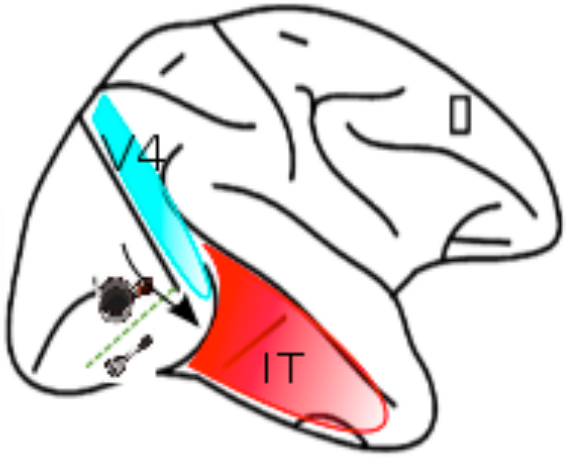
CHALMERS



Nature and scale of information transformation underlying visual object recognition in cortical areas V4 and IT in primates

Master of Science Thesis in Biomedical Engineering

PANTEA MOGHIMI

Department of Signals and Systems
Division of Biomedical Engineering
CHALMERS UNIVERSITY OF TECHNOLOGY
Göteborg, Sweden, 2012
Report No. EX026/2012

Nature and scale of information transformation underlying visual object recognition in cortical areas V4 and IT in primates

Pantea Moghimi
Master thesis performed at DiCarlo Lab
Department of Brain and Cognitive Sciences
Massachusetts Institute of Technology
Cambridge, MA, USA

Supervisor: James J. DiCarlo
Department of Brain and Cognitive Sciences
Massachusetts Institute of Technology
Cambridge, MA, USA

Examiner: Yngve Hamnerius
Department of Signals and Systems,
Chalmers University of Technology
Gothenburg, Sweden

November 2012
Master's thesis EX026/2012

Abstract

Object recognition is an easy task for humans. Although a particular object can produce an infinite number of images on the retina, depending on the position, size, pose, etc. of the image, human brain still has the ability to recognize them with no difficulty. In other words our brain has developed a tolerance to identity preserving transformations. Our visual system receives the pixel-like information from retina, and transforms it to a representation that is invariant to object's position, size, etc. and yet is capable of distinguishing between different objects. Our brain solves this problem in a hierarchical structure. Visual information propagates along this hierarchy, known as ventral visual pathway, consisting of retina, Lateral Geniculate Nucleus (LGN), primary visual cortex (V1), V2, V4 and inferior temporal cortex (IT). At higher stages neurons become more selective to abstract images and less selective to local features of the image. Yet, how the brain constructs this representation is unknown. In this study I focus on the two final stages: V4 and IT. I will try to elucidate the operation IT neurons do on top of V4 neurons. First I rule out the possibility that IT neurons are simply reducing neuronal noise. Then I will try to gain insight into scale and complexity of linear models that could potentially underlie the goodness of representation of IT over V4. In the end, I propose that our method could be used with larger data sets, to obtain more reliable results.

Acknowledgement

I would like to thank my thesis adviser Dr. James DiCarlo and the entire DiCarlo lab, especially Nuo Li and Nicolas Pinto, for all their help and support.
I also would like to thank Dr. Yngve Hamnerius, my thesis examiner, for reviewing my master thesis and helping me.
I would like to thank my lovely and supporting family for always being there for me.

Contents

Chapter 1

Introduction

In everyday life humans can identify objects regardless of their size, position, pose, illumination etc. In other words our ability to recognize objects is invariant to identity preserving transformations. However Invariant Object Recognition is a very hard task for machines [1]. The algorithm primates' brain exploits to solve this problem is yet unknown. Science of anatomy and study of activity latencies have elucidated which parts of the central nervous system are involved in invariant object recognition [2]. Currently it is believed that sensory information in the brain is mostly represented by the firing rate of neurons in specific temporal windows [3]. Yet the underlying algorithm is unknown.

When we look at an object, an image of that object is formed on our retina. Retina receives visual information by means of an irregular array of photoreceptors. Photoreceptors are sensory cells sensitive to light, even when light intensity is as low as one photon. These photoreceptors sample the image with a high enough frequency that provides us with a visual system that enables us to navigate through the external world. Exposure to light excites cons and rods in the retina causing them to depolarize and fire action potentials. This signal then goes to a specific part of thalamus, an area known as Lateral Geniculate Nucleus (LGN). LGN cells send their signal to cortex, to primary visual cortex (V1). V1 cells in turn excite cortical area V2. The activity then propagates to cortical area V4 and then inferior temporal cortex (IT). IT is the final area involved in object recognition. IT cells send their output to different areas of cortex including motor control areas and memory [2]. This pathway is called the ventral visual pathway also known as the "what" pathway. It is responsible for the task of invariant object recognition.

The image of the object on the retina is similar to that of a digital image consisting of pixels recorded by a digital camera. As visual information is processed along the ventral visual pathway, an invariant representation is built up in IT [4, 5, 6, 7, 8]. So from a computational point of view, we can define the invariant object recognition problem as building a good representation (tolerant to identity preserving transformations) from the pixel-like representation [9]. While a lot is known about structure and function of primary visual cortex [10, 11, 12], very little is known about how the final goal is achieved by IT neurons.

Retinal photoreceptors drive retinal ganglion cells. Majority of retinal ganglion cells respond optimally to light spots with dark surround or vice versa in their receptive field [1], called on-center and off-center ganglion cells respectively. LGN cells have the same shape preference, but they have bigger receptive fields. V1 cells show selectivity[2] to Gabor filters[3]. Each V1 cell has a preference for a particular size, frequency and orientation. So the population of V1 cells can span a range of different Gabor filters. Gabor filters act as edge detectors. There have been various studies on why V1 cells have this particular shape selectivity [11] and how it is learnt [15]. As we go up along the ventral visual pathway, neurons become selective to more complex shape features [16, 17, 18, 19]. V4 cells show selectivity to curvature at specific positions [16]. IT cells are mostly selective to conjunction of shapes and abstract objects [7]. In other words they are sensitive to presence of objects as a whole, or to semantics of the objects, and not particular features per se (figure 1.1).

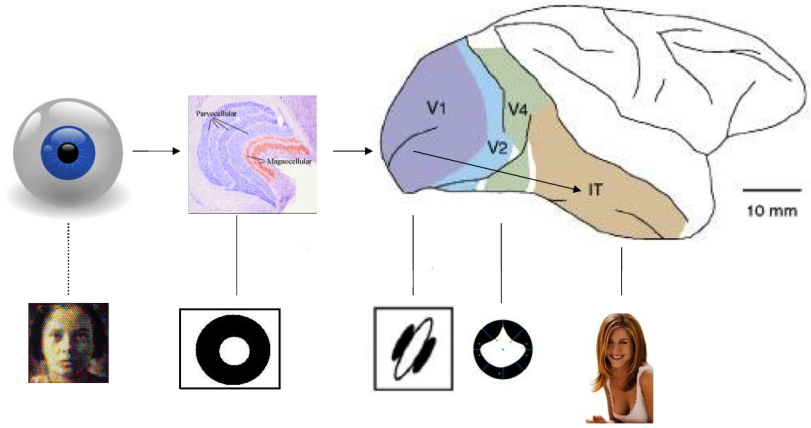


Figure 1.1: Visual information goes from retina to Lateral Geniculate Nucleus (LGN) in thalamus and then to primary visual cortex (V1). Ventral visual pathway, which is responsible for object recognition gets its input from V1 and passes information along to cortical areas V2, V4 and inferior temporal cortex (IT). Neurons in later stages of the hierarchy show selectivity to more complex images. Image courtesy of [9], [16] and [20]

From previous studies it is known that IT cells are tolerant to changes in size, position, etc. [6, 7]. We also know that this tolerance increases along the ventral visual pathway [7]. A comparison between areas V4 and IT has shown that IT cells show more invariance than V4 cells. One possible explanation is that the reason is IT cells have bigger receptive fields. It has been shown that although IT cells have bigger receptive fields than V4 cells, this is not the reason that representation of objects in IT is invariant [7]. The goal of

[1] Receptive Field (RF) of a cell is part of the visual field that can excite the cell if the proper stimulus is presented there.

