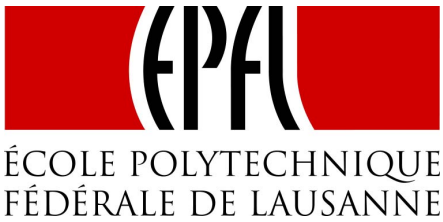
ECOLE POLYTECHNIQUE FEDERALE DE LAUSANNE
SCHOOL OF LIFE SCIENCES



Master project in Bioengineering

Behavioral and computational study on the recognition of novel occluded objects

Carried out in the Kreiman Laboratory
at Boston, Harvard Medical School
Under the supervision of Gabriel Kreiman, Associate Professor

Done by

CHARLOTTE MOERMAN

Under the direction of
Prof. Michael, Herzog
In the laboratory of psychophysics (LPSY)

EPFL

Lausanne, June 22, 2017

Contents

Acknowledgements

I would first like to thank Prof. Gabriel Kreiman not only for the incredible opportunity of doing my master thesis in his lab at Boston Children's Hospital, Harvard Medical School but also for his advice and guidance throughout my work. As he guided me through new sets of questions after each meeting, I was able to grow my self-confidence as an independent scientific researcher when adressing the challenges of the project.

I would also like to thank the entire group at the Kreiman lab, in particular Jiarui Wang for his help with IT issues, William Lotter for the occasional brainstorming and Megan Bendsen-Jensen for making the endless administrative process so smooth. I am of course also extremely grateful for all the people who agreed to take part to the psychophysics experiment, most of which were EPFL students who also came to Boston for their thesis.

Thank you as well to my supervisor from EPFL, Prof. Michael Herzog, for accepting to take part in this adventure and Pascale Zbinden for clarifying every deadline and administrative requirement for this exchange to happen.

Finally, I would like to thank friends and family back home for their messages and Skype calls throughout my stay, new friends from here for our discovery trips around Boston and all the people who made me smile through hard and fun times.

To all, thank you!

Chapter 1

Summary

Interpreting incomplete information is a critical aspect of intelligence. In the visual domain, humans efficiently recognize objects rendered partially visible due to noise, limited viewing angles, poor illumination or presence of occluders on a daily basis. However, it remains unclear if humans need extensive previous experience with whole objects and/or their occluded counterparts to perform efficient pattern completion or if this is an inherent property of the visual system, at least up to certain visibility levels. In the present thesis, we investigate if humans still robustly categorize heavily occluded renderings of artificially created novel objects when having only minimal training and no pre-existing partial object exposure. In parallel, we augmented state-of-the-art hierarchical feed-forward computational models with recurrent connections to assess if human-like performance could be reached for a particular categorization task. Previous studies using such networks were unable to match human results unless they were trained with occluded objects specifically and their generalization to novel categories is still questioned. However, they did perform significantly better than their bottom-up counterparts which are not robust to object occlusion, implying that recurrent connections can facilitate pattern completion. Our results show that although humans can still categorize partial objects above chance level for very low image visibilities, artificial neural networks augmented with recurrent connections on only one layer are now able to outperform behavioral results for all visibility levels. Although extensive previous experience with novel occluded objects is not essential for humans to be robust against novel object occlusion, it could maybe explain why recurrent models perform less well than humans for the same task involving everyday objects.

Chapter 2

Introduction

Although computers are very efficient at executing tasks involving deterministic numerical algorithms, the human brain still outperforms them in many domains involved in daily life activities. Some of the brain's characteristics of interest include robustness to noise or incomplete information, incorporation of contextual information present in the environment, plastic memory formation and fault tolerance. Therefore, studying the brain as a computational model will provide clues to help improve hardware and software in order to create better performing artificial intelligence algorithms. In the field of machine imagery for example, efforts have been made to better understand visual processing in the brain with the hope to increase performance for partial image recognition, a task for which humans were found to perform above chance for visibility levels as low as 15%, a result which state-of-the art feed forward neural networks fail to match[1].

A broad overview of current neuroscientific state-of-the-art regarding the visual cortex architecture and information processing will be presented in *Section 2.1*, focusing on recognition of objects given only partial information. Biological architecture and performance will then be compared to different computational models in *Section 2.2*. Finally, a summary of the scientific aim of this project by contrasting computational models and biological data will be presented in *Section 2.3*.

2.1 Biological context

As a first approximation, visual information can be depicted as traveling through the brain areas presented in schematic representation 2.1. Visual input is progressively transformed from a very specific format that is almost pixel-based to a more behaviorally useful representation detailed in sub-section 2.1.1. While both neurons of early brain areas and artificial neurons of computational models where found to be selective to low-level features such as edges and colors, single units of the hippocampus amongst other higher brain areas were found to be invariant to specific object categories or even individual faces[2, 3, 4]. However neuroscientists still have only limited understanding of the underlying mechanisms of these observed specificities. The brain's high processing speed and surprising combination of selectivity and robustness to variations of scale, occlusion, luminosity and angle only to name a few[2, 3, 5] remain an intriguing domain of modern research. In sub-sections 2.1.2, 2.1.3 and 2.1.4 biological knowledge about object completion tasks will be exposed. Indeed vision in a daily life setting rarely involves whole and/or perfectly isolated object recognition tasks and therefore recognition of occluded objects is very interesting for multiple artificial intelligence applications. Understanding the specificities of neural responses to occluded visual signals at different levels of the hierarchy can give new hints to improve existing computational models.

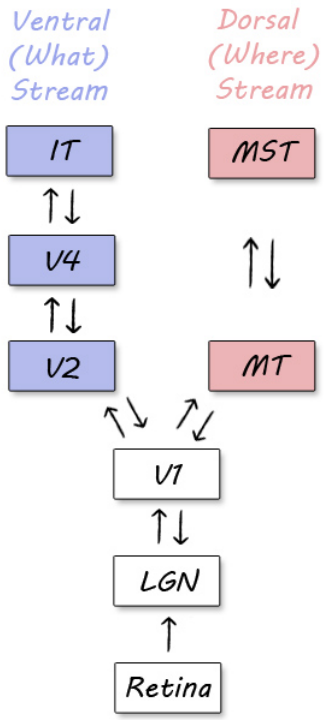


Figure 2.1: Highly oversimplified schematic of the visual system. The ventral pathway is shown in blue while the dorsal one is shown in pink. Boxes represent the main areas involved in image processing. Arrows show the direction of visual information flow.

2.1.1 The visual cortex hierarchy

The path from the retina to the cortex

Light reaches the eye and excites photoreceptor neurons of the retina, located at the back of the eyeball. These neurons subdivide in two types: the rods are mainly activated in dim light conditions while the cone are implicated in color and finer detail perception. The central region of the retina provides the highest resolution, being composed solely by cones. The above-mentioned neurons then transmit the signal through horizontal, bipolar and amacrine neurons to the retinal ganglion cells which compose the optic nerve. Information travels through the optic nerve to the lateral geniculate nucleus (LGN) located in the thalamus which provides the final link to the primary visual cortex (V1).

The primary visual cortex

In this area, neurons are arranged into six distinct layers which are perpendicular to columns sharing similar visual preferences[6, 7]. Visual signals are then split into the ventral and dorsal stream. What is of particular interest in V1 is the neuronal type subdivision which provided inspiration for early computational neural networks, especially HMAX which will be more detailed in section 3.3.1. At this level in the visual pathway, neurons can be generally divided into either simple or complex cells although complex cells can also be found in later visual areas such as V2 and V3. Hubel and Wiesel[8] were the first ones to explore how this specific architecture can explain orientation-tuning and position or scale invariance observed at the level of V1 in a simple and elegant way.

Simple cells were found to respond specifically to tasks involving low-level features such as color specificity or edge detection and have on or off center receptive fields which are usually modeled by Gabor functions[9] defined by:

$$F(x,y) = \frac{1}{2\pi\sigma_x\sigma_y} exp^{-\frac{x}{2\sigma_x^2}-\frac{y}{2\sigma_y^2}} cos(kx-\phi)$$

where σ_x and σ_y determine the spatial extent in x and y, k and ϕ are the preferred spatial frequency and phase respectively. On the other hand, complex cells receive inputs from multiple simple cells and therefore do not show simply defined excitatory or inhibitory responses anymore. These cells fire for inputs displayed in a certain orientation but irrespective of their exact location providing the invariance properties observed in the visual system. One example of more complex generated behavior are the end stopper cells which respond maximally when an oriented bar ends within the receptive field.

The dorsal stream

The dorsal stream will not be emphasized in this thesis but roughly corresponds to the action channel since it essentially processes spatial locations, stereopsis and object motion, eventually transforming this information into motor behavior. V3 is considered to be part of the dorsal stream.

The ventral stream

The ventral stream, also called the what channel, will be of special interest in this study since it is responsible for detailed recognition of the visual input. Information passes through the secondary visual cortex (V2) where it is transmitted through V4, sharing a similar organization to V1, and finally transfered to the inferotemporal cortex (IT). Neurons in the IT respond selectively to more complex shapes such as faces.

Anatomical top-down and recurrent connections

Although input up to area V2 is strongly feed-forward, back-propagations are in fact more abundant than purely feed-forward inputs when looking at the whole picture of the brain areas involved in vision[10, 11, 6]. Recurrencies are introduced in several ways such as horizontal connections within each area, bypass and top-down connections between areas and even connections between dorsal and ventral streams[12]. The high number and different type of connections between visual areas form a very complex network as presented for macaque monkey in **Figure2.2**[13]. The computational contributions and scope of these recurrences is not yet clearly understood. It was however argued that they are likely involved in the specific task of partial object recognition[1] which will be the focus of this thesis. They also play a role in many other interesting phenomena such as dynamical change in receptive fields properties like preferred orientation, position or size[14], providing evidence that even the primary visual cortex is not just a static feed forward spatio-temporal bank of filters.

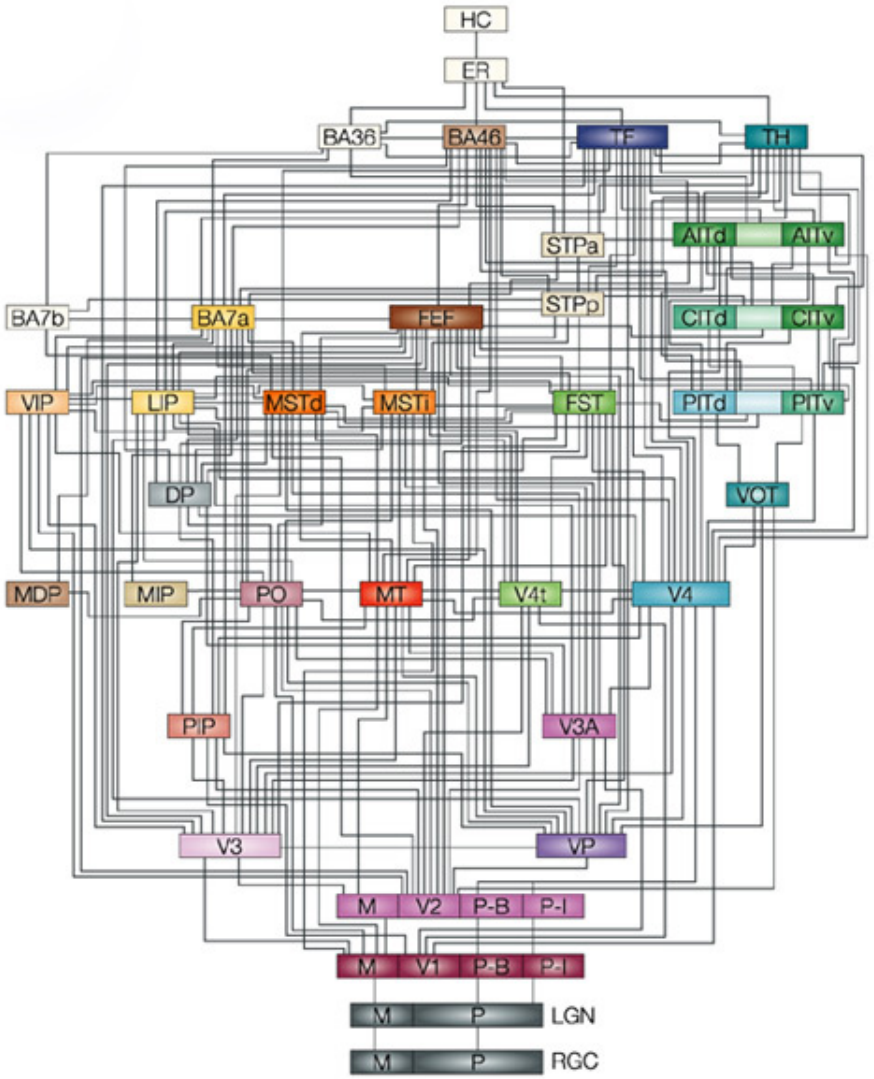


Figure 2.2: Layout and demonstrated anatomical connectivity of macaque visual cortex[13]. Information is gathered by the eye's retinal ganglion cells at the bottom of the hierarchy (RGG). It is then processed by up to thirty-two visual cortical areas, two sub-cortical visual stages and several non-visual areas before ending up in the hippocampus (HC). A total number of 187 anatomically demonstrated mostly reciprocal links connect these different areas, showing the extent of potential complexity neuroscientists are faced with when studying vision and the important to try to constraint, through limited timing or masking techniques, the number and type of connections used by the brain.

