Modeling Human Vision Using Feedforward Neural Networks

by

Francis Xinghang Chen

S.B., Massachusetts Institute of Technology (2015)

Submitted to the Dept. of Electrical Engineering & Computer Science
in partial fulfillment of the requirements for the degree of

Masters of Engineering in Computer Science and Engineering

at the

MASSACHUSETTS INSTITUTE OF TECHNOLOGY

September 2016

Author .....
Dept. of Electrical Engineering & Computer Science
September 21, 2016

Certified by .....
Tomaso Poggio
Eugene McDermott Professor, BCS and CSAIL
Thesis Supervisor

Accepted by .....
Christopher J. Terman
Chairman, Masters of Engineering Thesis Committee

Modeling Human Vision Using Feedforward Neural Networks

by

Francis Xinghang Chen

Submitted to the Dept. of Electrical Engineering & Computer Science
on September 21, 2016, in partial fulfillment of the
requirements for the degree of
Masters of Engineering in Computer Science and Engineering

Abstract

In this thesis, we discuss the implementation, characterization, and evaluation of a new computational model for human vision. Our goal is to understand the mechanisms enabling invariant perception under scaling, translation, and clutter. The model is based on I-Theory [50], and uses convolutional neural networks. We investigate the explanatory power of this approach using the task of object recognition. We find that the model has important similarities with neural architectures and that it can reproduce human perceptual phenomena. This work may be an early step towards a more general and unified human vision model.

Thesis Supervisor: Tomaso Poggio
Title: Eugene McDermott Professor, BCS and CSAIL

Acknowledgments

I never said this in lab, but I've wanted to work here since my freshman year at MIT. I thank Tommy Poggio, both for the opportunity to join the group, and for his guidance and support. Tommy has provided me with creative theories and suggestions, clear explanations, helpful feedback, and a friendly, welcoming research group, all of which have been foundational to my work.

Thanks also to my amazing mentor, Gemma Roig. When I began the project, she helped me through the difficult transition from ignorance to productivity, with patience and enthusiasm. Throughout the summer, she always took time for my (frequent) interruptions and questions, never failing to bring both a smile and a new perspective. Without her guidance, I'd probably still be trying to compile Caffe on the cluster. Thanks also to Xavier Boix and Leyla Isik. Along with Gemma, they have provided an invaluable foundation for my work, both in terms of code and conceptual insight.

Thanks to all members of the Poggio Lab—among all of my research experiences, this has been one of my favorites. I owe it to the whole group. Thanks especially to Andrea Tacchetti, Brando Miranda, Charlie Frogner, Ethan Meyers, Gadi Geiger, Georgios Evangelopoulos, Jim Mutch, Max Nickel, and Stephen Voinea, for informative discussions. Thanks also to Professor Thomas Serre for helpful suggestions.

Thanks to Tony Eng, my mentor and friend—he taught me most of what I know about public speaking, which was invaluable for my oral presentation. In addition, he gave me the opportunity to earn financial support for the M.Eng. as a 6.UAT TA. Thanks also to my friend Damon Doucet, for many insightful conversations about motivation and effort. These have helped me to evolve significantly.

Thanks to Anne Hunter of the EECS Department—I'm pretty sure she can resolve any M.Eng.-related student concern in seconds. And of course, thanks to all others at MIT who have been part of my experience.

I save my final paragraph to thank my family. My father, for decades of lessons on hard work and sacrifice. My mother, the best listener I know, for your time and your love. My twin sister, Florence, for your friendship. And all others, previous generations included, for making all of this possible. As is written in the epigraph of *For Whom the Bell Tolls*, no man is an island.

This work was supported by the Center for Brains, Minds and Machines (CBMM), funded by NSF STC award CCF - 1231216.

Contents

List of Figures

List of Tables

Chapter 1

Introduction

The human brain contains the most powerful visual system in the world. By understanding human vision, we can gain scientific insight into human cognition. We can also inspire engineering efforts that seek to build artificial intelligences. These benefits come from building *models*, abstractions that capture what we know about the brain. Computational models are of particular interest, since they are both precise and testable.

This thesis presents a new computational model, representing the *feedforward pathway* of the human visual system. This means that we only consider a 'one-way' flow of information, from the eye through the brain's *ventral visual stream* [57, 64]. In addition, we only consider performance resulting from a single *saccade*, or view of the world. We exclude longer-term feedback-based neural processing, as well as the potential for multiple saccades. Studies suggest that this subset of the visual system can account for key aspects of human and primate visual performance [57, 39, 26]. In particular, it explains human performance on *object classification* tasks.

In object classification, a test subject is shown a series of objects. For each object, they must categorize it as an exemplar from a *class* of objects. The set of classes is mutually exclusive and known ahead of time, and the test subject might be trained prior to the actual task. Successful classification requires both *selectivity* in mapping objects to classes, and *invariance* to image transformations [54]. Humans are known to be extremely skilled at such tasks, even in complex settings [64, 39]. In addition,

classification tasks can easily be generalized between humans and computers. Thus, object classification is a suitable medium for evaluating computational models against the human visual system.

In order to perform object classification, our model uses convolutional neural networks (CNNs). Recently, CNNs have achieved state-of-the-art performance in large-scale object classification [33, 61, 19]. Additionally, they represent the state-of-the-art for explaining primate neural activity in the inferior temporal cortex (IT) [64]. IT is one of the highest-level regions in the ventral stream [64, 57]. Thus, CNNs may provide an effective way of modeling the human visual system.

We present a CNN-based computational model of the feedforward pathway of the ventral stream. Our work builds on *I-theory* [50], which postulates mechanisms for transformation-invariant image representation in the human brain. Our model explains human psychophysic phenomena relating to object recognition. In particular, it demonstrates properties of translation- and scale-invariance, as well as performance in cluttered scenes, that are observed in human vision. The model also fits data regarding the structure and function of biological neural networks.

The remainder of this thesis discusses our work in detail. Chapter 2 reviews related work and provides conceptual grounding for our model. Chapter 3 discusses the actual implementation of the model. Chapters 4 and 5 focus on evaluating the model against human psychophysical phenomena. Chapter 6 assesses our work, compares it to previous approaches, and synthesizes contributions. Finally, Chapter 7 proposes future work.

Chapter 2

Background and Related Work

Our work relies on three main pillars: (1) studies of human vision, especially regarding the feedforward pathway of the ventral stream; (2) the HMAX computational model of the feedforward pathway [58, 57, 28, 27]; and (3) convolutional neural networks (CNNs) for object recognition [35, 33].

2.1 Human Vision Studies

The human visual system shows high levels of *selectivity* and *invariance*. We can correctly recognize a tremendous variety of objects, even when they are transformed (e.g. scaled and shifted) [53, 40]. In addition, we can recognize objects in a variety of contexts, i.e. in cluttered scenes (though this ability has limitations [63, 49]). These abilities generally do not require supervised training for every view of each object. We can see this intuitively—one can recognize a face from a photograph or from a small amount of experience. It is not necessary to see every possible transformation, in every possible context, while repeatedly being told, 'this is Francis.'

Thus, the brain must have a strong ability to *generalize*, learning a great deal from only a few samples. Logothetis et al. [40] note that it could be feasible, for instance, to store a few views of each object and *interpolate* between them to recognize all possible views. If we suppose that humans can combine this 'view-based' approach with scale- and translation-invariance, this results in a mechanism for strong generalization across

scales, shifts, and rotations. Biological studies have provided evidence in favor of this mechanism and have testified regarding its speed and accuracy.

2.1.1 Neural Mechanisms of Object Recognition

Non-human primates have some human-like brain structures [54, 57, 10] and exhibit similarly powerful object recognition capabilities [57]. Studies of them can therefore inform our understanding of human vision.

For instance, Logothetis et al. [40] studied the neural responses of monkeys during object recognition. They trained the monkeys to recognize new types of objects, which the monkeys had never seen before. During recognition, activations of neurons in the animals' inferior temporal cortex (IT) were recorded using electrodes, since IT is important to this task [40]. The researchers found many neurons whose activity correlated with particular views of particular objects. Notably, some of these *view-tuned units* (VTUs) [53] showed selectivity properties [40]. That is, their response was highest at some 'optimal view,' and decayed when an object was rotated away from this view. The decay was proportional to the size of the rotation. Such neurons therefore activated strongly in response to a relatively narrow subset of possible views [40], suggesting that the brain stores a few views and uses interpolation.

In addition, the same study observed translation- and scale-invariance. IT neurons would sometimes 'recognize' objects at scales and translations not previously seen, without requiring eye movements [40, 53]. In fact, the average IT neuron was found to be translation-invariant for ~2° shifts and scale-invariant for ~1-octave scalings in either direction [53].

Taken together, these results suggest (1) that neurons become adapted for object recognition through experience, e.g. training; (2) that *views* of objects are stored during learning; and (3) that the brain generalizes over scale and translations of an object, even when these have not been presented previously [40, 53]. This provides crucial hints regarding how we might model the human object recognition system.

Understanding the timing of visual processes is another key element informing the development of an effective model.

2.1.2 Speed of Object Recognition

Neural decoding studies suggest that the primate brain can recognize objects extremely rapidly. For example, Hung et al. [26] used a machine learning classifier to decode neural responses in the IT of monkeys during object recognition. The goal was to infer the category of a presented object, by only considering neural spike counts from a fixating test subject. 125 milliseconds (ms) after an object was presented, the classifier could predict its category with 70% accuracy (vs. 12.5% for random guessing). Serre et al. [57] note that this exceptionally short latency provides evidence in favor of a feedforward stage of object recognition, which they call *immediate recognition*. They assert that in general, eye movements and top-down processing take nontrivially longer. Thus, it is reasonable to study object recognition as a one-way flow of information from the eye through the visual cortex [57].

Liu et al. [39] performed a similar neural decoding study, recording neural activity from human subjects while showing pictures of objects. In this case, using a support vector machine (SVM) classifier on electrode recordings, the authors could infer the category of a presented object after ~100 ms of neural processing. This was possible even under scaling and depth-rotation of objects. 100 ms is not enough time, on average, for an eye movement (saccade)—subjects generally have one view of each object [39]. In addition, neural back-projections outside of the feedforward pathway are thought to have significant influence after ~200 ms of processing [39]. These findings suggest that the feedforward pathway is responsible for a significant portion of human object-recognition abilities, showing both selectivity and invariance.

The HMAX computational model explains these observations.

2.2 The HMAX Model

HMAX [57, 58, 28, 27] is a well-studied computational model of the feedforward pathway of primate vision. HMAX is based on Hubel and Wiesel's neurological research on the monkey's visual cortex [25, 57]. Given an input image, HMAX first passes it through a grid of S units. Each unit applies a *Gabor filter* to a small patch

of the image [28]. The Gabor filter is tuned, essentially, to detect alternating black-and-white bars [17] at a certain scale and orientation [58], and can be used e.g. for edge detection. Thus, the output of the first S layer, S_1, is a grid of filter responses over the entire image. Filters span a pre-defined set of orientations and scales. This models the behavior of *simple cells* in the primate brain [57], described in [25].

The output of S_1 is passed to C_1, a grid of C units [58]. Each C unit takes input from all S units within a small spatial region and a small set of scales. Its output is the maximum (MAX) over all of the S inputs [58]. Thus, the output of C_1 is a grid of 'MAX-pooled' filter responses over the image. This models the activity of *complex cells* in the visual cortex [57], described by Hubel and Wiesel [25]. MAX-pooling has been shown to be neurologically plausible, and to agree with biological and computational data [53]. In addition, it explains invariance to scale and clutter [53].

The basic version of the model follows S_1 and C_1 with layers S_2 and C_2 [58]. S_2 performs filtering, like S_1, but learns its filters by "imprinting" activations from training images (unsupervised learning) [57]. C_2 is another MAX-pooling layer, this time covering all scales and positions [58]. In the overall hierarchy, alternating S and C layers respectively build selectivity and invariance [58, 54], computationally meeting the demands of object recognition. From a neurological perspective, [56] enumerates plausible neural implementations for these two primitives, additionally showing that HMAX can explain biological data regarding the ventral stream. Furthermore, versions of HMAX have improved fidelity by calibrating parameters with physiological data [27, 57, 58]. The top layers of HMAX can be fed into machine learning classifiers (e.g. SVM) to perform object recognition from an image [58].

HMAX has exhibited impressive performance as a model. For instance, in [58], HMAX features were compared with features from scale-invariant feature transform (SIFT) and histogram of oriented gradients (HOG), two well-known computer vision techniques [2]. In several object recognition tasks, HMAX features were more descriptive, even in clutter [58]. In addition, HMAX responses correlated heavily with human performance in a rapid animal vs. non-animal image classification task [57]. This experiment used *backward masking* in an attempt to restrict processing to the

feedforward pathway and enable a fair comparison. Finally, activations of high-level HMAX units were found to predict neural activity in IT [57]. Synthesizing these results, it seems that HMAX is a reasonably representative model of the feedforward visual stream.

The advent of convolutional neural networks (CNNs) may enable models that are even more powerful and descriptive.

2.3 Convolutional Neural Networks (CNNs)

CNNs [33, 35] provide a powerful computing paradigm for object recognition. In a CNN, an input image is first processed by a *convolutional layer* [33]. This layer convolves the image with a set of *kernels*, outputting a grid of results called a *feature map* [35]. The feature map may then be processed by a *pooling layer*, also known as a *subsampling layer* [33, 35]. This type of layer combines outputs of the feature map that are close together in space, possibly using an average or MAX function [33, 35]. This results in a subsampled version of the feature map, with lower spatial dimension. Finally, CNNs can have *fully-connected* layers, where every output of the previous layer is connected to every input of the next layer, influencing it according to a parameter called the *weight* [33, 35]. Intuitively, this allows for the representation of arbitrary associations. Typically, a fully-connected layer generates the final output of a CNN [36, 30, 33]. In object recognition, the output is a probability distribution over possible class labels for an input image. Importantly, the kernels of convolutional layers and the weights of fully-connected layers are learned in a supervised manner, using a procedure called back-propagation [35, 23].

After supervised training with a set of class-labeled images, CNNs can significantly outperform than other object recognition approaches. For instance, when evaluated on the MNIST handwritten digit dataset [37], CNN-based approaches were superior to techniques based on linear classification and nearest-neighbors [38]. The best CNN-based approach yielded an error rate of only 0.7%, near the authors' estimated human error rate of 0.2% [38]. Even more impressively, in [33], the authors trained CNNs

to recognize objects from the ImageNet dataset [55]. This classification task involved 1.2 million training images and 1000 possible object classes. A CNN-based classifier achieved a top-5 error rate of 15.3%, compared to 26.2% for the nearest competing approach [33]. Subsequent CNNs have adopted new architectures and optimization methods [61, 19], achieving ImageNet error rates as low as 3.57% [19]. We believe that CNNs represent the most powerful existing approach for object recognition.

Moreover, CNNs naturally lend themselves to modeling neural activity in the brain. Convolutional layers are similar to the S layers of HMAX, which in turn represent simple cells in the brain. Though convolutional layers use learned kernels rather than pre-tuned Gabor filters, it is known that the first convolutional layer of a CNN frequently learns Gabor-like kernels [65]. Pooling layers in a CNN are analogous to C layers in HMAX and to complex cells in the brain. Finally, fully-connected layers allow for brain-like association learning, in an intuitive sense. They could possibly represent the *prefrontal cortex* (PFC) [57], a top-level ventral stream area. The PFC is described in [57] as "involved in linking perception to memory and action."

A recent study by Yamins et al. [64] quantifies the ability of CNNs to represent neural activity in the ventral stream. In this study, the authors trained a mixture of CNNs for object recognition, using it to predict neural activations in the IT of monkeys. They found that the CNN could predict this activity much more accurately than any other model, including HMAX [64]. Since IT is the part of the visual cortex containing the most selective and invariant responses to displayed images [64], this suggests that CNNs develop somewhat brain-like patterns of activity in object recognition. Thus, they may be a useful tool for modeling neural activity in the ventral stream.

2.4 Synthesis of Opportunity

Considering the literature reviewed above, we believe that CNNs provide a promising opportunity for modeling the feedforward human visual system, in the context of object recognition. This requires developing a CNN-based model that is both biologically

plausible and performant. We aim to have such a model explain the phenomena of translation- and scale-invariance [40, 53], and performance under clutter [63]. Other models have attempted to perform the same tasks. However, we believe that our work is the first to thoroughly evaluate a CNN-based model *across* the contexts of invariance and clutter, while maintaining close and explicit ties with the underlying biology.

Chapter 3

Our Model

We first discuss the *eccentricity theory* [50], a pre-existing body of work which provides the foundation for this thesis. The eccentricity theory is a subset of I-Theory. We then describe our CNN-based model, which mostly follows the eccentricity theory, but diverges slightly.

3.1 The Eccentricity Theory

As mentioned in Chapter 2, the brain can learn a great deal from a small amount of data. In object classification, one would say that it requires *low sample complexity* [50]. The eccentricity theory posits a key mechanism behind this phenomenon—the ability to recognize objects in a *transformation-invariant* way [50]. Specifically, the theory focuses on invariance to translation and scale.

In particular, the theory proposes that the brain prioritizes scale-invariance over translation-invariance [50]. It relies on the premise that for any given object, there exists a minimum scale s_{min} and a maximum scale s_{max} at which an organism can recognize the object [50]. As long as the optical hardware is not infinitely powerful, this must be true. Then, at the minimum scale s_{min}, there exists a set of translations (the "translation set" S_{min}), for which the organism can recognize the object [50]. A translation is defined by its eccentricity x—eccentricity is just the distance of an object from the center of focus. Since humans focus with the center of our field of

vision, we presumably achieve minimum-scale recognition where $x = 0$. Thus, we can think of S_{min} as all translations from $-x_{max}$ to x_{max}, for some maximum translation x_{max} (this will clearly be centered at $x = 0$) [50]. According to the eccentricity theory, *all elements of S_{min} should then be recognizable under arbitrary scaling (without translation), up to s_{max}* [50]. We call this the *scale-invariance requirement.*

We can understand the implications of this requirement by considering the geometry of scaling. Consider an object at scale s and position x. When this object is scaled (i.e. its distance to the observer changes), its apparent scale and position *both* change [50]. As a demonstration: one can try the following experiment:

1. Find a reasonably spacious room and locate an object on the wall (e.g. a light switch).

2. Visually fixate on a point on the wall, approximately 10 cm away from the object. Stand facing the wall, about one meter away, oriented towards the fixation point.

3. Try walking directly towards or away from the wall, maintaining the same point of fixation. Observe the target object using peripheral vision.

In the above experiment, one will notice two effects: (1) the object appears to grow larger or smaller (scaling); (2) the object appears to move towards the point of focus when decreasing in scale, and away from it when increasing in scale (translation). That is, the object's perceived *eccentricity* (distance from the center of focus) is linearly related to its *scale* (distance from the viewer) [50]. This effect disappears at the center of focus, where an object will not translate under scaling [50]. See [50] for further details.

Under these principles, for any recognizable object, the set of recognizable scales and shifts can be represented as an *inverted truncated pyramid* in the scale-space plane (Figure 3-1) [50]. The shape of the pyramid reflects the geometry of scaling—when an object outside of the center of fixation is viewed, its apparent eccentricity grows as its viewing distance decreases [50].

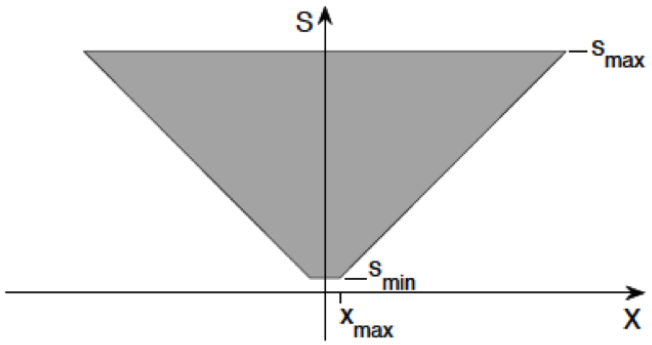


Figure 3-1: **Inverted truncated pyramid.** The X-axis is eccentricity; the S-axis is scale. Suppose an organism can recognize an object at some scale s_{min}, at eccentricity x_{max}. Then, as it moves closer to the object, the object's "trajectory" in the scale-space plane will follow the right edge of the inverted pyramid, up to s_{max} [50]. Suppose further that at scale s_{min}, the organism can recognize the object at all eccentricities x where $|x| \leq |x_{max}|$. This is called the 'translation set,' S_{min} [50] (this captures the shift-tolerance of the organism at the minimum scale). Then, the associated "scaling trajectories" will fill the pyramid, as shown [50]. If an organism can recognize everything in the pyramid, it will satisfy the scale-invariance requirement. (Figure is from [50].)