[2] Selectivity to some stimulus means the cell is the most active when the stimulus is presented. Cell's response to other stimuli declines as the stimuli gets more and more different from the preferred stimulus

[3] Gabor filters are a family of wavelet functions which are basically a Gaussian function multiplied by a sinus function. For further information on Gabor functions see [13, 14]

this study is to elucidate how IT neurons process visual information they receive from activity of V4 neurons. Each IT neuron is connected to several V4 neurons. All inputs to a single neuron are not integrated with the same weight. We want to know what determines the strength and quantity of these connections, i.e. each IT neuron receives information from which V4 neurons and what determines the weighting of inputs to the cell. In other words we want to know the essence and size of the computation IT cells do on top of V4 to build an "invariant" representation underlying our ability to recognize objects without difficulty (figure 1.2). Although several computational models have been proposed to model the visual system [21, 22], they are mostly based on what is known from early visual stages. The results of this study can be used on top of these models to boost the performance of machine vision algorithms.

First, I show that the underlying mechanism cannot be explained by simple noise reduction mechanism; i.e. each IT neuron is not just pooling from exact similar V4 neurons which only leads to a reduction in neuronal noise. Then I try to see if IT neurons are doing linear pooling from V4 neurons or a non-linear computation is necessary to explain the difference between IT and V4. Further more, if the difference can be explained by linear pooling, how many V4 neurons are connected to a single IT neuron and how are those neurons determined.

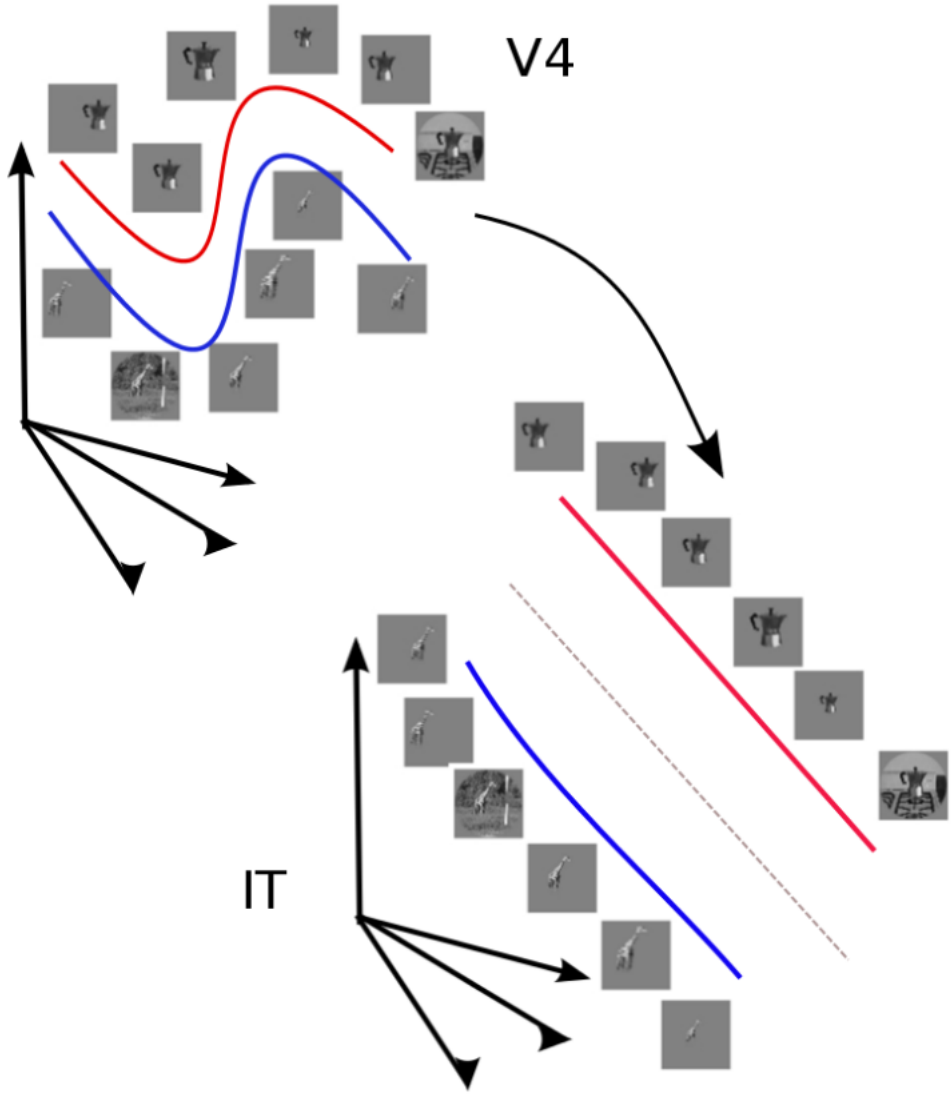


Figure 1.2: Representation of objects is more linearly separable in cortical area IT than V4 [7]. The nature of the transformation that maps representation of objects in V4 to IT is yet unknown. This is a simple schematic of what is meant by linear separability. Different objects undergo different transformations, yet representation of the same object in the brain supports invariant object recognition. Hypothetically we can assign a manifold to each object and its infinite possible transformations, in the neural space. In neural space each dimension is response of one neuron, and each point, is representation of one image in this space.

Chapter 2

Methods

2.1 Estimation of performance in the absence of noise

As previously pointed out by Rust et al. [7], IT is better than V4, not only at invariant object recognition, but also at object recognition itself in that it seems that object representation is more linearly separable in IT than V4. Their data set and approach was adopted for this work. They recorded electrophysiological activity from areas V4 and IT of two macaque monkeys while they were fixating at images shown on a monitor. Their eye movement was monitored by an eye-tracking camera. They used 60 images of 10 different objects, 6 different images of each object at different size and positions and object on a background image. Images were interleaved and each shown to each monkey 10 times i.e. they had ten different trials for each image of each object. They recorded from 140 V4 and 143 IT neurons in total. They compared the goodness of representation in IT and V4 by training linear classifiers on them. They treated each neuron as one dimension, so for example with 140 V4 neurons, we have 140 dimensions. Each trial of each image of each object is one point in this space. They then defined 10 binary tasks; each one was to separate all images of one object from all the other images. For each binary task, they trained a support vector machine (SVM) [23]. They randomly split their data set to training and testing subsets. They used half of the trials for training and the other half for testing. They repeated this random procedure 50 times to obtain a robust estimate. Having 10 objects, they trained 10 classifiers and had 10 outputs for each test data point. Label of each test data point, i.e. determining what object that data point belongs to, was the label of the classifier that had the highest output among all the classifiers for that data point (one vs. all approach). The real labels and classification labels were then compared and the performance was calculated. They observed a significant difference between IT and V4. Representation of objects in IT is more linearly separable than V4.

One explanation could be that the representation in IT is less noisy than V4 (IT Fano factor[1] for this data set: 1.43, V4 Fano factor for this dataset: 1.69 on average). Assume

[1] Fano Factor is a measure of dispersion and is defined as $F = \frac{\sigma^2}{\mu}$, Fano factor for Poisson distribution is 1.

the n dimensional space, in which each neuron is one dimension and each image of each object is a point in that space, determined by the firing rate of each neuron to that image. Each recorded trial for that image is a noisy measure of the true firing rate (i.e. firing rate in the absence of noise). It is well established that statistics of the firing rate variability is close to Poisson distribution [24, 25]. So the recorded data in that space would look like a cloud around the true value of each point in the space, the cloud points being the noisy measurements. It may be possible that IT is pooling from identical V4 neurons to reduce the noise. To test that, I decided to estimate the performances in the absence of neuronal noise.

