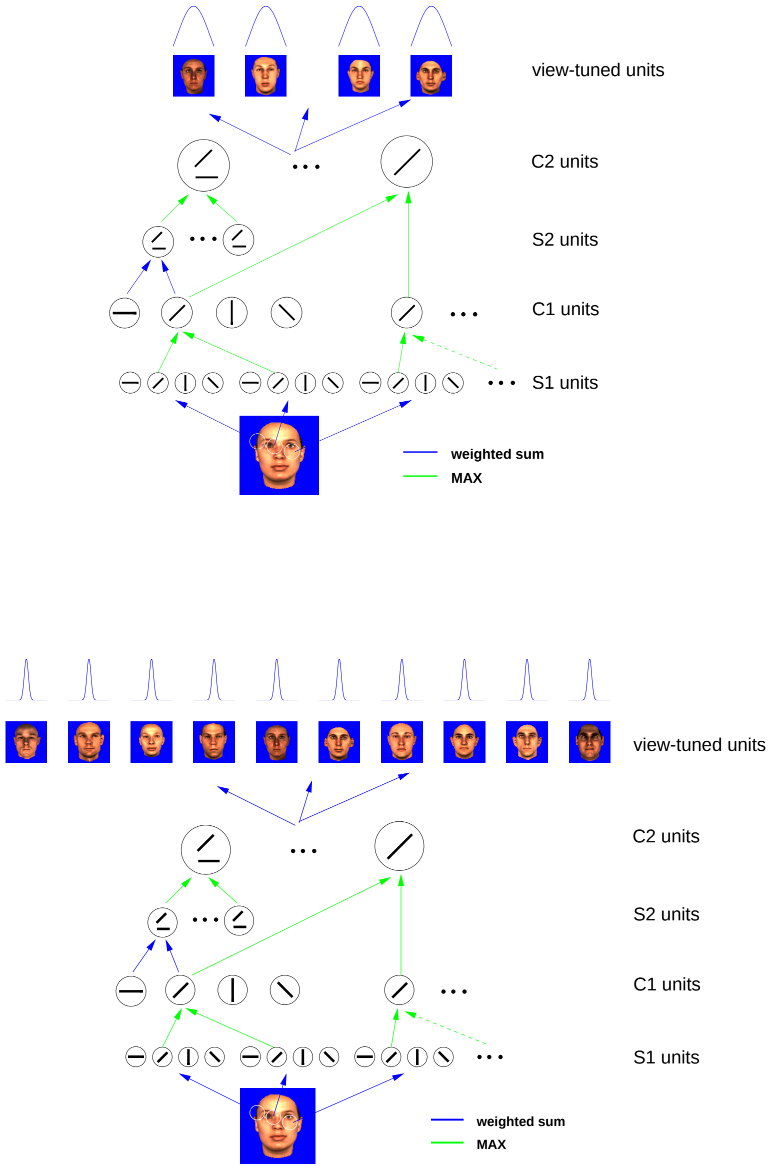

Figure 2.1: Schematic of the HMAX model. The bottom is an "expert", the top a "novice". See Methods.

9, and 12, respectively. Additionally, each C1 cell only pools over S1 cells with filters oriented in only one direction. The pooling that is done is a maximum operation. For example, the smallest scale band has S1 filters that are 7 × 7 and the 9 × 9. One specific C1 cell takes the maximum of all S1 cells oriented at 0 degrees in the smallest scale band in a 4 × 4 square. Thus, for each pooling region within a given scale band, there are 4 C1 cell types. C1 cells are not dense like the S1 layer, and the spacing of the C1 cells is given by the overlap parameter. In our simulations a value of 2 was used, meaning that half of the S1 units feeding into a C1 unit were also used as input for the adjacent C1 unit in each direction.

The S2 level is next, and these S2 cells do a combination of C1 cells from a given scale band. S2 cells group over 4 adjacent, nonoverlapping C1 cells arranged in a square. In this context, nonoverlapping means these C1 cells all had pooled over distinct, different S1 cells. There are 256 different types of S2 units in each filter band, which correspond to the 4^4 possible arrangements of C1 units. The S2 activations are computed by finding the distance of the C1 activations from the S2 target, and then exponentiating the result. We use 1 as the S2 target, so the output of this level ranges between e^{-4} and 1.

These S2 activations are then passed on to the C2 cells. These cells pool over all of space, and over all scale bands. Each C2 unit pools(takes the maximum) over all S2 units of a given type. Pooling these S2 cells into one C2 cell thus achieves position and size invariance. There are 256 of these C2 units as well, one for each S2 filter type.

C2 units are input to the view-tuned units (VTUs), known as such because they are tuned to a two-dimensional view of a three-dimensional object. The C2 → VTU connection is the stage of the standard HMAX model where learning occurs, and all lower units in the standard HMAX model have no learning capabilities. A single VTU unit is tuned to a stimulus by selecting the activities of some or all of the C2 units in response to that stimulus. This vector serves as the center of a 256-dimensional Gaussian response function. If a test face's C2 activation pattern exactly matches the C2 activation pattern evoked by the training stimulus, the response reaches its maximal value, 1. These VTUs form a radial basis function(RBF) network, a neural network approached by viewing the design as a curve fitting problem in high dimensional space. Learning is equivalent to finding a multidimensional function that provides a best fit to the training data. Within this RBF, we can vary the number of VTUs that we use, the standard deviation of the Gaussian, and the number of C2 afferents fed into the RBF.

The above describes the standard HMAX model; another variation involves changing the S2 targets to allow for perceptual learning at the S2 level, as motivated by Serre et al[23]. Standard HMAX has been shown to capture the shape tuning and invariance properties from physiological experiments, but fails in natural images. The low specificity of the hardwired C2 feature set yields units that do not show any particular tuning for faces versus background. To combat this, a layer of perceptual learning is added at the S2 level, substituting for the original targets. Specifically, these new S2 targets are clustered C1 activations, where the C1 cells have been stimulated by a face. Patches of these clustered C2 activations are chosen at random. Some examples of these learned S2 targets are shown in figure 2.2. With this addition, there is now learning at two different levels of the HMAX model.

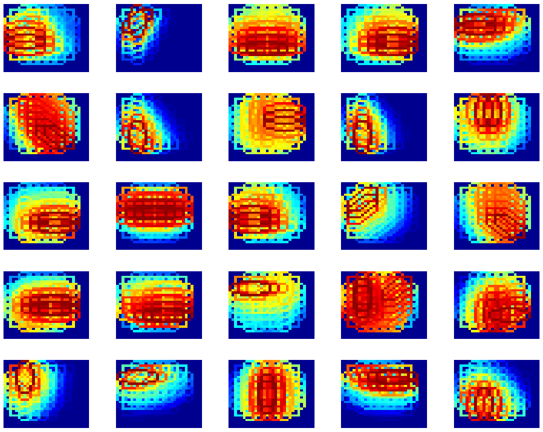

Figure 2.2: More reasonable 5 × 5 S2 targets. Ellipses on top of the filters allow for better visualization.

2.2 Stimuli

For stimuli to be used in the recognition task and as RBF centers for VTU units, we used grayscale faces that were 128 × 128 pixels. There were 182 distinct faces that served as VTU units, and an additional 18 faces for stimuli. These 18 faces were morphed with each other. Morphed faces are especially useful for recognition experiments for one can quantify the similarity between two different faces, and hence the relative difficulty of the recognition task at hand. The face morphing was implemented using software developed by Vetter et al[28]. Examples of some face VTU units and some face morphs are given in figure 2.3.

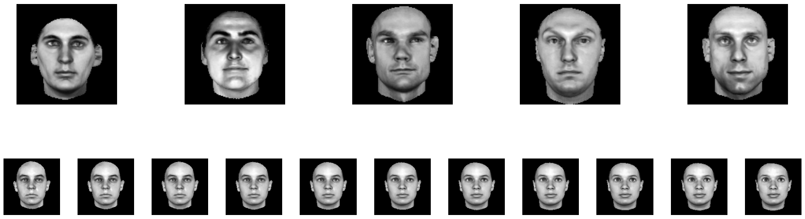

Figure 2.3: The top line are some examples of faces to which the VTU units are tuned. The bottom line moves moves along a morphline, where the two prototype faces are on the outside. For example, the fourth face(the third morph) is a mixture of 70% of the face on the far left and 30% of the face on the far right.

Cars were used as alternative stimulus to faces, and this morphing software was developed by Shelton[24]. These stimuli were 128 × 128 pixels as in the face simulation, but here we had only 13 cars to be used in VTU units. Also, there were 8 prototype cars, each morphed with every other, yielding 56 distinct car morphs. Some examples of these can be found in figure 2.4.

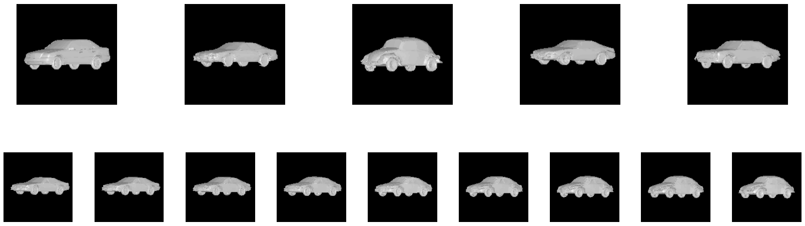

Figure 2.4: Some car stimuli used. The top line are cars used in VTUs, the bottom moves along a morphline.

Chapter 3

Face Inversion Effect as a Result of Expertise in a Shape Based Model

As discussed in chapter 1, faces have been thought of as special due to their disproportionate inversion effect and because of prosopagnosic patients. In a landmark paper[4], Carey and Diamond sought to show that faces were special, they were not in a class all their own. They tested out their theory on a different class of objects than faces, that of dogs. As subjects, they used dog show judges, experts in both the dog and face object classes. The control group was regular college students, who were allegedly experts on faces, but not on dogs. For each object class, the experiment consisted of a training phase, in which subjects were told to remember items, and a test phase, in which subjects underwent a forced-choice recognition task. Results from their experiment are given in figure 3.1. The dog experts clearly have a large inversion effect for faces and dogs, whereas the students only had the large inversion effect with faces. Subjects having a large inversion effect are known as experts in a specific object class, whereas subjects showing a small inversion effect are known as novices. These designations are due to hypotheses regarding the cause of the inversion effect.

They argued that the large effect of inversion on face recognition is a result of the major role of relational distinguishing features. They posit that the large inversion effect is a product of three conditions being met. First, members of the object class must share a configuration, or first-order relational properties. Second, observers must be able

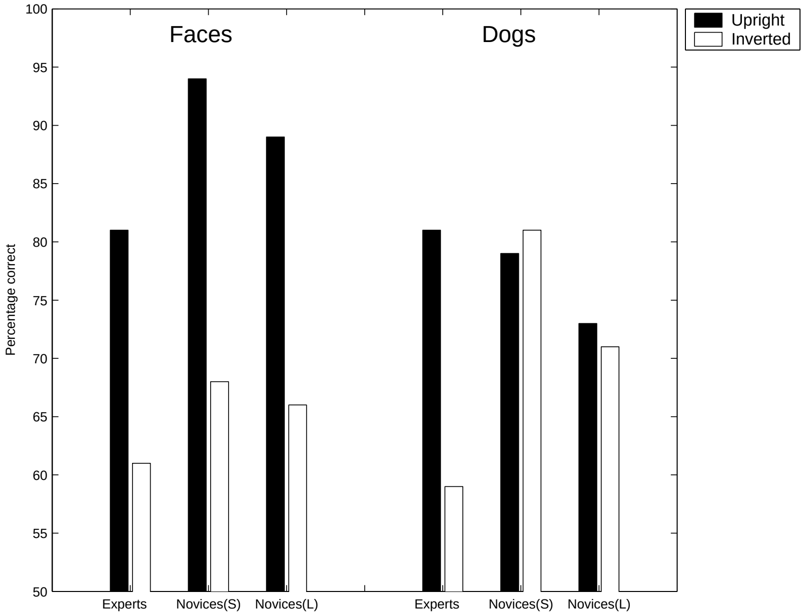

Figure 3.1: Performance of experts(dog show judges) and novices(college students) on faces and dogs presented upright and inverted. Novices(S) were given a small set size on dogs, whereas novices(L) were given the same large set size as were experts.

to use second-order relational features to individuate members of the given object class. Second-order relational properties are the relations among elements that define the shared configuration. In all faces, the first-order configuration is the same; for example, the nose is always above the lips. People are forced to discriminate between faces by using second-order properties. On the other hand, houses may have the garage towards the left or the right of the front door, so first-order properties are sufficient for discrimination. With this criterion necessary, it is impossible for landscapes or houses to have a large inversion effect because they can be individuated via first-order relational properties. Last, subjects must have the expertise to exploit these second-order features. Because of their familiarity with dogs, their dog show judges possess expertise on the dog object class. Also, dogs require second-order relational features for recognition. This explains why there is a large inversion effect there as given in figure 3.1.