The eccentricity theory additionally assumes that the scales and positions in the inverted truncated pyramid are *sampled* by Gabor filters, which provide optimal translation- and scale-invariance [50]. This is compatible with biological observations of simple cells in the brain [58, 24]. The model conjectures that the number of sample points at each scale is constant, as shown in Figure 3-2 [50]. This would allow seamless sharing of representations among different scales.

Of course, a logical question to ask is, 'Why prioritize scale-invariance?' In response, [50] observes that it is energetically inexpensive to move the head or eyes of an organism to adjust for object translation, but much more fatiguing to move the body and adjust distance to objects. Thus, it is reasonable to support maximum generalization over scale, under optical constraints.

The eccentricity theory concludes that the human eye only needs to support maximum-resolution processing in a small central area called the *foveola*, supporting small-scale recognition [50]. This is represented by the lowest and most densely sampled pyramid layer in Figure 3-2. By definition, the foveola is at the very center of the

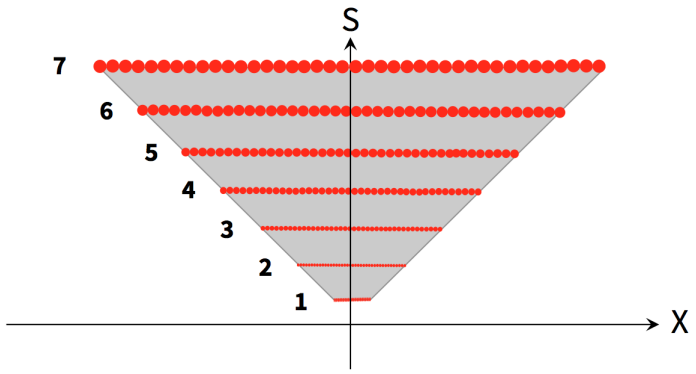


Figure 3-2: **Sample points in the inverted pyramid.** The X-axis is eccentricity; the S-axis is scale. Each red dot represents the center of a Gabor filter. In this specific example, each scale has 40 filters. There are 7 'scale channels' numbered from 1 to 7, increasing linearly in diameter from 7° at the bottom to 49° at the top. Gabor filter sizes also increase linearly. In the theory, filters have half overlap [50]. Together, the filters cover the eccentricities and scales in the pyramid, satisfying the scale-invariance requirement. This is based on a figure from [50].

fovea, which is the small central area (estimated to be $< 10°$ in diameter) of the visual field [50, 25, 1]. Besides the foveola, there can then be wider arrays of larger-scale, lower-resolution processing units, extending to the periphery (see Figure 3-2) [50]. This arrangement allows for generalization within the inverted pyramid, maintaining the scale-invariance requirement [50]. Notably, this does not require high-resolution processing across the entire visual field. The last property may be desirable due to resource constraints [50].

Given the sampling pattern of Figure 3-2, the eccentricity theory relies on pooling, both over space and scale, as a mechanism for achieving invariance [50]. It estimates about 4 stages of pooling, at major parts of the ventral stream [50]. As in the basic version of HMAX [58], each round of pooling follows a round of filtering, and processing is fundamentally feedforward [50].

Measurements of neural receptive field (RF) sizes in the ventral stream seem to support the theory. Figure 3-3 (originally from [10], displayed here for convenience) shows a summary of macaque monkey RF measurements. As the figure shows, RF sizes increase along the ventral stream, from V1 to V2 to V4. Furthermore, [7] asserts that IT neurons have RFs of 10° to 30° when measured with "standard RF mapping

methods." This is slightly larger than the range of V4 RFs in Figure 3-3, suggesting another RF size increase from V4 to IT. Since information flows from V1 to V2 to V4 to IT in the ventral stream [57], these data suggest at least one pooling stage per location. Intuitively, pooling increases spatial invariance, leading to larger RFs. This is consistent with the 4 stages of pooling in the eccentricity theory.

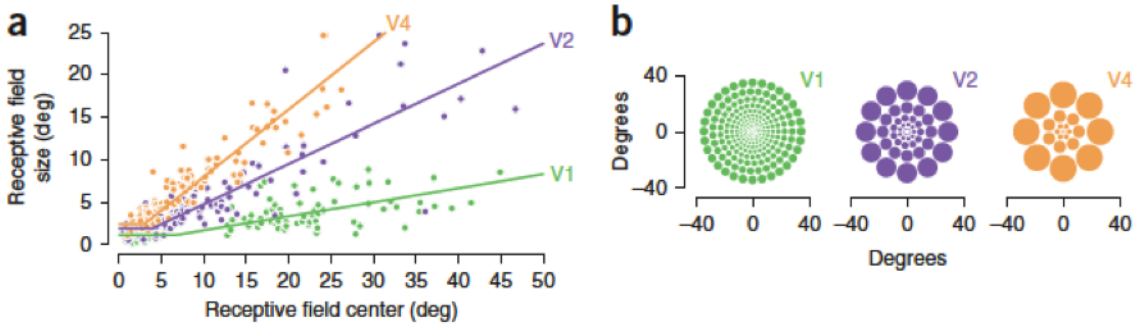


Figure 3-3: **a. Measured receptive fields (RFs) of macaque monkey neurons.** V1, V2, and V4 are progressively higher areas of the ventral stream; lines attempt to fit data points [10]. Increasing RF size in higher visual areas suggests progressive spatial pooling. **b. Probable spatial arrangement of RFs in V1, V2, and V4, considering biological data.** The larger the eccentricity, the larger the RF. The data are roughly consistent with *chevron sampling*, discussed in [50] and shown in Figure 3-4. Plots (a) and (b) are both from [10], and show data from [12] and [13].

Figure 3-3 also shows that RF sizes tend to increase linearly with eccentricity. This agrees with the theory (see Figure 3-2). However, if we consider the V1 RFs in Figure 3-3b, it seems that there are 'large' RF sizes that are represented in the periphery, but not in the fovea. This disagrees with Figure 3-2, which suggests that *all* RF sizes should be present in the fovea. The eccentricity theory explains this observation by conjecturing that larger scales of processing are *restricted to the periphery* [50]. This changes the inverted pyramid into a 'chevron' (Figure 3-4) [50]. This adaptation can be justified by considering the possibility of resource constraints in the brain [50]. With 'chevron sampling,' processing power is saved by restricting the number of scales processed at each eccentricity.

To summarize, the eccentricity theory unifies geometric and neurological observations into a biologically plausible model of visual processing. It makes a number of predictions regarding visual performance under scaling, shift, and clutter [50]. The

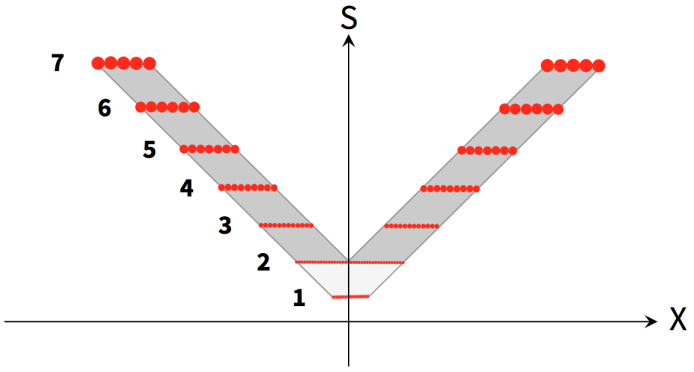


Figure 3-4: **Chevron sampling.** The X-axis is eccentricity; the S-axis is scale. The dimensions of scale channels and filters are the same as in Figure 3-2. However, some scale channels do not cover all eccentricities (e.g. observe the large gap in the 7th channel). The eccentricity theory imposes a "constant difference" requirement, stating that, "there is a constant difference between the largest and smallest scale at each eccentricity" [50]. In this example, at most two scales are active at any given eccentricity—we say that the *chevron parameter*, c, is 2. This means that the gap in channel 3 is exactly the size of channel 1, the gap in channel 4 is exactly the size of channel 2, etc. This satisfies the "constant difference" requirement mentioned above. Note that the light gray central region is an inverted pyramid, similar to Figure 3-2. This is based on a figure from [50].

following section describes our model, which implements the theory. We use the model to evaluate the predictions from [50].

3.2 The Model

Our CNN-based model is built to recognize grayscale handwritten digits from the MNIST dataset [38, 37]. This dataset is very lightweight, allowing for rapid training and testing. In addition, the task is difficult enough to reasonably evaluate the model. Here, we discuss the most important aspects of our architecture. Appendix A provides supporting details.

3.2.1 Architecture

Our model takes in grayscale images and outputs a 0-9 prediction of what digit is in a given image. Unlike a traditional 'flat' CNN operating at one scale, our model uses

a CNN operating at 7 different scales. It has 13 processing layers in total.

An input image first encounters 4 rounds of S-C-type processing. Each contains (1) a convolutional layer, followed by (2) a spatial pooling layer, followed by (3) a layer of pooling over scales. The last scale pooling layer is followed by a single fully-connected layer with 10 outputs (one for each possible digit class). The four convolutional layers and the fully-connected layer are learned with back-propagation. Note that our model has the same number of S-C layers as is estimated in the eccentricity theory [50]. This is also close to the estimated number of processing layers in the ventral stream ($O(10)$) [9, 57].

Figure 3-5 illustrates a simplified version of our model, emphasizing the main operations involved. We describe each processing stage in depth here.

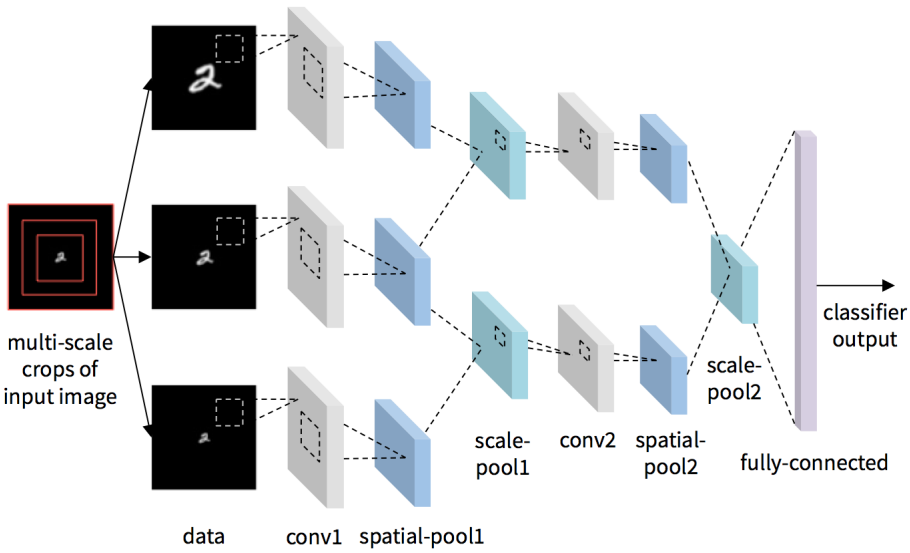


Figure 3-5: **Simplified version of our model.** This network contains the same basic building blocks as our final model, but is greatly simplified for clarity. `conv` layers perform convolution, `spatial-pool` layers perform spatial pooling, and `scale-pool` layers perform scale pooling. See Subsection 3.2.1 and Appendix A for more details. Note: this figure shows a model using 3 scales of input and performing 2 rounds of convolution and scale/space pooling. Our final model takes input at 7 scales and performs 4 rounds of convolution and scale/space pooling.

Input Data

As suggested by the eccentricity theory [50], our model processes input at many scales. More precisely, it takes in 7 *crops* of a given input image, one per scale channel. For our purposes, a crop is just a centered square cutout of an input image. Figure 3-6 shows the 7 input crops. Each crop is 83×83 pixels (px)—data are upsampled or downsampled as necessary to satisfy dimensions. Given the 28×28 px MNIST digits [37], an 83×83 px image size allows us to construct recognition tasks with scaling, translation, and clutter, without downsampling the digits beyond recognition. As shown in Figure 3-5, each crop is initially processed by a separate 'scale channel,' with its own convolution and spatial pooling layers. Subsequent scale pooling may combine the scale channels, so that fewer channels remain.

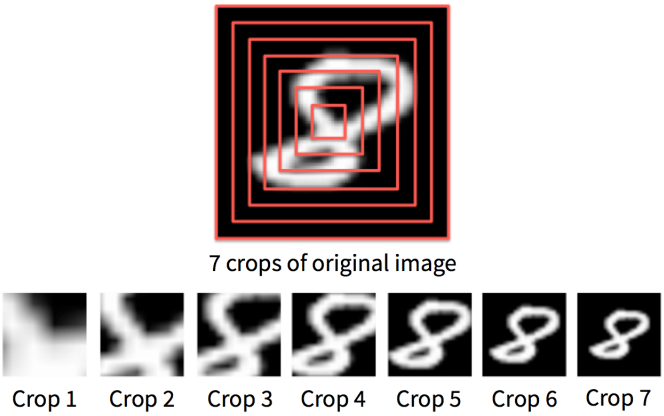


Figure 3-6: **Model input: 7 crops of an image.** Each crop is 83×83 px, with upsampling and downsampling conducted as necessary. Crops increase linearly in size from 1 (smallest) to 7 (largest)—crop 7 is exactly 7 times the size of crop 1. Crop i corresponds to scale channel i in Figure 3-2.

We use our input data to implement 'chevron sampling' (see Figure 3-4), constraining the network so that at most c scale channels are active at any given eccentricity. For a given crop, this simply means zeroing out all data from the crop c layers below. For example, consider crop 3 in Figure 3-6. With $c = 2$, we would take the area of crop 3 that is also covered by crop 1 (a square region in the center), and replace this area with all zeros (i.e. the black background). Thus, the first convolutional layer in crop 3 effectively does not 'see' anything in that area. We use $c = 2$ for every

simulation, except when exploring the effect of c in Section 4.4.

In order to facilitate comparison with human data, we developed a correspondence between image dimensions and physical dimensions. In our model, 83 px corresponds to 7° of visual angle, as viewed from a distance of 50 cm. Thus, 1° of visual angle is ≈12 px at the highest resolution. Crop sizes increase linearly in width by 7° per crop, from 7° at crop 1 to 49° at crop 7 (see Figure 3-6). This corresponds to a total field width of ≈46 cm at crop 7. We use 7 linearly spaced crops to approximate the linear increase in RF diameters with eccentricity [50]. The minimum crop size approximates the estimated fovea size in [50], and the maximum crop size reaches over 20° into the periphery. This allows us to replicate human experiments that involve both the fovea and the periphery [5, 1, 25, 50].

Note that our model's resolution is far lower than human vision. According to [42], the smallest human visual channel has at least 120 cone cells per degree of visual angle, an order of magnitude higher than our model's 12 px per degree. Though this reduces the fidelity of our model, it also allows for faster training and testing. This in turn allows for more simulations and iterations, to explore the parameter space. It would be relatively trivial to increase our model's resolution.

Convolution and Spatial Pooling Layers

The first convolutional layer, `conv1`, uses a kernel of size 5×5, with stride 2. This allows for 40×40 outputs in the first layer, with approximately half overlap between the filters, which follows the eccentricity theory's assumption [50]. Additionally, a 5×5 kernel allows for a rough approximation of the major features of a Gabor filter [17]. See Appendix A for examples of learned filters from `conv1`.

The first spatial pooling layer, `spatial-pool1`, uses a kernel of size 3×3, with stride 1. Thus, its RF diameter is approximately twice that of a convolutional unit. This follows the Hubel and Wiesel study on the primate brain [25], which found that complex cell RFs were about 1.5 to 2 times as wide as simple cell RFs. As in HMAX [53, 58], all pooling units perform the MAX operation, where the output is the maximum activation among the cells in the kernel. This basic paradigm of

convolution and MAX-pooling is known to perform well on digit recognition [30, 36] and natural image recognition [33].

In the following `conv` and `spatial-pool` layers, we use a 5×5 kernel with stride 1 for convolution, and a 3×3 kernel with stride 1 for pooling. Note that the kernels of these units always 'overlap,' even though the eccentricity theory suggests 2×2 'non-overlapping' pooling [50]. We have a few reasons for this choice. First, such 'overlapping' units intuitively preserve information and learning potential, by pooling over more sets of activations. Second, they have been used in high-performing neural networks [35, 38, 36, 33, 30]. Third, they mirror Hubel and Wiesel's finding of cells with overlapping RFs in the primate visual cortex [25].

Every `conv` and `spatial-pool` layer in each scale channel produces 32 channels of output. This allows for the learning of Gabor filters at various orientations and phases in `conv1`, and for the learning of various higher-level features in subsequent layers. In addition, `conv` units in the same layer are constrained to share weights across scale channels. We guarantee this during back-propagation by averaging the error derivatives over all scale channels, then using the averages to compute weight adjustments. We always apply the same set of weight adjustments to the `conv` units across different scale channels. All `conv` units use the ReLU non-linearity function, which has been shown to improve convergence rates in image recognition [33].

Table 3.1 shows the minimum and maximum possible RF diameter at each `conv` and `spatial-pool` layer, assuming no pooling between scale channels. Note that scale pooling would make RFs irregular and difficult to characterize, since they would contain information from multiple scales.

As an initial validation step, we plot our model's relationship between RF diameter and eccentricity (without considering scale pooling), comparing against the measurements in [10, 12, 13]. We assume that at most two scales are active at each eccentricity ($c = 2$, see Figure 3-4 and explanation of Input Data). Figure 3-7 shows the results. Note that the slopes and values of `conv1` and `spatial-pool1` are quite similar to the estimates for V1 in Figure 3-3. Similarly, V2 in Figure 3-3 has RFs similar to `conv2` and `spatial-pool2`, and V4 in Figure 3-3 has RFs similar to `conv3` and

Layer	Minimum RF Diameter	Maximum RF Diameter
`conv1`	0.4°	3.0°
`spatial-pool1`	0.8°	5.3°
`conv2`	1.4°	10.0°
`spatial-pool2`	1.8°	12.4°
`conv3`	2.4°	17.1°
`spatial-pool3`	2.8°	19.5°
`conv4`	3.5°	24.2°
`spatial-pool4`	3.8°	26.6°

Table 3.1: **Receptive field (RF) diameters in our model.** This assumes no pooling over scale and rounds to the nearest tenth. Since CNN RFs are square, 'diameter' refers to the side length. The minimum RF diameter occurs in crop 1's scale channel and the maximum occurs in crop 7's scale channel (see Figure 3-6).

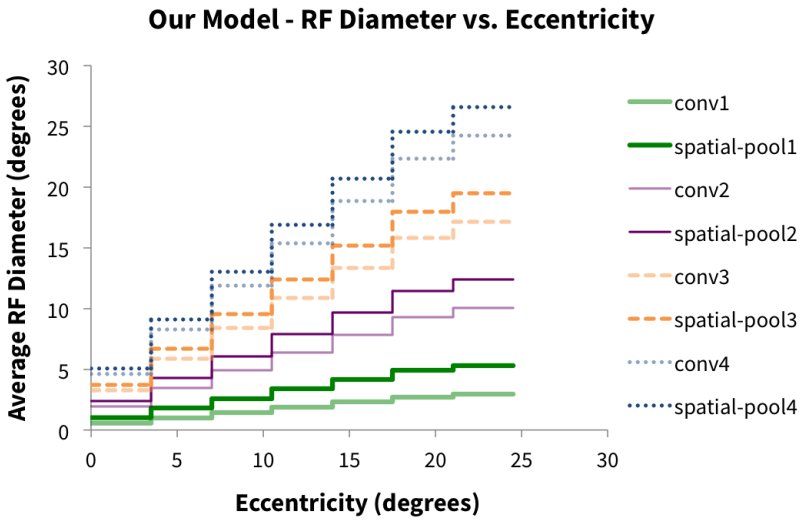


Figure 3-7: **Average receptive field (RF) diameter vs. eccentricity in our model.** The sharp 'cutoffs' in the plot indicate transitions between different 'eccentricity regimes.' This is a consequence of discretizing the visual field into 7 scale channels. Note that this ignores pooling over scales, to simplify RF characterization. Compare with Figure 3-3.