The problem I encountered was that measures like performance in percent correct or area under the ROC (Receiver Operating Characteristic) curve [26] would saturate and do not have enough power to show the difference between V4 and IT, specially in this case where the dimensionality is high relative to number of data points (140 dimensions and 60 data points); So I chose *d'* [26] as the performance measure. *d'* can be calculated when data points are distributed over only one dimension. To calculate *d'*, I trained a SVM (neurons being the dimensions and their response to each image being one data point quantified in terms of firing rate) and projected data points on one dimension determined by the weight vector obtained from SVM. I used the same 10 binary tasks explained above, obtaining one *d'* for each task. I had to introduce some cross-validation when calculating the performance to prevent over fitting and for the estimate of goodness of representation to be robust. Yet I could not cross-validate over trials, since I was averaging out the noise discussed further in the methods. One possibility is to use a subset of images from each object to train the classifier and use the remain for testing. What I was after was not the how the neurons can generalize over subset of images, but how linearly separable the representation is (figure 2.1). To avoid this dilemma, out of the 60 images I left only one image out, trained on all other images and applied the weights to the left out image. This way I obtained a one dimensional projection of all data points from the high dimensional neuronal space. I repeated that for all images, i.e. each image was left out once. I used a one vs. all approach [27] to train the classifier, i.e. I trained one classifiers for each object which is trained to separate that object from all the other objects. That would be training a total of 60 classifiers for each of the 10 binary tasks. Thus I will end up with one *d'* for each binary task. I used LibSVM library [28] with a C value [2] of 10000 although I did not see a significant difference with other C values. Throughout this work *d'* is always calculated using SVM unless stated otherwise.

I used the following formula to calculate *d'* for each binary task:

$$d' = \frac{\langle \text{points belonging to the object} \rangle - \langle \text{points not belonging to the object} \rangle}{\sqrt{0.5 \cdot (\sigma^2_{\text{points belonging to the object}} + \sigma^2_{\text{points not belonging to the object}})}}$$

I devised two different methods to estimate *d'* in the absence of noise: Empirical method and Analytical method[3]. I tested both methods with simulated data. I used Pixel values (as a model for retinal photo receptors), simulated V1-like neurons [1] and simulated IT neurons [29]. For Pixel and V1-like cells, I randomly chose a sub set of them

[2] C value is the parameter of support vector machines that should be preset. The optimal value that generates best generalization is usually found by exhaustive search.

[3] These methods were devised and tested by the author as part of the thesis work

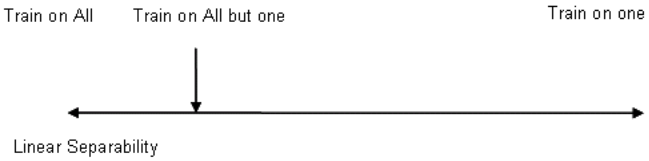


Figure 2.1: Linear separability is measured when the classifier is trained and tested on all images and shows if a linear hyper plane can distinguish between different objects, Generalization is when the classifier is trained only on one image and tested on all other images and shows how much that representation can generalize over other images having seen the data from one single image of each object. Along the linear separability-generalization continuum we had to introduced some generalization to get a robust estimate, but tried to stay near the linear separability extreme

(Since the number of Pixel and V1-like cells was very high, training a classifier on all of them was very time consuming and unnecessary for the purpose of current study. Also I wanted to keep the dimensionality of the simulated and real data at the same number so that dimensionality would not become a confounding factor), and then normalized each one randomly with respect to one random real IT neuron. I took the average of each simulated neuron over all images, and divided that by average of one random IT neuron over all images and trials. Then response of the simulated cell to all images was divided by this normalization factor. I did so to keep firing rates of the simulated and real cells at the same range, since with Poisson noise, variance of the response is equal to the mean. Also for Poisson distribution Signal to Noise Ratio (SNR) is higher for higher firing rates, I had to make sure that my simulations have the same SNR as real data. For V1-like neurons, I first had to add a bias value so that the minimum firing rate was shifted to zero, then normalize. In all of these three models, true firing rates are known and I can calculate *d'* in the absence of noise. I then added Poisson noise to raw values, and made a data set identical to the real neuronal data with ten trials. I then tested the accuracy of the estimation and its robustness to a number of different factors: number of images[4], the particular image set I have[5], number of trials used for estimation, noise model and noise amount. To do this, I varied number of images for each object from 2 images to all 6 images from each object. Images were chosen randomly three times. Also using all the images I used different number of trials (5, 7 and all 10) for estimation. To vary the amount of noise first I did the estimations with Poisson noise. Then I used Gaussian noise with a mean value equal to the raw value and varied the variance from equal to the mean value (Poisson like) to 1.5 times, 2 times and 10 times the mean value. Since I had the raw values I could compute the *d'* and compare that to the estimation. With the IT simulation I tried different parameters to span a broader range of true *d'*. Also for our V1 and Pixel simulations, I only picked features (neurons) that were responsive to at

[4]I randomly selected subsets of images to see if number of images for each object can affect the estimation, i.e. whether the estimation would be more accurate if I had for example 20 images instead of 6 for each object

[5]I tested the method with another image set to make sure our results is not dependent on the particular image set

least one image, which is the same criterion used when recording real electrophysiological activity. So I picked pixel or V1 cells that had nonzero response to at least one image, and then selected random subsets from these cells.

2.1.1 Empirical method

To average out the noise ideally we should have an infinite number of trials to estimate the real firing rates, which is not a tractable goal for obvious reasons. So I had to get a robust estimate with only 10 trials. First I estimated *d'* using different number of trials. I had 140 V4 neurons and 143 IT neurons. So formed a 140 dimensional space (for IT I randomly chose 140 neurons out of 143 several times). I averaged different number of trials to form one data point for each image, i.e. represent each image with the average of m number of trials for that image, m varying from 1 to 10. The trials were chosen randomly out of 10 trials (10 times and without replacement). I plotted *d'* vs. number of trials averaged together. Clearly more using more trials resulted in higher *d'* values. Ideally what I was after was *d'* at infinite number of trials. Then I fitted a function to this curve using the least square method and extrapolated the value of the function at infinity. I tried different functions such as tanh, exponential, logistic and hyperbolic and compared their performance with simulations.

Each point on the *d'* vs. number of trials curve is an average over several values from different random trial selections. To fit a curve, I did 20 bootstraps from values of each data point with replacement and computed the average, then used the data for least square estimation. Least square estimator needs an initial value to estimate the curve parameters. The estimator was initialized with random values 50 times and the best fit (out of 50) was chosen based on the RMS (root mean square) error of the fit. The average estimation over those 20 bootstraps would be the estimation of *d'* in absence of noise. Formulations of the tested functions are:
tanh:
$y = atanh(b.x) + c$
hyperbolic:
$y = a - \frac{1}{bx + c}$
logistic:
$y = \frac{a}{1 + e^{-(bx+c)}}$
exponential:
$y = a - be^{\frac{-x}{c}}$
where [a, b, c] are the parameters to be estimated.

I checked the method to see if it is robust to number of data points, number of trials used for estimation and amount of noise as described in the previous section.

2.1.2 Analytical method

When computing *d'*, within-class variance[6] consists of two components: within-class difference between different images in that class and noise. Assuming independent noise we

[6] Variance across data points all belonging to one binary class

can estimate variance of noise and subtract it. Consider one neuron case first. Assume we have I images in the class, each of them consisting of N_i trials. In total we will have N data points where: $N = \sum_{i=1}^{I} N_i$.