2.1.2 Properties underlying contour completion in early brain areas

It is thought that contour completion is one of the initial steps involved in a partial object identification task. But even this first approach is an ill-defined problem since an infinite number of solutions can be consistent with the visual input. Exactly how the brain is successful in finding the right solution in a fast and mostly reliable way even for very high percentages of occlusion remains unclear. However some characteristics of contour completion when edges are in close proximity have been discovered through numerous psychophysics studies. A summary based on the review of H. Tang and G. Kreiman[15] will be provided here.

Modal and amodal completion types

Modal completion takes place when an observer is able to build a mental model of the image thanks to illusory contours revealed with the help of inducers[16] as shown in **Figure 2.3 b)**. However since this setup does not happen in nature, focus will be set on amodal completion which happens when an explicit occluder is hiding part of the image, as presented in **Figure 2.3 a)**. For non extreme occlusion percentages, an observer would be aware of the overall shape despite not seeing some of its contours[17].

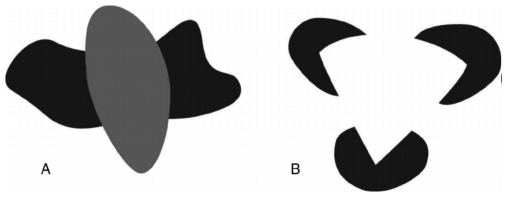


Figure 2.3: Figure from Wagemans et al. [18], adapted from work by Singh et al. [19]
A. Amodal completion of the black shape by a gray occluder.
B. Modal completion: a white triangle shape is seen although contours are illusory because of the three black inducers.

Contextual modulation

An interesting property of amodal completion is that it relies on identification of the different depths within the image inferring that information from outside individual receptive fields is combined to enable accurate surface-based representation [20].Contextual modulations is unsurprising in higher brain areas which have performed multiple pooling operations already but interestingly, it was also suggested to happen to some extent at the level of the primary visual cortex. Indeed, a small percentage of the orientation specific cells (12%) of V1 responded strongly to positive disparities depicting an occluder presented in front of the moving bar but not to zero or negative disparities [21]. This behavior could be explained mainly by lateral connection and potentially close-proximity feedback loops of adjacent areas. Modal completion experiments also favor the implications of such connections since illusory contour responses were measured fist in V2 and only later appeared in V1.

The importance of an explicit occluder

Another property of amodal completion was that the presence of an explicit occluder compared to just erasing part of the image was found to make the completion task easier. The B letters of Bregman, presented in **Figure 2.4** is a famous example of this effect. A similar observation was made while performing a forced categorization task on images presented through gaussian bubbles with either gaps filled by occluders or simple background as shown in **Figure 2.5**. The performance was significantly higher for high percentages of occlusion (more than 75% of the pixels missing) [22].

2.1.3 Responses of higher brain areas to occluded shapes

Higher visual area were also studied and some interesting observations were made. In V4, selectivity was maintained for various curvatures occluded by dots within a certain range providing evidence for the involvement of contour-based mechanisms in segmentation and subsequent recognition of partially occluded images[26].

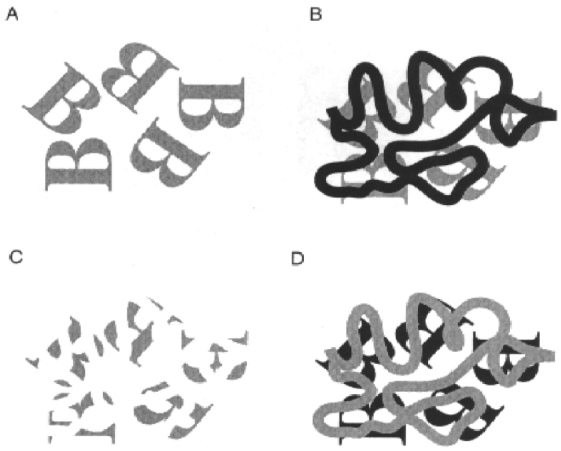


Figure 2.4: Bregman-Kanizsa Display [23, 16, 24]: B letter identification is easier when an occluder is filling the gaps compared to the deleted counterparts.
A. unoccluded B letters,
B. Occluded B letters,
C. B letters fragment counterparts. Identification is harder in this case.
D. occluded B letters with different contrast.
Figure from Kelly et al. [25]

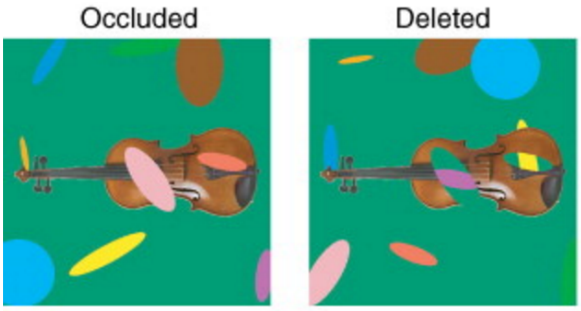


Figure 2.5: Example of sample images used by Johnson et al. [22] for their experiment. Presence of an occluder compared to simply deleting image pixels strongly increased subject performance for occlusion/deletion of 60% and 75% of pixels. The violin pictured here has 20% of missing pixels in both conditions.

Another study[27] focused on the IT neuronal responses and presented naturalistic scenes which were occluded by generating a bubble mask through which the image is seen. By exploring the effect of different bubble locations in a behavioral experiment involving humans and monkeys, they first showed that both species had similar results and further confirmed the intuition that some information in the image is more important for object recognition than other depending on the object type. In parallel, intra-cortical measurements were carried out in monkey using the same experimental paradigm [28]. Results showed that for occluded scenes where features with diagnostic value were conserved, firing rates and local field potentials of neurons in the IT remained mostly invariant to very high amounts of occlusion. In contrast, when only non diagnostic parts were shown the absolute magnitude of the responses varied linearly with the percentage of occlusion.

Finally, intra-cortical recording in humans exposed to naturalistic objects, again occluded using the bubble paradigm, measured neuronal responses in the fusiform gyrus that remained similar for images with up to 89% of occlusion[29]. Interestingly, some of these images shared no common pixel and still triggered similar neural local field potentials.

2.1.4 Spatio-temporal dynamics of object completion

Intra-cranial recording in epilepsy patients allowed Tang et al. [29] to evaluate how and when visually selective responses to occluded objects appeared. They observed that neural responses along the inferior occipital and fusiform gyri were still selective to partial objects showing only 9 to 25% visibility compared to the object's whole counterpart. Despite differential occlusion and therefore variation in specific feature presentation across trials, recorded intra-cranial field potentials (IFP) waveforms, amplitudes and object preferences were similar between whole and occluded conditions. However IFP responses were delayed by approximately 100 ms for partial condition as shown in **Figure 2.6**, which can be contrasted to image transformations such as scale, position or rotation which do not trigger delays [30, 31, 32, 33, 3]. The observed latency difference remained significant after controlling for variations in contrast, signal amplitude and selectivity strength. Moreover, consistency was maintained when using different frequencies bands and different statistical comparisons. These delays were particularly pronounced in higher brain areas within the ventral stream. When comparing with other studies, it is clear that it is the presence rather than the exact value of the delay that is characteristic of occluded stimuli across different experimental paradigms.In another experiment[34] analyzing amodal completion of more complex natural images such as faces, the delays were closer to 200 ms. Yet another study[32] focused on neural responses in areas not only involved in vision and observed delays ranging from 200 to 500ms.

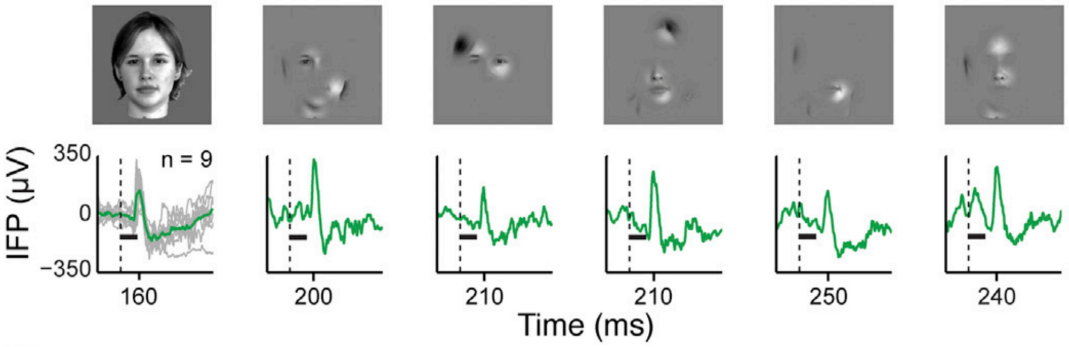


Figure 2.6: Figure from the work of Tang et al. [29] showing an example of intracranial field potential responses to a whole face stimulus (left) and its five occluded counterparts from an electrode in the left uniform gyrus. For the whole condition, average over 9 responses is in green while single trial traces are shown in gray. For the Partial condition single trial responses are in green and each trial's stimulus was a different occlusion pattern of the whole image presented left. The dashed line indicates the stimulus onset time and the black bar corresponds to stimulus presentation duration.

It is very unlikely that these delays are due to slower speed of information flow through a purely feed-forward visual hierarchy for partial objects compared to whole ones because early visual areas, such as V1, did not show significant delay in the response latency. Delayed responses could thus be used as indicators of recurrent or feedback modulations [35, 36, 37, 38].Moreover, the timing between long feedback loops connecting IT and V1 and shorter ones between V1 and V2 should also be distinguishable[32]. Therefore, timing of neural responses can be used to hypothesize the presence of proximal or distal feedback loops. Previous studies[11, 12] further argued that horizontal and feedback connections present throughout the visual cortex are the most probable components involved in these recurrent modulations . Finally understanding the specifics could ultimately

lead to creating more efficient algorithms for artificial intelligence since the observed temporal dynamics indirectly show restrictions on the number of underlying computations involved in the brain.

2.2 Computational context

2.2.1 Definition of Artificial Neural Networks

At the basis of every neural network is a single unit called a "neuron" which can be modeled by the very simple architecture presented in **Figure 2.7** (left). Mathematically, the activity of a single unit u is computed by applying a non-linear activation function f to the combination of an input vector X with a weight vector W (including a bias term) as defined by the following equation:

$$u = f(\sum_i w_i x_i)$$

Neural networks then combine many of these units, each defined by their specific weights, into rows, referred to as layers. In computer vision, each layer outputs a set of features to which its neuronal ensemble preferentially responds. Complexity is then increased throughout the network by stacking together multiple different layers, one layer becoming the input of the next as depicted by **Figure 2.7** (right). Such networks are then trained to extract useful features for image recognition using backpropagation, a process we will define later. During training, outputs from lower layers become sensible to certain edge orientations, colors or curvatures while higher layers will eventually respond selectively to faces for example.

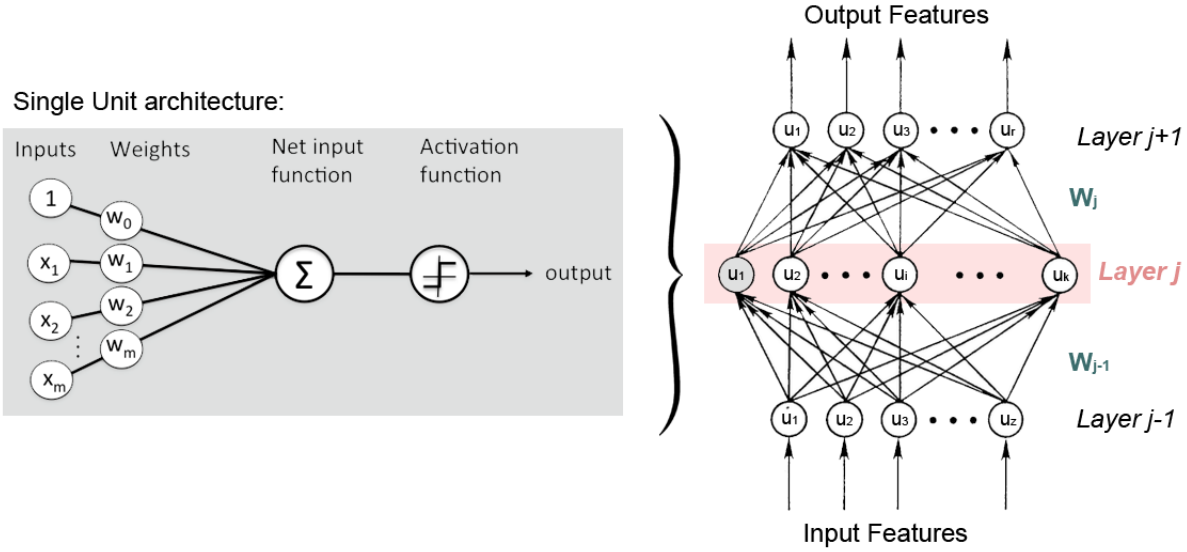


Figure 2.7: Basic architecture of a neural network. Several single units are combined in a layer. Layers are then stacked together, the output of one layer becoming the input to the next layer. In computer vision, complexity of extracted features thereby increases as information flows through the system, in a similar fashion as the feed forward hierarchy of the brain. The connections between layers are determined by weight matrices W, updated to minimize error during training of the network. At the single unit level, a non-linear activation function is applied to the combination of inputs and learned specific weights of a «neuron ».