`spatial-pool3`. In this correspondence, `conv4` and `spatial-pool4` might map to IT, and the `fully-connected` layer might correspond to the pre-frontal cortex (given the ventral stream model discussed in [57]). Thus, our model seems to reproduce some aspects of neural organization in the brain.

Note that we have chosen these parameters as a four-way compromise between

(1) the eccentricity theory, (2) biological data, (3) performance, and (4) simplicity and ease of understanding. Though we use this parameterization for the bulk of our simulations, we have explored some variations, as documented in Chapters 4 and 5. Note that the principal goal of this thesis is to explain and produce intuitions for human visual phenomena. Rigorous parameter optimization is beyond the scope of this work, but is discussed in Chapter 7.

For further details on convolution and spatial pooling layers in our model, see Appendix A.

Scale Pooling Layers

After each spatial pooling layer in our model, we conduct a round of pooling over scales. This is defined as MAX-pooling over corresponding activations among every group of s neighboring scale channels. Figure 3-8 is a diagram of this operation. Pooling over scales will reduce the number of scale channels by $s - 1$.

Clearly, there are many possible different methods for pooling over scales. While exploring this space of possibilities, we allowed for some scale-pooling layers to be inactive, preserving all input scale channels. For clarity, we introduce a convention for discussing instances of our model with different scale pooling. We refer to a specific instance with a capital 'N' (for 'network'), followed by four integers. Each integer denotes the number of scale channels outputted by each `scale-pool` layer (there are always 7 scale channels before `scale-pool1`). For example, consider a model instance that conducts no scale pooling until the final `scale-pool` layer, then pools over all scales. We would call this N7771. The bulk of our results were generated with N6421 (we call this the 'incremental' model), and N1111 (we call this the 'early' model). See Appendix A for thorough descriptions of their architectures. We also explore different scale pooling methods, as documented in Chapters 4 and 5.

Fully Connected Layer

The output of `scale-pool4` goes through a `fully-connected` (FC) layer with 10 outputs. The FC layer's outputs correspond to 0-9 predictions regarding which digit

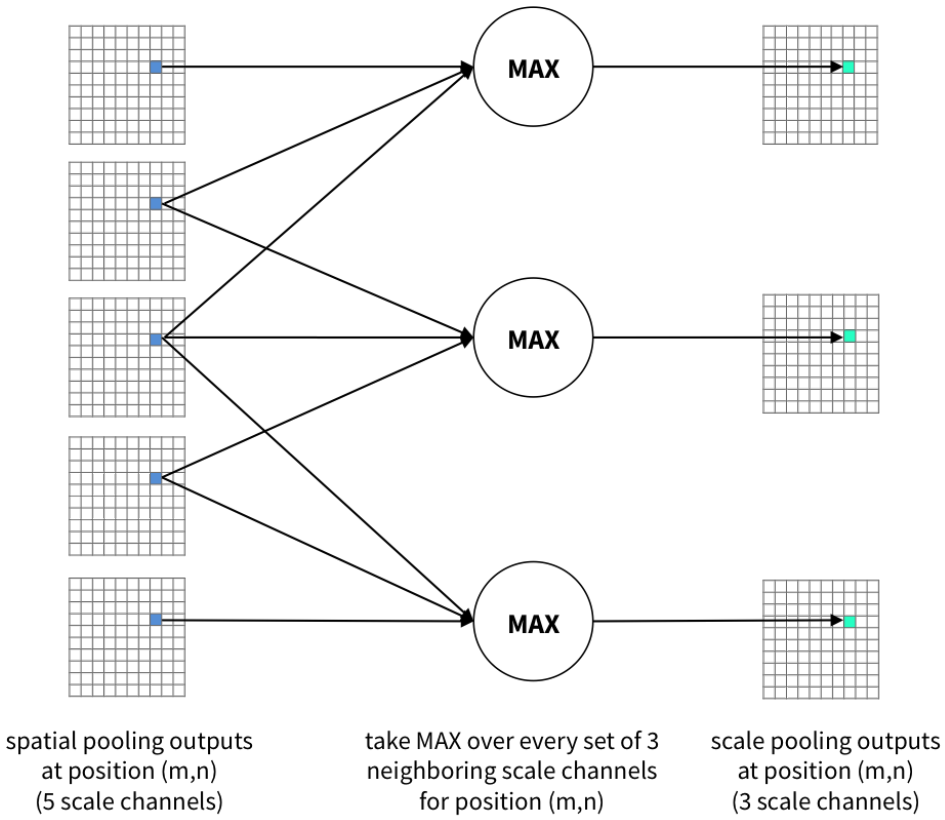


Figure 3-8: **Pooling over scale.** In this example, we have $s = 3$, such that we pool over every group of 3 neighboring scale channels. This reduces the number of scale channels from 5 to 3, i.e. by $s - 1 = 2$.

has been seen. This is a standard way of implementing a linear classifier [38]. Though some versions of the high-performing LeNet digit recognition CNN [38, 30] use multiple FC layers, we seek to confine learning mostly to S-C layers, since these map more directly onto known architectures in the brain. Our use of a linear classifier mirrors the neural decoding approaches in [39] and [26], which use support vector machine (SVM) classifiers with linear kernels.

3.2.2 Implementation

We trained and tested neural networks using the Caffe deep learning library [30]. Our image pre-processing and data formatting was done in C++, heavily relying on OpenCV [6] and using some modules from Caffe. Since Caffe does not implement scale channel operations (namely, averaged derivatives and pooling over scales),

we used pre-existing implementations of these operations, written in the lab using NVIDIA CUDA. We performed high-throughput simulation and automation using Python scripts, with the OpenMind High Performance Computing Cluster [43].

Appendix A contains more implementation details.

Chapter 4

Case Study: Transformation-Invariant Recognition

This chapter focuses on evaluating the model's performance under the transformations of scaling and translation. The general simulation paradigm is as follows: (1) train the model to recognize MNIST digits [37] at some position and scale, then (2) test the model's ability to recognize transformed digits.

This train-test paradigm follows the general approach of some scale- and translation-invariance experiments in human psychophysics [11, 46]. These experiments used short presentation times, below 200 ms, with backward masking. This time window is thought to allow mostly feedforward processing from a single saccade [39], and backward masking has been used to "block significant top-down effects" [57] and eliminate afterimages [8]. Thus, it is reasonable for us to treat these experiments as mostly feedforward, comparing them with our model.

In general, synthesizing the results of psychophysical studies [11, 46, 8] and neural decoding studies [39, 26, 40, 53], we expect to find scale-invariance over two octaves at least, while translation-invariance will likely be limited to shifts on the order of a few degrees (almost certainly less than 8° [8, 46, 40]).

4.1 Simulation: Invariance at the Fovea

This simulation mirrors [40], which used a train-test paradigm combined with neural recording from IT in primates. Specifically, monkeys were trained to recognize previously unfamiliar objects, through presentation of centered (foveal) objects at a 'default' scale [40, 53]. The subjects were then presented with scaled and translated versions of the stimuli. Electrode recordings found neurons that were scale-invariant, over a ~2-octave range on average [40, 53]. In addition, neurons exhibited an average translation invariance of ~2° from the 'default' centered position [40, 53].

4.1.1 Procedure and Results

We evaluated two instances of our model (See Section 3.2): the 'early' model (N1111), and the 'incremental' model (N6421). For comparison, we also tested a a low-resolution 'flat' CNN, and a high-resolution fully-connected network (FCN), both operating at one scale. See Appendix A for further details on these networks.

We trained the networks to recognize centered digits of 3° in height and width. We used the full MNIST training set of 60,000 digits [37], with around 6,000 labeled examples for each digit class.

Networks were tested using the full MNIST test set of 10,000 digits [37], with around 1,000 labeled examples per digit class. To test scale-invariance, we maintained the centered position of the digits while scaling by octaves (sizes rounded to the nearest pixel). To test translation-invariance, we maintained the 3° height of the digits while translating. Each test condition evaluated either scale- or translation-invariance, tested a single scale or translation, and used all 10,000 test digits. We used an equal number of left and right shifts in translation. Sample data is shown in Figure 4-1.

Note that MNIST digits are 20×20 px, centered in a 28×28 px black background [37]. Since we performed all of our calculations relative to the full 28×28 px images, measurements such as digit height and translation may be slightly skewed in all of our simulations. Since our goal is to perform *qualitative* evaluations, this

skew is unlikely to significantly affect our results.

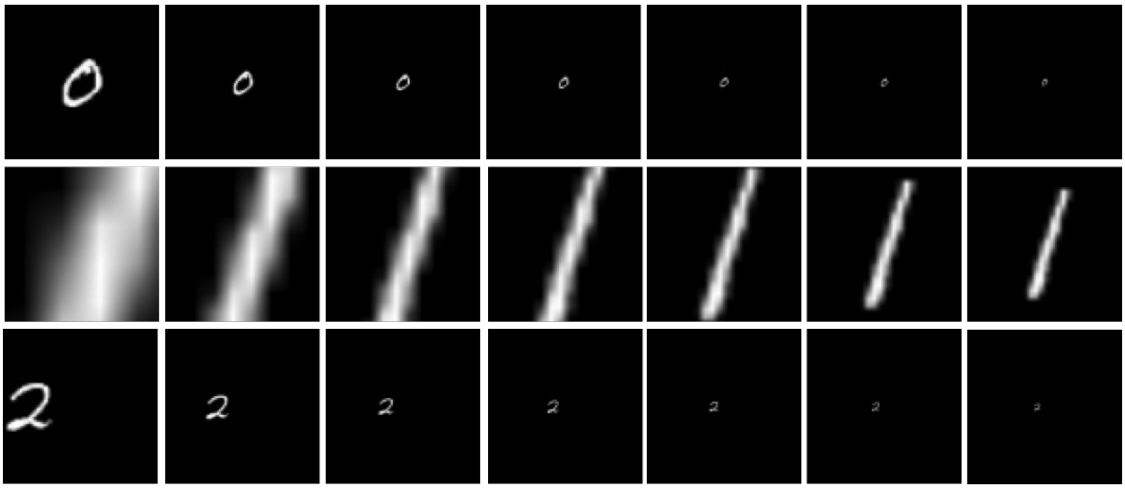


Figure 4-1: **Sample data for transformation-invariance simulations.** Each row shows the 7 crops seen by the CNNs, from 1 to 7 (see Figure 3-6). The top row shows the normal training digits. The middle row shows test digits, with 4 octaves of scaling. The bottom row shows test digits, translated by 28 px (12 px $\approx 1°$). The 'flat' CNN and FCN would see the the rightmost crops only, sampled to the proper resolution (see Appendix A). Though these seem barely legible on paper (this figure is not meant to represent the actual resolution of images), they are sufficient to allow over 90% accuracy in validation after training.

Results are shown in Figures 4-2 and 4-3. The exact amount of 'invariance' depends on our choice of threshold. For example, with a 70% accuracy threshold (well above chance), we observe scale-invariance for a 2-octave range, centered at the training height (at best). We observe translation-invariance for ~0.5° from the center. Our threshold may be less permissive than [53], which defined a monkey IT neuron as 'invariant' if it activated more for the transformed stimulus than for distractors.

4.1.2 Analysis

In general, our model shows prioritization of scale over translation, outperforming the other networks in scale-invariance, but not in translation-invariance. It shows scale-invariance over ~2 octaves, generally agreeing with psychophysical [11] and neurological [39, 26, 40, 53] results. It shows generally less translation invariance than is observed in neural decoding [26, 40, 53]. However, these values agree in spirit with

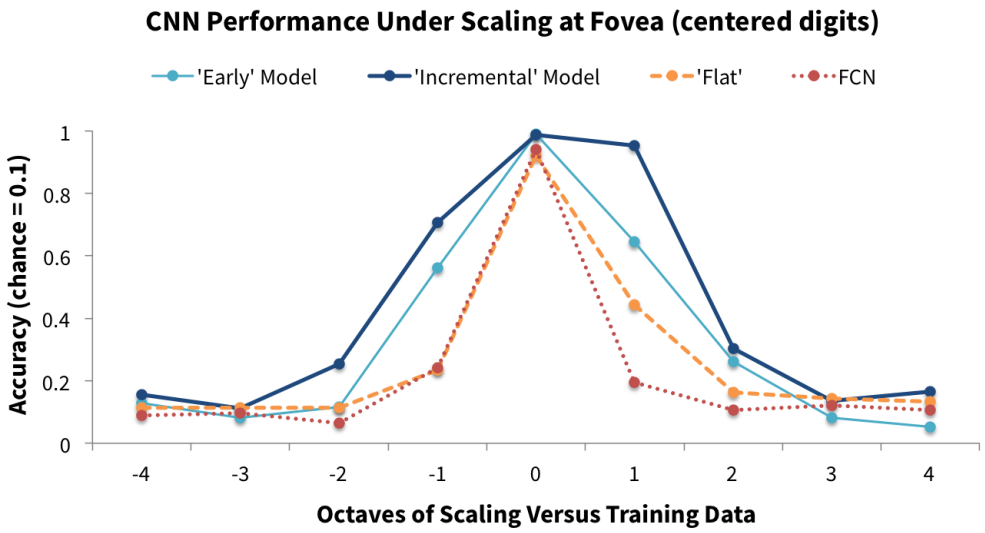


Figure 4-2: **CNN performance under scaling at fovea.** The 'incremental' model (N6421) shows the most scale-invariance (about two octaves centered at the training size, depending on threshold), followed by the 'early' model (N1111). The other neural networks perform well on the training scale, but do not generalize to other scales. Note the asymmetry, favoring larger scales. We hypothesize that this results from insufficient acuity (see Subsection 3.2.1).

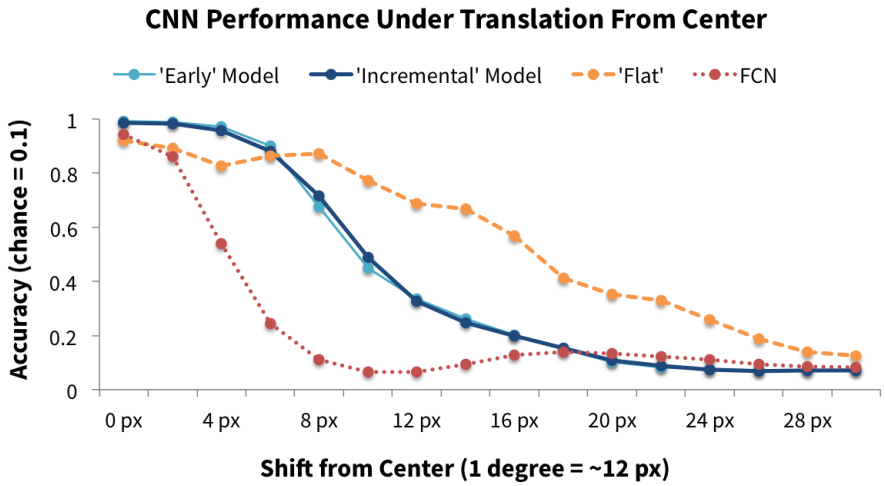


Figure 4-3: **CNN performance under translation from center.** The two instances of our model show similar translation invariance, which is generally less than the 'flat' CNN and more than the FCN. The 'amount' of translation-invariance depends on our choice of threshold, but in general, near-maximal accuracy is achieved within half a degree of the center.

the psychophysical study in [46]. Though [46] used training in the periphery instead of the fovea, it showed statistically significant classification performance variations for human subjects under translation. This could even occur for shifts of less than a degree [46], demonstrating incomplete invariance.

We hypothesize that the acuity of our model limits scale-invariance. This can be fixed by increasing the resolution at the cost of performance. Also, we find that different parameter settings lead to different invariance properties, as documented in Section 4.4. In particular, we can increase translation-invariance by simply increasing spatial pooling ranges.

4.2 Simulation: Limits of Acuity

Since neural RF sizes increase linearly with eccentricity [42, 50], it is reasonable to suppose that visual resolution decays accordingly, enforcing a bound on the amount of invariance that can be achieved. This effect was studied by Anstis [1]. Anstis plotted the "threshold letter height" of human subjects, i.e. the minimum height at which typed letters were "just identified," as a function of eccentricity [1]. This resulted in a linear relationship. In fact, the eccentricity theory predicts this behavior as a "special case" [50]. Our goal is to verify the correspondence between our model and Anstis' findings.

4.2.1 Procedure and Results

We sought to find, for a given eccentricity, the *minimum digit height* at which our model would learn with a certain 'threshold accuracy.' We chose 90% as our threshold, higher than the 70% threshold used in the previous simulation. Thus, we interpreted Anstis' study [1] to suppose that "threshold letters" are identifiable with high confidence.

We trained the 'incremental' model (N6421, see Subsection 3.2.1) on MNIST digits of various heights, at various eccentricities. (The 'early' model exhibited qualitatively similar results.) Each training condition contained the full MNIST training set [37],

at a single height and eccentricity. Eccentricity was measured as the distance between the center point and the innermost edge of the digit. Left and right shifts were used equally. Again, measurements may be slightly skewed by the small amounts of 'blank space' in MNIST data [37] (see Section 4.1).

For each training condition, we performed validation using the full MNIST test set [37], at the same height and eccentricity as for training. Some sample results are shown in Figure 4-4.

We used simple linear interpolation to estimate the minimum digit size needed for 90% validation accuracy, at each eccentricity. This allows a direct comparison with [1]. Results are shown in Figure 4-5.

4.2.2 Analysis

Clearly, our model shows the same qualitative trend as [1], with a linear relationship and higher thresholds near the periphery (presumably because fewer cells are active there). However, our slope and intercept for the regression line are significantly higher, despite having the correct order of magnitude. We have a few hypotheses for this behavior:

- The meaning of Anstis' "threshold height" [1] is uncertain when mapping this study to the computational domain. If we choose a lower threshold, instead of 90%, we might obtain a lower slope and/or intercept.

- As discussed in Subsection 3.2.1, our model has lower acuity than human vision—a tradeoff chosen to increase performance and allow more thorough testing and iteration.

- Handwritten digits are likely to be more difficult to recognize, compared to Anstis' typed letters [1], since they are much less uniform.

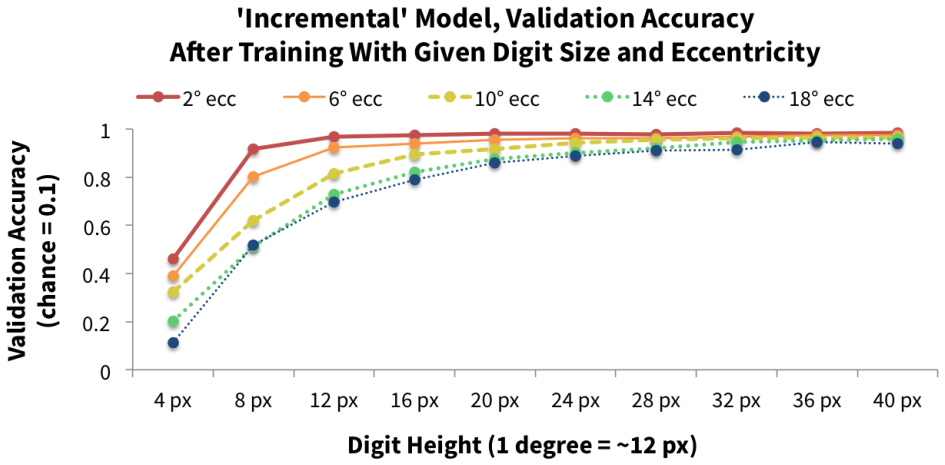
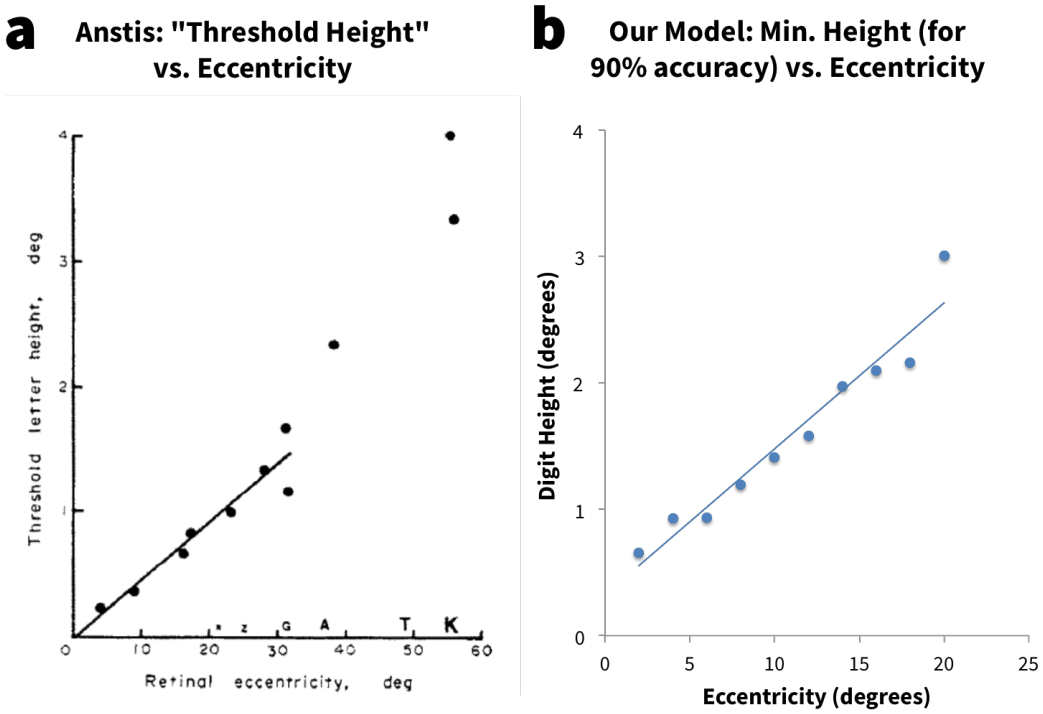


Figure 4-4: **Validation results from testing limits of model's acuity.** The higher the eccentricity, the larger the digit size needed for successful training and high validation accuracy.