Each data point is the response of the neuron to one image at one particular trial. Here I show the response to image i in trial j as r_{ij} where:

$$r_{ij} = v_i + n_{ij} \quad \text{with } i = 1{:}I \text{ and } j = 1{:}N_i \tag{2.1}$$

where v_i is the true firing rate of the neuron to image i and n_{ij} is the neuronal noise which is assumed a random variable with zero mean.
The average response of the neuron to all images would be:

$$\begin{aligned}
\bar{r} &= \frac{1}{N}(\sum_{i=1}^{I}\sum_{j=1}^{N_i} r_{ij}) \\
&= \frac{1}{N}(\sum_{i=1}^{I}\sum_{j=1}^{N_i} v_i + n_{ij}) \\
&= \frac{1}{N}(\sum_{i=1}^{I} N_i v_i + \sum_{i=1}^{I}\sum_{j=1}^{N_i} n_{ij}) \\
&= \frac{1}{N}(\sum_{i=1}^{I} N_i v_i)
\end{aligned} \tag{2.2}$$

The overall variance of the response would be:

$$\begin{aligned}
\sigma^2 &= \frac{1}{N}(\sum_{i=1}^{I}\sum_{j=1}^{N_i}(r_{ij} - \bar{r})^2) \\
&= \frac{1}{N}(\sum_{i=1}^{I}\sum_{j=1}^{N_i}(v_i + n_{ij} - \bar{r})^2) \\
&= \frac{1}{N}(\sum_{i=1}^{I}\sum_{j=1}^{N_i} v_i^2 + n_{ij}^2 + \bar{r}^2 + 2v_i n_{ij} - 2v_i\bar{r} - 2n_{ij}\bar{r}) \\
&= \frac{1}{N}(\sum_{i=1}^{I}(N_i(v_i^2 + \bar{r}^2 - 2v_i\bar{r}) + \sum_{j=1}^{N_i} n_{ij}^2 + 2v_i n_{ij} - 2n_{ij}\bar{r})) \\
&= \frac{1}{N}(\sum_{i=1}^{I}(N_i(v_i - \bar{r})^2 + \sum_{j=1}^{N_i} n_{ij}^2))
\end{aligned} \tag{2.3}$$

If N_i is equal for all i (which is the case for this data set), equation (2) is reduced to:

$$\bar{r} = \frac{1}{I}(\sum_{i=1}^{I} v_i) \tag{2.4}$$

Which is the within-class average in the absence of noise. So to estimate the true average firing rate of a class we can use equation (2) and simply compute the average over all images and trials for that class. Provided the above condition, equation (3) is also simplified to:

$$\begin{aligned}\sigma^2 &= \frac{1}{I}(\sum_{i=1}^{I}(v_i - \bar{r})^2) + \frac{1}{N}(\sum_{i=1}^{I}\sum_{j=1}^{N_i} n_{ij}^2) \\ &= \sigma^2_{within-class} + \frac{1}{N}(\sum_{i=1}^{I} N_i \sigma^2_{n_i}) \\ &= \sigma^2_{within-class} + \frac{1}{I}\sum i = 1^I \sigma^2_{n_i} \\ &= \sigma^2_{within-class} + \sigma^2_n\end{aligned} \tag{2.5}$$

where $\sigma^2_{n_i}$ is variance of noise for image i. So in the single neuron case, we can just compute the overall variance over images and trials within a class, then estimate variance of neuronal noise for each image from different trials that we have for that image. Since trial-to-trial variance is assumed to be due to neuronal noise, variance of different trials for each image is an estimation of the variance of noise. By substituting these values in equation (5) and subtracting the noise elements from the overall variance we can estimate the within-class variance in the absence of neuronal noise.
In a high dimensional, we should first project the data on one dimension. If we have D neurons, we can project that on one dimension with a weight vector $\vec{W}$ of size $D \times 1$.

$$r_{ij} = \sum_{k=1}^{D} w_k r_{ij_k} \quad \text{and } ||\vec{W}||^2 = 1 \tag{2.6}$$

The relationship between the mean and variance (for each class) of the projected data and individual neurons is (Assuming having equal number of trials for each image, i.e. N_is are equal and assuming that neurons are independent):

$$\bar{r} = \sum_{k=1}^{D} w_k \bar{r}_k \tag{2.7}$$

$$\begin{aligned}\sigma^2 &= \sum_{k=1}^{D} w_k^2 \sigma_k^2 \\ &= \sum_{k=1}^{D} w_k^2 (\sigma^2_{within-class_k} + \sigma^2_{n_k})\end{aligned} \tag{2.8}$$

Which is mathematically equivalent to just taking the projection, and do what we did for one neuron case.

To form the projection I left one image and its trials out, trained the classifier on other images, took the weights, normalized them, and applied those weights to the trials of the left out image. When training SVM I did not use the bias term, because I just needed to project the data on one dimension and bias point was just going to add more variance to estimations. Like before, I had a binary one vs. all task for each object.

2.2 Comparison of single IT neurons with populations of V4 neurons

From anatomical and latency studies it is known that V4 is afferent to IT. Yet the nature of the operation IT neurons do on top of V4 is unknown. The complexity of this computation can vary on a continuum, from simple linear pooling of V4 neurons to a non-linear operation.

I tried to gain insight into linear models that can explain the performance gain of IT over V4. Specifically I was interested to see if IT neurons are simply pooling from random V4 neurons or from V4 neurons that are good in each specific task (for instance giraffe vs. everything else) to form an IT neuron selective to giraffes. To do this I compared single IT neurons with subpopulations of V4 neurons, either selected randomly, or selected cleverly (based on single neuron performances). To do that I defined binary tasks, and for each task chose one single IT neuron that was good at doing that task. Then compared its performance with subpopulations of V4 neurons and examined how many V4 neurons can outperform the single IT neuron. My measure for performance was estimated *d'* in the absence of noise. I used the analytical method for the estimations. In the single neuron case, I did not train a SVM. I left each data point out once, computed the *d'* with all the other data points, and multiplied the sign of that *d'* to that data point. Choosing the best single neuron is prone to outliers that are a byproduct of random neuron selection, so I looked at the distribution of single neurons *d'*s for each task for both IT and V4 (figure 2.2) and decided to take the Q90 neuron[7]. I also tried other statistics (see results). In cases I wanted to choose a subpopulation of V4 neurons that were good at doing each task (i.e. choose cleverly), I chose neurons as good or worse than the Q90 neuron, excluding the outliers.

To choose the Q90 neurons, I cross-validated over trials. I randomly selected 5 trials (without replacement), estimated noiseless *d'* for all neurons I had, chose the neurons based on those values, and then calculated the noiseless *d'* with the remaining 5 trials. This process was repeated ten times. I also tried this procedure with using 1 trial for selection and 9 trails for calculation and did not see a significant difference.

In the case of random V4 neurons, I randomly chose neurons 20 times without replacement and took the average.

As mentioned above I looked at two linear models: clever pooling or random pooling from V4 neurons. I formed subpopulations of different sizes from random or good (chosen cleverly as described above) V4 neurons (figure 3.4). I fitted a curve to this plot. I tried both linear curve and hyperbolic curve with the mentioned formula. With this curve I could calculate how many V4 neurons could explain the performance of the single Q90 IT neuron (i.e. their performance was equal to that of single IT neuron). Obviously I obtained different numbers for the random pooling and clever pooling models (See results)

To see which model was more consistent I made the task harder by decreasing number

[7] Q90 neuron is the neuron which outperforms 90% of neurons

of images to span a broader range of performances for single IT and see if any of those models break down. Decreasing number of images makes the task harder because each time the classifier is trained, it seems fewer data points and can generalize less for the left out data points. I tried this approach with other classifier training methods as well. I used linear correlation coefficient classifier and random weights. In the case of random weights, weights were chosen randomly from a uniform distribution and then normalized.