This argument can account for their observations, but there is a far simpler way to explain their data. It is unlikely that there is a special mechanism that is used in face processing. We think that the large inversion effect inherent in face recognition is purely due to the tuning of the computational units involved in recognition. That is, the inversion effect is due to the parameter settings in the computation, but the methodology and architecture that is used is identical. People have more training and more experience with faces than with any other object class, and consequently in face recognition many, sharply tuned units are used. With these specific view units, a large inversion effect is observed.

This effect is displayed quite nicely in figure 3.2. In that figure, activations for an expert, with many, sharply tuned units are compared to those of a novice with fewer, broadly tuned units. Looking at the plot on the left, the performance for the expert falls off at a far sharper rate than that of the novice. Both expert and novice are tuned to the same face, but the expert is more tightly tuned. The difference in activations between the prototype face and one 5 morph units away is much bigger for the expert than the novice. Thus, in a recognition task, the expert would function better because there is a large difference in the activations of the two faces. On the other hand, on inverted faces, the

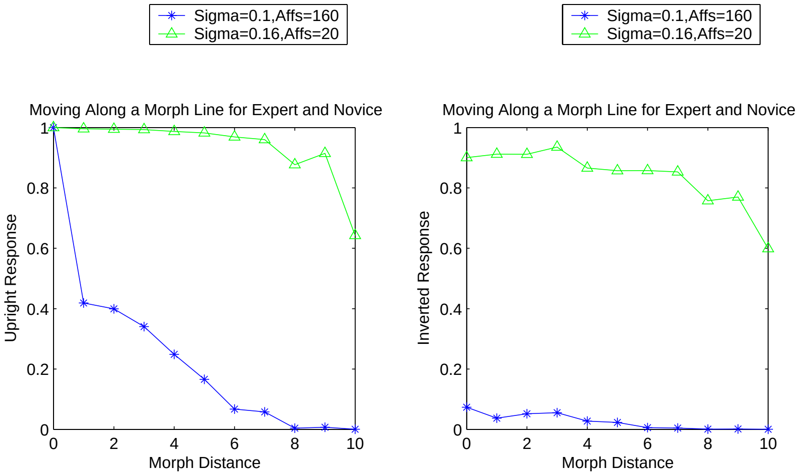

Figure 3.2: Moving along a morphline for an expert and a novice. This is the activation of one VTU unit. We start with the face to which the VTU was tuned(resulting in upright performance=1), then slowly move away from that face.

expert's activation curve changes such that the difference in activations between prototype and morph is small. The result is a huge drop in performance on inverted faces for an expert. The novice behaves differently. The difference in activations of prototype and morph is almost the same for upright and inverted faces, yielding a small inversion effect. It is very important to realize that the responses of these cells are based purely on shape, with no face-specific processing. Had another stimulus been artificially created possessing the same features as the test face, the unit would have responded in the same way because of their identical C2 activations.

In our simulations, we test this hypothesis by running simulations using the HMAX model. We can create experts and novices in recognition of any object class by varying the parameters of our model accordingly. Thus, even though people are well tuned to faces(hence, experts) we can create face experts or face novices using our model. As described above, training using a few, broadly tuned units would constitute a novice, while training with many, finely tuned units would characterize an expert. By recognizing faces in these two different ways, we show that the faces themselves are not what is special in face processing. Rather, it is the way in which the processing units are tuned to faces that is their special quality. Furthermore, by running identical tests on another object class we can show that these ideas do not only apply to faces.

3.1 Simulations

3.1.1 Representation

The HMAX model is described extensively in section 2, but the VTU level is not explained in full. This is the topmost level as shown in figure 2.1. The low-level machinery is identical in the expert and novice, but the tuning of cells in the VTU level differentiates the two.

Within the VTU level, there are five parameters that we can change to vary the specificity of the model. The first parameter to look at is the number of VTUs. Increasing the number of VTUs makes the model more specific, and the number of VTUs(known as SSCU) can be set to 1, 4, 8, 32, 64, 128, and 182. The subset of VTUs are picked at random from the 182 total VTUs. To ensure that the VTUs chosen are representative of the data set, VTUs are chosen randomly 5 times for each setting of SSCU. Additionally, one can select only the most strongly activated used VTUs(the nMostAct parameter) for recognition. The value of nMostAct can take on values of 1, 4, 8, 16, and 32. By increasing nMostAct, the model becomes more specific as well. Not only are the number of VTUs varied, but the settings within a given VTU can be changed as well. The two parameters here are the number of most activated C2 units to be selected as afferents for a VTU(the numAffs parameter), and the standard deviation of its Gaussian response function(σ). In general, specificity increases when numAffs increases and when σ decreases. NumAffs can assume values of 20, 80, 160, and 256, while σ can take on values of 0.1, 0.12, 0.14, 0.16, 0.18, 0.20, 0.40, and 0.80. Lastly, the standard deviation of the additive Gaussian noise can be varied as well. Although this parameter has nothing to do with the initial activations, it is of crucial importance when calculating Euclidean distances of responses. The standard deviation of the noise took on values of 0.1, 0.2, and 0.4. As expected, when noise increases, signal to noise ratio decreases, and hence a given unit will become less specific.

3.1.2 Constraints using data from fMRI fractions

There are an enormous number of possible settings of our five parameters(see 3.1.1), and the number of permutations increases even more when we set both an expert and a novice. Thus, it is necessary to reduce this number by constraining the parameters via other experiments that explore face inversion. One such experiment was performed by Kanwisher et al[14], in which they performed functional magnetic resonance imaging to study the effect of face inversion in the fusiform face area(FFA). Several imaging studies[12, 13, 17] have demonstrated that the FFA responds selectively to faces, compared to a wide variety of other stimulus types. The FFA is analogous to our expert because of its selectively to upright faces. They find the average activity, or fMRI fraction, to be 23% higher for upright faces than for inverted ones.

In that light, we ran HMAX on upright and inverted faces, and measured the average activity over all 182 VTUs. Note that the only parameters that we varied were number of afferents, noise, and σ. SSCU and nMostAct were meaningless as we averaged over all VTU units. As in their experiment, we observed the percentage change in activation for upright and inverted faces. Our “valid” expert settings were those that produced an fMRI fraction between 15% and 25%. The results of these simulations are given in 3.3, and from these data, the number of permutations of expert settings can clearly be reduced.

From the data in figure 3.3, we can see that there are a few values within admissible noise levels(noise = 0.1 or 0.2) that allow for the fMRI fraction to lie between 15% and 25%. There are four of them,

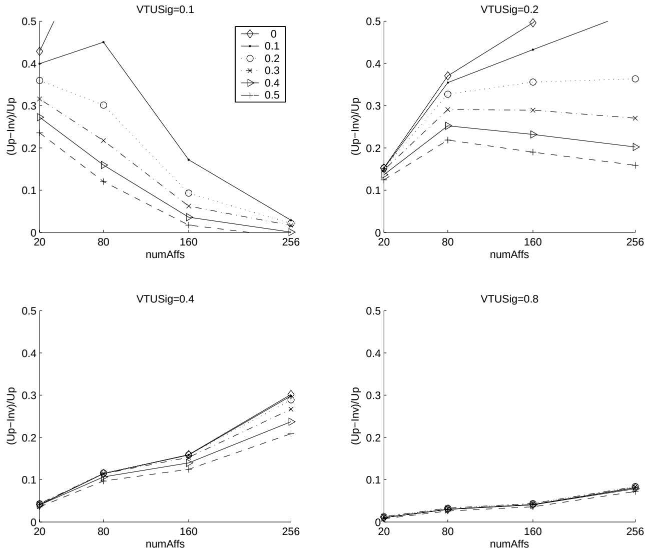

Figure 3.3: Adding noise to the fMRI fractions. Adding noise has the potential to decrease these fMRI fractions by a considerable amount, hence having a better fit to the values from the Kanwisher experiment. Each line on the graph adds Gaussian noise with σ given in the legend.

and in triplet form of $\{noise, \sigma, numAffs\}$, they are $\{0.1, 0.2, 20\}$, $\{0.1, 0.1, 160\}$, $\{0.1, 0.4, 160\}$, and $\{0.2, 0.4, 160\}$. Hence, by forcing the expert to agree with the physiological data in [14], the number of possibilities for the expert in that 3D parameter space is reduced from 48 to 4. Note that despite this reduction, in reality there are still many available permutations because the SSCU and nMostAct parameters are still unconstrained, as they don't affect the fMRI comparison.

3.1.3 Recognition Paradigm

The VTU units to be used in the recognition task were built by tuning them to 182 prototype faces. Taken together, these VTU units comprise the VTU layer. In the simulation, 18 prototype faces (different from the original 182 faces) were used. Of these 18 faces, 10 were used as prototypes and 8 were used for morphing. First, one of these 10 unmorphed prototype faces served as a training face, and the response of the VTU layer was recorded. Then we morph the given prototype face with the remaining 8 faces, at all 9 morph levels(the morphpos parameter), and record the response. Our recognition task tests which is closer(in Euclidean distance) to the original VTU response–the morphed response or the original response with additive Gaussian noise. This additive noise slightly corrupts the original activations to the prototype face. If the noisy, prototype face is closer, recognition occurs, and conversely if the morphed face is closer recognition fails. The mean percentage correct over all faces at a given parameter setting is the recognition performance. Faces in both testing and training phases were either both upright or both inverted, but the VTU units were always trained on upright faces.

Using this paradigm, performance on upright and inverted faces can be analyzed for different settings of the six(5 given in section 3.1.1, and morphpos) parameters which we can vary. In measuring the performance values for the six different parameters, we can attempt to understand what happens when the recognition units change their specificity. In order to keep all recognition tasks at the same difficulty, morphpos is set to 5, meaning the two faces in the recognition task are always 50% the same. Thus, as there are 10 prototype faces and 8 faces they are morphed with, there is a total of 80 trials for any one parameter setting.

3.1.4 One Parameter Changes

Before trying to understand the complexity of the 5 dimensional parameter space of our model, we must first entertain the simpler idea that changing one parameter can make the transformation between expert and novice. Perhaps multiple qualities of the units do not draw the line between experts and novices. Perceptual discrimination, or perceptual learning, improves with practice. This adjustmust of perception can be specific to the stimuli presented during training[20], indicating that practice may alter the response characteristics of cells in the visual cortex. What changes is a matter of debate, but the two obvious candidates are either the noise level or the signal strength. In particular, Lu and Dosher[5] argue that internal noise is the only factor that is altered by learning. As such, the noise level of the expert would be less than that of the novice because the expert has more training. That would be the only necessary change to the model presented here. In contrast, a study by Gold[9] tested the recognition of faces embedded in noise, and observed an increased performance over several days. Also, they measured the response consistency in the presence of different amounts of noise, and found it to be independent of magnitude of the noise. Thus, they argue that noise does not play a role, and it is the signal strength that changes in perceptual learning.

To address this, we plot the effect of moving one parameter on recognition performance. First, by looking at these plots, we can tell that changing only the noise parameter cannot change a unit from a novice to an expert. But there's a lot more here than meets the eye. These data can give us a handle as to the behavior of HMAX when parameters are changed. Starting with the plot in figure 3.4, these data suggest that one parameter changes cannot transform an expert into a novice.