$y = 0.046x - 0.031$ $y = 0.115x + 0.324$

Figure 4-5: **a. Acuity vs. eccentricity, as measured by Anstis [1].** The y-axis shows the minimum height at which subjects could "just identify" a typed letter; the x-axis shows eccentricity [1]. Equation of least-squares regression line [1] appears below. Figure is from [1]. **b. Acuity vs. eccentricity in our model.** Axes are in the same units; equation of least-squares regression line appears below.

4.3 Simulation: Scale vs. Translation

The eccentricity theory makes two predictions about the relationship between scale and translation [50]. First, it predicts "uniform scale invariance over a fixed range of scales independently of eccentricity" [50]. Intuitively, since scale channels overlap over almost all eccentricities, scale-invariance may also persist across eccentricities. Second, the eccentricity theory predicts that "there will be shift invariance that increases linearly with spatial wavelength" [50]. This can be explained by the notion that larger spatial wavelengths will be processed by coarser scale channels, which have wider pooling ranges. This simulation sought to test both predictions.

4.3.1 Procedure and Results

We only considered the 'incremental' model (N6421), since this showed more accurate scale-invariant behavior. We duplicated this procedure with the 'early' model (N1111), achieving qualitatively similar results.

To test the first prediction (consistent scale invariance at all translations), we trained our model using digits of 3° in height, randomly shifted along the horizontal meridian. We used all odd digits from the MNIST training set [37]—about 30,000 labeled examples. Using odd digits allowed us to use the same trained networks as in Section 5.1—see this section for an explanation. We would expect similar results if using all MNIST digits.

We tested the trained model using odd MNIST digits from the test set [37], at various scales and shifts (from the center). Each test condition contained about 5,000 exemplars at a single scale and shift. As before, left and right shifts were equally represented. Results are shown in Figure 4-6.

To test the second prediction (translation invariance is a constant multiple of spatial wavelength), we used scale as a proxy for spatial frequency. Intuitively, when a stimulus is at twice its original scale, all spatial wavelengths double. Thus, we trained instances of our model using centered MNIST digits [37] from the full training set. We used various training scales, with each instance receiving exemplars from only one

scale.

We tested the trained models using shifted digits, always at the same scale as the training digits. Each test condition used the full test set [37] at a single eccentricity and scale. Results are shown in Figure 4-7. For comparison, we performed the same training and testing on the 'flat' CNN and FCN, as in Section 4.1. Results appear in Figures 4-8 and 4-9.

4.3.2 Analysis

Figure 4-6 shows that the first prediction (consistent scale invariance at all translations) is *sometimes* correct. Namely, within 10° of the center, scales within an octave of the trained scale are consistently recognized with reasonably high accuracy. However, recognition of small digits falls off with shift—we hypothesize that acuity limits invariance. Considering the results in Section 4.2, this observation is unsurprising. In contrast, recognition of very large digits increases with shift—it is possible that the larger filters in the periphery are more adapted to this task. Thus, the prediction from [50] fails to capture some key aspects of performance.

Figure 4-7 shows that the second prediction (translation invariance is a constant multiple of spatial wavelength) is essentially upheld, allowing some room for interpretation. In other words, though we do not see a perfect linear relationship, the plots qualitatively support the prediction.

Also, comparing Figure 4-7 with Figures 4-9 and 4-8, we observe that the second prediction is somewhat upheld by the single-scale networks as well. However, our model is much more scale-invariant—unlike the single-scale networks, it converges for all digit sizes. In addition, it is more translation-invariant than the FCN and smoother than the 'flat' CNN in terms of decaying performance with eccentricity. This suggests that the eccentricity-dependent architecture helps to allow for a reasonable amount of translation invariance, while smoothing the decay of recognition with eccentricity.

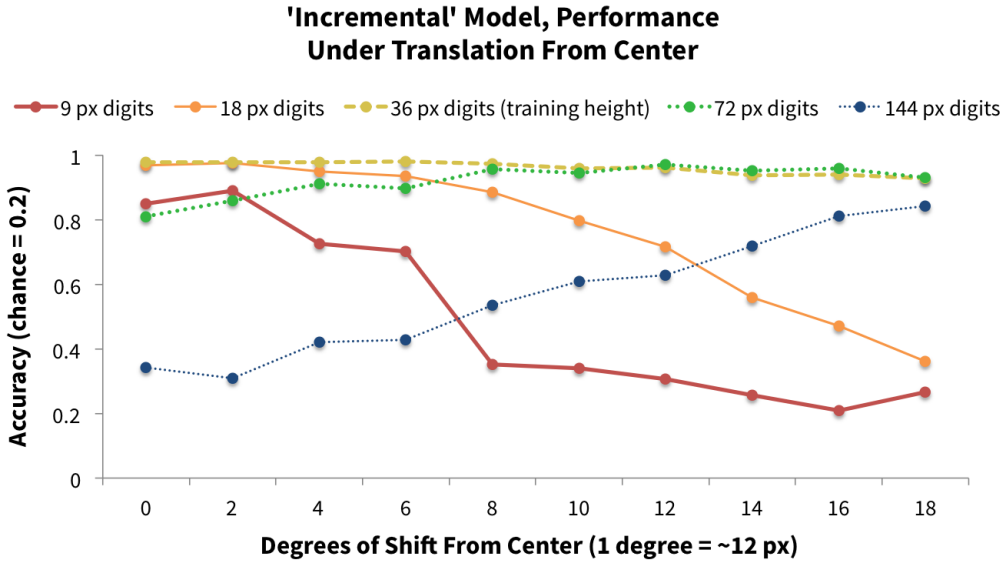


Figure 4-6: **Scale invariance vs. shift from center in our model.** The model shows reasonable scale-invariance within an octave of the training height, given a shift of $\leq$10 degrees from the center. We hypothesize that acuity effects account for the sloped lines at 9, 18, and 144 px.

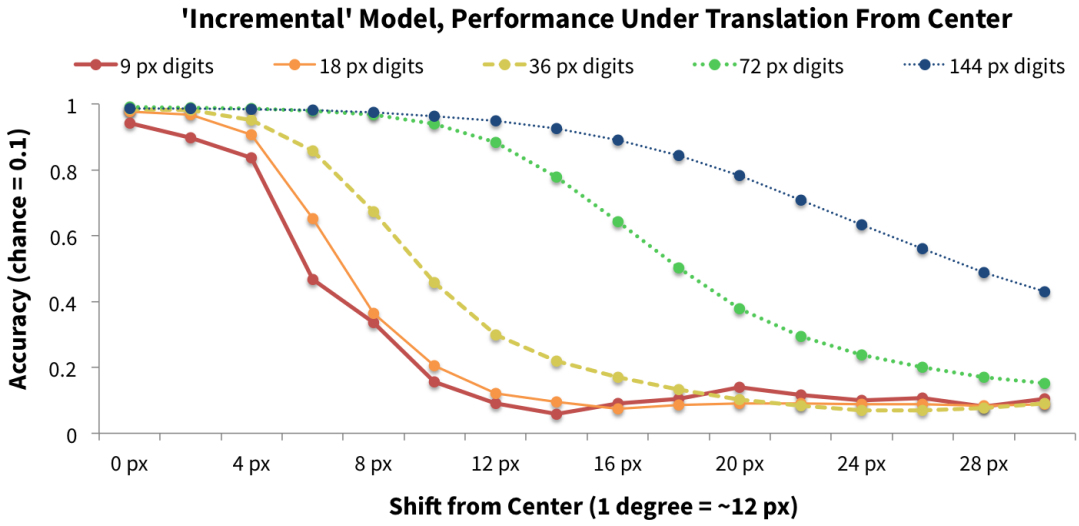


Figure 4-7: **Translation invariance vs. scale in our model.** Digit size increases in octaves. Consider the difference between the '50% accuracy intercept' along the 9 px line, vs. along the 18 px line. For digit sizes $\geq$ 18 px, doubling the digit size also approximately doubles the distance between the '50% accuracy intercept' for the larger size and the '50% accuracy intercept' for 9 px.

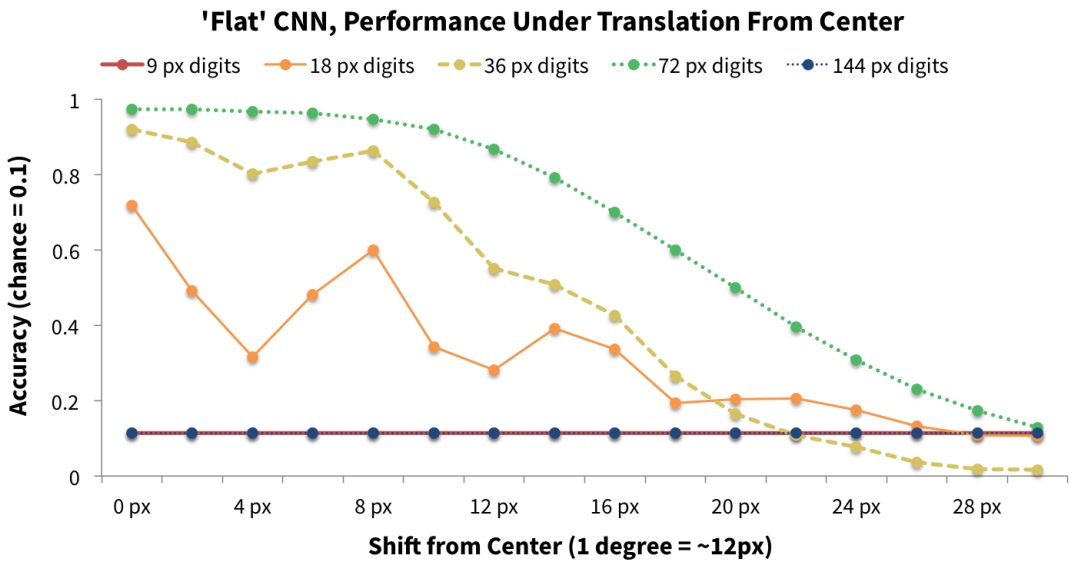


Figure 4-8: **Translation invariance vs. scale in 'flat' CNN.** Digit size increases in octaves. The 'flat' CNN can display more translation-invariance than the model, but has much more 'jagged' behavior under translation. In addition, it does not converge for the most extreme digit sizes of 9 px and 144 px (flat lines at bottom of plot).

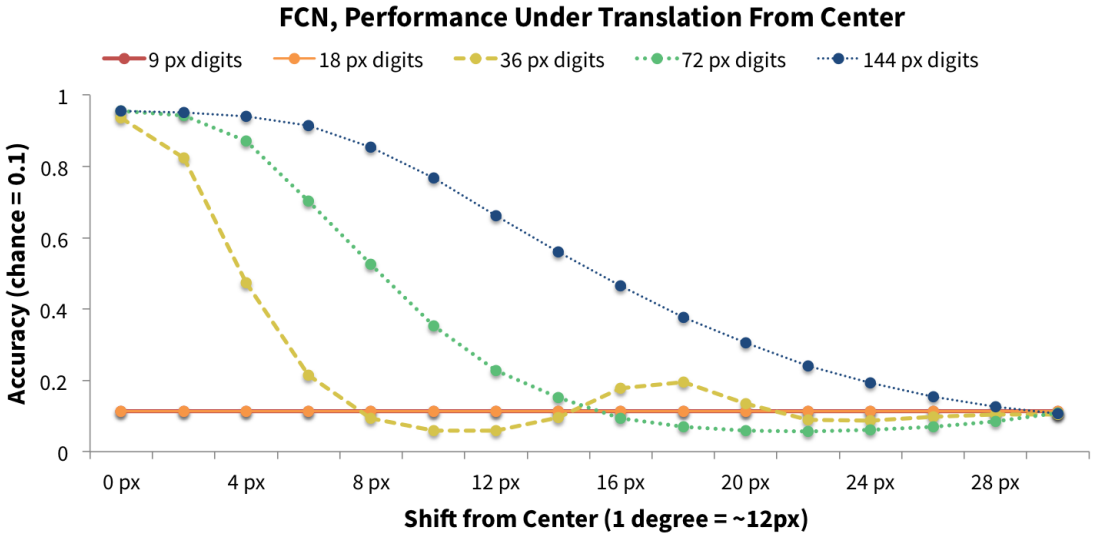


Figure 4-9: **Translation invariance vs. scale in FCN.** Digit size increases in octaves. The FCN displays approximately the same relationship as the model, but with less translation-invariance in general. In addition, it does not converge for the lowest digit sizes (flat lines at bottom of plot). We attribute the 'bump' for 36 px digits at 18 px shift to chance alignments of filters, producing slightly higher accuracy in an extraneous way.

4.4 Parameter Exploration

We repeated some of the above simulations with different parameter settings, to explore their effects. This could be useful for tuning the model to achieve desired properties. We varied only one parameter per simulation set, assuming mostly convex behavior in the parameter space. Though this may not necessarily be fully correct, it may provide some general intuitions. Training and testing procedures are the same as in Section 4.1. Our main findings are as follows:

- **Chevron parameter (see Figure 3-4) directly relates to scale-invariance.** A 'thinner' chevron has fewer active scales operating at each eccentricity and less scale-invariance. A 'fuller' chevron therefore has more scale-invariance. A full inverted pyramid (Figure 3-2) has the most scale-invariance. See Figure 4-10 for simulation results.

- **Larger spatial pooling ranges increase translation-invariance.** Intuitively, with wider spatial pooling, features can potentially occupy a larger 'field of possibilities,' without affecting recognition. See Figure 4-11 for simulation results.

- **Later scale pooling increases scale-invariance.** See Figure 4-12 for simulation results. Note that the 'incremental' model is actually not the most scale-invariant, though it is a high-performer. We chose to evaluate it both because it performs well and because it is neurologically plausible, given the biological evidence in favor of incremental pooling (see Section 3.1).

- **'Incomplete' scale pooling can result in lower scale-invariance.** 'Incomplete' scale pooling means that the model has multiple scale channels entering the `fully-connected` layer. See Figure 4-13 for simulation results. Although the eccentricity theory suggests pooling to $\approx$1 scale at the end [50], results imply that this may not be necessary.

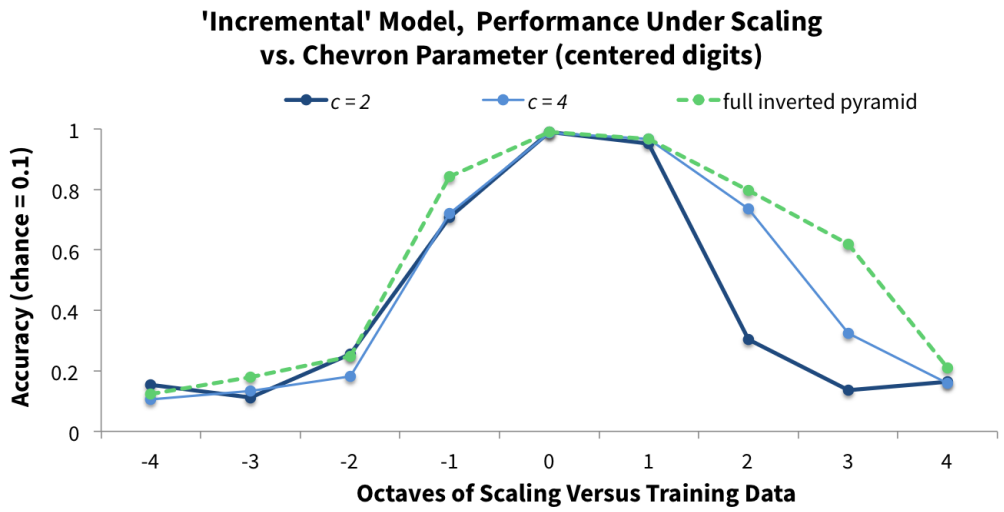


Figure 4-10: **Scale invariance vs. chevron parameter in our model.** See Figure 3-4 for the definition of the chevron parameter c. 'All scales' corresponds to a full inverted pyramid (Figure 3-2). Scale invariance increases with chevron parameter, reaching a maximum in the full inverted pyramid. Simulations done with 'incremental' model (N6421).

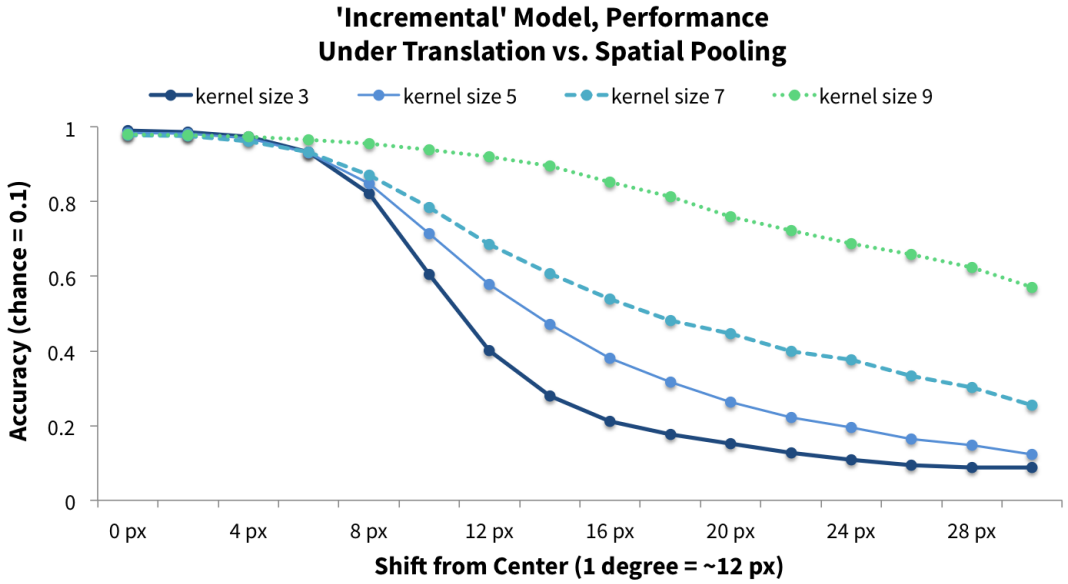


Figure 4-11: **Translation invariance vs. spatial pooling range in our model.** Wider spatial pooling ranges lead to more translation-invariance. We always use stride 1 pooling. Simulations done with 'incremental' model (N6421), $c = 2$ (see Subsection 3.2.1).