Then I tried to test the predictability of these models (Clever or random pooling with different sizes). For each pooling strategy, I considered different number of V4 neurons that IT was pooling from to build an artificial IT neuron. For each model, I took the noiseless *d'* estimation of that number of V4 neurons (either best or random for the corresponding strategy) and plotted that against the Q90 IT noiseless *d'*, each point being one task. For this analysis I used SVM to compute the projection of data points on one dimension (see results).

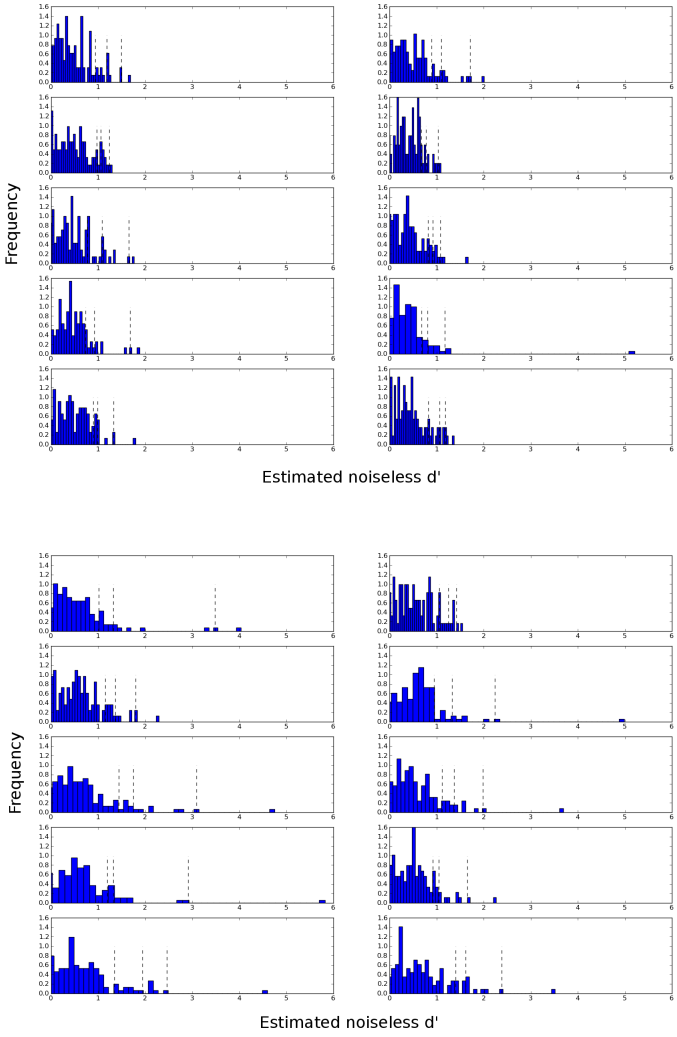


Figure 2.2: Normalized histogram of single neuron *d'* in the absence of noise for V4 (up) and IT (bottom), estimated with the analytical method. Dotted bars show Q90, Q95 and Q99 *d'*s from right to left respectively. Each subplot shows one task.

Chapter 3

Results

3.1 Goodness of estimation of performance in the absence of noise

Before analyzing the real data, I decided to estimate the robustness of the method to 1) number of images, 2) number of trials used for estimation and 3) different noise levels. I did simulations with image pixel values, V1-like cells [1] and simulated IT neurons [29] (See methods). The results of the simulations for the four functions mentioned in the methods for the empirical method are shown in figure 3.1, top. The method is clearly unbiased and robust to the variations we made i.e. number of images, number of trials and amount of noise. The estimator only failed at the noise amount with a Fano factor of 10, which is far from Fano factor values for the real data set. Those data points are not plotted. For high d's for simulated IT, the method is under estimating the true value. This is because at that d' range, the data given to estimator is so sensitive to noise. The reason is that there are only 60 data points in a 140 dimensional space, so SVM is so sensitive to noise and therefore cannot generalize properly. As can be seen in figure 3.2, error bars shrink or do not change size as number of trials increases. This was not the case for those data points deviating from the identity line, so those underestimations are because we are feeding the estimator with bad data. This proves the empirical method reliable to be used on real data. I did simulations with V1-like and IT like cells with analytical method as well and compared the two methods. The results are shown in figure 3.1, bottom. As it is evident there is no bias or significant difference between the two methods.

3.2 Comparison of IT and V4 in the absence of noise

I applied the analytical estimation method to real neuronal data for different number of neurons for V4 and IT. Neuron subsets where chosen randomly without replacement 50 times. The difference between IT and V4 cannot be explained solely by simple noise reduction (Figure 3.3). This shows that IT neurons are doing a more complicated operation on input they received from V4 neurons. The nature and scale of this operation is not quite known. It can vary from random linear pooling to clever linear pooling to nonlinear

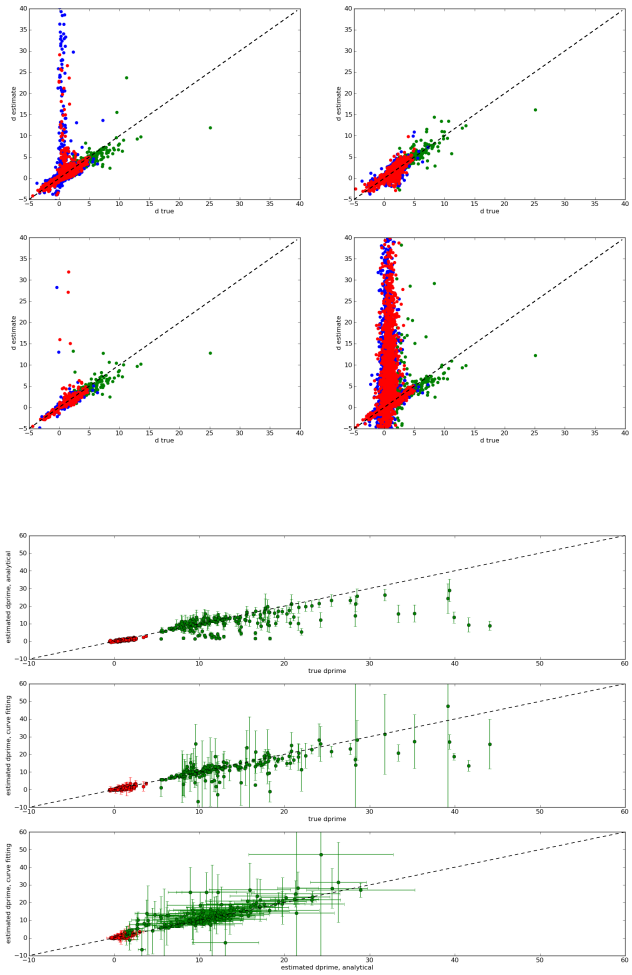


Figure 3.1: **Top:** Robustness check for Empirical estimation, x axis shows the true *d'*, y axis shows the estimated *d'*, each point is one task with particular noise model and amount, number of images and number of trials. Blue, red and green points show pixel, V1 like cells and simulated IT neurons respectively. From left to right the plots show the estimation with logistic, hyperbolic, exponential and tanh curves fitted to data respectively. **Bottom:** Simulations for both analytical (up) and empirical (middle) methods and their comparison (bottom) for V1 like cells (red) and simulated IT neurons (green). Error bars reflect standard deviation across different bootstraps of values.

pooling of V4 neurons (Figure 3.4). Also we do not know each IT neuron is computing on how many V4 neurons in each of these models.