Artificial neural networks were originally inspired by the biological observations[8, 39] described earlier but quickly became dominated by mathematical optimization[40] rather than advances in research about the brain's anatomy and connectivity. Despite this divergence, performance achieved by several deep convolutional networks on the ImageNet large-scale dataset[41] increases every year, becoming comparable to human data [42]. Examples are listed in table **??** from which one can observe that adding layers seems to be necessary to reach higher performance. Increasing network depth does however require additional precautions to address new challenges such as the vanishing gradient problem which will be presented later. Moreover, adding layers to the architecture while keeping the feed-forward reflects only a very minor percentage of the connectomics of the brain since most connections are horizontal or feedback connections[14] as argued earlier. Recent studies[1] have hinted that this might be the explanation of feed forward model's poor performance with more complicated tasks requiring pattern completion or context awareness. Just adding complexity to the model without making advances in the underlying learning mechanisms leads to models that are harder and harder to predict or understand. Nonetheless, neural networks of increasing depth can perform a variety of tasks by combining different concepts and techniques, some of which will be presented below[43].

Year	Neural Network	Number of Layers	top-5 error
2012	Alexnet	8	16.4%
2013	ZFNet	8	11.7%
2014	GoggleNet	22	6.7%
2015	ResNet	152	3.6%

Table 2.1: State-of-the-art neural networks that won the ImageNet contest [44] . The number of layers gives an idea of their complexity while their performance is depicted by the top-5 object classification rate on the ImageNet test set. Top 5 error is less conservative than other error measurements since it only requires the network to narrow down the output to 5 potential labels which must contain the correct one.

2.2.2 Supervised Learning in Artificial Neural Networks

Backpropagation and Gradient Descent

Backpropagation[45] is a supervised learning technique that repeatedly adjusts the weights of the connections in the network by minimizing an objective function quantifying the error the model makes when predicting the label of the input. The most commonly used functions serving this purpose are the L_1 loss and L_2 loss, also called mean squared error (MSE) and are defined by:

$$L_1 = \sum_i \mid y_i - \hat{y}_i \mid$$

$$L_2 = \sum_i (y_i - \hat{y}_i)^2$$

where $\hat{y}$ is the output predicted by the model and y is the ground truth provided in addition to the dataset. The L_2 metric can be interpreted geometrically as the euclidean distance between the two vectors. Its gradient is the difference between the prediction and the true label and the L_2 loss is therefore very sensible to outliers.

During backpropagation, errors computed with an objective function L are propagated back iteratively through the network by applying stochastic gradient descent (SGD) to the weights using the following mathematical formula[**?**]:

$$W = W - \mu \nabla_W L_W(X_i, Y_i)$$

where W is the weight matrix, μ is the learning rate and $\nabla_W L_W(X_i, Y_i)$ the current gradient approximation for input/output pair i from the training set. The weight parameters update along the direction of the gradient of the objective function is iterated until a minimum is reached. The particularity of SGD compared to classic batch gradient descent is that it computes an approximation of the gradient for one input/output pair at a time. SGD therefore performs updates with higher variance, enabling it to jump to new solutions but also causing instability through high fluctuation. The mini-batch gradient descent combines advantages of both batch and SGD by updating the weights for a subset of training examples. This approach allows computations to take advantage of big matrix multiplications which are highly optimized in GPUs [46]. Momentum[47] can also be added to the gradient descent to prevent oscillations. It can be incorporated by simply adding a fraction of the update vector v of the past time step to the current time step as shown in the following equation:

$$W = W - v$$

$$v = \gamma v - \mu \nabla_W L_W(X_i, Y_i)$$

where μ is the momentum coefficient, representing the memory of previous gradient directions. Therefore momentum is amplified in directions where the objective function was persistently decreased over multiple time steps [48] and convergence is faster.

Backpropagation can also be used to help visualize and better understand what exactly deep neural networks learn by performing gradient descent in the input images space [49] rather than weights space. One can thereby find the optimal stimulus for each unit and which features maximally activate a given layer but this method does not give information about the unit's invariance.

Activation functions

Activation functions applied to the weight-input combination aim to introduce a non-linearity after successive element-wise summations and multiplications. Some examples are provided in **Figure 2.8**.

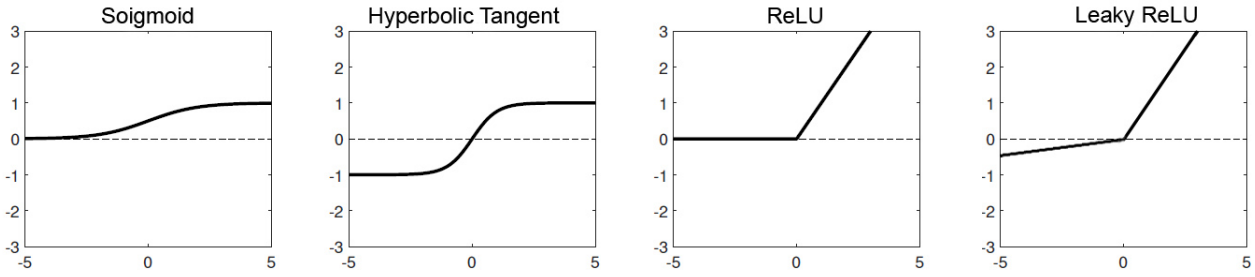


Figure 2.8: Different frequently used activation function. A common and necessary feature is non linearity. Sigmoid and hyperbolic tangent are saturating, increasing the problem of vanishing gradient while ReLU provides faster learning and sparser weights. Leaky ReLU is an attempt to minimize the risk of "dead" ReLU were the system is unable to update because of negative input and zero gradient.

Although sigmoid or hyperbolic tangents are still frequently used, choosing rectified linear unit (ReLU) was found to decrease training time while reaching comparable accuracies[50]. It is often argued that ReLUs are more biologically plausible since studies indicate that cortical neurons rarely exhibit a maximum saturation regime. Its most interesting properties stem from its mathematical definition:

$$f(x) = max(0, x)$$

One immediate advantage is that the gradient will not vanish during backpropagation since it will have a constant value of one or zero. For sigmoids on the contrary, the gradient would decrease exponentially through the layers thereby slowing down the learning process significantly. The impact of hard saturation at zero on optimization with ReLU can of course also be negative if too many neuronal contributions are canceled since, once the gradient is null and the input is negative, there is no way for the network to recover and update its output. Although this issue rarely happens, a common precaution is to use a leaky ReLU by using $f(x) = max(0.01x, x)$ for example. The vanishing gradient problem is then again present but still greatly diminished compared to sigmoids. However, there are also advantages to keeping the initial definition of the ReLU. Indeed, it will generate sparser weights, meaning that less neurons will be used in the network and features will be less inter-dependent which greatly reduces overfitting. In a randomly initialized network for example, it was found that about 50% of the hidden units had a non zero output[51]. In contrast, most other activation functions are saturating and will always output a very small non zero value keeping a small contribution for all neurons and generating very dense solutions.

Weight Initialization

As the network is trained, it will update its weights in order to maximize its performance on the given training set. If the weight matrix W is properly normalized it can be expected that the number of negative and positive weights would cancel out. However initializing all the weights at zero would not yield good results since every neuron would compute the exact same output, leading to computation of the exact same gradient during backpropagation of the error and thus the same parameters. To break the symmetry, it is best to update the weights to small random values close to zero so that each neuron generates a random but unique output.

A robust initialization method[52] for a single neuron for deep models using the ReLU activation function is given by the formula:

$$w = random(n) \times \sqrt{\frac{2}{n}}$$

where n is a random number are drawn from a Gaussian distribution and the variance is divided by the number of input units n. The scaling of the variance is important so as to stop it from increasing with the number of inputs.

Another interesting initialization technique is batch normalization[53] which makes normalization a part of the model's architecture. Indeed during training of deep neural networks, the distribution of a layer's input changes over time because of the stacking of the layers. This problem, called the internal covariate shift, may lead a significant slow down of learning because the distribution then moves towards the saturation regime of activation functions such as sigmoids. Batch normalization is inserted at the end of every layer of the model, just before applying the non-linearities and forces the activations to take on a unit Gaussian distribution by normalizing

features with respect to the mean and variance of each mini-batch independently. It is therefore more robust to sub-optimal initialization, reduces the internal covariance shift and allows higher learning rates.

2.2.3 The building blocks of neural networks

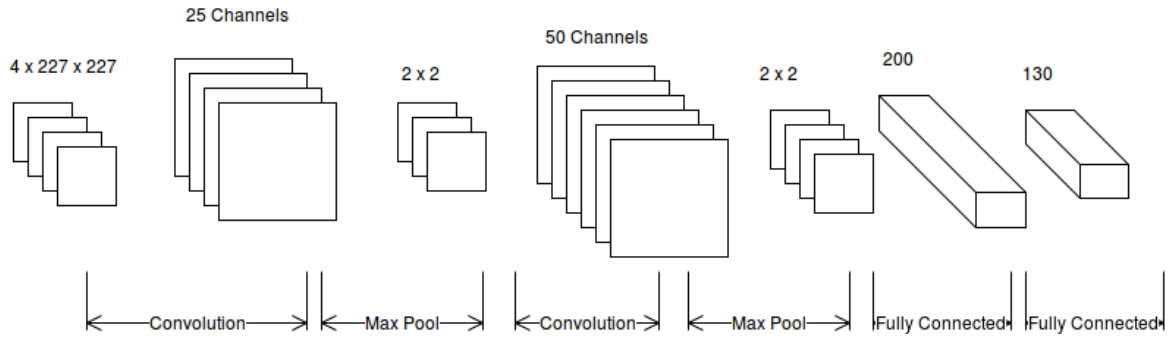


Figure 2.9: Example of a neural network architecture containing two convolutional layers, two pooling layers and two fully-connected layers. Image was taken from `https://filebox.ece.vt.edu/~aroma/web/cv_project_15/approach.html`.

Convolutional Layers

A convolutional layers consist of a three dimensional arrangement of parameters called neurons. Each of these neurons is locally connected to every pixel of a small specific input volume called its receptive field. Neurons linked to the same receptive filed are organized into a depth column. Aligning such depth columns along each image volume defines a depth slice. An underlying hypothesis for dimensionality reduction in the case of convolutional layers is that some features, especially low level ones, that are relevant for some area of the input have high chances to also be important for other areas. Therefore, neurons of a single depth slice share the same weights and they can be implemented as a spatially small filter that is slid across the input's height and width while extending throughout the input's depth (three channels for RGB images) in order to be mathematically consistent. The output of the layer is a two dimensional activation maps for every filter used. These maps are generated by computing the dot product between the weights of the filter and the input while sliding the filter across each receptive field location. The hyper-parameters to be chosen and optimized include the amount of filters used per convolutional layer K determining the depth of the output volume, their spatial extent F also called receptive field size, their stride length S and the extent of zero padding P in order to extract information from the borders of the input space. Thanks to weight sharing, the number of weights to optimize is decreased to $W = K \times F \times F \times D$[43].

During training, the network learns specific weights for each filter using backpropagation. As a result, each filter ultimately learns to detect different low level features such as curvature or edges along certain orientations for early layers as presented in **Figure 2.10** or more complex features such as faces or wheels for higher level layers.