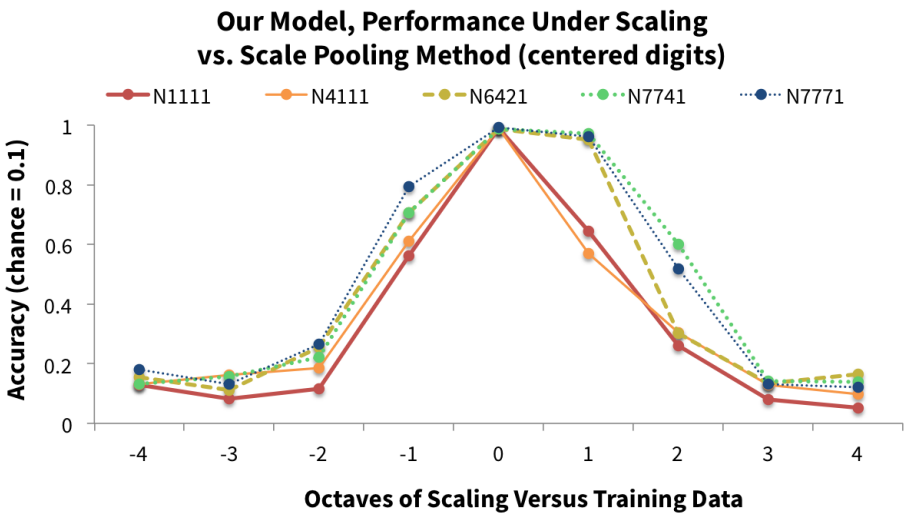


Figure 4-12: **Scale invariance vs. scale pooling method in our model.** Later scale pooling leads to better performance. The 'incremental' model, N6421, is among the high-performers, and is also neurologically plausible (see text in Section 4.4). Simulations done with $c = 2$ (see Subsection 3.2.1).

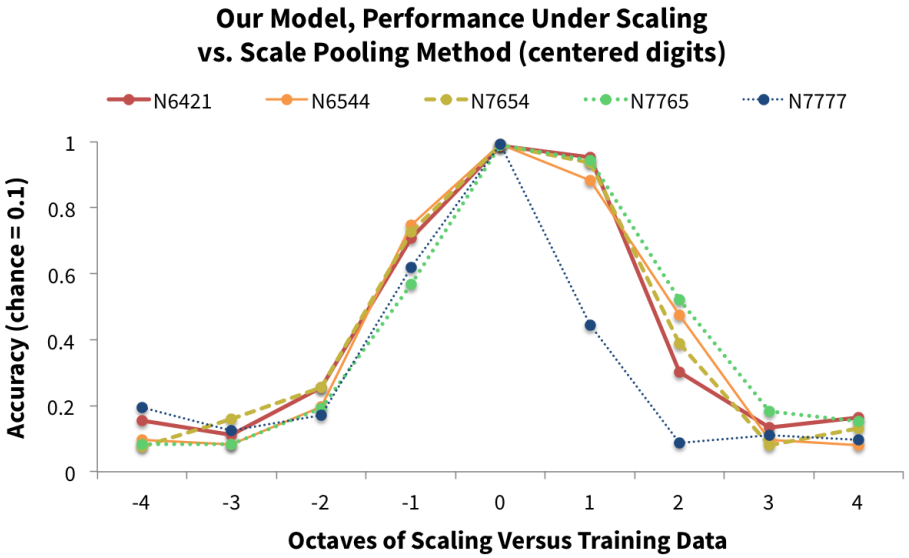


Figure 4-13: **Scale invariance vs. scale pooling method in our model, considering 'incomplete' pooling.** The model with no scale pooling, N7777, performs much worse, suggesting that some scale pooling is required for scale-invariance. Simulations done with $c = 2$ (see Subsection 3.2.1).

4.5 Synthesis

Our model exhibits significant scale-invariance and limited translation-invariance. This generally agrees with psychophysical [11, 46, 8] and neurological [40, 39, 26, 53] studies. Though the psychophysical studies somewhat vary in terms of timing and masking approaches, many were able to produce results while imposing short ($<$200 ms) time constraints on recognition-related processes [11, 46, 8, 39], suggesting that it is valid to compare them to a feedforward model.

We have also characterized our model in relation to the predictions from the eccentricity theory [50], mostly finding expected behavior. Acuity effects may account for much of the unexpected behavior observed in our simulations—this could be remedied with a higher-resolution model. In addition, the observations in Section 4.4 could allow for tuning of the model to better fit biological data.

Qualitatively, these results suggest that the model represents a viable approach for modeling the feedforward pathway of the human visual system. In particular, it demonstrates the key properties of scale- and translation-invariance, at approximately the right levels.

Chapter 5

Case Study: Recognition in Clutter

Whitney et al. [63] refer to the "deleterious effect of clutter on peripheral object recognition" as *crowding*. In other words, crowding is the decay of a subject's ability to recognize an object, when it is near other objects [63]. The study of crowding in human vision has intuitive value, since everyday recognition occurs almost exclusively in cluttered scenes. In addition, crowding is thought to relate strongly with dyslexia and other vision-related disorders [15, 63, 49], such that understanding it may have clinical value [63, 15].

Fortunately, the study of crowding is tractable from a feedforward perspective. A large number of significant crowding studies have used short presentation times ($\leq$200 ms) to produce their results [5, 20, 14, 15, 16, 32]. Some have also used backward masking, as mentioned in the previous chapter [32, 16]. Differences in specific timing and masking approaches make it difficult to exclusively attribute these studies' observations to feedforward processing. However, we look to [39] to justify applying our feedforward model as a first pass. In particular, it is estimated that average saccade times fall within ~300 ms, and that back-projections take ~200 ms to have significant impact [39]. Thus, it is reasonable for us to examine crowding from a feedforward perspective.

Of course, the feedforward model is imperfect. In the absence of a backward mask (occurring some or all of the time in [15, 5, 20, 32]), subjects may still be able to perform visual processing in the interstimulus interval (ISI) between the short

presentation and the report [57]. Thus, the impact of back-projections may not be entirely excluded [57, 39]. Kooi et al. [32] assessed this effect by performing crowding experiments both with backward masking (90 ms presentation) and without (150 ms presentation). Though the backward mask produced stronger crowding effects, qualitative results were similar. This further suggests that feedforward crowding models can capture key aspects of crowding. We additionally observe that other feedforward models [4, 59] have been able to explain crowding-related effects. Thus, we assert that our feedforward approach is a valid step in the study of crowding.

We note that crowding is a complex phenomenon, thought to involve multiple locations along the ventral stream [63]. Unsurprisingly, there are many models and explanations for crowding, using population coding [18, 62], statistics [31, 4, 10, 45], and attention [20, 45], among other concepts. In fact, the original HMAX model reproduced crowding effects by implementing eccentricity-dependent receptive fields [28, 27].

Our goal in these simulations is not to propose a new unifying model of crowding; rather, it is to demonstrate the viability of our approach. CNNs have a unique advantage over pre-existing tools—they are more powerful and general than other object recognition techniques [33], and can explain IT responses more than any other known model [64]. Additionally, as discussed in Chapter 2, CNNs have some functional similarities with cells in the ventral stream, especially when combined with the eccentricity theory [50]. This combination of generality and neurological plausibility is well-suited to the study of crowding, which is known to both (1) apply to many different stimuli, and (2) vary strongly based on stimulus characteristics. For example, crowding is known to affect perception of Gabor patches, letters, and faces [63], which intuitively seem to be different combinational 'levels' of stimuli. Also, its strength is known to depend partially on stimulus type (letters vs. keyboard symbols in [16, 63]), and partially on stimulus characteristics like color and shape [32, 63]. A general computational model, capable of recognizing diverse object types, would clearly be an invaluable tool for the study of crowding.

We performed three simulations to evaluate our model's ability to explain crowding. In addition, we investigated the effect of different parameterizations on the

model's performance.

5.1 Simulation: Bouma's Law

Bouma's Law [5] is sometimes referred to as the essential distinguishing characteristic of crowding [59, 48, 49]. Bouma's Law states that for recognition accuracy equal to the uncrowded case, the spacing between flankers and the target must be at least half of the target eccentricity (see Figure 5-1). This is often expressed by stating that the *critical spacing ratio* must be at least b, where $b \approx 0.5$ [63]. Note that stimulus characteristics can influence the value of b [63, 16, 32].

Figure 5-1: **Explaining Bouma's Law of Crowding.** The cross marks a subject's fixation point. The 'T' is the target, and each 'F' is a flanker. For recognition accuracy equal to the uncrowded case, flanker spacing (the length of *each* red arrow) must be at least half of target eccentricity. Note that we measure target eccentricity and flanker spacing in an 'edge-to-edge' manner, for compatibility with the original statement of Bouma's Law [5, 63, 59]. While it is currently more common to measure these distances in a 'center-to-center' manner, this would require us to express the Law slightly differently [59]. Note that this choice of convention could slightly skew our results, relative to other findings.

Canonically, a crowding experiment might show a target and two flankers to the subject, much like in Figure 5-1 [63, 5]. The subject is asked to name the target (the middle stimulus). If the results are plotted on curves with constant flanker spacing, they will resemble Figure 5-2 (from [5], shown here for convenience).

This experimental condition is difficult to replicate with a conventional CNN, which only gives one classification output. This is because it is non-trivial to specify a specific stimulus (e.g. the 'middle' one) for a feedforward CNN to report, given an input such as in Figure 5-1. Thus, we used a different task to evaluate our model.

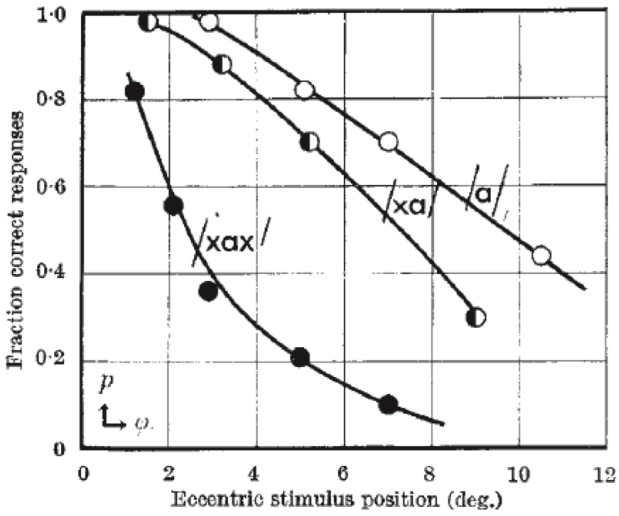


Figure 5-2: **Original Bouma's Law results.** Flanker spacing is held constant. The bottom curve represents two flankers, as in Figure 5-1, the middle represents a single flanker on the target's foveal side, and the top represents an unflanked target. Note the downward slopes with increasing target eccentricity. Figure is from [5].

5.1.1 Procedure and Results

We trained CNNs on all odd digits within the MNIST training set [37]—a total of around 30,000 digits, with ~6,000 exemplars per class. Training digits were randomly shifted along the horizontal meridian. Crowding experiments often assume that subjects can identify eccentric letters without requiring extensive training [5, 16, 15]. Thus, it is reasonable for us to prepare our CNNs in this way.

When testing networks in clutter conditions, we used even digits as flankers and odd digits as targets, drawing both from the 10,000 digit MNIST test set [37]. Flankers were always identical to one other. In order to evaluate our model against Bouma's Law, we tested several different target eccentricities (1°, 3°, 5°, 7°, and 9°) and flanker spacings (0°, 2°, 4°, 6°, and 8°). We evaluated all 25 possible combinations. Each test condition used all ~5,000 odd exemplar targets and involved a single target eccentricity and flanker spacing. Targets used an even distribution of left and right shifts, and we always placed two flankers radially (see Figure 5-1). Sample data is shown in Figure 5-3.

Since the CNNs were not trained to identify even digits, they would essentially

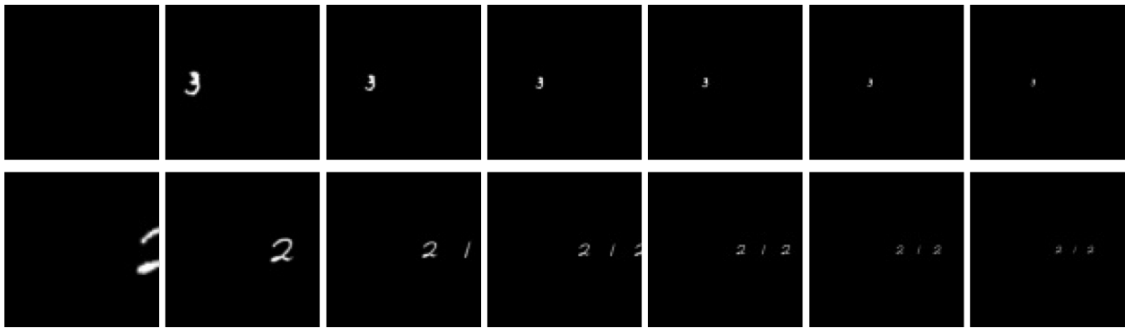


Figure 5-3: **Sample data for crowding simulations.** Each row shows the 7 crops seen by the CNNs, from 1 to 7 (see Figure 3-6). The top row shows shifted training digits (note that the shift moves the digit outside of crop 1). The bottom row shows test digits—the '1' is the target; each '2' is a flanker. Target eccentricity is 7°; flanker spacing is 2°. The 'flat' CNN would see the the rightmost crops only, sampled to the proper resolution (see Appendix A). Though these seem barely legible on paper (this figure is not meant to represent the actual resolution of images), they are sufficient to allow over 90% accuracy in validation after training.

never report an even digit class. Thus, we could monitor whether the information from the odd digit 'survived' crowding enough to allow correct classification. In general, even digits served to clutter odd digits only. The psychophysical analogue of this task would be an instruction to 'Name the *odd* digit', with a subject having no prior knowledge of its spatial position.

We evaluated 3 CNNs: the 'early' model (N1111), the 'incremental' model (N6421), and the 'flat' CNN. We excluded the FCN because it failed to achieve convergence during training. Results at two target eccentricities are shown in Figures 5-4 and 5-5. The 'early' model shows some Bouma-like behavior.

We can view our results from another angle by considering Whitney and Levi's description [63], shown in Figure 5-6 for convenience. For comparison, we plot our results for the 'early' model using the same convention in Figure 5-7. Our results show a similar trend.

5.1.2 Analysis

We do not directly compare the model's results to Bouma's original plots—since our model lacks a selection mechanism, this may not be a worthwhile comparison.

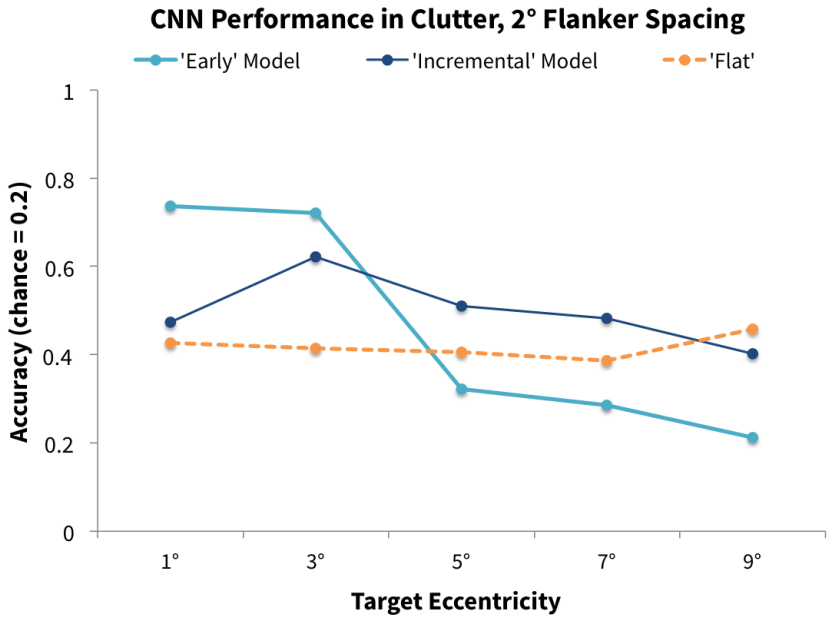


Figure 5-4: **Evaluating our model against Bouma's Law, 2° flanker spacing.** For the 'early' model, note the transition from high to low accuracy from 3° to 5° target eccentricity. This would be consistent with $0.4 \leq b \leq 0.67$. The other CNNs do not share this property.

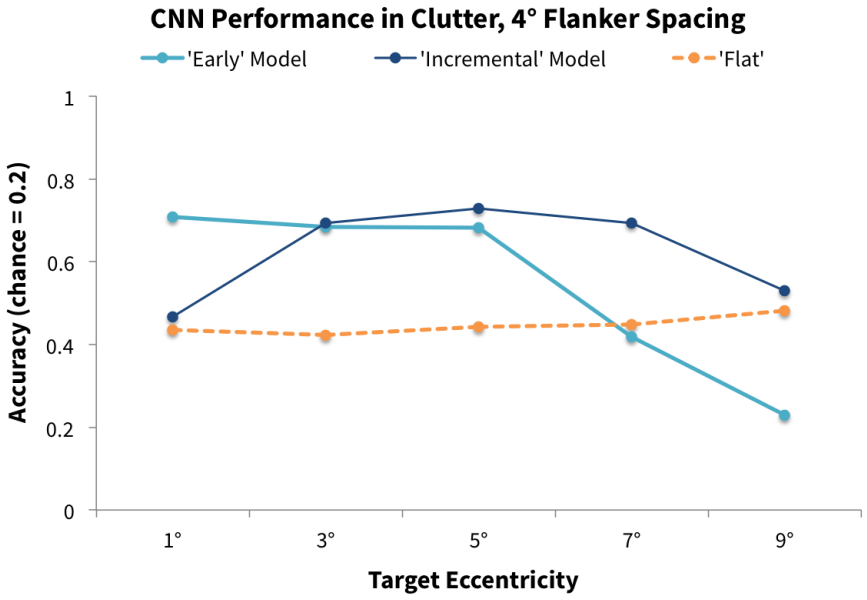


Figure 5-5: **Evaluating our model against Bouma's Law, 4° flanker spacing.** For the 'early' model, note the transition from high to low accuracy, from 5° to 7° target eccentricity. This would be consistent with $0.57 \leq b \leq 0.8$. The other CNNs do not share this property.

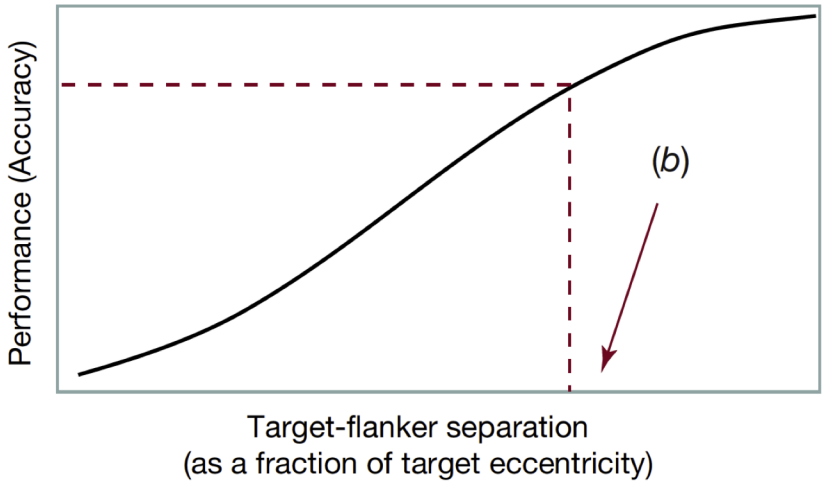


Figure 5-6: **Whitney and Levi's description of Bouma's Law [63].** When target eccentricity is held constant, we see a quasi-logistic curve of performance, which saturates at around b. Figure is from [63].

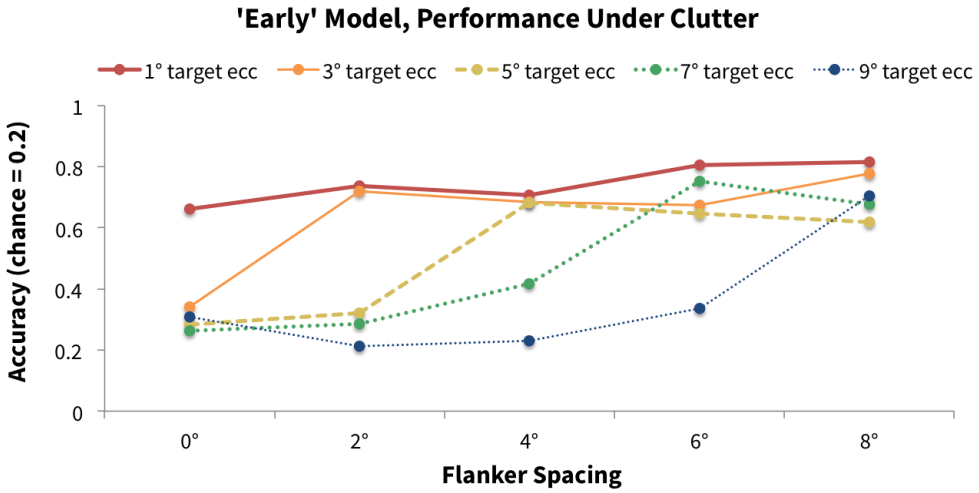


Figure 5-7: **'Early' model performance under clutter.** Each line has constant target eccentricity. We observe the same quasi-logistic trend as is shown in Figure 5-6. Since the model achieves 94% accuracy in the training condition, we believe that the 'saturation' at 70% occurs because we have no mechanism for selecting which digit to report. Thus, the model sometimes attempts to classify the flankers.