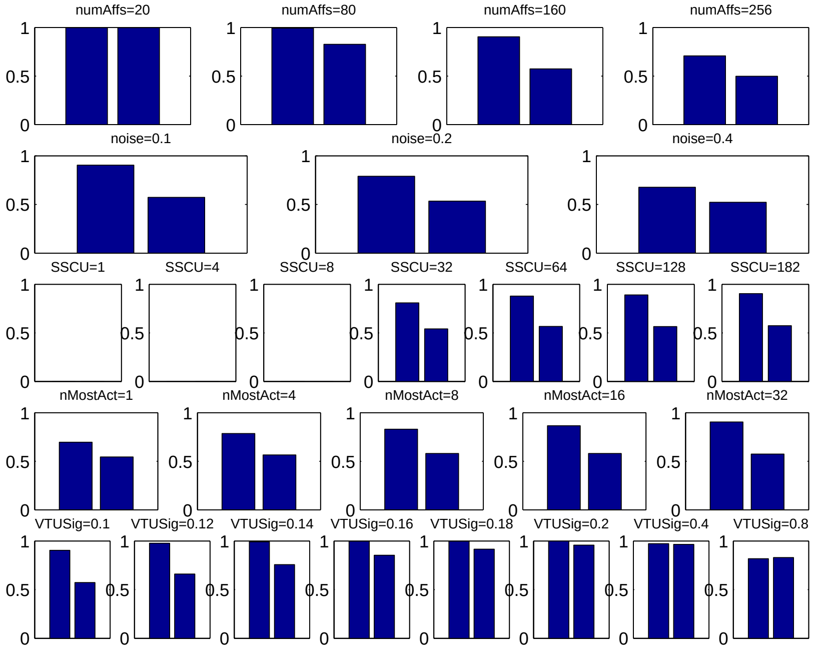

Figure 3.4: Here, we start with an expert given by parameters numAffs=160, noise=0.1, SSCU=182, nMostAct=32, and $\sigma = 0.1$. Upright performance is the left bar, inverted is the right bar. This expert's upright performance is 0.9045, while inverted performance is 0.5733. This shows what happens to both upright and inverted performance as we change one parameter. Each row moves a different parameter, keeping all others constant.

Analyzing each parameter closely, one first notices that the performance of both expert and novice decreases as more C2 afferents are added. This corresponds to the units becoming more specific, for more

C2 cells are fed into each VTU unit in this representation. With this gain in specificity comes a loss in generalization, for the VTUs are trying to match too closely to the test face, so the performance decreases. In contrast, moving along the third row, when the number of SSCU's is increased, it becomes easier to generalize across more units, resulting in a higher performance for both upright and inverted faces. This same logic works for increasing the number of nMostAct units along line 4. Also, moving toward the right on line 2 is just a result of decreasing the signal to noise ratio by increasing the noise, making the units too unspecific, and consequently decreasing performance. Initially, line 5 looks paradoxical, but that is just a product of units reaching optimum specificity, then going past it to yield a decrease in performance. Beginning at the left, the tuning is very sharp, so moving right the tuning broadens, facilitating better recognition. However, when we move right too much, σ is too high, making units tuned too loosely, so any stimulus can produce a high activation, leading to a lower performance. In figure 3.5 we can see the impossibility of changing a novice to an expert by only changing one parameter, and the trends follow the same logic as discussed above.

Another way to understand the effects of moving one parameter is by analyzing scatter plots of the data. Basically, starting with an expert and novice, now we attempt to create a novice from an expert simply by changing more than one parameter. This is a generalization of the data in figures 3.4 and 3.5. There, expert and novice differ by only one parameter, but here the expert and novice can differ by any number of parameters. Two of the most interesting scatter plot are given here in figure 3.6, and all scatter plots are given in figure A.1. The left plot of figure 3.6 is one of the many plots in figure A.1 that show that the most important two parameters are numAffs and σ. With those set, and the other three varying, performance always decreases for upright faces. Notably, performance decreases regardless of the other parameters. It is impossible to glean any information about the model from the other plots where these values are not set, as the points are not constrained to any region of the parameter space.

The subplot on the right side of figure 3.6 shows that when σ increases the performance on upright faces increases as well. Having the highest performance possible on upright faces is obviously favorable, so because a higher σ was not naturally selected, other factors are probably at play. In this analysis, analyzing the inverted performance as well is not necessary since recognition of inverted faces is not a necessary task for primates. A possible explanation is that another force governing the evolution of the structure and function of the brain is energy

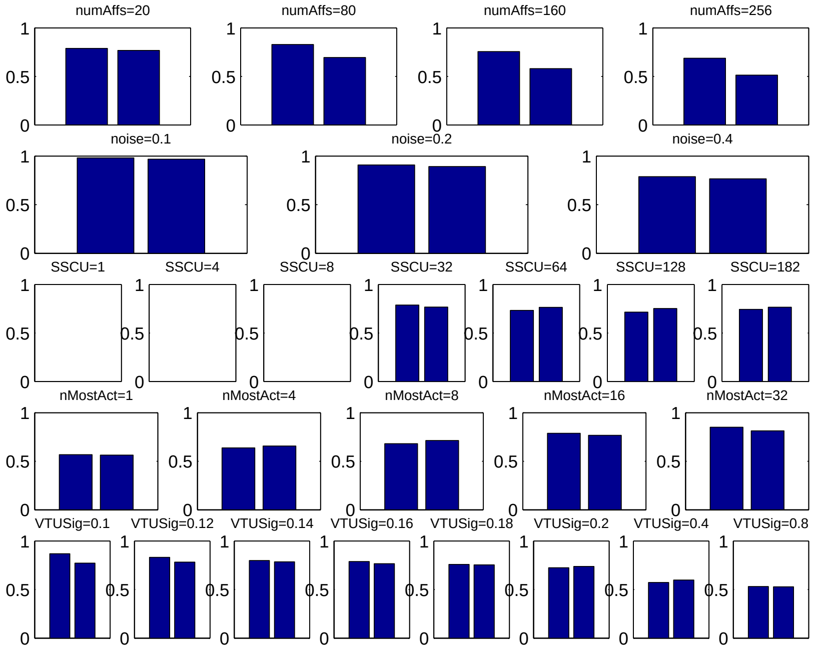

Figure 3.5: This is the same type of plot as is shown in 3.4, except here we start from a novice. The parameters for the novice are numAffs=20, noise=0.4, SSCU=32, nMostAct=16, and $\sigma = 0.16$. For this novice the upright performance is 0.7880, inverted performance is 0.7665.

efficiency [15, 1, 16]. 25% of total body glucose utilization is spent by the brain, and this energy needs to be spent economically. As such, maximizing the functionality of the visual cortex while minimizing the brain's enery usage is evolutionarily favorable. Given that, a 10% increase in performance may have been less important than a decrease in the energy usage that a lower σ provides.

3.1.5 Moving to multiple parameter space

From section 3.1.4 it is evident that in our representation, experts and novices differ by more than one parameter, so here we investigate the effects of multiple degrees of freedom. There are many possible experts and novices, so to assuage the number of combinatorial possibilities, one expert is chosen based on the fMRI data[14] and the data from Carey and Diamond's Experiment[4]. The expert that we choose is the

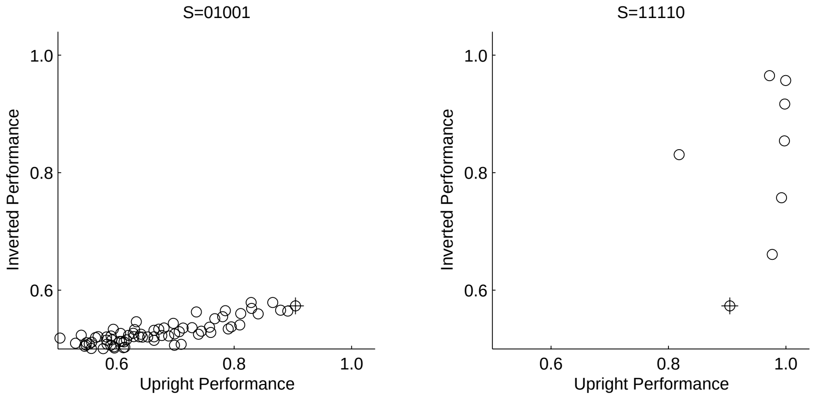

Figure 3.6: Scatter plot of moving to Novice from Expert via moving different parameters. The expert's parameter settings are numAffs=160, noise=0.1, SSCU=182, nMostAct=32, $\sigma = 0.1$. The original expert is marked with a cross, all other permutations are circles. The "S value" in the title of each subplot represents which variables are kept at the expert setting(set to 1), and which are unconstrained(set to 0). From least significant to most significant bit, the variables in S are σ, SSCU, nMostAct, numAffs, and noise. The plot on the left constrains σ and numAffs to the expert value, while the plot on the right constrains everything but σ.

same as used in figure 3.4. From this expert, possible novices can be analyzed from histograms and scatter plots as above, and from these plots we can deduce the sufficient and necessary qualities required for a novice.

Again, it is a difficult task to probe this parameter space. Our method is to control one parameter, letting all others range over all possible values. By examining the trends in these data, it is possible to make inferences as to which parameters are important as to transform as expert into a novice, and which directions the given parameters might move as to undergo this transformation. The qualities that we are looking for in a novice as observed by Carey and Diamond are a small inversion effect, and performance of the expert being greater than that of the novice on upright faces. To visualize this in our 5 parameter space, the only restriction that we place is for performance of expert on upright faces must be higher than that of novice on upright faces, or in other words $perf_{novice}(upright) < 0.9045$.

Displayed here in figures 3.7 and 3.8 are scatter plots of upright and inverted performance. For each subplot, each cross represents one unit, with upright and inverted performance on the x and y axes, respectively. For simplicity, units were separated by upright performance, and only units that have an upright performance between 75% and 80% are plotted here. The rest of the units and their scatter plots are displayed in appendix A.2. Because upright performance is restricted in this plot, all of the crosses lie in a neat vertical pattern. The interesting thing to look at here is the vertical location and clustering of these hatch marks.

As observed in section 3.1.4, the two key parameters in tuning these units are σ and numAffs. Where Noise, SSCU, or nMostAct parameters remain constant(figures A.3, A.4, and A.5, respectively), the distribution of the inverted performance remains the same. This is not the case with numAffs and σ, as is displayed in figures 3.7 and 3.8. As those two parameters get smaller and larger, respectively, the variance of the inverted performance decreases. Even more important, with respect to these two parameters, the inverted performance grows, thus decreasing the inversion effect. It appears that in order to raise inverted performance while lowering upright performance, it is sufficient to do a combination of raising σ and lowering numAffs. This is equivalent to lowering the specificity by decreasing the number and broadening the tuning of the individual units. Complete plots for all values of upright performance for numAffs and σ are given in appendix A.2 and A.6.

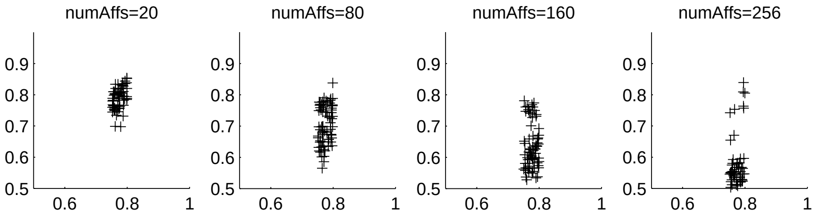

Figure 3.7: A scatter plot of upright versus inverted performance. As described in section 3.1.5, the only restriction on the representation for each of the subplots is that numAffs is fixed. All of these cells have an upright performance between 75% and 80%.

It is even easier to visualize these trends in terms of inversion effect, as is done in figures 3.9 and 3.10. The plots depict a histogram of the inversion effect as numAffs or as σ changes, letting all other parameters run free. Note that the "hump" of these plots decreases

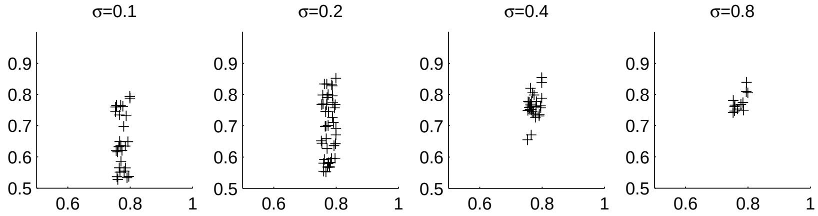

Figure 3.8: The same type of scatter plot as in 3.7, except σ is fixed here.

when numAffs in decreased or when σ is increased, no matter how the other parameters are set. The hump of the histogram is equivalent to the clustering of data points in the previous scatter plot. These shifts in performance suggest that these two parameters have the power to distinguish an expert and a novice, as the hump moving towards zero indicates that the inversion effect decreases. Additionally, the specificity of the model is most strongly affected by these two parameters working in synergy. As these histograms show, a novice can be created by changing the parameters to decrease the expert's specificity. For example, in the representation given by numAffs=20, noise=0.4, SSCU=32, nMostAct=16, and σ=0.16, upright performance is 78.9%, and novice performance is 76.7%.