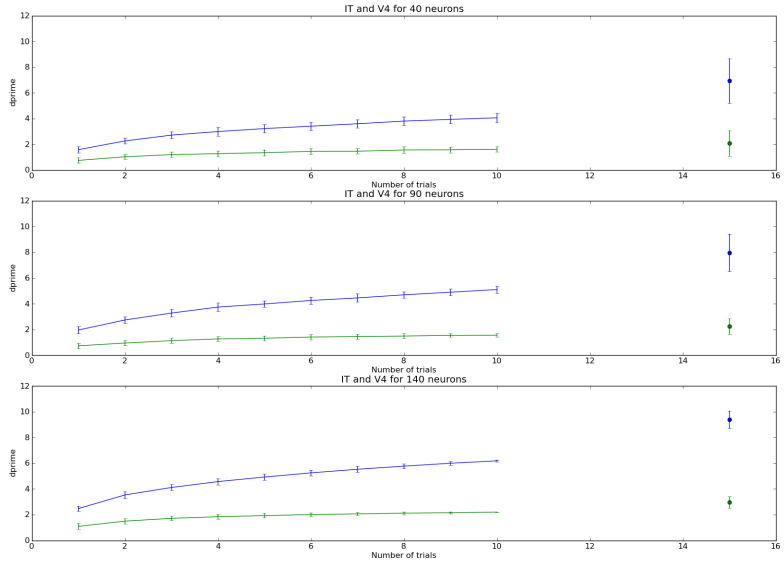


Figure 3.2: *d'* of IT (blue) and V4 (green) cells as a function of number of trials averaged together to compute the mean. The mean value for each image is the one data point used to compute *d'*. Final end points are no noise estimations extrapolated with hyperbolic function. Error bars reflect variation across different neuron subset and trial selections for the curves and different bootstraps from data for the estimated no noise points. This analysis is done for different number of neurons: 40 (top), 90 (middle) and 140 (bottom).

I considered two linear pooling models for further analysis: each IT neuron is pooling from random V4 neurons or each IT neuron is pooling from best V4 neurons for each specific task. I examined each of these models to see which one was more successful at describing the data and what was the size of pooling for each model. Although the results may not seem consistent across different tasks, they will give us an insight about the scale of the model. In other words, the space of probable models (from basic random pooling models to clever models to more sophisticated nonlinear models with any scale) shrinks to the smaller space of possible models.

3.2.1 Linear models of pooling from V4 to IT

To see which of the two considered models better describes the performance difference between IT and V4, for each binary task (see methods) I compared best IT neuron that could do that task with subpopulations of V4 neurons. I did this analysis with the estimations of *d'* in the absence of noise. First I looked at the distribution of single IT and V4 neurons performance, and decided that Q90 was robust to outliers for choosing the best neuron (see figure 2.2). So I compared Q90 IT with random subpopulations of V4 neurons or V4 neurons chosen as good as or worse than the Q90 V4 neuron (see methods).

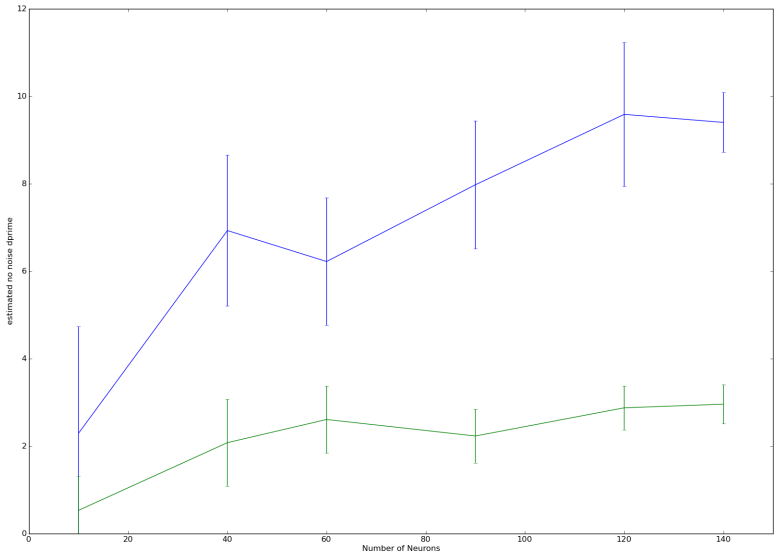


Figure 3.3: Estimation of *d'* in the absence of noise as a function of number of neurons for IT (blue) and V4 (green) with empirical method. Error bars reflect different neuron subset selections and bootstraps of data for estimation.

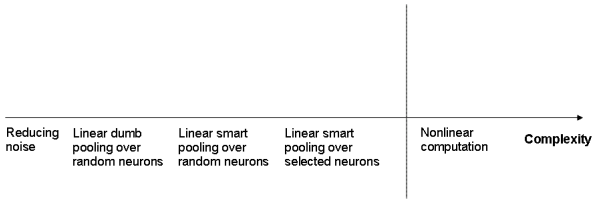


Figure 3.4: Complexity continuum of the models: It can vary from very basic of noise reduction to complex non-linear operations

I plotted noiseless *d'* as a function of number of neurons for V4 neurons for each model. I fitted a hyperbolic function to each curve and computed number of V4 neurons at which the curve is closest to the noiseless Q90 IT.

I then plotted this number vs. IT noiseless *d'* for each task. To see which model is more consistent to describe the difference between IT and V4, I decided to vary the difficulty of the task and span a broader range for noiseless IT *d'* cells (see methods). The results are shown in figure 3.5.

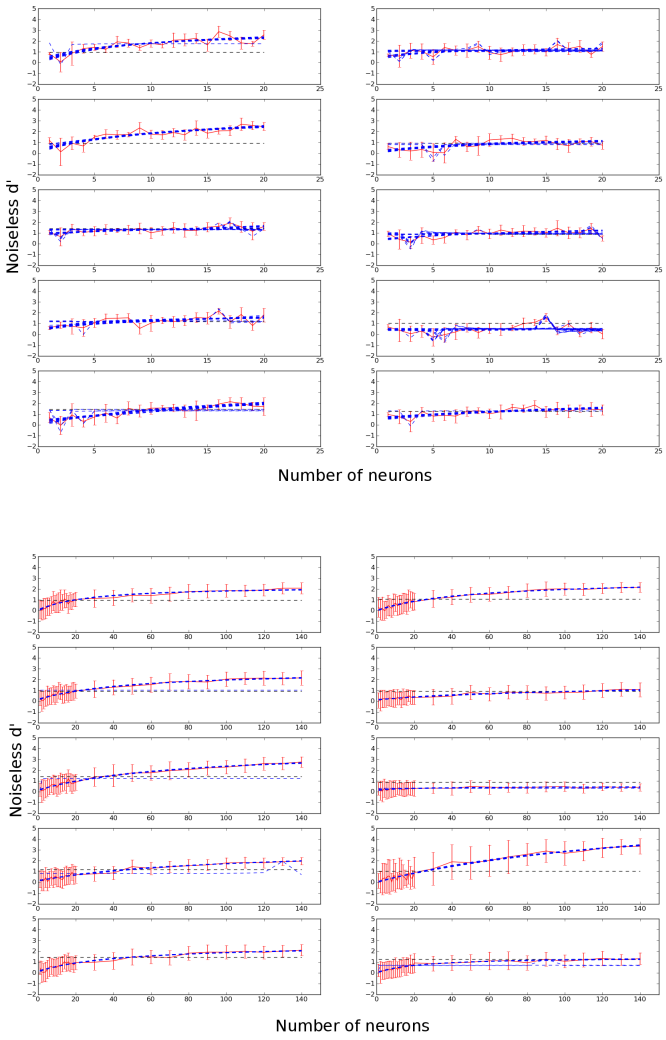


Figure 3.5: Estimated noiseless *d'* for Q90 V4 neurons (top) and random V4 neurons (bottom) as a function of number of neurons with analytical method. The blue curves show the curve fitted to the plot (Hyperbolic case), each of them corresponding to one bootstrap from data. The dashed black line shows no noise Q90 IT neuron for comparison with subpopulations of V4 neurons. Each subplot shows one task. Error bars reflect different trial and neuron subset selections for random selection case.