Nevertheless, the 'early' model exhibits qualitatively reasonable behaviors, with a dropoff of accuracy approximately at the expected points. This cannot be explained solely by a flanker being in the model's 'fovea'—(1) the model was trained with random shifts covering all eccentricities, so it has no explicit reason to be fovea-biased; (2) if this were true, then we would expect low accuracy in Figure 5-4 with target eccentricity 3°, and in Figure 5-5 with target eccentricity 5°. These cases would place a flanker directly on the model's 'fovea,' yet they exhibit high accuracy.

Considering Figures 5-4 and 5-5, our estimates for Bouma's constant b tend to be somewhat higher than the canonical value of 0.5. However, they are generally within or close to the interquartile range of 0.3 to 0.7 from published results, found by Pelli [48, 59]. However, our b-estimates tend to increase with target eccentricity, which could indicate an issue with our model. We have not yet produced a principled hypothesis for this effect.

The most interesting question to ask is, '*Why* does early scale pooling seem to reproduce the Bouma effect?' To answer this question, we note that the *cortical magnification factor*, i.e. the amount of cortex devoted to processing a given amount of visual area, decreases with eccentricity [50]. Also, some studies suggest that the 'critical cortical spacing' in V1 is constant (~6 mm) at all eccentricities—if radially adjacent signatures from two objects are closer than this in cortical space, they will crowd [49, 47, 63].

In our model, early pooling over scale essentially performs an eccentricity-dependent mapping from visual space onto cortical space, such that signals from different eccentricities are combined in the cortical domain. The subsequent convolutional layer, `conv2`, may then 'decide' whether the scale-pooled signals are crowded, demanding a constant amount of 'cortical spacing.'

This hypothesis also provides an intuition behind the value of b. To explain this, we first note that the eccentricity theory " 'predicts' " Bouma's Law as a consequence of the linear relationship between RF size and eccentricity [50]. Using a value of $b \approx 0.4$, it further observes that this constant matches the slope of V2 RF size, when plotted against eccentricity [50]. This was also observed in [10], which succeeded in

computationally predicting crowding effects by modeling V2 (see Figure 3-3). This implies that V2 may 'determine' whether or not signals are crowded. The supposed involvement of V2 fits neatly with our data—since all scales are combined before V2 in the 'early' model, V2 can indeed 'determine' whether features crowd, using a constant 'critical cortical spacing' and approximately the right RF size (see Figure 3-7).

5.2 Simulation: Anisotropy

Crowding is known to be *anisotropic*—radial flankers crowd more than tangential ones (see Figure 5-8) [63]. Explaining this effect has been somewhat difficult—it has been asserted that, "Currently, there is no satisfactory explanation for radial-tangential anisotropy" [45]. The authors of [45] present a possible hypothesis based on saccades, showing that it can explain radial-tangential anisotropy.

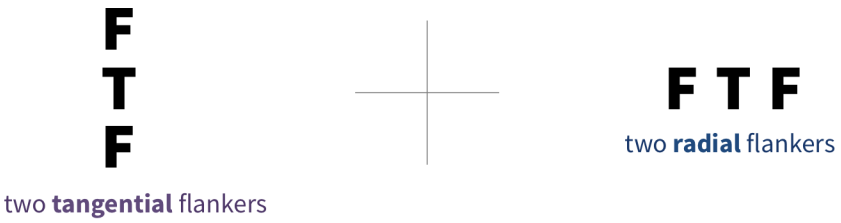


Figure 5-8: **Radial versus tangential flankers.** The cross marks a subject's fixation point. Each 'T' is a target, and each 'F' is a flanker. The left side shows tangential flanking; the right side shows radial flanking. In this case, target eccentricity and flanker spacing are approximately the same for both. Given proper fixation, we expect the target on the right to be harder to recognize [63].

Our model enables an alternative hypothesis that does not depend on saccades, and is therefore more generalizable to the short presentation times of crowding studies. Specifically, consider the 'early' model. Since this model pools over all scales as early as possible, it may pool more aggressively in the radial direction than in the tangential direction. This is because scale pooling always combines signals in the same direction from the center of focus, while spatial pooling operates approximately equally in all directions. Thus, at least at `conv2`, radial signals may interfere over longer distances, compared to tangential signals. This could be a direct result of the full scale pooling

before `conv2` in the 'early' model. Such a hypothesis agrees with the observations in [47], which states that the V1 'critical cortical spacing' is only ~1 mm in the "circumferential direction" (near the tangential direction), while it is ~6 mm radially.

We tested this hypothesis using a paradigm similar to that of Section 5.1.

5.2.1 Procedure and Results

The training and test sets here were nearly identical to those from Section 5.1, with two principal differences. First, the training digits were randomly shifted vertically *and* horizontally, to avoid biasing the model. Second, for each target eccentricity and flanker spacing, we generated a full test set representing 2 tangential flankers, in addition to the full test set representing 2 radial flankers (see Figure 5-8). We only considered the 'early' model, N1111, since this produced more correct behavior in the previous simulation.

We considered all 25 combinations of target eccentricities and flanker spacings, as in Section 5.1. As before, left and right shifts were equally represented. This yielded 25 test conditions in total. Results are shown in Figure 5-9.

5.2.2 Analysis

We observe the correct radial-tangential anisotropy in our model, as predicted by our explanation. However, any conclusions at this point are tentative at best—we have not thoroughly evaluated the diverse possibilities of flanking (e.g. one flanker, various stimulus types, etc.). We also have not implemented a selection mechanism, as discussed previously. Nevertheless, this represents a principled explanation of radial-tangential anisotropy, a key effect in crowding which has historically been difficult to understand [63, 45].

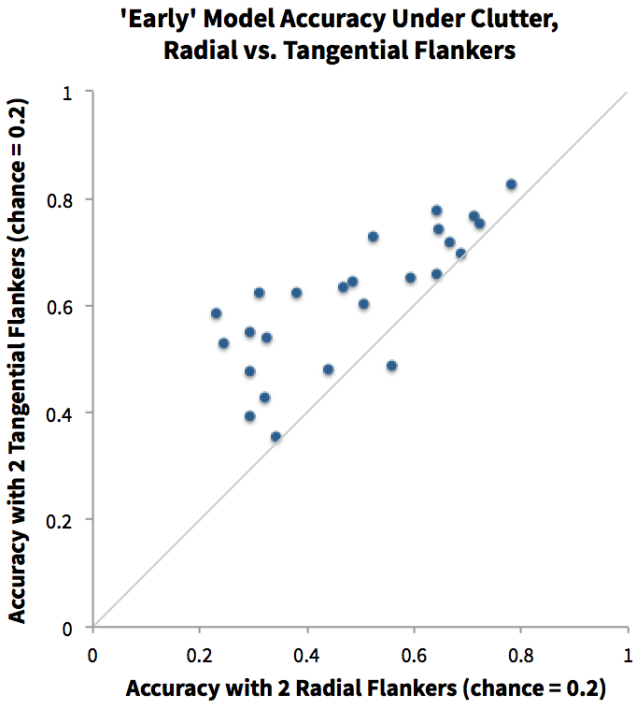


Figure 5-9: **'Early' model performance under clutter, radial vs. tangential flankers.** In all but one of the 25 test conditions (each with a unique combination of target eccentricity and flanker spacing), tangential flankers do not crowd as strongly as radial ones, allowing for better recognition.

5.3 Simulation: Asymmetry

Crowding is *asymmetric*—in the case of radial flankers, a single flanker towards the periphery will crowd more than a single flanker towards the fovea, with equal flanker spacing [63, 5]. Whitney and Levi explain this neatly, observing that, "the far flanker is actually closer to the target than the near flanker after mapping to cortical space" [63]. A population coding model of crowding successfully used this property to reproduce the asymmetric effect of crowding [62].

The analogous observation in our model is that pooling regions are larger towards the periphery. Thus, it may be more likely for peripheral flankers to interfere with the target, compared to foveal flankers. When we test our model, however, it generally does *not* show correct asymmetry properties.

5.3.1 Procedure and Results

Again, our training and test sets were nearly identical to those from Section 5.1 (we actually used the same trained networks as for Section 5.1). The difference was that for each target eccentricity and flanker spacing, we generated a full test set representing one foveal flanker and a full test set representing one peripheral flanker. We considered both the 'early' model, N1111, and the 'incremental' model, N6421, using the same set of 25 test conditions as in the previous simulations.

We find that both models produce the incorrect asymmetry quite consistently. The only exception is for small flanker spacings and small target eccentricities, where the 'incremental' model sometimes succeeds (see Figure 5-10).

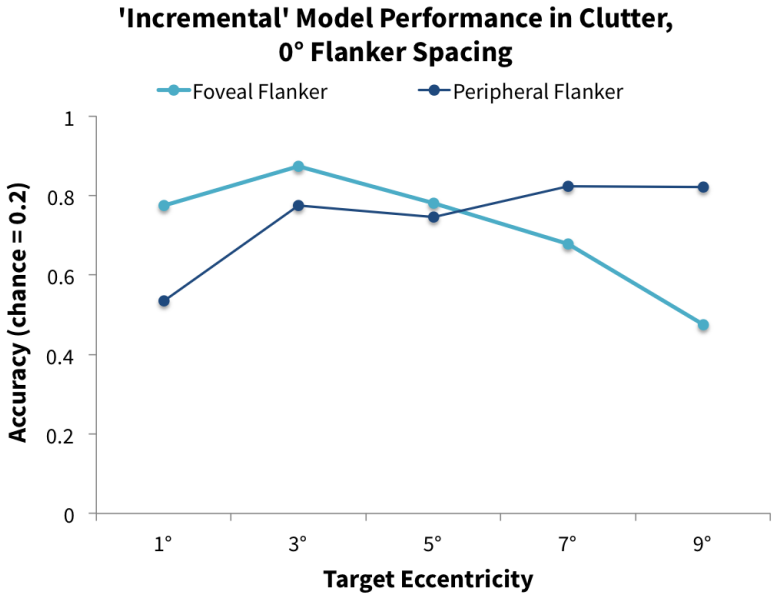


Figure 5-10: **'Incremental' model performance under clutter, foveal vs. peripheral flanker, 0° flanker spacing.** For small flanker spacings and target eccentricities, the model displays the correct asymmetry. However, it is incorrect in most cases—when either flanker spacing or target eccentricity grows larger.

5.3.2 Analysis

Though our model largely fails to explain asymmetry, the 'incremental' model can occasionally demonstrate it for small eccentricities and small target spacings. This suggests that some small part of the phenomenon may be explained by the model. We

hypothesize that the model's lack of a selection mechanism, as discussed previously, inhibits its ability to explain asymmetry.

5.4 Parameter Exploration

As in Section 4.4, we explored the effects of some different parameterizations. We summarize a few general trends, assuming mostly convex behavior in the parameter space. Training and testing procedures are the same as in Section 5.1.

- **General Bouma's Law trends persist with larger spatial pooling, but details somewhat change.** See Figure 5-11 for simulation results. The general eccentricity-dependence remains correct, but 'cliffs' do not necessarily occur at the same time. This could suggest different values of b for each parameterization.

- **'Early' scale pooling explains Bouma's Law most effectively in our model.** See Figures 5-12 and 5-13 for simulation results. The 'early' model displays the most qualitatively correct pattern of eccentricity dependence, by far. This striking observation implicates early scale pooling as the leading candidate for explaining Bouma's Law.

5.5 Synthesis

Our model exhibits some of the properties of human vision under clutter—namely, Bouma's Law and radial-tangential anisotropy [63]. However, it fails to reproduce foveal-peripheral asymmetry [63].

A prominent shortcoming of our current approach is the lack of a selection mechanism, as discussed previously. Even without this mechanism, our model demonstrates promise in explaining aspects of crowding with a general and biologically plausible object recognition technique. Future work (see Chapter 7) could address this issue and further extend our model's explanatory capabilities.

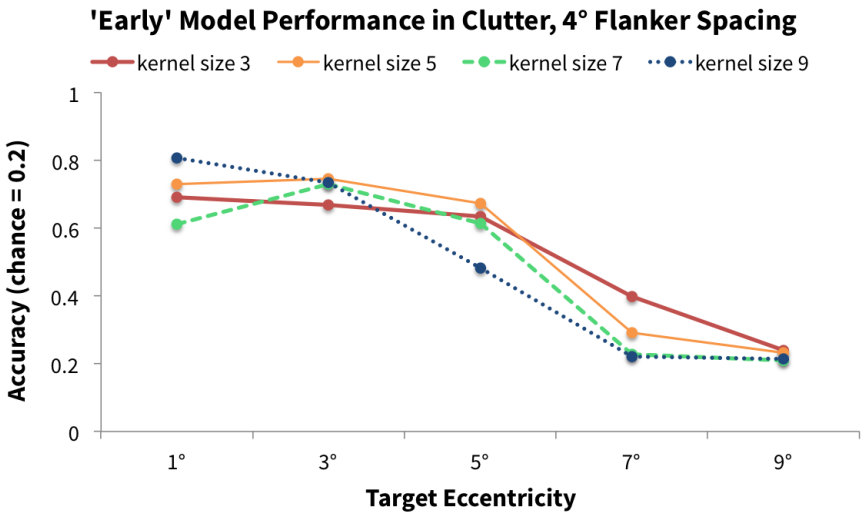


Figure 5-11: **'Early' model performance under clutter vs. spatial pooling, 4° flanker spacing.** All pooling uses stride 1. General trends persist across pooling methods, but the 'cliff' occurs at different points, possibly corresponding to different values of b. For kernel size 7, we attribute the variations before the 'cliff' to noise, caused by mis-alignment of filters and digits.

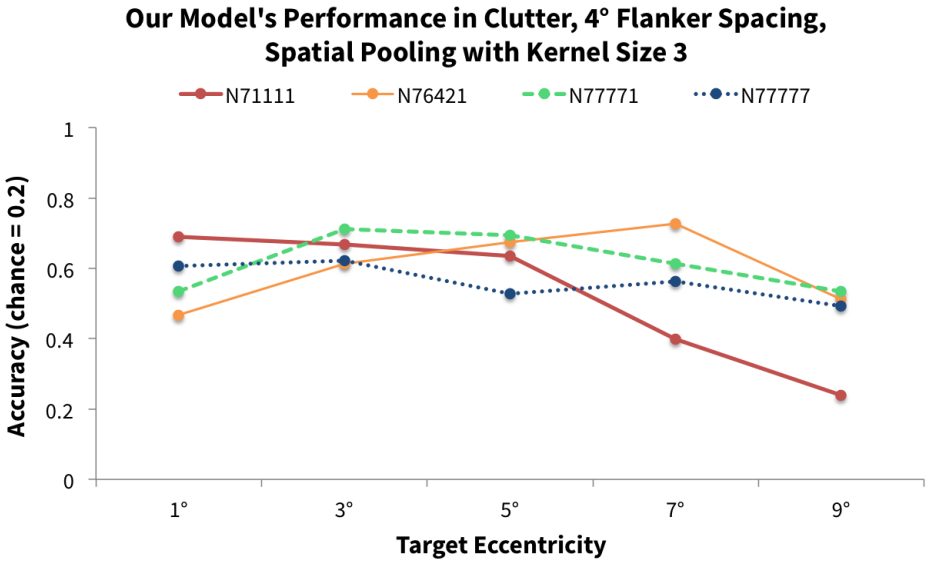


Figure 5-12: **Performance under clutter vs. scale pooling method, spatial pooling with kernel size 3, 4° flanker spacing.** Only N1111, the 'early' model, clearly displays qualitatively correct eccentricity-dependent behavior.

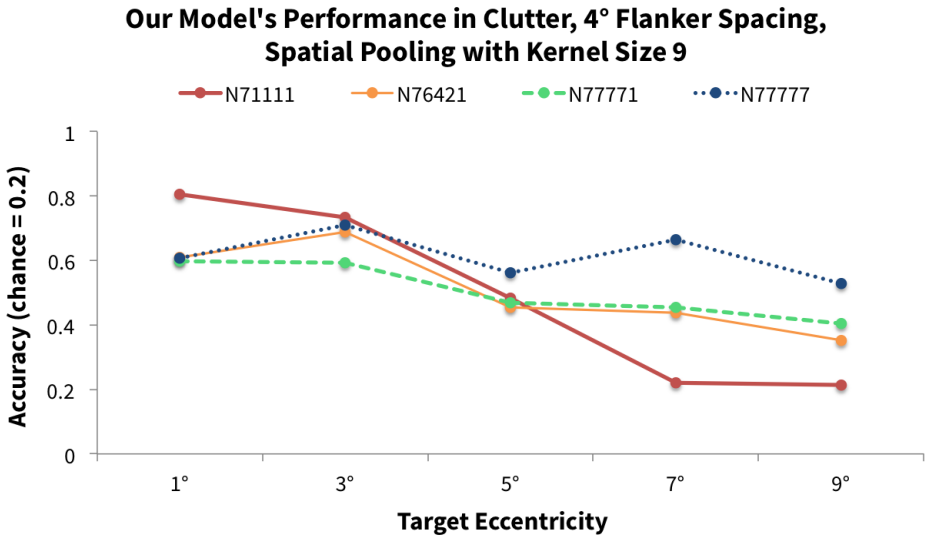


Figure 5-13: **Performance under clutter vs. scale pooling method, spatial pooling with kernel size 9, 4° flanker spacing.** N1111, the 'early' model, displays the clearest eccentricity-dependent behavior. All models, except for N7777, seem to display more correct behavior than Figure 5-12 in general—this implicates spatial pooling in crowding.

Chapter 6

Discussion

We have presented a new CNN-based computational model of the feedforward ventral stream, implementing the eccentricity theory [50]. As documented in Chapters 4 and 5, our model achieves several notable parallels with human performance. In addition, it uses biologically plausible computational operations [58, 56] and RF sizes (see Figure 3-7).

By considering two different methods of pooling over scale, both using the same fundamental CNN architecture, we are able to explain some key properties of the human visual system. The 'incremental' model, N6421, explains scale- and translation-invariance properties, while the 'early' model, N1111, explains Bouma's Law and radial-tangential anisotropy. The last achievement could potentially be significant—previous literature has not produced a widely-accepted explanation of radial-tangential anisotropy [45, 63]. Furthermore, we obtained all of our results without having to use a rigorous data-fitting or optimization approach to parameterize our model.

We find that the predictions of the eccentricity theory [50] are mostly correct in spirit. However, more rigor is required in both (1) the computational optimization procedure, and (2) the psychophysical backing, in order to establish a solid correspondence between the model and the visual system. In addition, there are two notable weaknesses in our model: (1) low acuity relative to human vision, and (2) the lack of a 'selection mechanism' for target-flanker simulations. We address approaches for these in Chapter 7.

Nevertheless, our model has significant advantages relative to comparable approaches.

6.1 Comparison to Existing Approaches

There are three general classes of models to compare against: (1) HMAX, (2) CNNs, and (3) crowding models.

6.1.1 HMAX

The HMAX model is an important basis of comparison for our work. HMAX displays similar or slightly higher amounts of translation- and scale-invariance, compared to the 'incremental' model [53]. We could tune our model to increase both types of invariance, under the constraint of biological plausibility (see Section 4.4).

Also, like our model, HMAX has high relevance to the study of crowding. Implementations of HMAX have outperformed benchmark approaches when providing features for recognition in clutter [58]. Additionally, a peripheral extension of HMAX obtained qualitative results agreeing with Bouma's Law [28, 27].

Finally, HMAX can use unsupervised learning techniques [57]. This gives it an advantage in biological plausibility over our model, which is fully supervised.

Our model uses the same primitives as HMAX—multi-scale filtering, MAX-pooling, and a linear classifier at the end [58]. Its main advantages lie in its generality and explanatory power.