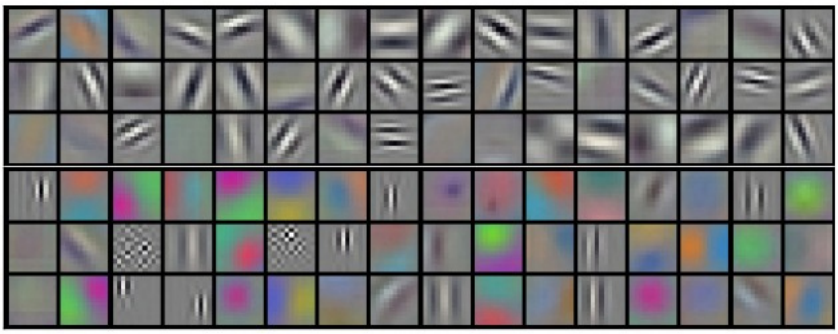


Figure 2.10: Examples of filters learnt by the first convolutional layer of Alexnet. Each of the 96 filters composing the convolutional layer is of size [11x11x3] for RGB images and each filter is shared by the 55*55 neurons in one depth slice. Image was taken from the course `http://cs231n.stanford.edu/`

Pooling layers

Pooling layers are frequently introduced between successive convolutional layers to reduce width and length of the representation and thereby reduce the computational cost by lowering the amount of parameters to optimize. It commonly downsamples the output of each convolutional layer filter by applying a max operation with filters of size 2x2 and stride 2 as shown in **Figure 2.11**. Other pooling functions such as average pooling or L2-norm pooling can also be applied. The depth of the convolutional layer output (i.e. the number of filters) remains of course unchanged and no additional weights are introduced in this process.

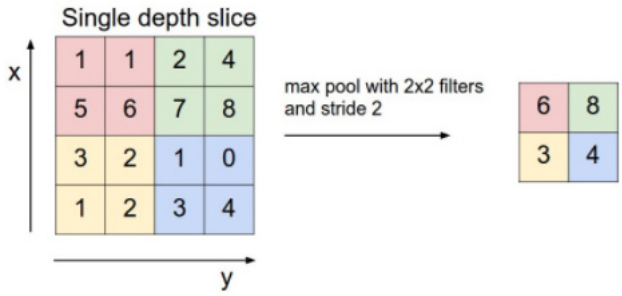


Figure 2.11: Example of a 2x2 filter slid with a stride of 2 across the height and length of the activation map of a specific filter of a convolutional layer[43]. Image was take from the course `http://cs231n.stanford.edu/`

Fully connected layers

Fully connected (Fc) layer neurons have full pairwise connections with neurons of the previous input layer but no self connections. The output is thus simply a matrix multiplication followed by a bias offset.The number of weights to be learned is therefore determined by the number of input neurons multiplied by the number of output neurons. Fully connected layers are mostly used to learn non-linear combinations of the high-level features of the last convolutional layers and reduce dimensionality to a one dimensional vector. The last fully connected layer usually classifies the features into the different classes of the dataset labels. It has frequently been observed that they are less generalizable than convolutional layers because of their higher feature specificity.

Recurrent layers

One of the limitations of feed forward networks is that they have a hard-wired number of computational steps determined by the number of layers as well as a fixed input and output size. Recurrent layers (RNN) on the contrary allow for operations over sequences of vectors and have a memory which captures information of previous computation steps. Interestingly, any input can be converted into a sequence and benefit from recurrent layers. With visualizations algorithms for example, images can be read patch by patch from one direction to the other.

Mathematically the internal state of these layers h is updated by combining information from the input x at time t and the previous state of the system at time $t-1$.

$$h_t = f(Wh_{t-1} + Ux_t)$$

where U and W are the specific combination weights which enable to fine tune the importance given to information in the past compared to information in the present. The function f is a chosen activation function, usually a ReLU. A prediction y is then given by multiplying the current state with a weight matrix V following the formula:

$$y_t = Vh_t$$

It was argued earlier that convolutional layers allow for complexity reduction by sharing weights through space. Recurrent layers also exhibit a similar advantage but by sharing the weights (U,V and W) over time. Therefore recurrent layers can be unfolded into multiple stacked fully connected ones which would in this case all share the same weight matrices as shown schematically in **Figure ??**. This approach is the most common way of training RNNs in practice: each time point corresponds to another fully connected layer and is stacked on top of the previous ones.

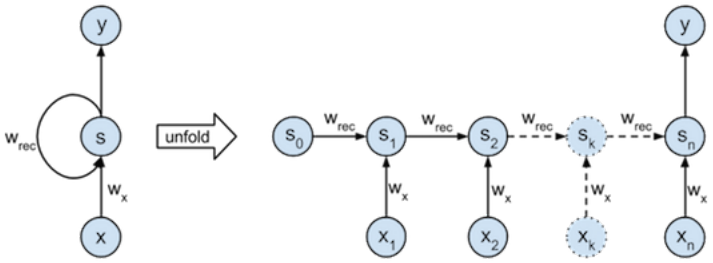


Figure 2.12: Unfolding of a recurrent layer into its feed-forward equivalent. Notice how the weights w_{rec} are shared across time and the input weight w_x is kept constant. The state of the network is updated at each time steps through the combination of the input and the recurrent contribution. Image was take from `http://peterroelants.github.io/posts/rnn_implementation_part01/`

Since h and y are updated at every time step, each new state of the system should contain traces of all states that proceeded it. Although this memory is in theory infinite, in practice it is limited because of the vanishing gradient problem to which RNNs are particularly sensible[54] because error is now backpropagated through time[55]. This backpropagation technique is implemented in practice by running standard backpropagation through the previously described equivalent feed forward artificially deep network. Weight values are typically selected to be somewhat distributed

across neurons, by the L2 regularization for example, and are therefore usually very small. As the number of time steps gets larger, the number of added layers increases and the number of times these weights are multiplied also increases which decreases the value of the weights more and more, eventually leading to an exponential decay of the gradient. The vanishing gradient then pushes weight values even closer to zero and training can no longer function efficiently.

Networks architectures such as Long-Short-Term-Memory (LSTM)[56] have recently bypassed this problem by adding gated memory units to each layer which enable fine-tuning of the amount of memory exposure and when to forget information or reset memory. Contrary to standard RNNs, LSTMs update the current state of the system using additions rather than multiplications and do not apply an activation function. Specific features can thereby be remembered without being "deformed" by the activation function and the error can be backpropagated efficienly, reducing the gradient decrease over time.

2.3 Project aim

The speculation made by Tang et al.[29] is that observed neural delays can be attributed to recurrent computations relying on prior knowledge about objects to be recognized[57]. By using novel objects, the aim of this thesis is to minimize the prior knowledge of human subjects progressively strengthened through extensive and repeated exposure to images. Behavioral experiments mainly involving occluded stimuli were conducted and contrasted to previous experiments from the Kreiman Lab using familiar objects[1]. These results will then be linked to performance of both feed forward state-of-the-art neural networks, such as Alexnet and HMAX, and neural networks with added recurrent connections using the same dataset as the one designed for the psychophysics experiment. Through this work, it is hoped that increased performance of recurrent neural networks generalizes well to completely novel shapes, providing a robust implementation for artificial intelligence applications involving recognition of occluded shapes.

Chapter 3

Materials and methods

3.1 Novel Images used

Novel object were created by manipulating visual attributes so as to generate non-familiar stimuli for human subjects of behavioral experiments. By using these artificial objects, scientists aim to minimize the influence of subject's previous experience on behavioral effect. They can thereby be more confident that observations are mostly due to the researcher's manipulations on the stimuli. Moreover, these stimuli are also a powerful tool for investigating human's learning ability. Five hundred unique novel objects used for the experiment were chosen from the Center for the Neural Basis of Cognition stimulus repository[1]. An equal number of selected objects were taken from five different categories described below containing 4 families each. In total, each family comprised twenty-five unique exemplars, adding up to a hundred images per category.

The Fribble stimulus sets was built using the work of Pepper Williams. The shape and texture of the main bodies as well as the approximate location and inter-relationships between appendage parts remain constant for all exemplars of a particular category. The main diversity between exemplars is introduced by the variation of the appendage parts' shape and texture. Each exemplar is thus defined by a unique conjunction of appendage parts while its category can be simply inferred from looking at the main body's shape. For our purposes, 3 different categories were chosen: one having a horizontal cylindrically shaped main body referred to as α, one having a ball-like shaped main body referred to as γ, and one having a vertical cylindrically shaped main body referred to as β. Greek letter names were chosen in our experiment so as not to influence subjects from the behavioral experiment with potential meaning of an invented name. Three exemplars belonging to the same family but different categories are presented in **Figure 3.1**

Stimuli from the Gribbles and Yu's Un-Facelike Objects (YUFOs) stimuli datasets were selected to compose our categories δ and ϵ respectively. These artificial images were originally designed by Scott Yu for different investigations. Indeed, three dimensional shapes of each exemplar are more consistent inside each specific category than for the Fribble objects mentioned above. Instead variation is mainly introduced through different object viewpoints and lighting conditions. An object identification task will therefore require more attention to details, as would be the case for faces, but symmetry is not respected here, making the images "un face-like". Three examplars of each category are shown in **Figure 3.2**.

[1]Novel objects, Center for the Neural Basis of Cognition, `http://wiki.cnbc.cmu.edu/Novel_Objects`

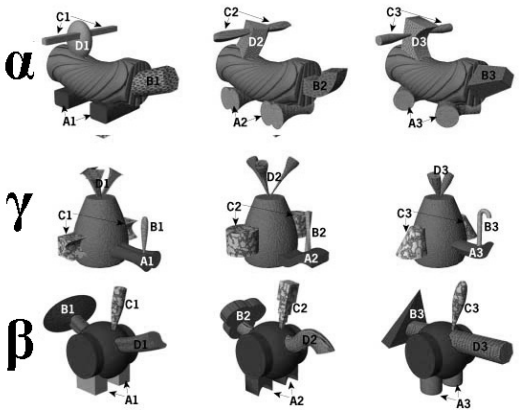


Figure 3.1: Categories of the Fribble Stimulus Set from the work of Pepper Williams. Presented categories α, β and γ were used for our experiment and can be identified thanks to the shape and texture of the main body. Three different exemplars per category with a unique combination of main body and appendage parts are displayed labelling specific appendage parts A, B, C and D.

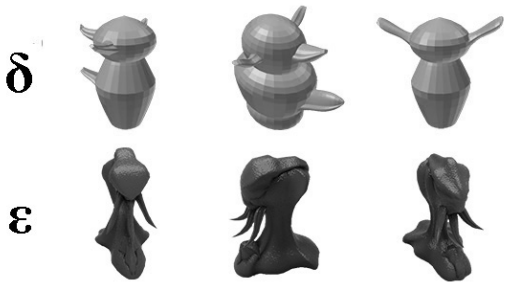


Figure 3.2: Categories of the Gribble (named δ, shown at the top) and YUFO (named ϵ, shown at the bottom) stimulus sets from the work of Scott Yu. Three different examplars of presented categories $\delta$and ϵ are shown. Main intra-category variation is due to lighting conditions and change in view point of the 3D model.

3.2 Psychophysics Experiment

A psychophysics visual recognition experiment was designed using Matlab's Psychtoolbox to assess human performance on a categorization task comprising a range of heavily occluded (used for about 85% of the trials) and whole (used for about 15% of the trials) versions of images described in Section 3.1.

3.2.1 Experimental setup

Subjects were recruited (n=23, ages ranging from 20 to 34, 11 female) and had to perform 5 way alternative forced choice categorization (AFC) on the novel objects. Since they have no previous experience with the presented stimuli, a training phase had to be validated before assessing partial object performance. A compromise between minimal exposure and correct classification into the 5 different categories (α, β, γ, δ and ϵ) for whole images had to be made. Therefore, the experiment first consisted of a short demonstration of one fully visible example of each category and its corresponding name. Then a training phase was implemented comprising of 10 whole novel images (2 from each category but from unique families) and subjects were required to get at least 5 times in a row 8 out of the presented 10 images names right before being allowed to proceed to the experiment assessing human performance on occluded version of the objects. Subject needed on average 80 exposures to pass the test (i.e. to get at least 8 images right out of 10 five times in a row). The standard deviation among subjects was 40 exposures. The same test was repeated once more in the middle of the experiment with the pass condition lowered to three trials. This control was implemented to verify that subjects remembered the categories of full images well so that performance can be fully explained by the object occlusion effect. As expected, all subjects passed the test with the minimal imposed number of trials.

During the experiment that followed, subject were presented with 1,000 stimuli resulting from uniquely renderings of 500 contrast-normalized and grayscaled novel images, resized to 256x256 pixels so as to span about 5° of the visual field. Amount of presentation time of the stimuli, termed stimulus onset asynchrony (SOA), was randomly chosen to equal 25, 50, 85, 100 or 150 ms for each stimulus and image presentation was followed for 500ms by a gray screen (unmasked condition) or spatial noise pattern (masked condition) chosen with equal probability. A choice screen was then displayed and subject's answer was recorded using a gamepad. The classification performance is defined simply as the percentage of correctly labeled categories and compared to chance level which is 20% since 5 categories are present. Eye movement was also recorded during the whole experiment for 21 subjects out of the 23 using an infrared camera eyetracker at 500Hz from Research Ontario, Canada. It was mainly used to ensure subject attentiveness during the task by monitoring fixation on a cross positioned at the center of the screen for at least 500ms before allowing stimulus presentation. A summary of the behavioral experiment setup can be found in **Figure 3.3**.