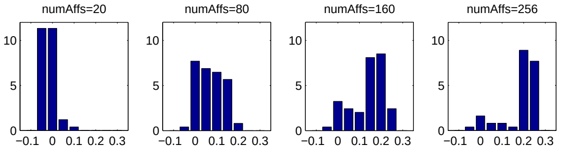

Figure 3.9: A sister plot to figure 3.7, the same data is plotted here as a histogram of the obtained inversion effect.

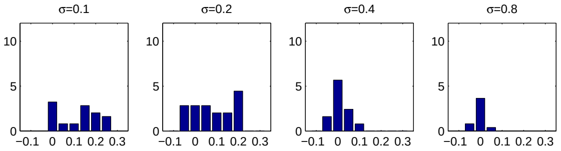

Figure 3.10: As in figure 3.8, σ is fixed.

3.1.6 Using Cars As Stimuli

In addition to the simulations using faces, we also ran identical simulations using cars as stimuli as an alternative object class. This is to show that the inversion effect is not inherent to faces, but rather applies to any class of objects. The expertise effect is a characteristic of the settings of the parameters within the model. It is not a quantitative effect due to features of a given object class or an object class dependent computational mechanism. Thus, the observed performance is dependent on the tuning and hence the parameter settings of the units, not on the type of stimuli.

A description of the cars used is given in section 2. Using these cars as stimuli, with morphed cars always 50% the same as the original prototype car, we observe the same trends as with faces. Increasing numAffs, increasing the number of SSCU's, and increasing the nMostAct parameter increase specificity, while increasing σ and the amount of noise make units less specific. With this in mind, we can construct an expert, having many, finely tuned units, and a novice, having fewer, widely tuned units. Of course the actual values compared with faces will be different because this is a different object class, but the effect turns out to be the same. Setting the parameters of an expert unit to numAffs=160, noise=0.2, numSSCU=13, nMostAct=13, $\sigma = 0.1$, yields upright and inverted performances of 0.9550 and 0.6957 respectively. Turning this expert into a novice, numAffs is changed to 20, and σ is changed to 0.13. With these modifications, upright performance is 0.9039, but inverted performance is 0.8914. Again, the decreased sensitivity of the novice enabled it to recognize inverted faces at a far higher performance level than the highly specific expert. Although cars and faces differ in their relational features, experts of both object classes share the same large inversion effect because of the tightly tuned units that they have in common.

3.1.7 Simulations Using More Realistic Patches

A possible critique of this model is that the features that are used are not realistic. That is, on the lowest level cells should be tuned to bars[11], but on higher levels there probably should be some sort of feature learning involved in the system. Perhaps the behavior that we observe is an artifact of this unrealistic model that has been used in the simulations. If this were true, the results obtained could not be compared with parallel psychophysics experiments. With this in mind, we want to see if the results obtained using the standard HMAX model also hold for more realistic features, so we alter the standard HMAX model to support feature learning at the S2 level. As shown by Serre et al[23], these features increase the sensitivity of the C2 units to faces as oppose to nonface objects. Consequently, this facilitates face recognition in cluttered scenes involving myriad nonface distractors, as oppose to the poor performance of the standard HMAX model. These new S2 targets are described in section 2 and examples of the new filters are shown in figure 2.2.

The experiments from section 3.1.3 were duplicated using the new S2 targets. The performance on both upright and inverted faces was comprable to the original data. Increases and decreases in specificity follow the same trends as outlined in 3.1.4, and the performance curves have the same shape for both upright and inverted faces. The shapes of these performance curves are given in appendices B.1 and B.2 for standard and realistic HMAX, respectively. These data show that the performance on faces is independent of S2 target type, and that the effect that we observe is not limited to the feature set of the standard HMAX model.

Chapter 4

Transfer of Learning Using a View-Based System

4.1 Background

In the real world, recognition of objects is not limited to two dimensions. Real objects can be rotated in any direction, and in a recognition task the original, unrotated object is compared with the same object rotated in an arbitrary fashion. In two dimensions, individual features can be compared, and on that basis an individual can discriminate between the two objects. Under three dimensional rotation, an architecture such as the one given in sections 2 and 3.1.1 fails because individual features are not invariant under rotation. If an object is rotated, these changing features yield a different activation pattern than the original object, making any comparison between the two images meaningless.

Two major schools of thought attempt to compare features that have been rotated, despite the changes that happen under rotation. The first is that of a structural representation, where a 3D model of the object in question is built. One example of this sort of structural representation is Biederman's recognition by components (RBC) theory[2], which states that any object can be decomposed into small, discrete parts called geons. Under rotation, the viewer trains on a single view, and mentally rotates the geons and their relative positions in a recognition task. This predicts viewpoint invariant object recognition as long as the object views can be used to obtain a structural representation. Although this representation may seem appealing, it is extremely hard to implement this computationally, or even imagine how it might be done. Biederman's representation is critiqued by Tarr and Bulthoff[25], who find that Biederman's conditions for viewpoint invariance lack the generality to characterize a wide range of recognition phenomena. Also, the massive amount of information involving relative locations of features and the computations involved in rotating these features is unfeasible.

Instead, Tarr and Bulthoff favor an exemplar-based multiple-views mechanism as the vital component of recognition. Rather than store all information about a scene, a sparse scheme that learns from a small set of perspective views is far more efficient. Note that there is no 3D model of the object, but rather novel views are recognized by interpolation between a small number of stored views. From this small set of views, Poggio and Edelman[19] showed that a function can be constructed that can map any view to a standard view. Using this mechanism, a network constructed in this way can recognize the training object from any viewpoint. Their paper showed that such a mapping was theoretically possible, and in 1995, Vetter et al[29] described a view-based model of object recognition. They outlined an RBF network and described how the model could be used to learn a face. Soon after, Edelman[6] implemented an RBF model of this sort and showed that under rotation similar objects yielded similar activations.

The above studies laid the groundwork for simulations that use a view-based model for a recognition task involving rotated faces. In chapter 3, we showed that we could train the network with faces, and recognize novel faces at the same orientation. This succeeds through a direct comparison of features from the training and test faces. With rotation, the features are no longer the same in the training and test phases. When novel views with which the system is not familiar are input, the model is forced to interpolate from the trained views.

In our simulations we exploit this view-based representation, tuning VTUs to different views of the same face. We then test the model using faces that are rotated at novel views, or at different angles than those used in training. We show that learning transfers over the shape transformations, indicating that generalization across all views is possible given only a small number of training views. This type of learning is crucial in tasks where the face changes slightly, such as in the case of changing facial expressions. These simulations show that it is unnecessary to store all possible configurations of a face in order to account for every possible featural change. Rather, it is sufficient to store a small, representative number of configurations of a given face from which other, unstored, configurations of features can be extrapolated.

4.2 Simulations

4.2.1 A New Representation

As described in sections 2 and 3.1.1 the representation given in the standard HMAX model is only useful in comparing features that are shown at the same orientation. Any hierarchical level of that representation will have very different activations when stimulated with the same face at varying angles. This is a consequence of individual features changing under rotation. The activity of a cell that is highly responsive to a feature oriented at a frontal view will be less active when the stimulus is rotated. Because the activity varies under rotation, object recognition becomes an impossible task if this representation is used. It becomes necessary to augment the representation of the standard HMAX model to respond invariantly under rotation to the same face.

With respect to the HMAX model, implementing this view-based model entails training VTUs on multiple views of the same object; a detailed explanation follows below. First, we store the responses of a set of VTUs to a given face rotated at a set of angles v. Then the responses of these VTUs v to the same face at a set of training views t is recorded. The parameters of these VTUs can be varied as explained in section 3.1.1, such as numAffs or σ. The matrix A contains these responses, with entry row i, column j containing the response of VTU v_j to training view t_i. Then we solve the matrix inversion problem for $Ab = w$, given that $w = 1$. This vector b is known as the VIU(view invariant unit) weight vector. This new implementation calculates VTU responses as before, but now the VTUs that are tuned to the same face, rotated are multiplied by their corrresponding VIU weight vector. In theory, this yields an equal response toward all views. This architecture is shown in figure 4.1.

When tuning view invariant units, we must be very careful when setting out parameters. On the one hand, it is desirable to make σ extremely low, while increasing the number of afferents as much as possible. This yields units that are very specific, facilitating object recognition and hence increasing performance. When these quantities are moved as to yield a specificity that is high enough to enable recognition, the matrix A begins to have entries that are close to 0, rendering it almost singular. Consequently b may have entries that are very large. If b has large entries the goal of generalizing from the

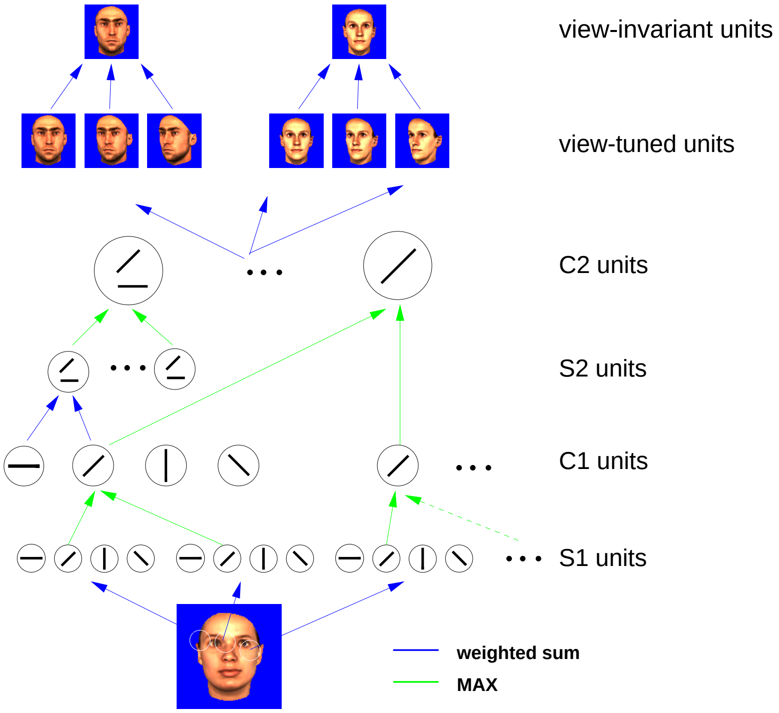

Figure 4.1: The HMAX model with a level of view invariant units.

training views to novel views becomes impossible because of the erratic entries in this matrix. To counteract that it is necessary to increase the number of views used for any one VIU. Moving the views closer together makes the fall-off from 1 on the diagonal of A less steep, and thus b will be better behaved. For both v and t we used angles of $\{0, -5, -10, -15, -25, -35, -40, -45\}$. There were 95 faces that we had for VTUs, resulting in a total of $95 \times 8 = 760$ VTUs, 95 8×1 b vectors, and 95 VIUs.

4.2.2 Recognition Task and Experiment

An experiment done by Moses et al[18] is a good basis for our recognition experiment. Additionally, we can make predictions based on their results. They explore the capacity of the human visual system to generalize the recognition of faces to novel images, and investigate the level at which this generalization occurs. Their approach compares the recognition performance between rotated faces that are either upright or inverted. In their experiment, they train the viewer on one of three faces in a set. They then test the subject on the identity of any of the faces in the set, at varying angles. They observe that subjects generalize easily for upright faces but not for inverted ones, meaning performance was high for upright faces regardless of viewing angle, but when viewing angle changes performance on inverted faces decreased.

These results support the view-based model theory. Using a view-based model, generalization to novel views for upright faces occurs through interpolation of the stored views. On the other hand, interpolation is not possible for inverted faces. The stored, upright faces have too little in common with the inverted test faces for any logical comparisons to be made. We agree with the view-based representation, so we simulate an identical experiment in the HMAX model to show that these results follow given the view-based representation.