In terms of generality, as described in Chapter 2, CNNs are much more powerful than any other object recognition technique [33, 19]. This could allow models like ours to be used in a variety of settings with relatively small amounts of task-specific adaptation. This is key to the study of crowding, which is highly stimulus-dependent [63, 16, 32]. Even more importantly, it could help to enable analysis of more natural and diverse scenes, which would significantly empower studies of the human visual system.

In terms of explanatory power, as mentioned previously, CNNs represent the state-of-the-art in explaining IT neural activity [64]. They have outperformed other approaches, including HMAX, by far [64]. This suggests that CNNs could be used to build models of the human ventral stream that are significantly more representative than competing approaches.

6.1.2 CNNs

As was demonstrated in Chapters 4 and 5, our model has transformation-invariance and crowding properties that are much more human-like than a 'flat' CNN and an FCN. Traditionally, such 'single-scale' networks rely on *data augmentation* [33, 52] to improve invariance. Data augmentation feeds CNNs with transformed versions of data during training, allowing them to 'memorize' transformations and improve recognition odds. In this sense, our model may be advantageous, because it requires lower sample complexity to achieve invariant performance. In fact, I-Theory proposes that decreasing sample complexity is one of the keys to achieving human-like performance [50].

It is important to note that our model is optimized for digit recognition, while other CNNs have tackled the much harder task of natural image recognition [33, 61, 19]. We have begun preliminary work in generalizing our methods to the ImageNet dataset [55] (see Chapter 7)—it is clear that demonstrating similar results will be non-trivial.

We also note that current state-of-the-art CNNs can be much 'deeper' than ours, with hundreds to thousands of layers [19]. Such networks can achieve performance far exceeding that in [33] on ImageNet classification [19]. However, the goal of our model is *human-like* performance, rather than performance optimization. Thus, we believe it is reasonable to limit the depth of our model to $O(10)$ in the near term [9, 57]. This seems to correlate well with human performance and with the architecture of the ventral stream [9, 57]

The work in [3] uses multiple resolutions of processing for a CNN, like our model. It seeks to recognize multiple digits from an image. [3] also uses a *recurrent neural*

network to enable several "glimpses" at a large scene. This is somewhat reminiscent of the notion of saccades in reading [5]. Though our work explores the concept of multi-resolution processing in a much more in-depth and human-like way, it uses only one feedforward pass. The use of multiple glimpses in [3] seems to be a key step in further study of the human visual system.

6.1.3 Crowding Models

As in Chapter 5, we enumerate two major categories of recent computational crowding models: population coding [18, 62] and statistics [31, 10, 45, 4].

Population coding models typically compute the responses of a large population of neurons, translating them into probability distributions that predict human subjects' responses [18, 62]. The work in [62] was able to accurately predict critical spacing and reproduce foveal-peripheral asymmetry. [18] successfully demonstrated Bouma's Law, and additionally showed better performance compared to models of attention, substitution, and averaging [18].

Unlike these models, ours has no obvious mechanism for spatial attention [18, 62]. By 'attention,' we simply mean the ability to select one signal among many, e.g. choosing a target among flankers. This is an important weakness—even in a short (<200 ms) presentation setting, attention-related effects have been implicated in crowding [20], such that it is difficult to completely exclude them from the feedforward pathway. The lack of a selection mechanism likely contributes to our model's inability to reproduce foveal-peripheral asymmetry.

On another axis, however, our model may represent a *superset* of population coding models. In some sense, a population code can be approximated with a CNN, given the correct learned weights. In addition, a CNN can generalize this idea to diverse settings without pre-configuration. This is unlike [18] and [62], which are specifically geared towards simple object types.

Unlike population coding models, statistical models often analyze images with a set of image-related statistics (e.g. correlations) [10, 4, 31, 51]. These statistics describe the 'effective representation' of a crowded object in the ventral stream [10, 4,

31]. Typically, such models are evaluated by synthesizing images known as "mongrels" [31, 4] or "metamers" [10]. Such synthetic images statistically match a given scene, but appear visually different. They are meant to represent perception in cluttered scenes. [51] presents an example image synthesis algorithm, used in [31, 4]. By evaluating people's foveal or unconstrained recognition performance on these synthesized images, studies can determine whether the statistics capture the information lost and preserved by crowding [10, 4, 31].

This method has the advantage of being both general and easy to evaluate psychophysically. However, image synthesis can be computationally intensive, such that a single image takes hours to synthesize [10, 31]. Though our model is not as easy to map to psychophysics (as discussed in Section 5.1), it can produce predictions much more quickly. Specifically, both of our CNN-based models take less than 2 hours to train and less than a minute to test (see Appendix A). Even a CNN as complex as the one in [33], which takes about a week to train, can classify over 500 images *per second* on a GPU [30]. In addition, CNNs are more explicitly tied to the underlying neural architecture (see Chapter 2), and can potentially make direct predictions on the neural level [64].

The statistical model in [45] takes a different approach, modeling statistical computations that involve V1 cells. Like our model, it leads to an explanation for radial-tangential anisotropy. However, it relies on an assumption regarding the timing of saccades and attention [45]. Our model does not require this assumption, exclusively considering single-view feedforward processing. In addition, our model represents more of the ventral stream, and can readily be extended to diverse object categories. Thus, it could potentially be more broadly applicable, especially considering that crowding effects are (1) neurologically difficult to isolate, and (2) stimulus-dependent [63, 16, 32].

We must emphasize that crowding is quite complex, and that there are some effects which are difficult to capture with *any* model. For instance, [14] shows a phenomenon which they call "demasking." This means that showing a copy of a target at the fovea can enhance a subject's ability to identify the same target among

flankers. This seemingly violates Bouma's Law in a setting involving relatively short ($\leq$150 ms) presentations. Similarly, it has been shown that additional flankers can *decrease* crowding, even in short (150 ms) presentation settings [22, 41]. In fact, [22] asserts that crowding currently cannot be reduced to a computational effect, because visual *grouping* is too difficult to model.

We do not know of a computational model that accounts for these observations; nevertheless, they must somehow be incorporated in the quest for a unifying understanding of crowding.

6.2 Implications and Applications

In addition to the main findings described previously, our work has the potential for additional positive impacts.

6.2.1 Modeling Degeneracy

Our results suggest that the feedforward pathway of human vision is *degenerate* [60]. That is, there are multiple processing routes that accomplish the same general goal. This is desirable in part because it allows a host of redundant mechanisms, each of which may (1) evolve and specialize independently, or (2) accommodate for the failure of another mechanism [60]. In our particular model, we find two different scale pooling strategies (N6421 and N1111) that explain different aspects of human visual performance.

Additional evidence for degeneracy comes from [64]. This work used a *mixture* of CNNs, with various depths and compositions, to achieve state-of-the-art results on IT prediction. This suggests that the actual neural architecture may be closer to a mixture of networks, rather than a single network. Similarly, [57] used *bypass routes* in HMAX, as described in [44], achieving a higher-fidelity representation of the ventral stream. This enabled multiple pathways of processing. Crowding studies have also observed that the addition of flankers can cause 're-emergence' of information thought to be lost in pooling [22, 41]. This further implies that information can

be obtained from multiple pathways. Finally, studies of IT neurons have revealed nontrivial heterogeneity [7, 40].

We propose that degeneracy should be a cornerstone in subsequent modeling of the visual system, and that our 'early' and 'incremental' models can serve as a starting point.

6.2.2 Engineering Applications

As stated in [50], one of the principal motivators for the eccentricity theory is learning with low sample complexity. Our model demonstrates this, to some degree. Furthermore, our model incorporates mechanisms which could be used to increase computational efficiency. For example, the eccentricity-dependent resolution of processing could conserve resources, supporting high-resolution only where absolutely necessary. In addition, 'chevron sampling' (see Figure 3-4) might further reduce the amount of computation required. Note that this preserves significant scale-invariance (see Figure 4-10) and has enabled the main results of this thesis. By introducing techniques for reducing sample complexity and saving computational resources, this work could contribute to the development of more data-efficient and performant CNNs.

6.2.3 Predictions, Psychophysics, and Physiology

Most importantly, the generality and neurological grounding of our model can enable the generation of various new kinds of predictions for psychophysical evaluation and neurological comparison. For instance, in Section 4.3, we used the predictions from [50] to generate computational results, which can now be directly compared against psychophysics. In addition, we have enumerated two explanatory scale pooling approaches, which can be examined physiologically.

In turn, biological results can be used to further calibrate and improve the model. This feedback paradigm could be used to develop an increasingly more general and more explanatory model of the ventral stream.

Chapter 7

Future Work

We list several major areas of future work, ordered from most immediate to most long-term.

7.1 Calibration

Most immediately, we can adopt simple measures to improve our model's fidelity. First, as mentioned in Chapters 3 and 4, we can increase the model's resolution. An order-of-magnitude increase might bring the model closer to human visual resolution [42], allowing for better simulations.

In addition, considering the distribution of RF sizes in Figure 3-3, we hypothesize that the human visual system has many more scales than our model. Implementing a model with more scales could reduce discretization artifacts (e.g. in Figure 3-7 and possibly in Figure 5-11), improving the quality of simulations.

To achieve even greater fidelity, we could adapt our model for the fovea and foveola. This is likely to be an important step—these areas are thought to account for the highest resolutions of visual perception [50]. According to the estimates in [50], the foveola spans only ≈26 minutes of visual angle, much less than the smallest scale channel in our model. Thus, matching the foveola would require much higher resolution. Additionally, the 'scale channels' in the fovea seem to display a roughly exponential pattern of size increase [42, 50]. Modeling this phenomenon may yield

additional insights into human visual properties.

Ultimately, through simulation and calibration, we could choose a set of scale channels that follows biological data more closely and allows more human-like performance.

7.2 Other Datasets

As mentioned in Chapter 6, we have started to apply our modeling approach to the ImageNet dataset [55]. We use Caffe's version of AlexNet [33, 30] as a baseline. Since ImageNet provides bounding boxes for some images [55], we can train and test on properly centered images as a first pass. In order to reduce training time and improve the probability of convergence, our strategy is to start with a pre-trained network, provided by Caffe [30], and apply supervised fine-tuning [34].

Applying the model to ImageNet would unlock more possibilities in terms of evaluation and iteration. It could possibly be worthwhile to consider other types of data, e.g. natural scenes from the Places dataset [66], faces as in [21] (also used in [58]), and keyboard letters/symbols, as in [16].

7.3 Optimization

It is currently difficult for us to perform rigorous optimization, because the behaviors of interest are sometimes qualitative in nature. It would be interesting to express these effects in mathematical terms. This would allow us to 'score' models based on their ability to reproduce the right effects. Though it is not clear how this approach should relate to absolute performance in a scoring metric, some combination of these two criteria could be informative. A purely quantitative performance metric could in turn allow more thorough exploration of the parameter space, through optimization methods as in [64].

7.4 Degeneracy and Attention

CNNs already implement 'bypass layers'—GoogLeNet, for instance, uses so-called "inception modules" and achieves an extremely low 6.67% top-5 error on ImageNet [61]. Since Caffe provides concatenation layers to enable this [30, 61], we could easily create a degenerate feedforward CNN with multiple pathways. The true challenge lies not in creating more pathways, but in implementing mechanisms for selection among their outputs. Thus, the problem of degeneracy is strongly coupled with the notion of implementing attention in our model.

According to [29], the primate visual system takes in far more information than the brain can thoroughly process. Thus, the brain necessarily focuses on a small subset, using attention to control the flow [29]. Even in the absence of saccades and top-down neural signals, "bottom-up" mechanisms may regulate the allocation of processing resources on visual input [29]. The authors of this study implemented a computational *saliency map* to model such "bottom-up" mechanisms. Similar efforts would undoubtedly increase the fidelity of our model.

We also note that attention can have many meanings. In addition to spatial saliency, there may also exist a notion of selecting among many processing pathways, which may pool information differently [22]. In addition, there is the simple notion of naming a target in a crowding simulation, mentioned in Section 5.1. Finally, [57] admits that feedforward timings do not exclude the possibility of "local feedback loops." These could modulate allocation of visual processing resources. It could be worthwhile to explore computational implementations of such functionalities.

7.5 Psychophysics

It is difficult to draw broad conclusions from our psychophysical reference literature, due to the heterogeneity of timing and masking strategies used in experiments. Thus, it would be extremely informative to conduct a consistent set of psychophysical experiments, aimed at assessing our model's predictions. By assembling a set of

psychophysical observations that can be easily compared and analyzed against our model, we can more reliably meet the feedforward assumption of processing and more effectively develop a correspondence between model and reality.

7.6 Unsupervised Learning

An important drawback of our model, especially in relation to human performance, is its reliance on fully supervised learning. In contrast, humans clearly do not need full supervision (i.e. labeling of every input data point) to learn object recognition. Approaches exist for implementing unsupervised learning in neural networks [23, 34]. By implementing such capabilities in our model, we might gain a better representation of the human visual system.

7.7 Multiple Saccades

As mentioned in Chapter 6, previous work has used a recurrent neural network (RNN) to combine information from multiple "glimpses," with a multi-resolution input at each [3]. This allows for the recognition of multiple digits in an image [3]. This is an early step along the ultimate path of vision research—integrating information from multiple saccades, as humans do [5, 63].

Our model approaches this path from another angle by shedding light on the feedforward pathway. In the future, combining the feedforward model with an RNN could bring us closer still to developing a more holistic and detailed understanding of human vision.

Appendix A

CNN Details

We present the detailed architectures and training procedures for our CNNs. We also provide some supplemental analyses on their behaviors. This appendix builds upon the description of our model in Section 3.2.

A.1 Network Architectures

As mentioned in Chapter 4, we perform simulations with 4 CNNs:

1. the 'early' model (N1111)
2. the 'incremental' model (N6421)
3. a 'flat' CNN operating at one scale
4. a fully-connected network (FCN) operating at one scale

Both the 'early' and 'incremental' models use 'chevron sampling' with $c = 2$ (see Subsection 3.2.1), for every simulation except for the chevron parameter simulation in Section 4.4.

In general, our CNNs use the same parameters as the version of LeNet included with Caffe [36, 30].

Layer	L × W × D	# Scales	Kernel Size / Stride / Pad
`data`	$83 \times 83 \times 1$	7	- / - / -
`conv1`	$40 \times 40 \times 32$	7	5 / 2 / 0
`spatial-pool1`	$38 \times 38 \times 32$	7	3 / 1 / 0
`scale-pool1`	$38 \times 38 \times 32$	$7 \rightarrow 1$	- / - / -
`conv2`	$38 \times 38 \times 32$	1	5 / 1 / 2
`spatial-pool2`	$36 \times 36 \times 32$	1	3 / 1 / 0
`scale-pool2`	$36 \times 36 \times 32$	$1 \rightarrow 1$	- / - / -
`conv3`	$36 \times 36 \times 32$	1	5 / 1 / 2
`spatial-pool3`	$34 \times 34 \times 32$	1	3 / 1 / 0
`scale-pool3`	$34 \times 34 \times 32$	$1 \rightarrow 1$	- / - / -
`conv4`	$34 \times 34 \times 32$	1	5 / 1 / 2
`spatial-pool4`	$32 \times 32 \times 32$	1	3 / 1 / 0
`scale-pool4`	$32 \times 32 \times 32$	$1 \rightarrow 1$	- / - / -
`fully-connected`	$1 \times 1 \times 10$	1	- / - / -

Table A.1: **Layers of 'early' model (N1111).** Data flows from top to bottom. We use a standard softmax [23, 30] to convert the `fully-connected` layer outputs into classification probabilities. Every convolutional (`conv`) layer uses ReLU non-linearity [33]. For simplicity, we use padding such that only `spatial-pool` layers reduce the spatial dimension.

A.1.1 Layers

The principal layers of the four CNNs mentioned above are listed in Tables A.1, A.2, A.3, and A.4, respectively.

A.1.2 Layer Details

As discussed in Chapter 3, all pooling layers use the MAX operation. All learned layers (`conv` and `fully-connected` layers) use the same initialization as in Caffe's LeNet [36, 30]:

- Learning rate multiplier of 1 for filters and 2 for biases
- Weights initialized with Caffe's `xavier` algorithm [30]
- Biases initialized to zero

Also, as Chapter 3 mentions, convolutional filters are constrained to be identical across scale channels.

Layer	L × W × D	# Scales	Kernel Size / Stride / Pad
`data`	$83 \times 83 \times 1$	7	- / - / -
`conv1`	$40 \times 40 \times 32$	7	5 / 2 / 0
`spatial-pool1`	$38 \times 38 \times 32$	7	3 / 1 / 0
`scale-pool1`	$38 \times 38 \times 32$	$7 \rightarrow 6$	- / - / -
`conv2`	$38 \times 38 \times 32$	6	5 / 1 / 2
`spatial-pool2`	$36 \times 36 \times 32$	6	3 / 1 / 0
`scale-pool2`	$36 \times 36 \times 32$	$6 \rightarrow 4$	- / - / -
`conv3`	$36 \times 36 \times 32$	4	5 / 1 / 2
`spatial-pool3`	$34 \times 34 \times 32$	4	3 / 1 / 0
`scale-pool3`	$34 \times 34 \times 32$	$4 \rightarrow 2$	- / - / -
`conv4`	$34 \times 34 \times 32$	2	5 / 1 / 2
`spatial-pool4`	$32 \times 32 \times 32$	2	3 / 1 / 0
`scale-pool4`	$32 \times 32 \times 32$	$2 \rightarrow 1$	- / - / -
`fully-connected`	$1 \times 1 \times 10$	1	- / - / -

Table A.2: **Layers of 'incremental' model (N6421).** Data flows from top to bottom. Identical to 'early' model, except for scale pooling. See caption from Table A.1 for additional notes.

Layer	L × W × D	Kernel Size / Stride / Pad
`data`	$83 \times 83 \times 1$	- / - / -
`conv1`	$79 \times 79 \times 32$	5 / 1 / 0
`spatial-pool1`	$77 \times 77 \times 32$	3 / 1 / 0
`conv2`	$73 \times 73 \times 32$	5 / 1 / 0
`spatial-pool2`	$71 \times 71 \times 32$	3 / 1 / 0
`conv3`	$67 \times 67 \times 32$	5 / 1 / 0
`spatial-pool3`	$65 \times 65 \times 32$	3 / 1 / 0
`conv4`	$61 \times 61 \times 32$	5 / 1 / 0
`spatial-pool4`	$59 \times 59 \times 32$	3 / 1 / 0
`fully-connected`	$1 \times 1 \times 10$	- / - / -

Table A.3: **Layers of 'flat' CNN.** Data flows from top to bottom. This operates on only one scale of input. For simplicity, we use the same convolution and pooling strategy at all layers and do not apply padding. We use softmax [23, 30] after the final layer and ReLU [33] after each `conv` layer, as discussed in the caption of Table A.1.

Layer	L × W × D	Kernel Size / Stride / Pad
`data`	$581 \times 581 \times 1$	- / - / -
`fully-connected1`	$1 \times 1 \times 32$	- / - / -
`fully-connected2`	$1 \times 1 \times 32$	- / - / -
`fully-connected3`	$1 \times 1 \times 32$	- / - / -
`fully-connected4`	$1 \times 1 \times 32$	- / - / -
`fully-connected5`	$1 \times 1 \times 10$	- / - / -

Table A.4: **Layers of FCN.** Data flows from top to bottom. This operates on only one scale of input. We use softmax [23, 30] after the final layer and ReLU [33] after the first four `fully-connected` layers, as discussed in the caption of Table A.1.

Parameter	Value
`base_lr`	0.01
`momentum`	0.9
`weight_decay`	0.0005
`lr_policy`	`inv`
`gamma`	0.0001
`power`	0.75
`max_iter`	10,000

Table A.5: **Solver parameters.** This set of parameters was used to train all of our networks in Caffe [30]. It is identical to the solver parameters used for Caffe's LeNet on MNIST [36, 30].

A.2 Training Procedure

Our training procedure uses the same parameters as for Caffe's LeNet on MNIST [36, 30], listed in Table A.5. [30] implements these parameters, and provides some documentation on them. [23] also provides some insight regarding their importance.

We use a batch size of 64 during training, identical to Caffe's LeNet on MNIST [36, 30]. We train using standard backpropagation [30], explained in [23]. Training is done on GPUs, using the OpenMind computing cluster [43]. Sample running times for training and testing are shown in Table A.6. Figures A-1, A-2, A-3, and A-4 show sample convergence behavior for the 'early' model, the 'incremental' model, the 'flat' CNN, and the FCN, respectively.