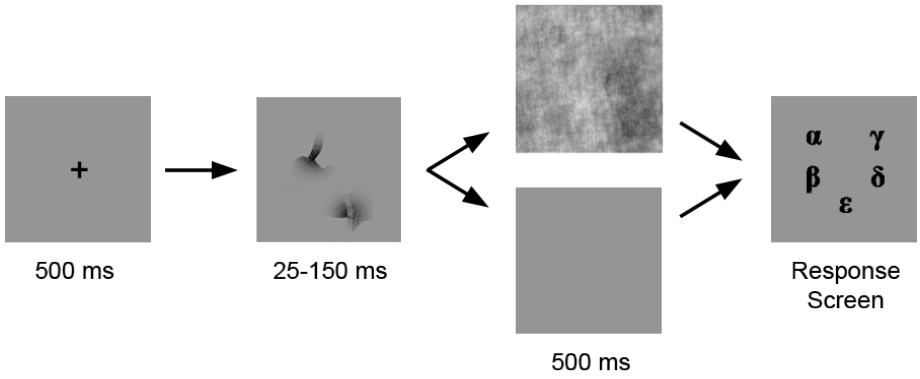


Figure 3.3: Novel Objects Occlusion psychophysics experiment setup: fixation was required for at least 500ms in order to display occluded (shown here) or whole image for SOAs varying from 25 to 150 ms. Image presentation was either followed by a neutral background (unmasked condition) or a spatial noise mask (masked condition) for 500ms. The choice between the two conditions was made with equal probability. Finally they had to classify the presented image into one of the 5 categories shown on the response screen using a gamepad.

3.2.2 Building the dataset

The stimuli dataset was built from a database of 500 images uniformly subdivided into 5 categories named α, β, γ, δ and ϵ. Examples can be found in Appendix 5.3. Image background was extracted for each image using a combination of histogram bin counting, [3x3] median filtering and erosion by a [2x2] disk for category ϵ and a [3x3] square for the other categories. Through this process, 500 background masks were defined in order to enable computation of specific image visibility after occlusion without the contribution of uninformative background pixels. Next, all images were contrast normalized using the `histMatch` matlab function from the `SHINE` toolbox to reduce potential influence of low-level features especially relevant for the feature extraction which will be performed by computational models on the dataset in a complementary experiment. This function equates the luminance histogram of sets of images to a computed input specific average target histogram while ignoring the image background specified by the background mask. It additionally uses the SSIM method of Avanaki[58] which optimizes histogram specifications for structural similarity.

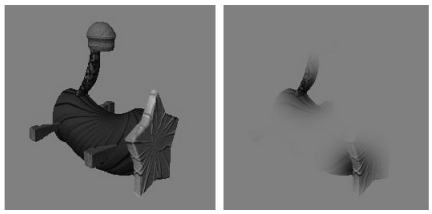


Figure 3.4: Typical whole object (category α, left) and occluded counterpart (shown right)

In the next step, a thousand unique renderings per subject were obtained by applying different occlusion patterns to the original images, resulting in a total of 23,000 different stimuli. An example of a whole object and its occluded counterpart is shown in **Figure ??**. Subject specific image presentation, masking condition, SOA and whole or occluded condition orders were attributed by random index permutation. Object occlusion was implemented by generating randomly positioned Gaussian bubbles through which the image is shown. This means that any part of the image outside the defined bubbles was hidden. The bubble paradigm used[59] enables evaluation of spatial integration of multiple parts to achieve recognition. The number of bubbles was fixed to 5 and the standard deviation of the Gaussian was experimentally chosen to be 24 so as to optimize task difficulty, trying to maximally span heavily occluded images while keeping the subject motivated for the experiment. Bubble centers locations were generated through random uniform sampling and bubbles were created using the following algorithm:

```
function [bubbledImage, mask] = ...
    AddBubbles(img, bubbleCenters, bubbleSigmas, color)
% centers are indexed over numel(img)
myeps = 10^-8;
bubbledImage = double(img);
mask = zeros(size(img));
[y, x] = ndgrid(1:size(img, 1), 1:size(img, 2));
[yc, xc] = ind2sub(size(img), bubbleCenters);
for i = 1:length(xc)
    maskt = exp(- ((x - xc(i)).^2 + (y - yc(i)).^2) / 2 / bubbleSigmas(
        i)^2);
    maskt = maskt / max(maskt(:));
    mask = max(mask, maskt);
end
mask(mask < myeps)=0;

foreground = color / 255;
m = max(255, max(bubbledImage(:)));
bubbledImage = bubbledImage / m - foreground;
bubbledImage = bubbledImage .* mask + foreground;
bubbledImage = uint8(bubbledImage * 255);
end
```

where:

– *img* is the array of the whole grayscaled image

– *bubbleCenters* is the array with the coordinates of the center of each of the 5 bubbles

– *bubbleSigmas* is the standard deviation which determines the size of the bubble

– *color* is the background color in grayscale

– *bubbledImage* is the occluded image

– *mask* is the bubble mask though which the image will be seen

Image visibility was defined as the ratio of total amount of visible object pixels after and before adding the bubbles to the image. Background pixels were not considered in this computation since they are uninformative of object category.

3.2.3 Experimental conditions

We were particularly interested in human performance for high occlusion conditions since this is where purely feed forward computational models fail to match human scores in this context of perfectly isolated objects. Most image renderings therefore resulted in occlusion patterns spanning the 0 to 40% image visibility range as shown in **Figure 3.5**. As a control, 15% of image renderings were presented to the subject without occlusion (whole condition).

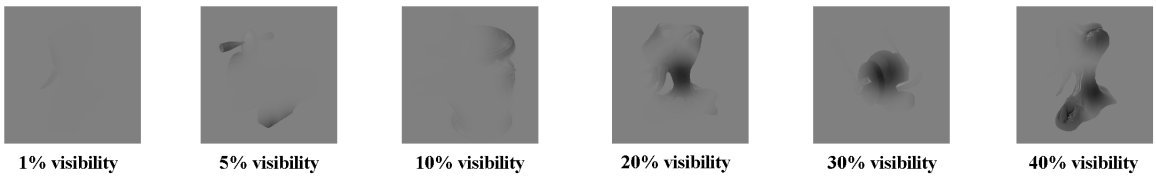


Figure 3.5: Examples of partial object with different visibility percentages

Moreover, we wanted to analyze the effect of presenting a mask directly after image presentation, depending on amount of occlusion and SOA. Indeed, backward masking and short SOAs have been used in several psychophysics experiments as tools to approximately isolate the feed forward information flow from most of the numerous horizontal and feedback connections that modulate it[60]. Several studies[61, 62, 63, 64, 65, 66] previously showed that presenting a high-contrast spatial noise mask stimulus directly after an image presented with a short SOA efficiently interrupted any additional processing, thereby strongly limiting modulation by recurrent connections in the visual pathway. Therefore, each of the 500 images was occluded twice in our experiment, generating two unique renderings per image with one rendering used for the masked and one for the unmasked conditions. A specific spatial noise mask was computed for each rendering using its corresponding whole image. This approach was motivated by the observation that mixing the spectrum noise bands with information of the image's spectrum is more effective at interrupting neural responses[67].

The formula used for each grayscaled 256x256 whole image was thus:

$$mask = \Re(\mathcal{F}^{-1}(A \times exp^{iI}))$$

where:

A is the average across images of the complex modulus of the fourier transform of each image,

$I = atan(\Im(Z), \Re(Z)) + atan(\Im(H), \Re(H))$
with Z being the fourier transform of the grayscaled 256x256 image and H being a 256x256 matrix of pseudorandom values drawn from the standard uniform distribution on the open interval $(0, 1)$. The $\Re$ and $\Im$ symbols refer to the real and imaginary part of a complex matrix while $\mathcal{F}$ denotes the Fourier transform.

3.3 Feature extraction using feed forward neural network models

3.3.1 HMAX

For a simple task such as whole and isolated object identification, visual processing in the cortex can be represented by a hierarchical feed forward information flow with increasing complexity as explained in Section 2.1. The HMAX model[39] was the first quantitative computational model to conform to anatomical and physiological constraints, extending the principle of Hubel and Wiesel's model[8] of simple and complex cells to other brain areas than V1. This early model is purely feed forward and its architecture was inspired by the visual cortex from V1 to IT by combining simple cells (S layers) performing linear template matching and complex cells (C layers) with invariance properties as shown in **Figure 3.6**.

Simple cells layers perform linear template matching using a dot product operator between their preferred stimuli and a small image patch mimicking their receptive field. They thereby are sensible to stimuli such as bar orientations. Invariance to position as illustrated in **Figure 3.6** and scale are then realized by C1 and C2. To achieve this fundamental property, complex cells layers uses an idealized nonlinear pooling mechanism with the maximum operation rather than a linear sum operation as is done in layers S. Summing increases response with increasing number of inputs and the limitation of this operator becomes clear in case of clutter, presence of multiple objects in a cell's receptive field or if size invariant responses are desired. Max pooling on the other hand is more robust against these problems since only the most active afferent determines the post-synaptic response. The MAX-like mechanism is moreover biologically plausible since, when multiple stimuli are present in an IT neuron's receptive field, its response is dominated by the stimulus that also produces highest firing rate in isolation. In conclusion, the combination of linear operations performed by the S layers and the max pooling operation performed by the complex layers provides selectivity while preserving feature specificity. The top layer of the hierarchy are composed of view tuned cells which receive pre-processed features from C2 and are trained to map these to images labels of the dataset, acting as the read out layer.

In this thesis, final feature layer C2b was tested and output was classified using a support vector machine (SVM) described in sub-section 3.6.

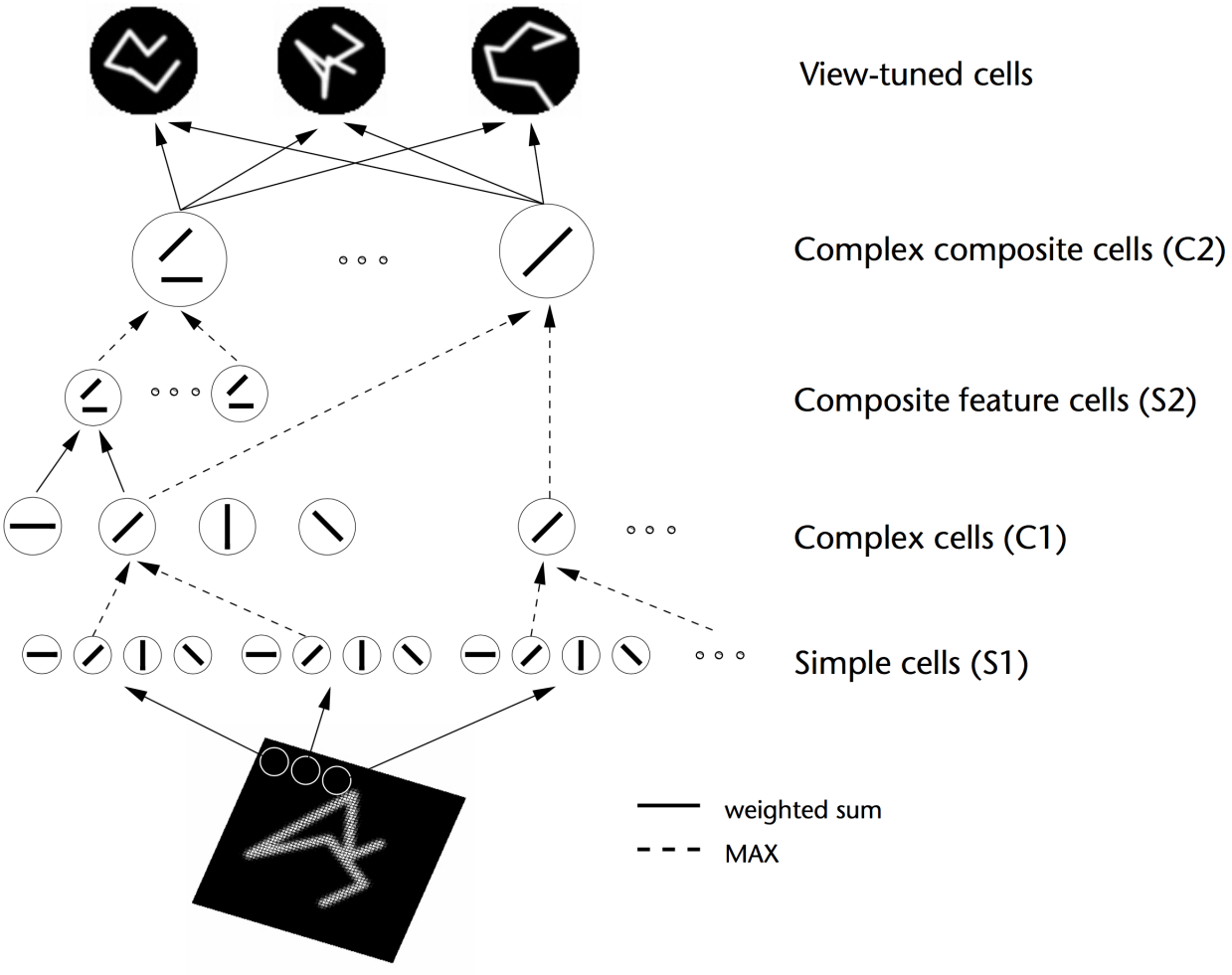


Figure 3.6: HMAX architecture and composing units[39]: simple cells (named S) perform linear template matching (solid lines) while complex cells (named C) implement non linear MAX pooling operation (dashed lines) by combining multiple S templates give the model invariance to translation. By combining multiple small receptive fields from the bottom layers, pattern selectivity can be achieved in the top layers. Also note that connections can be skipped in the hierarchy since C1 has some projections which map directly to C2, bypassing composite feature layer S2. Feature extractor cells here described as view tuned are then trained on labeled images of the dataset.