For our recognition task, 6 different faces morphed at 9 different degrees were used. Each of these faces was rotated at angles of $\{0, -7.5, -15, -22.5, -30, -37.5, -45\}$. These provided novel angles and faces on which the VTUs and VIUs had not been trained. The recognition paradigm is similar to that mentioned in section 3.1.3, but there are slight differences. We train on a frontal view of a given face. The recognition task is a 2-way forced choice between two faces rotated by the same amount. One is a morphed version of the original test face, and the other is the original itself. An example of a face rotated at different angles that we use for stimuli is given in 4.2. As in chapter 3, if the activation of the original, rotated face is closer, recognition is correct, and conversely, if the morphed face is closer recognition fails.

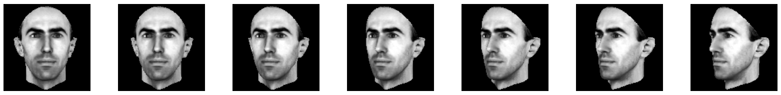

Figure 4.2: Rotated faces used

The results of this recognition experiment are plotted in figures 4.3 and 4.4. The performance is that of the 760 VTUs and the 95 VIUs. The first conclusion we draw involves a comparison of VTU performance with VIU performance. The plots show that VIU units give a clear advantage over VTU units in both the upright and inverted case. This shows that the VIU level has a positive influence on recognition. The low performance of the VTU units on the rotated faces shows that additional computation besides VTUs is necessary to process rotated faces.

Next, assuming that the brain uses VIU like units for object recognition under rotation, we can draw a parallel between the performance of VIU units and the Moses[18] experiment. In comparing the VIUs on upright and inverted faces, we see a similar result as the one they obtained. As expected, the VIUs, a view-based network, have high performance on upright faces, but a lower performance on inverted faces. For upright faces, the VIUs can learn the behavior of features under rotation. Thus, when it is rotated to a novel view, the VIUs are still capable of recognizing a given face by interpolating from the known views. Although the VIUs are capable of generalizing across different views for upright faces, this same sort of generalization for inverted faces is impossible. The features of an inverted face change differently under rotation, resulting in very low recognition performance.

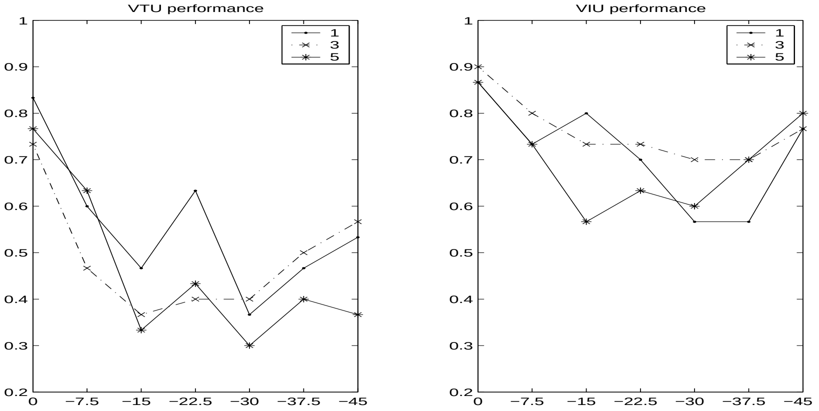

Figure 4.3: Performance of VTUs(left) and VIUs(right) on rotated, upright faces. The legend indicates the morphpos of the test face. A higher morphpos means the test face is further away from the training face, indicating a harder recognition task, and hence lower performance. The parameters are the same as those of the expert in chapter 3, σ=0.1, numAffs=160, nMostAct=32, and noise=0.1. All available VTUs and VIUs are used, so there is no SSCU parameter.

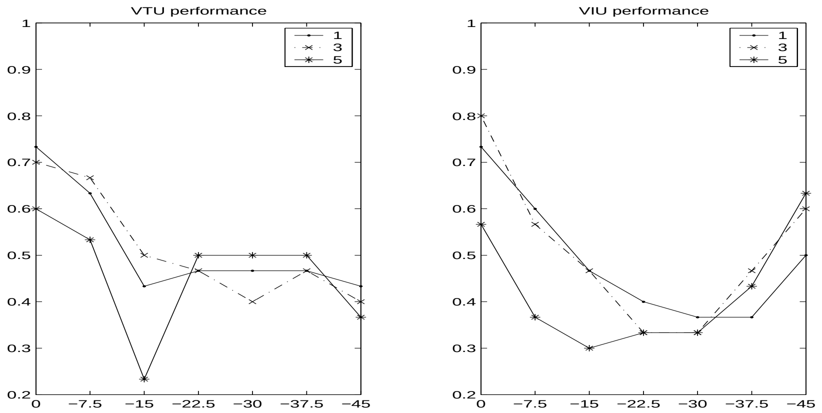

Figure 4.4: Performance of VTUs(left) and VIUs(right) on rotated, inverted faces. The legend is the same as in figure 4.3.

Chapter 5

Discussion

5.1 Are Faces Special?

In chapter 3 we investigated whether faces are special. The model that we used for simulations was entirely shape based and has no internal understanding of faces. Rather the model searches for the presence of shapes in the training stimuli. By changing the parameters of the model, we were able to vary the sensitivity to these shapes. These changes in sensitivity control the performance of the model on upright and inverted faces, generating inversion effects of varying magnitudes.

Hence, it's sufficient to vary the parameters to produce a change between novice and expert. While the parameters change, the computational mechanism remains the same. Thus, the inversion effect due to experience is a quantitative change in the representation rather than a qualitative one. In other words, the components and interconnections of the model do not change, but the strength of the connections and the number of units at different levels can change. To show that this is not unique to faces we also ran the same simulations with cars and obtained the same results. Concisely, faces are no different than any other object class. The reason they were thought of as special is because of the highly specific units characteristic to faces in people.

5.2 Learning and Representation

Chapter 4 is a generalization of the work in chapter 3. In that section, VTUs that are tuned to prototype faces provide the model with a general idea of the features normally present in a face. Then the model makes a direct comparison of the features in the training face to those in the test face. In the best case, if the training and test faces were the same, the activations would be identical. This is obviously not the case under rotation, as the features in the face change when a face is rotated.

We showed that a recognition of a face oriented at a novel view is feasible if the model is tuned to prototype faces rotated at multiple views. Edelman[6] proved that this aatchitecture could generate similar activations for rotated faces, and this takes that result further by showing that recognition is possible. Even though the VIUs have no internal information about the behavior of features at a novel view, they are able to intimate the similarity of test and training faces. This sort of generalization probably would not work with objects that have very different 3D structure, such as landscapes. With that type of object class, the model would not be able to deduce the featural changes that occur at an untrained view. Using the performance of our view-based model on upright and inverted faces, we give credence to the hypothesis of Moses et al[18]. They think that the generalization in face recognition occurs at a class based level, and at this level upright and inverted faces are members of two distinct object classes. This is a product of class-level processing, as our VIUs had far lower performance on inverted faces because of the different ways in which features of upright and inverted faces change under rotation.

5.3 Future Work

The simulations support our claim that a large face inversion effect is due to expertise. The next logical step involves duplicating the same type of simulations in psychophysics experiments. We hope to create experts and novices in a novel object class by varying the training in this object class between two sets of people. Our simulations predict that we can control the inversion effect for this object class by varying the amount of training. The low σ and high number of afferents in the expert translates to a large amount of training. More training produces an increase in experience and hence tighter tuning, which should result in a large inversion effect.

Appendix A

Supplemental Figures

A.1 Scatter Plots

Each row of the plots restricts the scatter plots of the upright performance. That is, the four rows separate upright performance of the novice to be between 70-75%, 75-80%, 80-85%, and 85-90.45%.

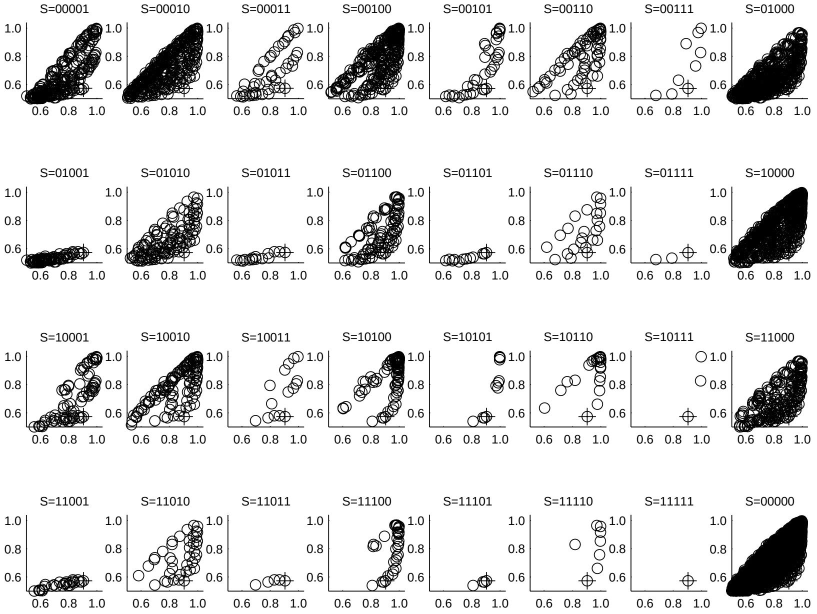

Figure A.1: Scatter plot of moving to novice from expert via moving different parameters. The expert's parameter settings are numAffs=160, noise=0.1, SSCU=182, nMostAct=32, σ = 0.1. The original expert is marked with a cross, all other permutations are circles. The x-axis is upright performance, the y-axis the inverted performance. The "S value" is as in figure 3.6. There are 32 plots here, corresponding to the 2^5 ways in which S can be set.