I did this analysis with various classifiers for the original task to see the range of number of V4 neurons explaining IT (see methods). The results are shown in tables 3.1 and 3.2.

Sample	Learn	Q90	Q95	Q99
Best	RAND	x	x	x
Best	SVM	$10.35 \pm 13.70^{\diamond}(0.3)$	5.23 ± 2.33	$7.50 \pm 5.95^{*\diamond}(0.2)$
Best	ρ	$9.96 \pm 13.98^{\diamond}(0.3)$	8.44 ± 4.18	$8.03 \pm 3.99^{*\dagger}(0.6)$
Random	RAND	x	x	x
Random	SVM	$54.10 \pm 43.60^{\dagger}(0.1)$	$129.73 \pm 332.84^{\dagger}(0.2)$	$176.34 \pm 175.31^{\dagger}(0.5)$
Random	ρ	$92.39 \pm 50.40^{\dagger}(0.4)$	$52.82 \pm 24.66^{\dagger}(0.8)$	$278.07 \pm 79.75^{\dagger}(0.8)$

Table 3.1: Hyperbolic curve fitted. Numbers show where the fitted curve meets IT (for choosing the best, the criterion is the same as IT, i.e. Q90 IT is compared to Q90 V4, Q95 IT to Q95 V4 and so on) in the mean ± std format. Numbers in () show how many of the curves were excluded form the analysis. *: Curve was excluded because the fitted curve was above IT. †: Curve was excluded because V4 curve almost never met IT, in hyperbolic case it cut IT at negative values and in linear case it had near zero slope. $\diamond$: Curve was excluded because the function was not a good fit for the curve hence excluded from the analysis.

Sample	Learn	Q90	Q95	Q99
Best	RAND	x	x	x
Best	SVM	$13.56 \pm 13.61^{*}(0.1)$	$6.13 \pm 2.31^{*}(0.2)$	$10.31 \pm 5.73^{*\dagger}(0.4)$
Best	ρ	11.79 ± 3.96	$10.63 \pm 4.55^{*}(0.1)$	$15.73 \pm 7.79^{*\dagger}(0.2)$
Random	RAND	x	x	x
Random	SVM	90.81 ± 97.64	139.84 ± 167.47	289.44 ± 423.152
Random	ρ	$251.04 \pm 407.36^{\dagger}(0.2)$	$474.01 \pm 972.60^{\dagger}(0.2)$	$461.00 \pm 566.08^{\dagger}(0.2)$

Table 3.2: Linear curve fitted. Numbers show where the fitted curve meets IT (for choosing the best, the criterion is the same as IT, i.e. Q90 IT is compared to Q90 V4, Q95 IT to Q95 V4 and so on) in the mean ± std format. Numbers in () show how many of the curves were excluded form the analysis. *: Curve was excluded because the fitted curve was above IT. †: Curve was excluded because V4 curve almost never met IT, in hyperbolic case it cut IT at negative values and in linear case it had near zero slope. $\diamond$: Curve was excluded because the function was not a good fit for the curve hence excluded from the analysis.

To test the predictability of each model (each model is specified with the strategy of pooling and size of pooling, this analysis was only done with SVM), I plotted Q90 IT performance vs. prediction of each model for different sizes for each strategy (see methods). The results are shown in figure 3.6. Unfortunately I did not see a consistency in those models, so I could not find a model that could predict the performance of IT cells for all tasks. This could have different reasons:
1. There is a sampling bias in this neuronal data set. Recorded neurons maybe not revealing the underlying operation. As we can see for single Q90 IT vs. single Q90 V4 plot (figure 3.6, top, upper left plot), there is no correlation between single IT and V4 neurons. A task hard for IT is not necessarily hard for V4 and vice versa. So the variation of d's we observe for different tasks is because of this particular sample of neurons and not the intrinsic difficulty of the task due to lower level features and image statistics (see figure 3.7).
2. The task is not difficult enough to show the difference between IT and V4. As we can see single V4 neurons are almost as good as single IT neurons (see figure 2.2), but in single IT cells distribution, we can see more outliers very good at doing the task.

These two reasons can be further investigated with a better data set, i.e. a harder task with more images for each object and a sample with more neurons.

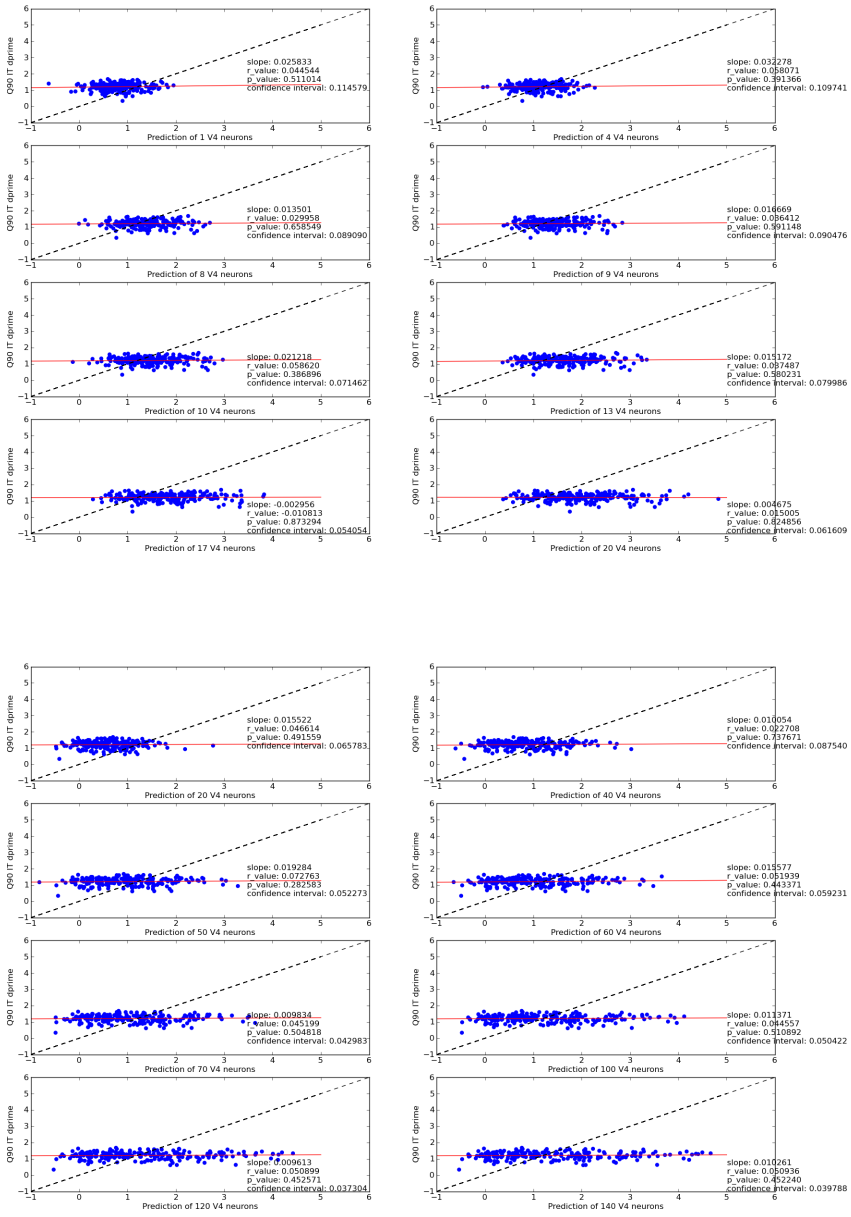


Figure 3.6: Prediction of Q90 (up) and random (bottom) V4 models vs. real Q90 single IT *d'*. Each subplot shows a different model with different number of neurons.