3.3.2 Alexnet

Thanks to the evolution of GPUs and optimization of 2D convolution implementation, training of convolutional neural networks (CNNs) on extensively large image datasets became possible and AlexNet[68] was the first deep convolutional neural network to win the ImageNet competition. Its architecture is composed of 8 sequential layers, the first 5 being convolutional and the last three being fully connected as depicted in **Figure 3.7**.

The network was trained on the ImageNet dataset[41] comprising 15 million high resolution images belonging to about 22,000 cateogories labelled using Mechanical Turk crowd sourcing tool. The only pre-processing steps applied to the input images are rescaling and demeaning. In the first

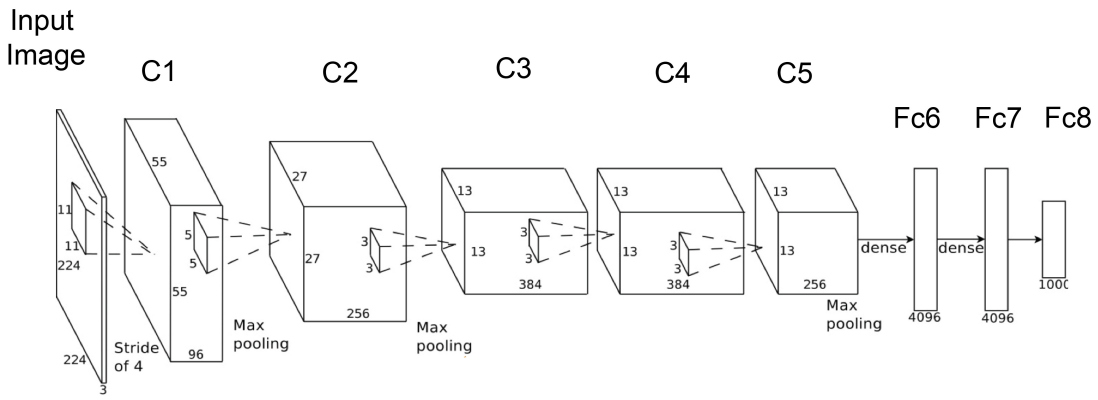


Figure 3.7: Alexnet architecture: the five first layers (C1-C5) are convolutional and process the input image with filters of different sizes and steps. The output of C5 is then fed into three fully connected layers (Fc6-Fc8) where Fc8 is performing the final read-out (by classifying images into one of 1,000 possible categories. Max pooling steps occur after layers C1, C2 and C5. Spatial size is progressively reduced throughout the convolutional layers as the number of filters used is increased. Computations were shared between 2 GPUs, one using the top of the image as an input (as depicted in this figure) and one using the bottom (not shown). GPUs communicate only in the fully connected layers. Image was adapted from `http://www.cc.gatech.edu/~hays/compvision/proj6/`

step, the RGB images are scaled down to 244x244x3 and are filtered by the first convolutional layer (C1) using 96 kernels of sizes 11x11x3 and step size of 4 pixels. The output of this operation is then response-normalized and max-pooled to reduce dimensionality and filtered by the second convolutional layer (C2) with 256 kernels of sizes 5x5x48, also applying response-normalization and max pooling at the end of this step. Convolutional layer 3 (C3) uses 348 kernels of sizes 3x3x256, layer 4 (C4) subsequently uses 328 kernels of sizes 3x3x192 and finally layer five (C5) uses another 256 kernels of size 3x3x192. The output of C5 is again max pooled. The first two fully connected layers (FC6 and FC7) then reduce the dimensionality sequentially to a 4096x1 vector while the last layer (FC8) serves as a read out layer and outputs the final 1000x1 vector containing the category labels of the image. Visualization of the output weights of different layers is presented in **Figure 3.8**.

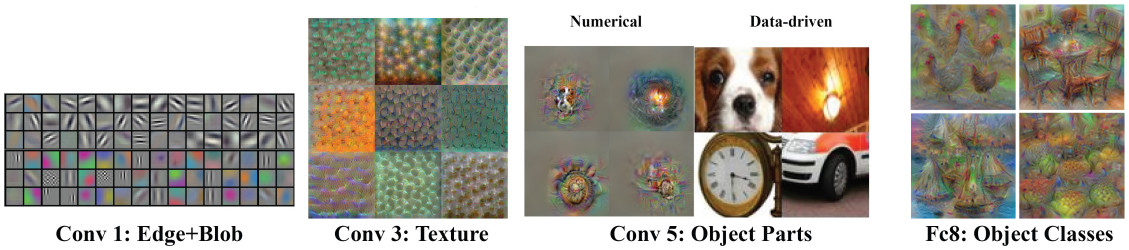


Figure 3.8: Visualization of computed weights of convolutional layers 1, 3 and 5 and final read out layer Fc8. The first convolutional layer performs basic edge detection while the third convolutional layer is sensible to texture and the fifth one can detect complex shapes in small receptive fields. The Fc8 layer provides the final classification step, given a label to each image. Image was taken from `http://www.cc.gatech.edu/~hays/compvision/proj6/`

Because AlexNet has 60 million parameters and 650,000 neurons, reducing overfitting was a primordial concern and a combination of techniques have been used for this purpose. Data augmentation through translations, horizontal reflections and alternation of RGB channels intensities not only proved to be good ways to enlarge the dataset but additionnally provided human-like properties to the network such as invariance to illumination intensity or color. Another technique that was used is called dropout[69] and consists in changing the output of hidden neurons to zero with a 50% probability thereby reducing interdependence between neurons as they can no longer rely on the output of specific other neurons. Finally, faster training can also reduce overfitting and therefore AlexNet's usage of the rectified Linear Unites (ReLU) non linearities were also useful for this purpose. They were used at the outputs for the max pooling steps after C1, C2 and C5.

The version that was used in this thesis was trained using Caffe[70]. Alexnet layers 5 and 7, respectively the last convolutional layer and last fully connected layer before read out, were tested on the same image dataset as used for the psychophysics experiment, consisting of 23,000 images of partial objects generated from 500 whole novel images distributed in 5 categories.

3.4 Feature extraction using recurrent neural network models

3.4.1 General Architecture and aim

Recurrent connections were added on top of either the last convolutional layer (pool 5) or the last fully connected layer (fc7) in all-to-all fashion to explore the effect on performance of combining different types of connections rather than just adding complexity to a feed-forward network. Only one layer was augmented at a time and all other layers remained kept their set of fixed feed-forward weights obtained from pre-training on ImageNet. This setup is of course far from reflecting the true diversity and density of connections in the brain but provides more interpretable results due to its simplicity.

In an RNN, the state vector h is updated at every timestep t through a non-linear weighted combination of the input x_t at time t and the previous state at time $t-1$. Mathematically this update equation is, in our case, defined as:

$$h_t = f(W_h h_{t-1}, x_t)$$

where

- h_t is the state vector at the current time t in our case representing either the updated [$9216x1$] pool 5 or the [$4096x1$] fc7 feature vector at time t.

- W_h is the learned weight matrix of the RNN.

- x_t is the feature vector of the previous layer (either pool 4 of fc6) multiplied by the transition weight matrix, respectively $W_{4\to5}$ and $W_{6\to7}$. In our case its value will remain fixed.

- f is a function introducing a non linearity, in our case a ReLU

Using the approach described above, different RNNs were obtained by the manner of updating the weight matrix W_h and the chosen activation function f.

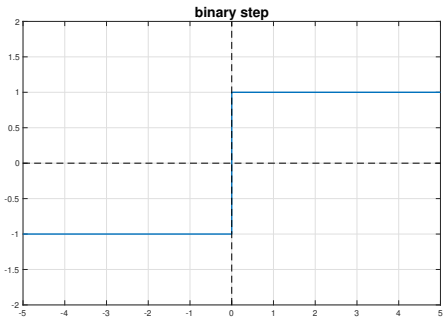
3.4.2 The Hopfield recurrent neural network

A Hopfield network[71] (RNNh) stores information, in our case whole object features extracted from AlexNet, in the attractor states corresponding to energy minima and can then retrieve this information when presented with incomplete data such as occluded object features by converging to the appropriate attractor. This learning method was recently shown to be equivalent to the way previously mentioned feed forward networks learn if the derivative of this energy function is used as the activation function[72]. It however relies on convergence to learned attractors and several precautions need to be taken in order to assure stability and decrease the probability of spurious attractors.

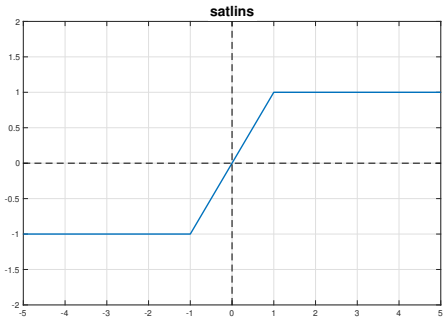
Units of the Hopfield network take only binary values of either -1 or 1 and are initialized to a start pattern. Their states are traditionally updated synchronously or asynchronously using a binary step function defined by:

$$s_i = \begin{cases} +1, & \text{if } \sum_j W_{ij} s_j \geq \theta_i \\ -1, & \text{otherwise} \end{cases}$$

where s_i is the state of the i-th unit, θ is the threshold for the state flip of the unit and W_{ij} is the weight given to the connection between unit i and j. An alternative activation function called the saturated linear transfer function (satlins) is most often used for modern implementations of RNNh because it outputs continuous values in the $[-1, +1]$ range. It is mathematically defined by $satlins(y) = max(min(1, y), -1)$ and presented in **Figure ??**. Using a continuous transition between the binary values enable an easier analysis of the network. The original Hopfield network is in then described by first order linear differential equations and weights can be computed using singular value decomposition [73].



(a) Binary step activation function between -1 and +1 which was originally used in Hopfield networks



(b) Satlins activation function which is used by matlab for the RNNh implementation

Figure 3.9: Visualization of the activation functions used in the Hopfield network

The weight matrix W is constrained to be symmetric ($W_{ij} = W_{ji}, \forall i, j$) and to have a zero diagonal ($W_{ii} = 0, \forall i$). The symmetry property implies that there is a single bi-directional connection between any two units reducing oscillations while forbidding self-connections is important for the system's stability by avoiding units to receive permanent feedback from their own state. The associated energy of each unit is then computed using the following formula:

$$E = -\frac{1}{2}\sum_{ij} W_{ij} s_i s_j + \sum_i \theta_i s_i$$

The state's energy is high when it far from the attractors but decreases as it converges closer to them. During training, each weight W_{ij} is updated according to the pairwise similarity between the units of n features using a rule which stems from Hebbian theory[74]:

$$W_{ij} = \frac{1}{n}\sum_{p=1}^{n_p} x_i^p x_j^p$$

with $n_p = 500$, the patterns of whole objects to be stored and x_i^p the activity of unit i in response to feature p. This learning rule has two biologically plausible properties. First, units, which can also be called neurons, respond locally since they take into account information only from the units they are connected with, being "blind" to the whole network state. Second, the update does not require reference to previous patterns to learn a new training pattern. This incremental behavior is different from most deep neural networks which are exposed to the whole dataset at once.