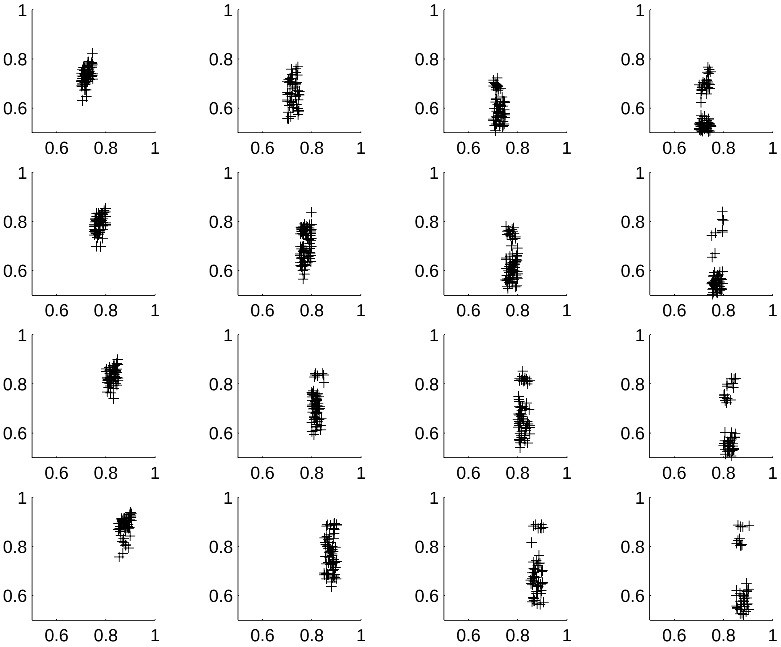

Figure A.2: Afferents

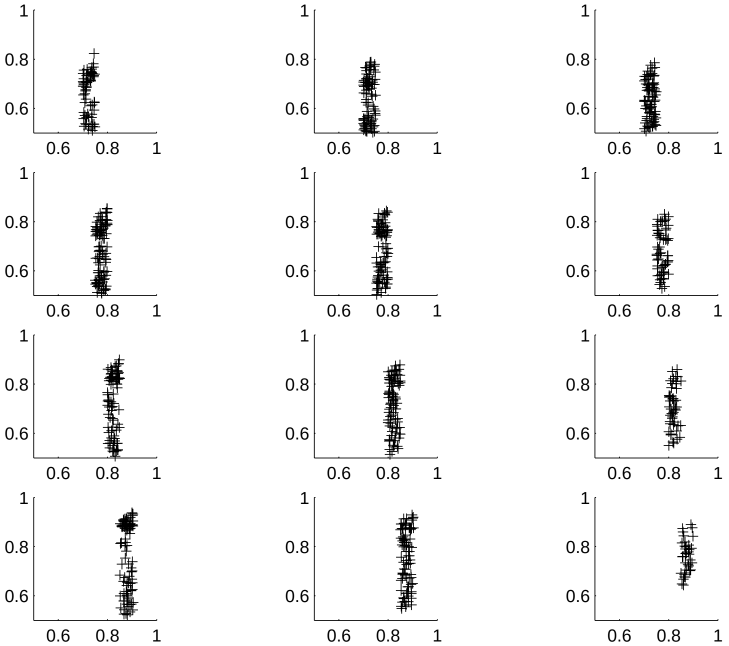

Figure A.3: Noise

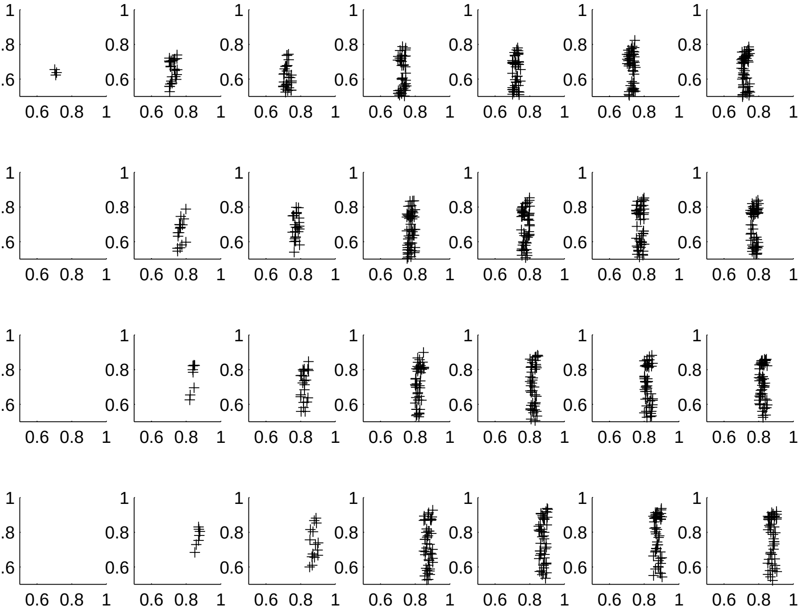

Figure A.4: SSCU

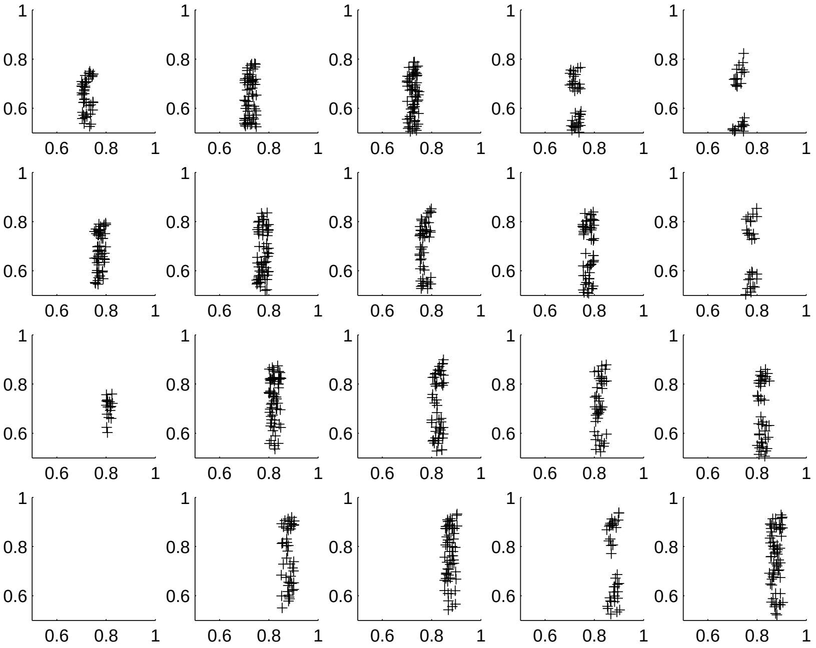

Figure A.5: NMost

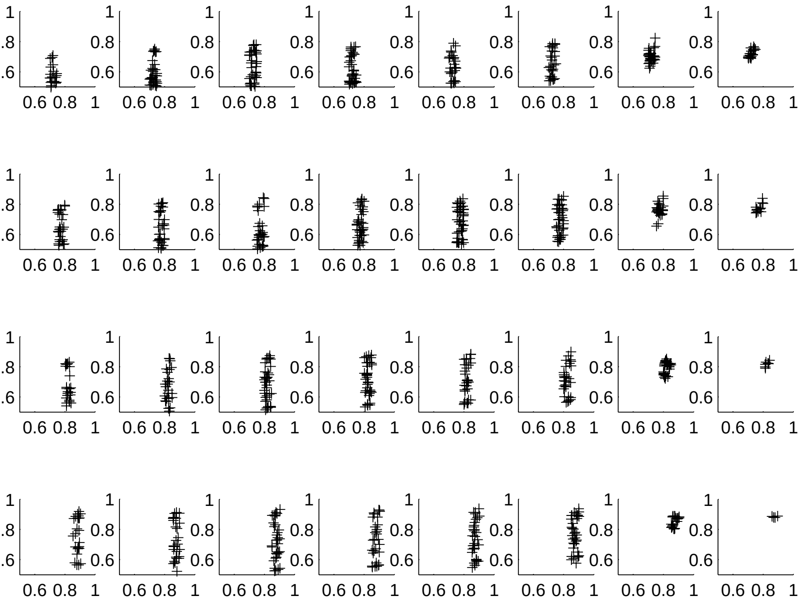

Figure A.6: Sigma

A.2 Histograms

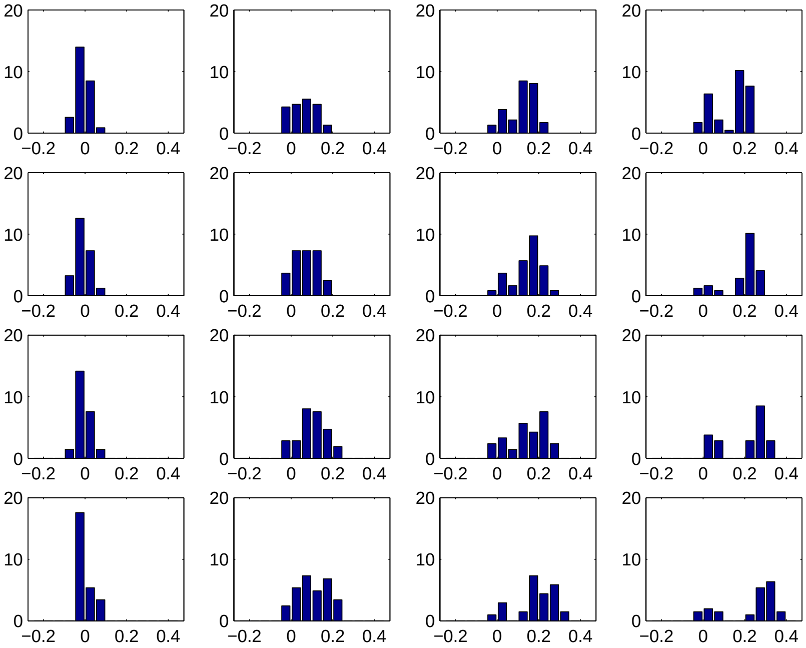

Figure A.7: Afferents, Inversion Effect

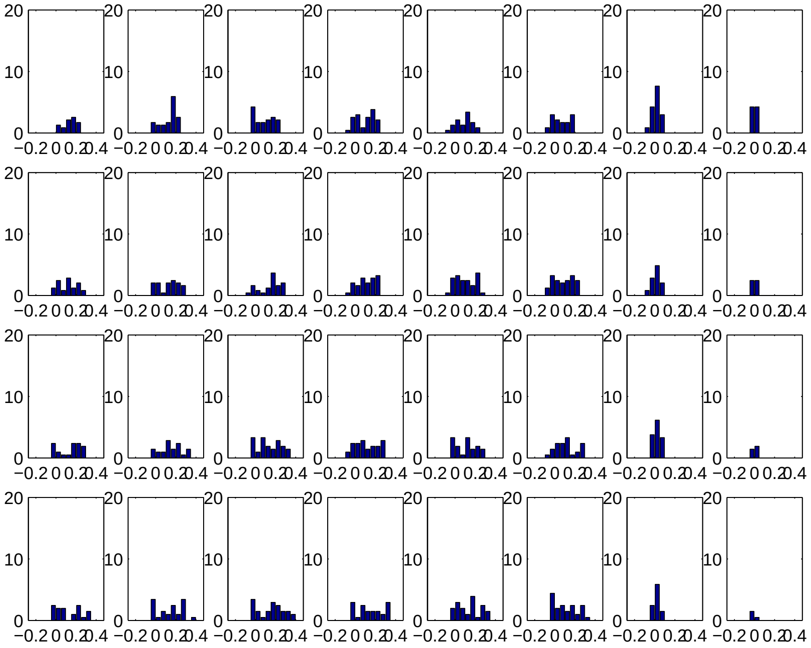

Figure A.8: Sigma, Inversion Effect

Appendix B

Performance Curves

In each figure, noise is increased as one moves vertically down subplots on values 0.1, 0.2, and 0.4. Within each subplot, the different lines are different settings of nMostAct, which are given in the legend. Moving along the x-axis varies the number of afferents. In order to display a digestable number of subplots, we only plot data for σ=0.1 and σ=0.4.