Network	Time to Train (sec)	Time to Test (sec)
'early' model, N1111	1,905	38
'incremental' model, N6421	6,986	33
'flat' CNN	2,238	88
FCN	7,900	25

Table A.6: **Approximate running times.** These measurements were conducted on a single run only. Training used the full MNIST training set [37], with centered digits of 3° height and 10,000 iterations. Testing used the full MNIST test set [37], with data from one scale-invariance test condition (see Section 4.1).

In terms of iterations, it seems that the model converges as quickly as the 'flat' CNN and much faster than the FCN. As expected, later scale pooling leads to more computationally intensive training. From both the scientific and engineering perspectives, it would be interesting to compare convergence rates and performance in more depth.

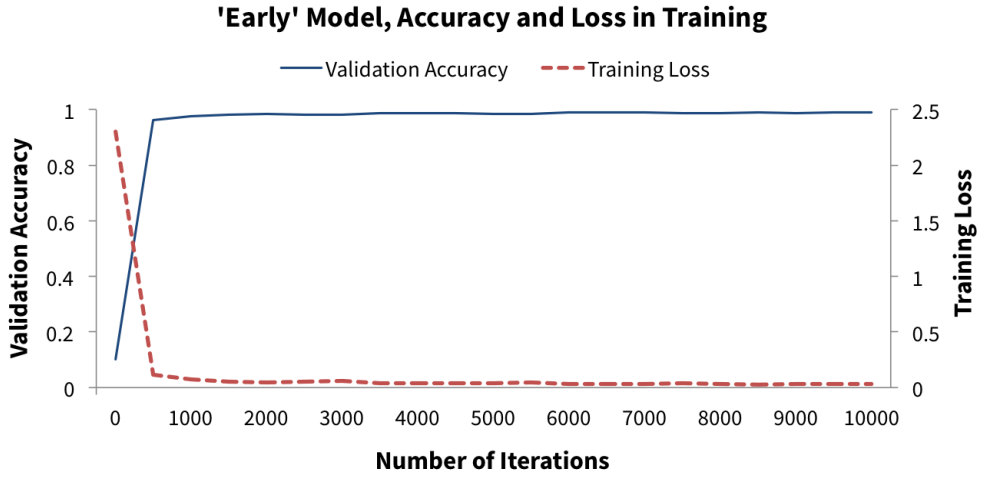


Figure A-1: **Convergence behavior of 'early' model.** Note the rapid convergence. These measurements were done when training on the full MNIST training set [37], using centered digits of 3° height. Validation used the full MNIST test set [37], with the same properties.

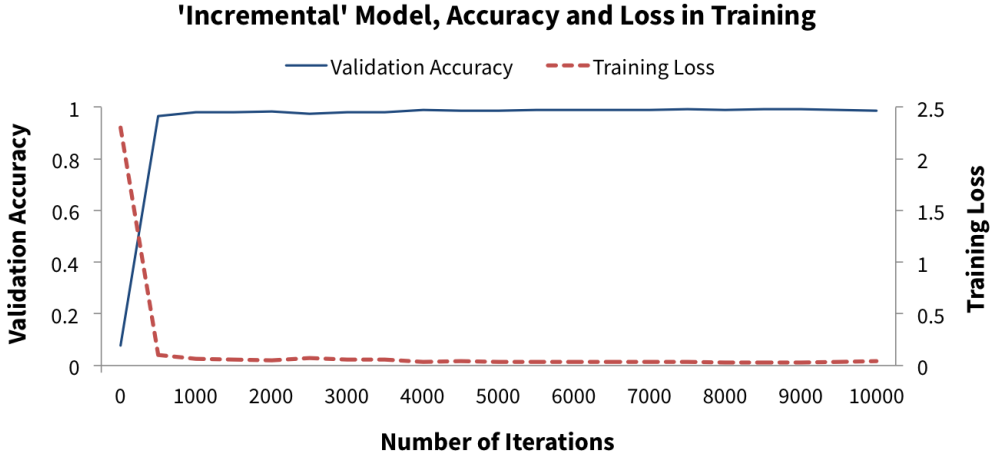


Figure A-2: **Convergence behavior of 'incremental' model.** Again, note the rapid convergence. These measurements were done when training on the full MNIST training set [37], using centered digits of 3° height. Validation used the full MNIST test set [37], with the same properties.

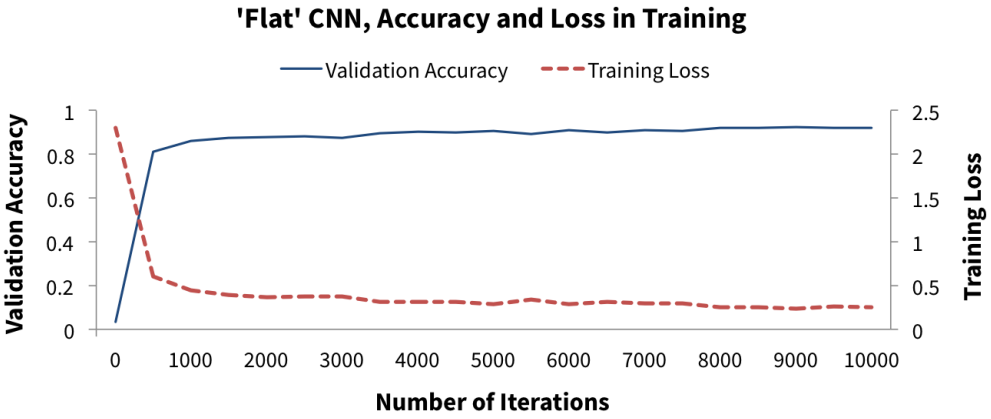


Figure A-3: **Convergence behavior of 'flat' CNN.** The network converges rapidly in the initial phases, then makes slow improvements. Low resolution may give it a disadvantage versus our model (our goal is to show the advantage of multiple scales, rather than direct performance comparison). These measurements were done when training on the full MNIST training set [37], using centered digits of 3° height. Validation used the full MNIST test set [37], with the same properties.

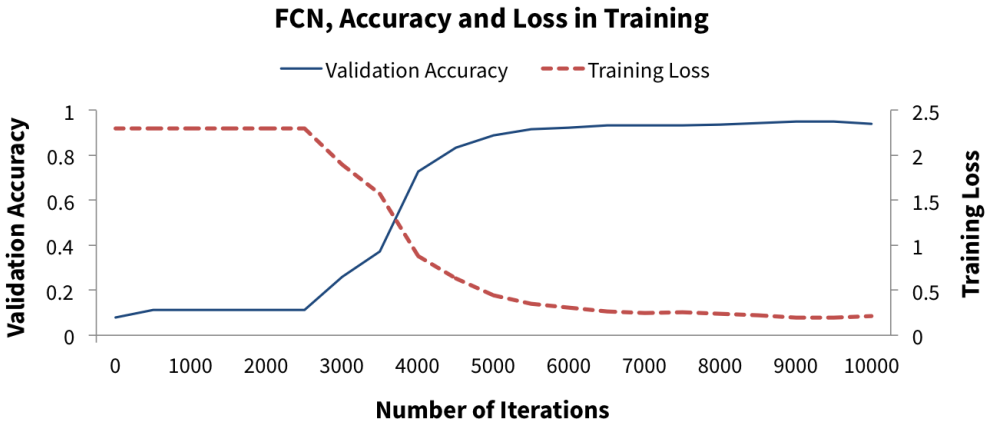


Figure A-4: **Convergence behavior of FCN.** The network fails to converge for a while, then learns relatively quickly. These measurements were done when training on the full MNIST training set [37], using centered digits of 3° height. Validation used the full MNIST test set [37], with the same properties.

Bibliography

[1] S.M. Anstis. A chart demonstrating variations in acuity with retinal position. *Vision Research*, 14:589–592, 1974.

[2] Yusuf Aytar and Aude Oliva. 6.819/6.869 Advances in Computer Vision. MIT course, 2015.

[3] Jimmy Lei Ba, Volodymyr Mnih, and Koray Kavukcuoglu. Multiple object recognition with visual attention. *International Conference on Learning Representations*, 2015.

[4] Benjamin Balas, Lisa Nakano, and Ruth Rosenholtz. A summary-statistic representation in peripheral vision explains crowding. *Journal of Vision*, 9(12):13.1–13.18, 2009.

[5] H. Bouma. Interaction effects in parafoveal letter recognition. *Nature*, 226:177–178, April 1970.

[6] G. Bradski. OpenCV library. Dr. Dobb's Journal of Software Tools, 2000.

[7] James J. DiCarlo and John H.R. Maunsell. Anterior inferotemporal neurons of monkeys engaged in object recognition can be highly sensitive to object retinal position. *Journal of Neurophysiology*, 89:3264–3278, 2003.

[8] Marcus Dill and Shimon Edelman. Imperfect invariance to object translation in the discrimination of complex shapes. *Perception*, 30:707–724, 2001.

[9] Sven Eberhardt, Jonah Cader, and Thomas Serre. How deep is the feature analysis underlying rapid visual categorization? arXiv:1606.01167 [cs.CV], June 2016.

[10] Jeremy Freeman and Eero P. Simoncelli. Metamers of the ventral stream. *Nature Neuroscience*, 14(9):1195–1201, September 2011.

[11] Christopher S. Furmanski and Stephen A. Engel. Perceptual learning in object recognition: Object specificity and size invariance. *Vision Research*, 40:473–484, 2000.

[12] R. Gattass, C.G. Gross, and J.H. Sandell. Visual topography of V2 in the macaque. *The Journal of Comparative Neurology*, 201:519–539, 1981.

[13] R. Gattass, A.P.B. Sousa, and C.G. Gross. Visuotopic organization and extent of V3 and V4 of the macaque. *The Journal of Neuroscience*, 8(6):1831–1845, June 1988.

[14] Gad Geiger and Jerome Y. Lettvin. Enhancing the perception of form in peripheral vision. *Perception*, 15:119–130, 1986.

[15] Gad Geiger, Jerome Y. Lettvin, and Olga Zegarra-Moran. Task-determined strategies of visual process. *Cognitive Brain Research*, 1:39–52, 1992.

[16] Jonathan Grainger, Ilse Tydgat, and Joanna Isselé. Crowding affects letters and symbols differently. *Journal of Experimental Psychology: Human Perception and Performance*, 36(3):673–688, 2010.

[17] Mohammad Haghighat, Saman Zonouz, and Mohamed Abdel-Mottaleb. CloudID: Trustworthy cloud-based and cross-enterprise biometric identification. *Expert Systems with Applications*, 42:7905–7916, 2015.

[18] William J. Harrison and Peter J. Bex. A unifying model of orientation crowding in peripheral vision. *Current Biology*, 25:3213–3219, December 2015.

[19] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. arXiv:1512.03385 [cs.CV], December 2015.

[20] Sheng He, Patrick Cavanagh, and James Intriligator. Attentional resolution and the locus of visual awareness. *Nature*, 383:334–337, September 1996.

[21] Bernd Heisele, Thomas Serre, Massimiliano Pontil, Thomas Vetter, and Tomaso Poggio. Categorization by learning and combining object parts. *Advances in Neural Information Processing Systems*, 2002.

[22] Michael H. Herzog, Bilge Sayim, Vitaly Chicherov, and Mauro Manassi. Crowding, grouping, and object recognition: A matter of appearance. *Journal of Vision*, 15(6):5.1–5.18, 2015.

[23] Geoffrey E. Hinton. Neural networks for machine learning (online course materials). Coursera, 2012.

[24] D.H. Hubel and T.N. Wiesel. Receptive fields, binocular interaction and functional architecture in the cat's visual cortex. *Journal of Physiology*, 160:106–154, 1962.

[25] D.H. Hubel and T.N. Wiesel. Receptive fields and functional architecture of monkey striate cortex. *Journal of Physiology*, 195:215–243, 1968.

[26] Chou P. Hung, Gabriel Kreiman, Tomaso Poggio, and James J. DiCarlo. Fast readout of object identity from macaque inferior temporal cortex. *Science*, 310:863–866, November 2005.

[27] Leyla Isik, Joel Z. Leibo, Sang Wan Lee, Jim Mutch, and Tomaso Poggio. A hierarchical model of peripheral vision. Poster: Society for Neuroscience, Washington D.C., 2011.

[28] Leyla Isik, Joel Z. Leibo, Jim Mutch, Sang Wan Lee, and Tomaso Poggio. A hierarchical model of peripheral vision. Technical Report MIT-CSAIL-TR-2011-031 CBCL-300, MIT Computer Science and Artificial Intelligence Laboratory, June 2011.

[29] Laurent Itti and Christof Koch. A saliency-based search mechanism for overt and covert shifts of visual attention. *Vision Research*, 40:1489–1506, 2000.

[30] Yangqing Jia, Evan Shelhamer, Jeff Donahue, Sergey Karayev, Jonathan Long, Ross Girshick, Sergio Guadarrama, and Trevor Darrell. Caffe: Convolutional architecture for fast feature embedding. arXiv:1408.5093 [cs.CV], 2014.

[31] Shaiyan Keshvari and Ruth Rosenholtz. Pooling of continuous features provides a unifying account of crowding. *Journal of Vision*, 16(3):39.1–39.15, 2016.

[32] Frank L. Kooi, Alex Toet, Srimant P. Tripathy, and Dennis M. Levi. The effect of similarity and duration on spatial interaction in peripheral vision. *Spatial Vision*, 8(2):255–279, 1994.

[33] Alex Krizhevsky, Ilya Sutskever, and Geoffrey E. Hinton. ImageNet classification with deep convolutional neural networks. *Advances in Neural Information Processing Systems*, 2012.

[34] Hugo Larochelle, Yoshua Bengio, Jérôme Louradour, and Pascal Lamblin. Exploring strategies for training deep neural networks. *Journal of Machine Learning Research*, 1:1–40, 2009.

[35] Y. LeCun, B. Boser, J.S. Denker, D. Henderson, R.E. Howard, W. Hubbard, and L.D. Jackel. Handwritten digit recognition with a back-propagation network. *Advances in Neural Information Processing Systems*, 1989.

[36] Yann LeCun, León Bottou, Yoshua Bengio, and Patrick Haffner. Gradient-based learning applied to document recognition. *Proceedings of the IEEE*, November 1998.

[37] Yann LeCun, Corinna Cortes, and Christopher J.C. Burges. The MNIST database of handwritten digits. Online (yann.lecun.com/exdb/mnist), accessed in 2016.

[38] Yann LeCun, L.D. Jackel, León Bottou, Corinna Cortes, John S. Denker, Harris Drucker, Isabelle Guyon, Urs A. Müller, Eduard Säckinger, Patrice Simard, and Vladimir Vapnik. Learning algorithms for classification: A comparison on handwritten digit recognition. *Neural Networks: The Statistical Mechanics Perspective*, pages 261–276, 1995.

[39] Hesheng Liu, Yigal Agam, Joseph R. Madsen, and Gabriel Kreiman. Timing, timing, timing: Fast decoding of object information from intracranial field potentials in human visual cortex. *Neuron*, 62:281–290, April 2009.

[40] Nikos K. Logothetis, Jon Pauls, and Tomaso Poggio. Shape representation in the inferior temporal cortex of monkeys. *Current Biology*, 5(5):552–563, 1995.

[41] Mauro Manassi, Bilge Sayim, and Michael H. Herzog. When crowding of crowding leads to uncrowding. *Journal of Vision*, 13(13):10.1–10.10, 2013.

[42] D. Marr, T. Poggio, and E. Hildreth. Smallest channel in early human vision. *Journal of the Optical Society of America*, 70(7):868–870, July 1980.

[43] McGovern Institute at MIT. OpenMind: High performance computing cluster, 2014.

[44] Hiroyuki Nakamura, Ricardo Gattass, Robert Desimone, and Leslie G. Ungerleider. The modular organization of projections from areas V1 and V2 to areas V4 and TEO in macaques. *The Journal of Neuroscience*, 13(9):3861–3691, September 1993.

[45] Anirvan S. Nandy and Bosco S. Tjan. Saccade-confounded image statistics explain visual crowding. *Nature Neuroscience*, 15(3):463–469, March 2012.

[46] Tatjana A. Nazir and J. Kevin O'Regan. Some results on translation invariance in the human visual system. *Spatial Vision*, 5(2):81–100, 1990.

[47] Denis G. Pelli. Crowding: A cortical constraint on object recognition. *Current Opinion in Neurobiology*, 18(4):445–451, August 2008.

[48] Denis G. Pelli, Melanie Palomares, and Najib J. Majaj. Crowding is unlike ordinary masking: Distinguishing feature integration from detection. *Journal of Vision*, 4:1136–1169, 2004.

[49] Denis G. Pelli and Katharine A. Tillman. The uncrowded window of object recognition. *Nature Neuroscience*, 11(10):1129–1135, October 2008.

[50] Tomaso Poggio, Jim Mutch, and Leyla Isik. Computational role of eccentricity dependent cortical magnification. Memo 17, Center for Brains, Minds and Machines, June 2014.

[51] Javier Portilla and Eero P. Simoncelli. A parametric texture model based on joint statistics and wavelet coefficients. *International Journal of Computer Vision*, 40(1):49–71, 2000.

[52] Ali Sharif Razavian, Hossein Azizpour, Josephine Sullivan, and Stefan Carlsson. CNN features off-the-shelf: An astounding baseline for recognition. *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR) Workshops*, pages 806–813, 2014.

[53] Maximilian Riesenhuber and Tomaso Poggio. Hierarchical models of object recognition in cortex. *Nature Neuroscience*, 2(11):1019–1025, November 1999.

[54] Maximilian Riesenhuber and Tomaso Poggio. Neural mechanisms of object recognition. *Current Opinion in Neurobiology*, 12:162–168, 2002.

[55] Olga Russakovsky, Jia Deng, Hao Su, Jonathan Krause, Sanjeev Satheesh, Sean Ma, Zhiheng Huang, Andrej Karpathy, Aditya Khosla, Michael Bernstein, Alexander C. Berg, and Li Fei-Fei. ImageNet large scale visual recognition challenge. *International Journal of Computer Vision*, 115(3):211–252, December 2015.

[56] T. Serre, M. Kouh, C. Cadieu, U. Knoblich, G. Kreiman, and T. Poggio. A theory of object recognition: Computations and circuits in the feedforward path of the ventral stream in primate visual cortex. AI Memo 2005-036, CBCL Memo 259, MIT Computer Science and Artificial Intelligence Laboratory, December 2005.

[57] Thomas Serre, Aude Oliva, and Tomaso Poggio. A feedforward architecture accounts for rapid categorization. *Proceedings of the National Academy of Sciences (PNAS)*, 104(15):6424–6429, April 2007.

[58] Thomas Serre, Lior Wolf, Stanley Bileschi, Maximilian Riesenhuber, and Tomaso Poggio. Robust object recognition with cortex-like mechanisms. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 29(3):411–426, March 2007.

[59] Hans Strasburger, Ingo Rentschler, and Martin Jüttner. Peripheral vision and pattern recognition: A review. *Journal of Vision*, 11(5):13.1–13.82, 2011.

[60] Gerald Jay Sussman. Building robust systems: An essay. Massachusetts Institute of Technology, January 2007.

[61] Christian Szegedy, Wei Liu, Yangqing Jia, Pierre Sermanet, Scott Reed, Dragomir Anguelov, Dumitru Erhan, Vincent Vanhoucke, and Andrew Rabinovich. Going deeper with convolutions. *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2015.

[62] Ronald van den Berg, Jos B.T.M. Roerdink, and Frans W. Cornelissen. A neurophysiologically plausible population code model for feature integration explains visual crowding. *PLoS Computational Biology*, 6(1), January 2010.

[63] David Whitney and Dennis M. Levi. Visual crowding: A fundamental limit on conscious perception and object recognition. *Trends in Cognitive Sciences*, 15(4):160–168, April 2011.

[64] Daniel L.K. Yamins, Ha Hong, Charles F. Cadieu, Ethan A. Solomon, Darren Seibert, and James J. DiCarlo. Performance-optimized hierarchical models predict neural responses in higher visual cortex. *Proceedings of the National Academy of Sciences (PNAS)*, 111(23):8619–8624, June 2014.

[65] Jason Yosinski, Jeff Clune, Yoshua Bengio, and Hod Lipson. How transferable are features in deep neural networks? *Advances in Neural Information Processing Systems*, 2014.

[66] B. Zhou, A. Lapedriza, J. Xiao, A. Torralba, and A. Oliva. Learning deep features for scene recognition using places database. *Advances in Neural Information Processing Systems*, 2014.