RNNh was implemented using a matlab built-in function called `newhop`. Values of input vector x were first binarized to -1 if negative or +1 else. The weights W_s are then learned for each state by using solely the features of the AlexNet extracted for the whole objects. It is important to emphasize that the model does not have any parameters trained with partial objects so it never explicitly learned about occlusion. The features s are then updated by RNNh using the following iteration until convergence was reached:

$$s_t = satlins(W_s h_{t-1}) + b \text{for} t > 0$$

with b a constant bias and $s_0 = x$ for $t = 0$ with $x = W_{4\to5} \times pool_5$ or $W_{6\to7} fc_6$ so that ho belongs to $\{-1, +1\}^{9216}$ or $\{1, +1\}^{4026}$.

3.4.3 RNN5

Unlike humans, RNNh was never familiarized with the concept of occluded objects. Therefore, a different type of RNN was implemented by directly training the weights W_h to minimize the feature distance between partial objects and their whole counterparts over time. The considered network, named RNN5[1], was exposed to examplars of all five categories during training. Since the number of weights to be learned is very high compared to the number of training images, additional precautions are needed to prevent overfitting. Therefore, a subset of the objects, which was never shown to the network during training, was saved for performance testing. The RNN5 model was implemented once on top of features of the pool 5 layer and once on top of the fc7 layer of Alexnet which determined the constant input x. The states were then updated with the following equation:

$$s_t = ReLU(W_s \times h_{t-1} + x) \text{for} t > 0$$

where $s_0 = x$ for $t = 0$ with $x = W_{4\to5} \times pool_5$ or $W_{6\to7} fc_6$ The objective function to be minimized can be interpreted as the mean squared euclidean distance between the features from the partial objects and the features from the whole objects Five fold cross validation was used during training in combination with the gradient descent optimization algorithm called RMSprop[]. To further reduce overfitting, early stopping at 10 epochs was additionally implemented and the weights for the epoch with lowest validation error were selected. RNN5 was trained with the RMSprop optimizer and with 5 fold cross validation so that each object occurs in testing set exactly once. Early stopping is also done to counteract RNN overfitting its high number of weights on the small number of training examples. Performance on validation set was computed after each epoch and use weights where validation error is minimal.

3.5 Parallel Pooling and Orchestra

A parallel pool is a set of Matlab workers on a computer cluster. When using the matlab notation `parfor` instead of a classical for loop, a parallel pool is started automatically. This enables to distribute computations along different computer cores, running sections of the parfor loop in parallel and thereby speeding up the execution time.

Feature extractions were conducted on the Orchestra High Performance Compute Cluster of Harvard Medical School[2]. This NIH-supported shared UNIX-based facility consists of thousands of processing cores and terabytes of associated storage. Using Orchestra did significantly speed up computation by distributing the computations among several cores using the integrated parallel pooling code.

3.6 Image Classification

A Support vector machine (SVM) with a linear kernel is a supervised learning algorithm which constructs a set of maximum margin hyperplanes to separate the different categories present in the dataset as presented in **Figure 3.10** in the case of two categories. The hyperplane should be positioned where its distance with the nearest point of either category is maximized.

In this thesis, the previously described linear SVM was used for classification after feature extraction using the above mentioned different networks to evaluate tolerance to occlusion. Like in other studies exploring invariance to object transformations such as size and position changes, the SVM was trained on one condition and tested on the others. The decision boundary was thus trained on the 500 whole images and evaluated on the 23 000 uniquely occluded renderings of these images. Cross validation was done 5 fold across the 500 objects. In each fold, 400 objects were used for training, 100 for validation or testing assuring each object was used only once in the validation and testing splits and each split contained a roughly equal number of objects from each category.

[2]chestra High Performance Compute Cluster at Harvard Medical School, partially provided through grant NCRR 1S10RR028832-01, `http://rc.hms.harvard.edu`

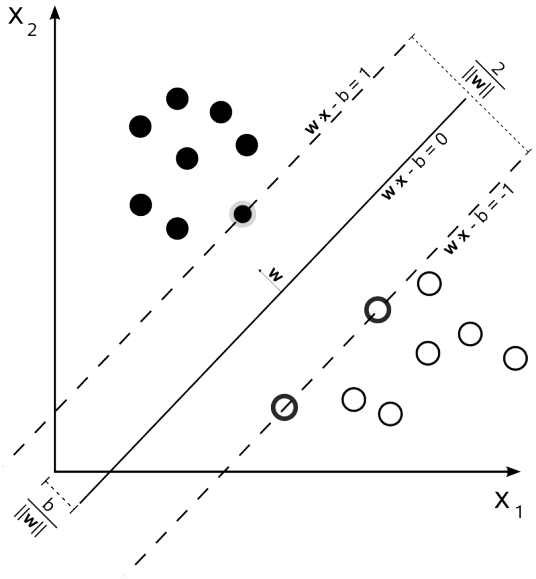
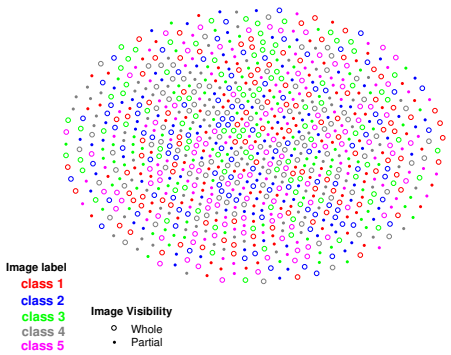


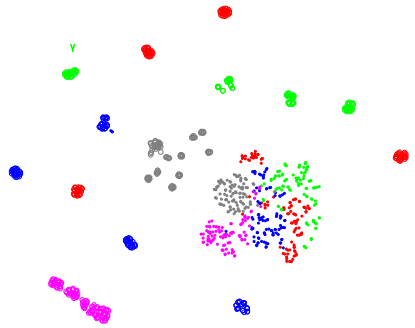
Figure 3.10: Example of SVM classification trained on samples from two categories. The dashed lines represent the two hyperplanes that separate the two categories where their difference is maximal. The datapoints positioned closest to the two hyperplanes are called the support vectors. The maximal margin hyperplane used for classification lies in the middle of these two margins. Image was taken from `https://en.wikipedia.org/wiki/Support_vector_machine`.

3.7 T-distributed Stochastic Neighbor Embedding

While PCA is widely used to find lower dimensionality combinations variables in a dataset which explain a high percentage its the variance, it is not ideal to analyze a classifier's performance. Indeed, it is a linear method focusing on preserving the distance between widely separated datapoints rather than nearby ones. On the other hand, T-distributed stochastic neighbor embedding[76] (t-SNE) is a non-linear dimensionality reduction technique used to visualize high-dimensional datasets in two or three dimensions and is widely used in machine learning. It starts by computing pairwise Euclidean distances between the datapoints of the high dimensional space and converts them into a matrix of conditional probabilities representing pairwise similarities. To do so, it defines a personalized standard deviation for each high dimensional point while conserving a pre-determined constant perplexity. An initial set of low-dimensional points is then created and iteratively updated by minimizing the Kullback-Leibler divergence between a Gaussian distribution in the high dimensional space and a heavy tailed t-distribution in the low dimensional space. The perplexity is a hyperparameter which allows to balance importance given to local structures with respect to global ones such as the presence of clusters at different scales. It can be compared to the expected number of nearest neighbors for each datapoint. The authors of this method have argued that values drawn from the $[5, 50]$ range conserve robust t-SNE performance. However if it is set too high, t-SNE will try to display all points as being equidistant as presented in **Figure 3.11**. This is only one example of a possible effect due to sub-optimally chosen parameters and care should always be taken not to over-interpret two dimensional displays of higher dimensional data. Since t-SNE performs different transformations on different local regions, it is very sensible to the curse of the intrinsic dimensionality of the data [77]. Moreover, the cost function of t-SNE is not convex meaning that different results can be obtained for different runs on the same data.



(a) Case for which the t-SNE algorithm fails. The perplexity hyperparameter was set most probably chosen too high. All datapoints are positioned so as to minimize their distance with respect to each other.



(b) Case where t-SNE gives more interpretable results, in this case the categorization of novel objects after feature extraction by the last convolutional layer of Alexnet

Figure 3.11: Presentation of two different outputs of the t-SNE dimensionality reduction algorithm.

Chapter 4

Results

As described in the **Materials and Methods** chapter 3.2, twenty-three subjects were recruited to gather human performance data for categorization of partial novel images described in *section 3.1*. After an initial training phase, the visual recognition task involved categorization of heavily occluded rendering of 500 different novel objects into 5 categories. In 85% of the trials a varying percentage of the image was shown through Gaussian uniformly distributed bubbles, while 15% of the images were presented in the whole condition to serve as a positive control. Two other conditions were also explored: the stimulus onset asynchrony (SOA) measuring the time of subjects exposure to the image was varied in the $\{25, 50, 85, 100, 150\}$ interval while a spatial noise mask was presented after the image in 50% of the trials (masked/unmasked conditions). First results in the unmasked condition will be described in section 4.1. These results will then be contrasted to the masked condition in section 4.2.

In parallel, human results were compared to performance of feed forward and recurrent computational neural networks (CNNs) when exposed to the same dataset comprising a total of 23,000 randomly occluded versions of 500 whole novel images. Performance of early purely feed-forward object recognition networks such as HMAX and Alexnet are presented in *section 4.3* while performance of such networks after augmentation with recurrent neural networks are presented in *section 4.4*.

4.1 Recognition of partially visible objects by humans is also robust with novel objects

In the absence of a mask, subjects recognition of whole objects is near ceiling (above 90%, see the points at 100% visibility) as presented in **Figure 4.1** meaning the performance dominantly reflects the behavioral effect of object occlusion. However, performance degrades when subjects are exposed to poorly visible stimuli compared to whole images ($(p< 10^{-30}$, paired left tailed t-test). Surprisingly, humans still categorize images above chance level ($58 \pm 6\%$ versus 20%) for image visibility as low as $5 \pm 2.5\%$ ($p< 10^{-50}$, paired right tailed t-test) despite very limited information provided. Regarding the SOA condition, **Figure 4.2** (full line) shows that there was a small but significant improvement in performance for the longest (150ms) SOA for the partially visible objects ($p=0.0083<0.05$, Spearman's correlation coefficient: $s=0.9$, permutation test). In conclusion, human still reached surprisingly high performance for very low image visibilities even though their previous exposure to novel whole objects was minimal and they had no previous experience with the corresponding occluded renderings.

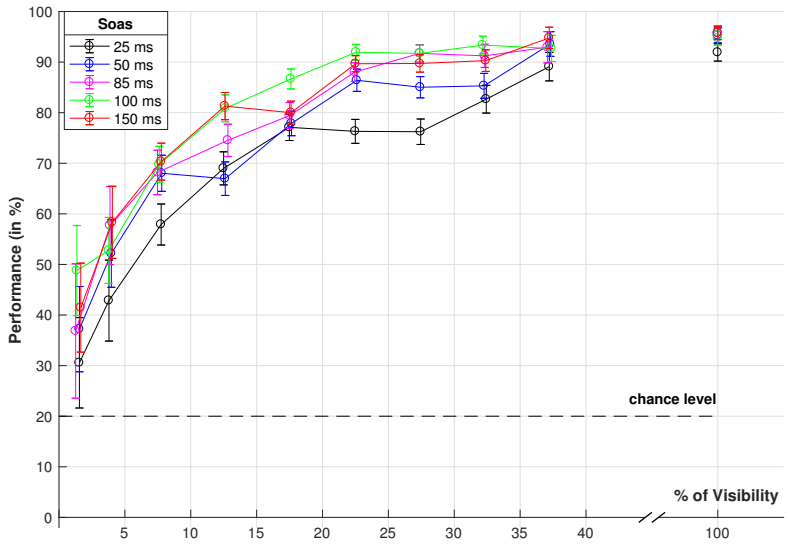


Figure 4.1: Human behavioral performance as a function of image visibility for the unmasked condition meaning that no spatial noise mask was presented after the image. Each curve denotes a specific stimulus onset asynchrony (SOA) represented in a different color. Error bars were computed using standard error of the mean (s.e.m.). This psychophysics experiment involved 23 subjects. Horizontal dashed line represents chance level performance (20%). Bin size was chosen to be 5%. A discontinuity was introduced in the x axis in order to show whole object performance as a positive control.