B.1 Regular HMAX

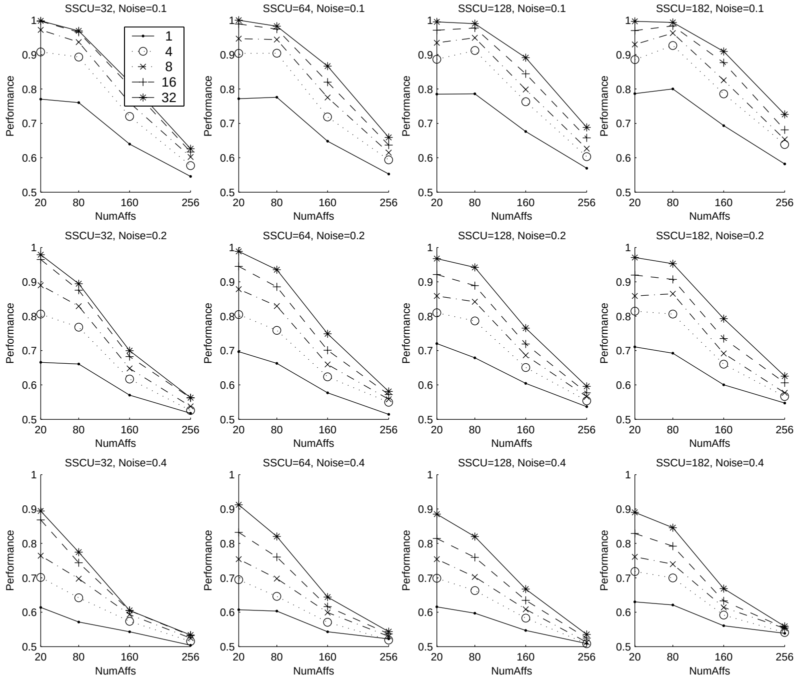

Figure B.1: HMAX on upright faces. Sigma = 0.1

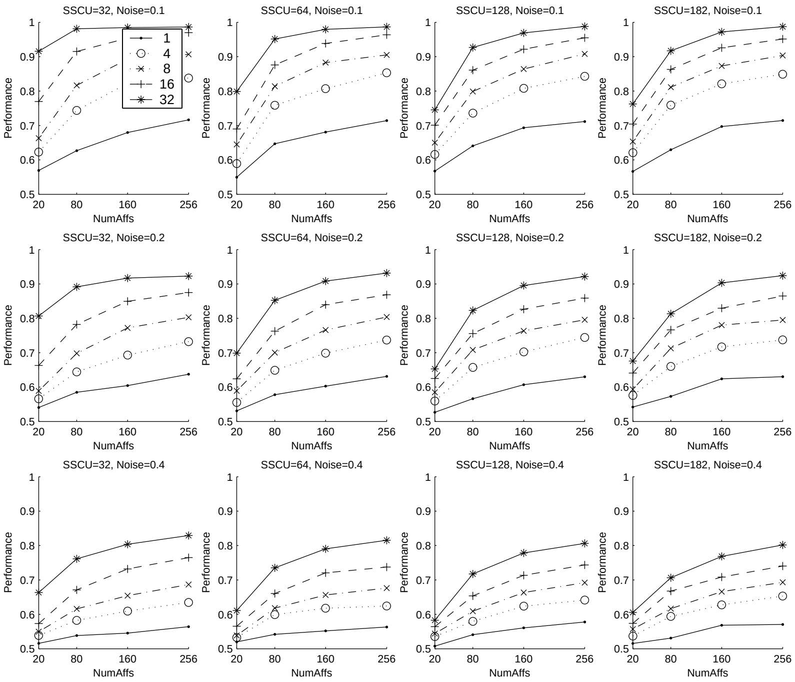

Figure B.2: HMAX on upright faces. Sigma = 0.4

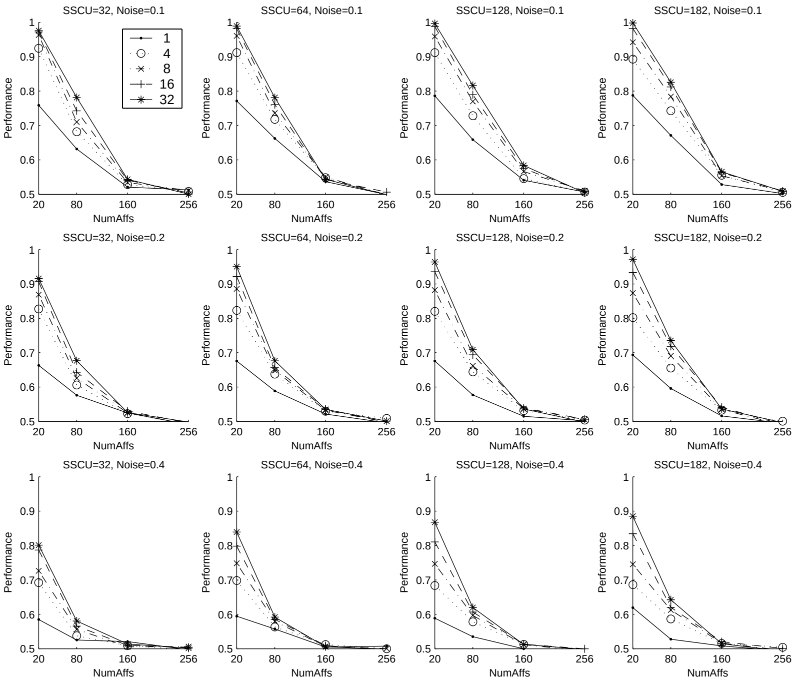

Figure B.3: HMAX on inverted faces. Sigma = 0.1

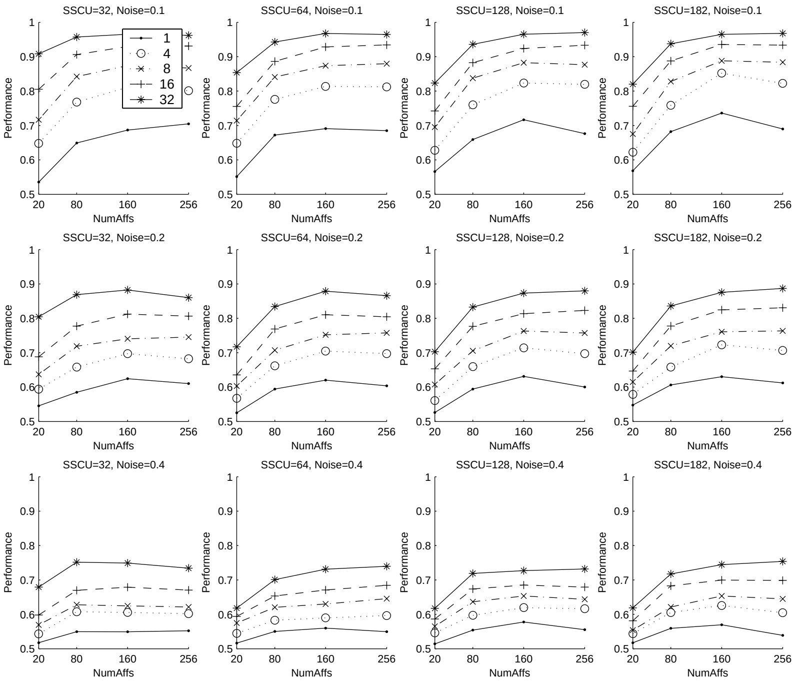

Figure B.4: HMAX on inverted faces. Sigma = 0.4

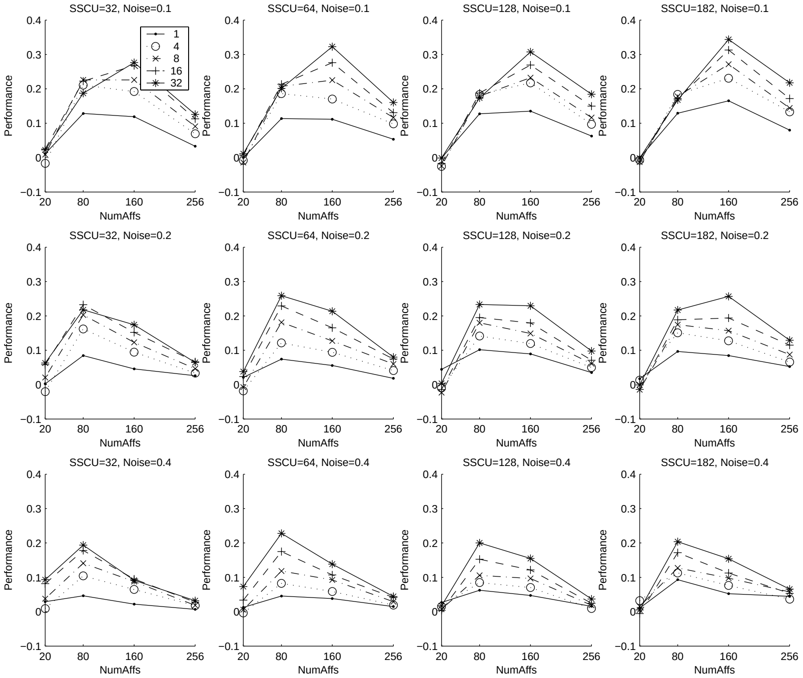

Figure B.5: Difference of Upright and Inverted Performance of HMAX on Faces, Sigma = 0.1

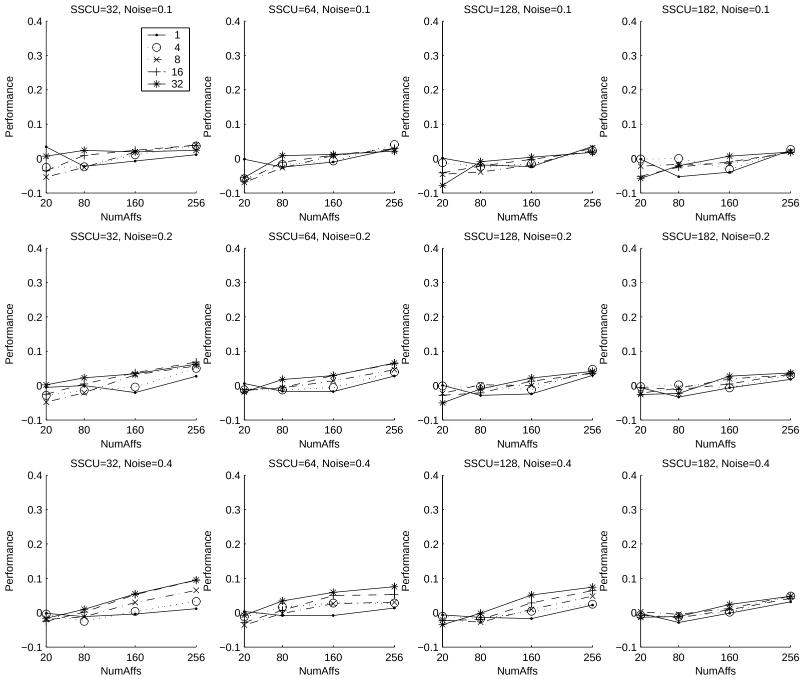

Figure B.6: Difference of Upright and Inverted Performance of HMAX on Faces, Sigma = 0.4