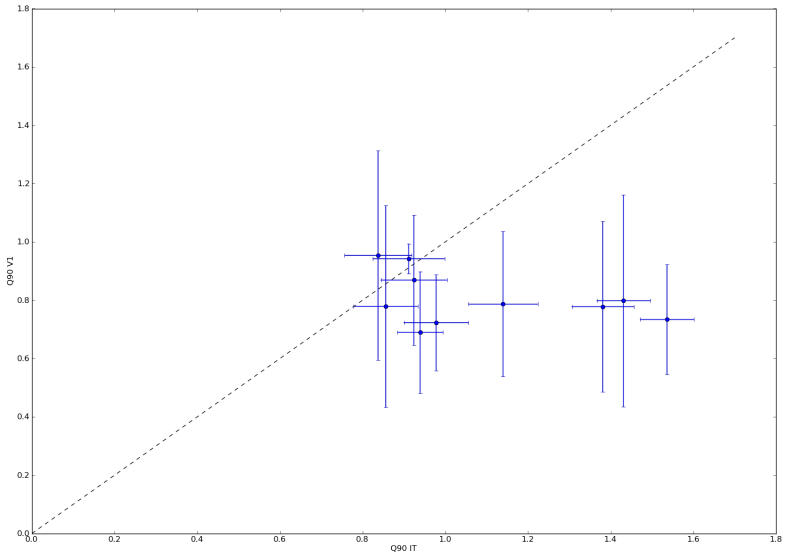


Figure 3.7: Comparison of the difficulty of the task for simulated V1 like cells and recorded IT cells. Each point shows a different task. Error bars reflect variation across different noiseless *d'* estimations. Performance of V1 like cells is more or less the same for all tasks.

Chapter 4

Conclusion

In this work I tried to get an insight into possible models that might be able to explain the difference in goodness of representation between visual cortical areas, IT and V4 (Figure 1.2). In the first part, I ruled out the simplest model (simple noise reduction) that potentially could explain that difference. Although this model seemed improbable because as previously shown before [7], IT neurons are more sensitive to feature conjunctions and semantics of the object and not just the shape, while V4 neurons carry more shape specific information. Yet this model was to be tested, because those studies were done with noisy data points.

In the second part, using the noiseless performance estimation method, I proposed a method that has the potential to extract the form and scale of the computation IT neurons do over their V4 afferents. With the proper data set, it is possible to elucidate the underlying computation. The proposed method has several advantages. First, it averages out an important confounding factor: Neural Noise. Second, the method gives a robust measure of goodness of representation that can deal with limited data sets. Third, the particular choice of performance measure (d') has the power to elucidate differences even when other measures saturate, i.e. it can to some extent deal with high dimensionality problem. Finally the results are not specific to current data set and not sensitive to amount of collected data (number of images or trials).

The results will help us understand how a "good" representation is built along the ventral pathway, i.e. a representation that can easily be used by later stages receiving input from IT. Also we will be able to elucidate how the brain solves the problem of invariant object recognition. This could inspire machine vision models to support various detection tasks such as pedestrian detection, face detection, etc. more efficiently. It could also be of clinical use in the future.

Bibliography

[1] N Pinto, DD Cox, and JJ DiCarlo. Why is real-world object recognition hard? *PLoS Comput. Biol.*, 4:0151–0156, 2008.

[2] ER Kandel, JH Schwartz, and TM Jessell. *Principles of Neural Science*. McGraw-Hill, 4th edition, 2000.

[3] AJ Parker. Sense and the single neuron: Probing the physiology of perception. *Annu. Rev. Neurosci.*, 21:227–277, 1998.

[4] M Ito, H Tamura, I Fujita, and K Tanaka. Size and position invariance of neuronal responses in monkey inferotemporal cortex. *J Neurophysiol*, 73:218–226, 1995.

[5] I Fujita. The inferior temporal cortex:architecture, computation, and representation. *Journal of Neurocytology*, 31:359–371, 2002.

[6] CP Hung, G Kreiman, T Poggio, and JJ DiCarlo. Fast readout of object identity from macaque inferior temporal cortex. *Science*, 310:863–866, 2005.

[7] NC Rust and JJ DiCarlo. Selectivity and tolerance (invariance) both increase as visual information propagates from cortical area V4 to IT. *J Neurosci*, 30(39):12978–12995, 2010.

[8] J Pauls, E Bricolo, and N Logothetis. chapter View Invariant Representations in Monkey Temporal Cortex: Position, Scale and Rotational Invariance, pages 9–42. Oxford University Press, 1996.

[9] JJ DiCarlo and DD Cox. Untangling invariant object recognition. *Trends in Cognitive Sciences*, 11:333–341, 2007.

[10] Y Dan, JJ Atick, and RC Reid. Efficient coding of natural scenes in the lateral geniculate nucleus: experimental test of a computational theory. *J Neurosci*, 16:3351–3362, 1996.

[11] BA Olshausen and DJ Field. Emergence of simple-cell receptive field properties by learning a sparse code for natural images. *Nature*, 381:607–609, 1996.

[12] RL De Valois, DG Albrecht, and LG Thorell. Spatial frequency selectivity of cells in macaque visual cortex. *Vision Res*, 22:545–559, 1982.

[13] JP Jones and LA Palmer. An evaluation of the two-dimensional Gabor filter model of simple receptive fields in cat striate cortex. *J Neurophysiol*, (58):1233–1258, 1987.

[14] D Gabor. *Theory of communication*. Institution of Electrical Engineering, 1946.

[15] A Caponnetto, T Poggio, and S Smale. On a model of visual cortex: learning invariance and selectivity from sequence of images. Technical Report 2008-030, CSAIL, Massachusetts Institute of Technology, Cambridge, MA, 2008.

[16] A Pasupathy and CE Connor. Shape representation in area V4: Position-specific tuning for boundary conformation. *J Neurophysiol*, 86:2505–2519, 2001.

[17] JL Gallant, J Braun, and DC Van Essen. Selectivity for polar, hyperbolic, and Cartesian gratings in macaque visual cortex. *Science*, 259:100–103, 1993.

[18] JL Gallant, CE Connor, S Rakshit, JW Lewis, and DC Van Essen. Neural responses to polar, hyperbolic, and Cartesian gratings in area V4 of the macaque monkey. *J Neurophysiol*, 76:2718–2739, 1996.

[19] J Hedge and DC Van Essen. Selectivity for complex shapes in primate visual area V2. *J Neurosci*, 20, 2000.

[20] M Carandini, JB Demb, V Mante, DJ Tolhurst, Y Dan, BA Olshausen, JL Gallant, and NC Rust. Do we know what the early visual system does? *J Neurosci*, 25:10577–10597, 2005.

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

[22] SM Stringer and ET Rolls. Invariant object recognition in the visual system with novel views of 3D objects. *Neural Computation*, 14:2585–2596, 2002.

[23] V Vapnik. *The nature of statistical learning theory*. Springer-Verlag, 1995.

[24] DJ Tolhurst, JA Movshon, and AF Dean. The statistical reliability of signals in single neurons in cat and monkey visual cortex. *Vision Res*, 23:775–785, 1983.

[25] MN Shadlen and WT Newsome. The variable discharge of cortical neurons: Implications for connectivity, computation and information coding. *J Neurosci*, 18:3870–3896, 1998.

[26] NA MacMillan and CD Creelman. *Detection Theory: A User's Guide*. Psychology Press, 2005.

[27] R Rifkin and A Klautau. In defense of one vs all classification. *Journal of Machine Learning Research*, (5):101–141, 2004.

[28] CC Chang and CJ Lin. Libsvm: a library for support vector machines. *ACM Transactions on Intelligent Systems and Technology*, 2:27:1–27:27, 2011. Software available at
url{http://www.csie.ntu.edu.tw/ cjlin/libsvm}, retreived September 2009.

[29] N Li, DD Cox, D Zoccolan, and JJ DiCarlo. What response properties do individual neurons need to underlie position and clutter invariant object recognition? *J Neurophysiol*, 18:360–376, 2009.