B.2 HMAX With Realistic Features

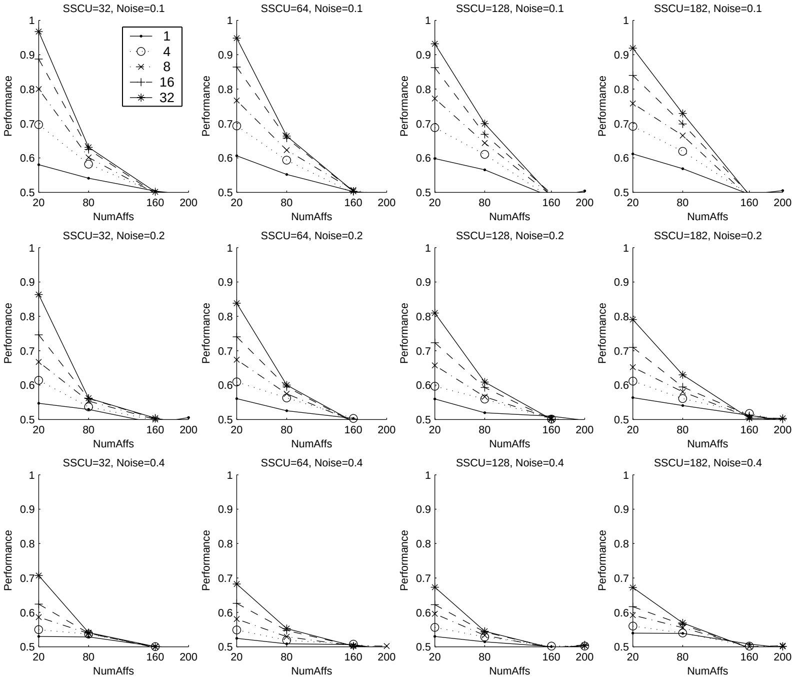

Figure B.7: Realistic HMAX on upright faces. Sigma = 0.1

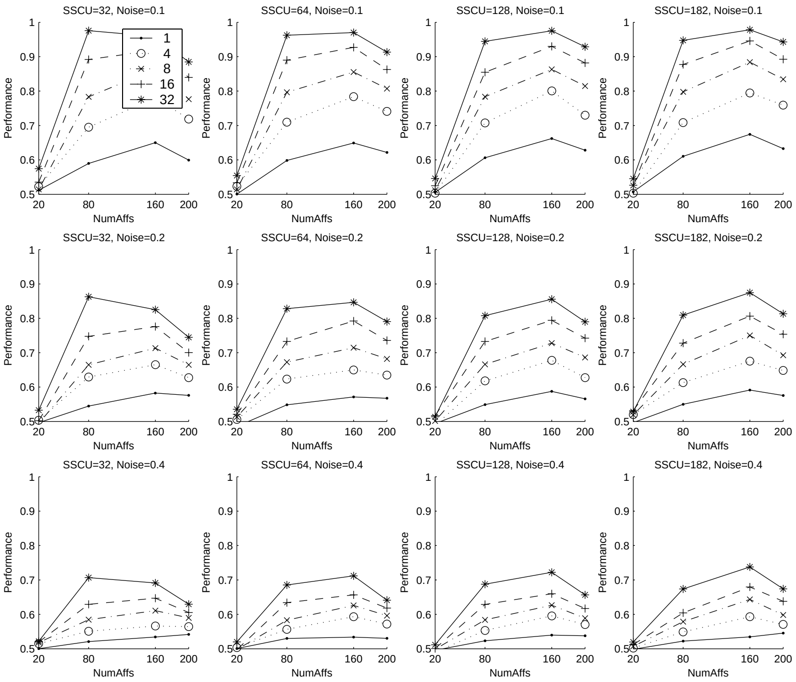

Figure B.8: Realistic HMAX on upright faces. Sigma = 0.4

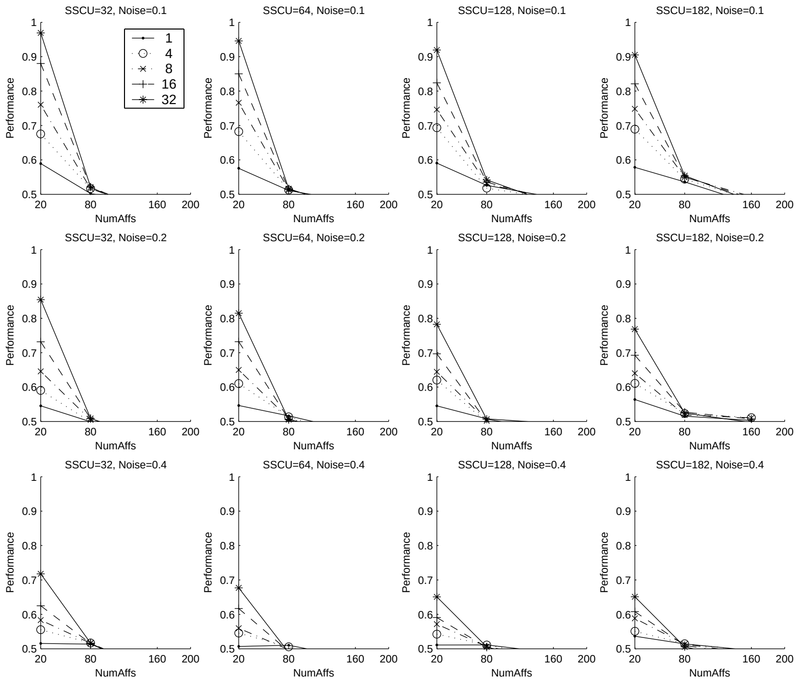

Figure B.9: Realistic HMAX on inverted faces. Sigma = 0.1

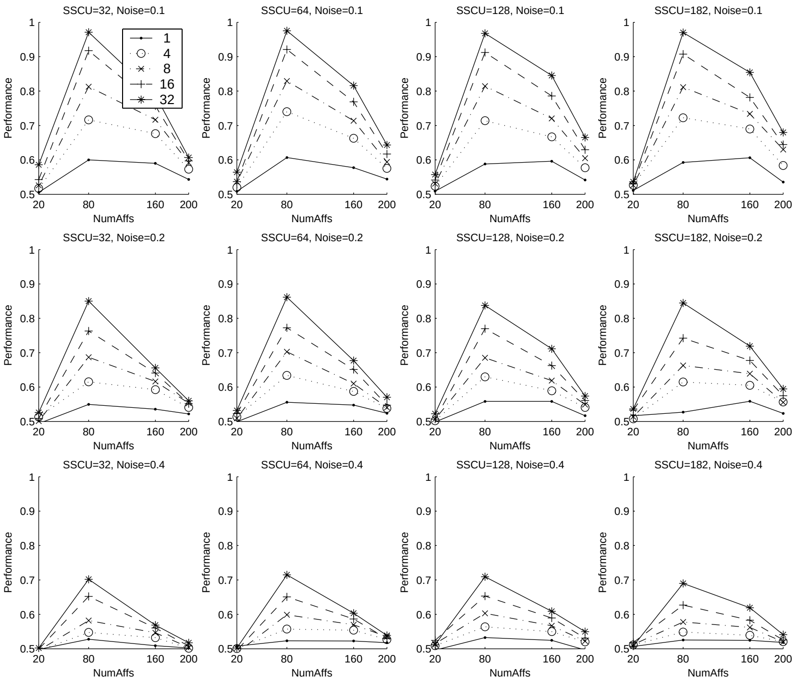

Figure B.10: Realistic HMAX on inverted faces. Sigma = 0.4

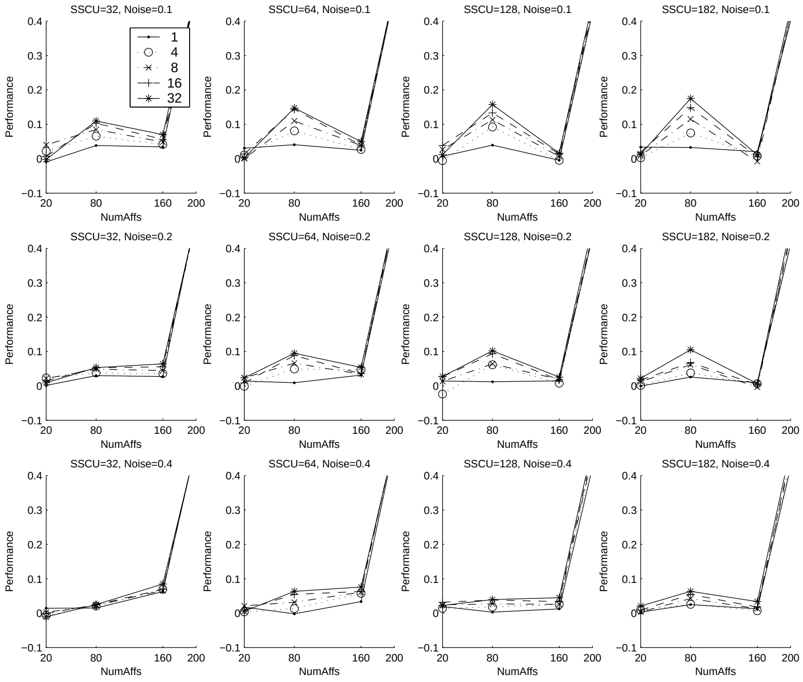

Figure B.11: Difference of Upright and Inverted Performance of Realistic HMAX on Faces, Sigma = 0.1

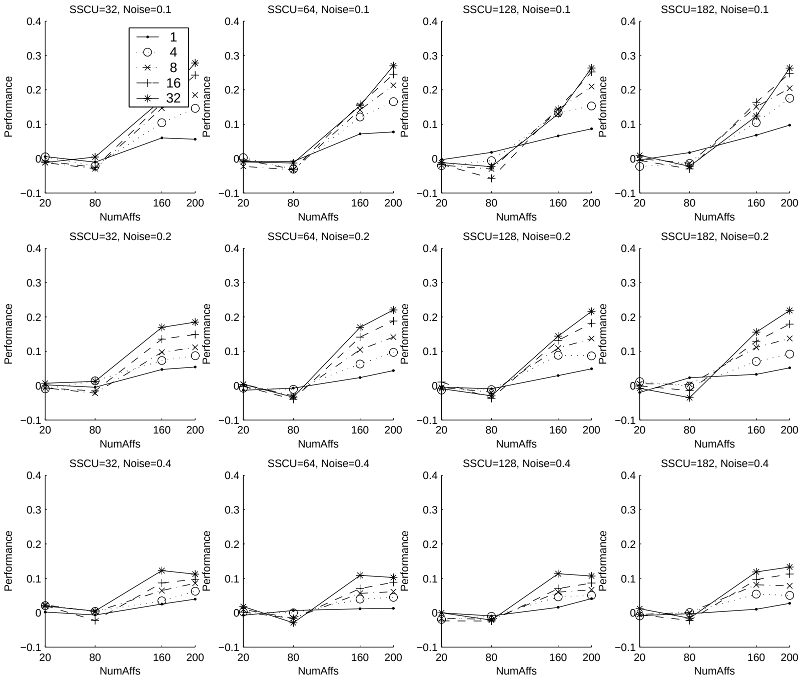

Figure B.12: Difference of Upright and Inverted Performance of Realistic HMAX on Faces, Sigma = 0.4

Bibliography

[1] V. Balasubramanium, D. Kimber, and Michael J. Berry II. Metabolically efficient information processing. *Neural Comp.*, pages 799–815, 2001.

[2] I. Biederman and P.C. Gerhardstein. Recognizing depth-rotated objects: Evidence and conditions for three-dimensional viewpoint invariance. *J. Exp. Psych.: Hum. Percept. Perf.*, pages 1162–1182, 1993.

[3] S. Carey and R. Diamond. From piecemeal to configurational representation of faces. *Science*, 195:312–314, 1977.

[4] R. Diamond and C. Carey. Why faces are and are not special: an effect of expertise. *J. Exp. Psych.:General*, 115:107–117, 1986.

[5] B.A. Dosher and Z.L. Lu. Perceptual learning reflects external noise filtering and internal noise reduction through channel reweighting. *Proc. Nat. Acad. Sci. USA*, 95:13988–13993, 1998.

[6] Shimon Edelman. *Representation and Recognition in Vision*. MIT Press, 1999.

[7] M.J. Farah. Is face recognition 'special'? Evidence from neuropsychology. *Behavioral Brain Research*, 76:181–189, 1996.

[8] I. Gauthier and N. K. Logothetis. Is face recognition not so unique after all? *Cog. Neuropsych.*, 17:125–142, 2000.

[9] J. Gold, P. J. Bennett, and A. B. Sekuler. Signal but not noise changes with perceptual learning. *Nature*, 402:176–178, 1999.

[10] http://www.ai.mit.edu/projects/cbcl/hmax.

[11] D. Hubel and T. Wiesel. Receptive fields, binocular interaction and functional architecture in the cat's visual cortex. *J. Phys.*, 160:106–154, 1962.

[12] A. Ishai, L. Ungerleider, A. Martin, J.M. Maisog, and J.V. Haxby. fMRI reveals differential activation in the ventral object vision pathway during the perception of faces, houses, and chairs. *Neuroimage*, 5:S149, 1997.

[13] N. Kanwisher, J. McDermott, and M.M. Chun. The fusiform face area: a module in human extrastriate cortex specialized for face perception. *J. Neurosci.*, 17:4302–4311, 1997.

[14] N. Kanwisher, F. Tong, and K. Nakayama. The effect of face inversion on the human fusiform face area. *Cognition*, 68:B1–B11, 1998.

[15] S.B. Laughlin, R.R. de Ruyter can Stevenick, and J.C. Anderson. The metabolic cost of neural information. *Nat. Neurosci.*, pages 36–41, 1998.

[16] W.B. Levy and R.A. Baxter. Energy efficient neural codes. *Neural Comp.*, pages 531–43, 1996.

[17] G. McCarthy, A. Puce, J.C. Gore, and T. Allison. Face-specific processing in the human fusiform gyrus. *J. Cogn. Neurosci.*, 9:604–609, 1997.

[18] Y. Moses, S. Ullman, and S. Edelman. Generalization to novel images in upright and inverted faces. *Weizmann Institute CS-TR 93-14*, 1994.

[19] T. Poggio and S. Edelman. A network that learns to recognize three-dimensional objects. *Nature*, 343:263–266, 1990.

[20] T. Poggio, M. Fahle, and S. Edelman. Fast perceptual learning in visual hyperacuity. *Science*, 256:1018–1021, 1992.

[21] M. Riesenhuber and T. Poggio. Hierarchical models of object recognition in cortex. *Nat. Neurosci.*, 2:1019–1025, 1999.

[22] M. Riesenhuber and T. Poggio. Models of object recognition. *Nat. Neurosci.*, 3:1199–1204, 2000.

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

[24] C.R. Shelton. Three-dimensional correspondence. Master's thesis, Massachusetts Institute of Technology, 1998.

[25] M.J. Tarr and H.H. Bulthoff. Is human object recognition better described by geon structural descriptions or by multiple views? *J. Exp. Psych.: Hum. Percept. Perf.*, pages 1494–505, 1995.

[26] T. Valentine. *Encoding processes in face recognition*. PhD thesis, University of Nottingham, 1986.

[27] T. Valentine and V. Bruce. The effect of race, inversion, and encoding activity upon face recognition. *Acta Psychologica*, 61:259–273, 1986.

[28] T. Vetter and V. Blanz. A morphable model for the synthesis of 3D faces. In *Proceedings of SIGGRAPH'99*, pages 187–194, 1999.

[29] T. Vetter, A. Hurlbert, and T. Poggio. View-based models of 3D object recognition: invariance to imaging transformations. *Cereb. Cortex*, 5:261–269, 1995.

[30] R. K. Yin. Looking at upside-down faces. *J. Exp. Psych.*, 81:141–145, 1969.