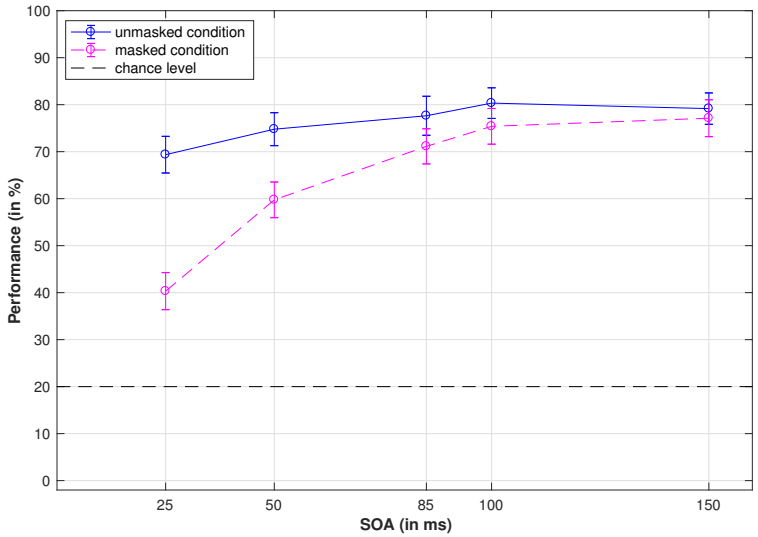


Figure 4.2: Behavioral image recognition performance as a function of stimulus onset asynchrony (SOA) averaged across image visibility for all the collected data except the whole images. Error bars were computed using standard error of the mean (sem). The solid blue line represents the unmasked condition while the dashed magenta line represents the masked condition. Chance level is shown by the black dashed line (20%)

4.2 Backward masking disrupts recognition as a function of object visibility and soa

Behavioral effects of backward masking can be understood by contrasting results presented in **Figure 4.3** to previously described **Figure 4.1**. It can be noted that only performance of stimuli presented at the shortest SOA (25ms) is affected in the whole condition ($p=0.0020<0.05$, paired two tailed t-test, Bonferoni corrected). Regarding the occluded objects, backward masking significantly impairs object categorization performance at low visibility levels since values decreased compared to the unmasked condition. This effect is especially obvious for shorter SOA times corresponding to the black, blue and magenta curves which are drawn farther apart from each other while without masking their differential impact is less clear. These observations are confirmed by a two-way ANOVA table analyzing the effect of SOA and masking on human performance. The interaction of these two factors is found to be statistically significant ($F(mask,SOA)=18.83$, $p<10^{-10}$). **Figure 4.2** also enables clearer visualization of the increasing strength of backward masking as SOAs are shortened. Significance of this effect was achieved for all SOAs smaller or equal to 100ms (the p-values of SOAs $\leqslant$ 100ms were inferior to 0.0001 using a paired one-tailed t test, Bonferoni corrected).

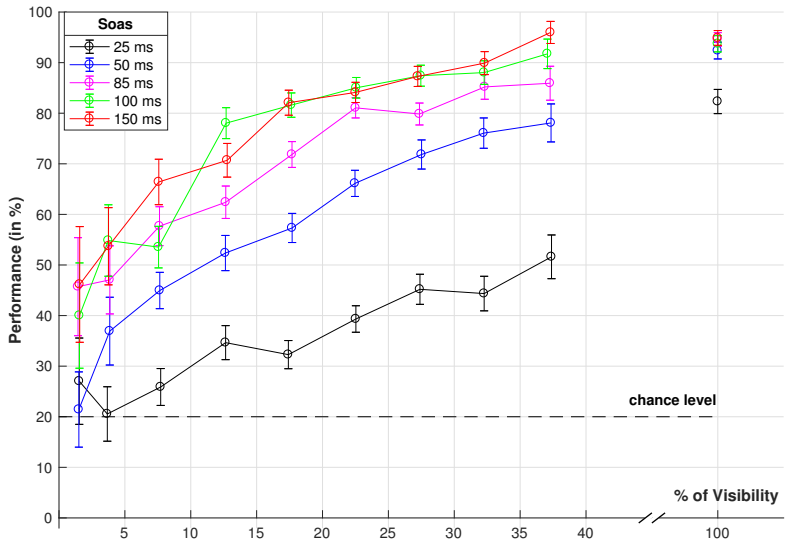


Figure 4.3: Human behavioral performance as a function of image visibility for the masked condition meaning that image specific spatial noise masks was presented after the image. Each curve denotes a specific stimulus onset asynchrony (SOA) represented in a different color. Error bars were computed using standard error of the mean (s.e.m.). This psychophysics experiment involved 23 subjects. Horizontal dashed line represents chance level performance (20%) Bin size was chosen to be 5%. A discontinuity was introduced in the x axis in order to show whole object performance as a positive control.

4.3 Recognition of partially visible objects by feed forward models is not robust

State-of-the-art neural networks such as HMAX or Alexnet reach very high performance for object recognition in computer vision competitions. They are characterized by a hierarchical purely feed-forward architecture with increasing receptive field sizes and selectivity. Interestingly, when using everyday isolated objects, they reached invariance to certain object transformations such as scale or orientation[39, 5, 65] but not to occlusion[1]. By using these networks on the same dataset as the one used in the psychophysics experiment, we wanted to investigate how well they generalize over novel objects and confirm that their architecture is not sufficient to explain the behavioral observations described in *section 4.1*. Presence or absence of similarity between human and model performance could indeed give a hint about the underlying computational mechanisms of pattern completion. Features were extracted using HMAX's final feature layer C2b and Alexnet's last convolutional layer pool 5 and last fully connected layer before classification Fc7. The corresponding number of dimensions were 1000, 9216 and 4096 respectively. A SVM was then used to classify the different features by training the decision boundary on the whole images and testing on the occluded ones. Pixels were also directly classified without further processing to account for the effect of potential low-level differences between categories such as contrast or object area.

The performance of the classifier when using raw image pixels did not perform significantly above chance level as shown in Figure 4.4 ($p>0.1$, right-tailed t-test). Performance of HMAX was found to be below chance level for occlusion percentages higher than 80% ($p=1$, right-tailed t-test) but slightly above lower occlusions.

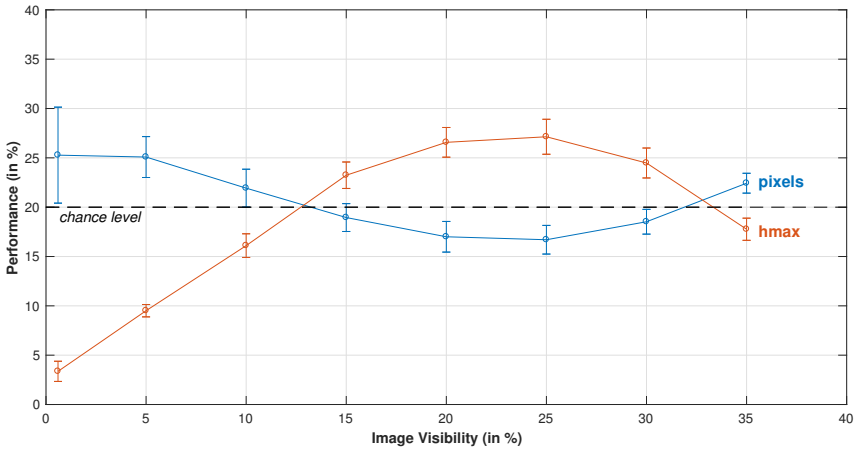


Figure 4.4: Performance of the HMAX computational model (red) compared to raw image pixels classification (blue). Chance level at 20% is represented by a dashed line. Extracted features by the HMAX model and pixel value were classified using a SVM trained on whole images and tested on occluded renderings. Occluded counterparts of objects used to train the classifier were not used in the test set. Error bars are the s.e.m across the 5 fold of cross validation.

In contrast, the AlexNet pool 5 and Fc7 feed-forward models presented in Figure 4.5 performed well above chance level. They reached 50% performance for image visibilities greater than 16% and 8% respectively. However, both models have a significantly lower performance than humans at visibility levels below 40% ($p<10^{-4}$, Chi-squared test). While humans almost reach 60% performance for image visibilities of only $5 \pm 2.5\%$, AlexNet Fc7 performs slightly above 30% and AlexNet pool 5 reaches about 38%. Interestingly, it can also be observed that AlexNet fc7 has significantly lower performance for all shown visibility levels than the earlier pool 5 layer ($p<2^{-13}$, Chi-squared test) which contrasts results obtained previously[1] using everyday objects resembling the ones composing ImageNet.

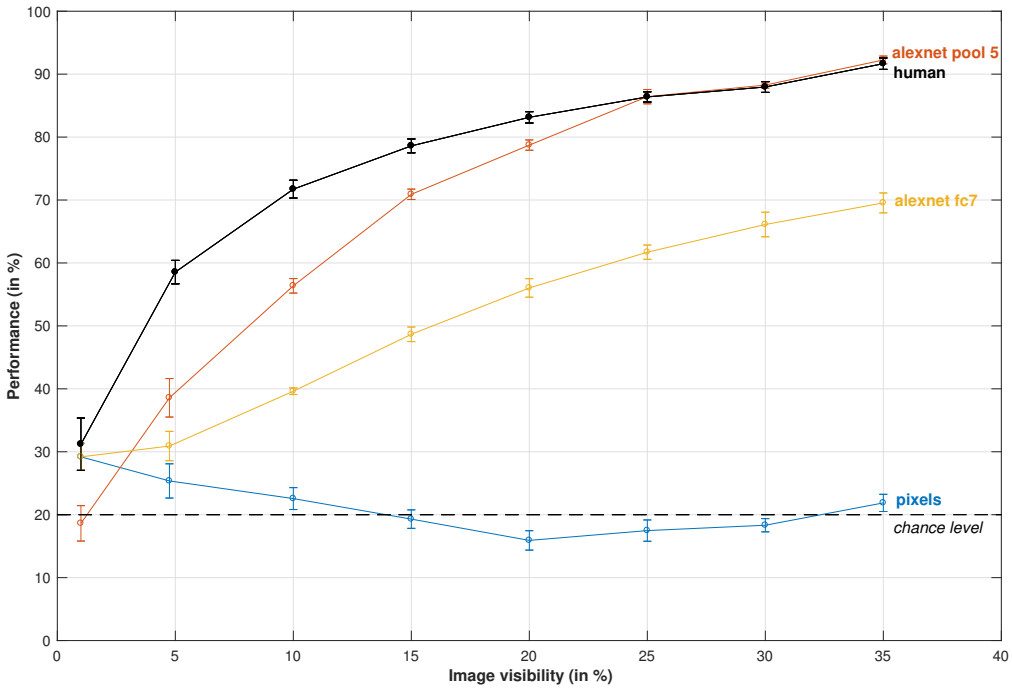


Figure 4.5: Performance of pool 5 (red) and fc7 (blue) layers of the feed-forward computational model named AlexNet. They can be contrasted to chance level (dashed line), raw pixel classification (blue) and human performance on the same dataset (black). Extracted features by Alexnet were classified using a SVM trained on whole objects and tested on partial renderings of the objects absent from the training set. Error bars are the s.e.m (5-fold cross validation).

To graphically display why object invariance was not reached by the AlexNet based feed-forward models, stochastic neighborhood embedding (t-SNE)[76] was used to project the pool 5 and Fc7 features onto two dimensions as shown in **Figure 4.6**. As expected, whole objects (denoted by open circles) group together well and are visually easy to separate. Even families within the α, β and γ categories can be distinguished, confirming the high performance of AlexNet for whole images recognition. However, the partial images represented by full circles are not close to their whole counterparts and do not form clear clusters which explains the observed decreased performance. We can thereby confirm that feed forward models such as AlexNet or HMAX are not robust to object occlusion and suggest that altering their hierarchical structure might help for this task.

(a) AlexNet pool 5 model

(b) AlexNet Fc7 model

Figure 4.6: Two dimensional visualization of different alexnet model features using the t-SNE dimensionality reduction technique. Whole objects (open circles) can be easily separated and one can even distinguish the different families within a category for α,β and γ. Partial objects however do not form clearly distinguishable clusters and are not close to their whole counterparts, explaining why performance is low for small image visibilities.

We also verified that sub-human performance of the AlexNet bottom-up models for partial image recognition tasks depends strongly on the amount of removed pixels. Indeed AlexNet models performance reached ceiling levels for image visibilities greater than 70% for AlexNet Fc7 and greater than 50% for AlexNet pool5 as shown in **Figure ??**. We also displayed the representation of features computed by AlexNet Fc7 using t-SNE in **Figure 4.8**. This time, occluded images were clustered with their whole counterparts, in agreement with the high obtained performance. This confirms that when large amounts of pixels are missing from an object, the model's representation of partial rendering is pushed farther and farther away from their whole counterparts. The distance between whole and partial cluster means can be interpreted as a measure of the impact of object occlusion on the image representations.

In conclusion, AlexNet was found to achieve successful feature extraction for whole novel objects that were very different from examplars of the ImageNet dataset showing its generalization power. It was also found to be invariant to light occlusion levels. However, its last fully connected layer fc7 generalized less well than its last convolutional layer pool 5 when the task was made more challenging by increasing partial image deletion. We also confirmed that these models are not robust to object occlusion for low visibility renderings below 40% and did not match human performance in pattern completion even though humans had only minimal training with the whole objects.

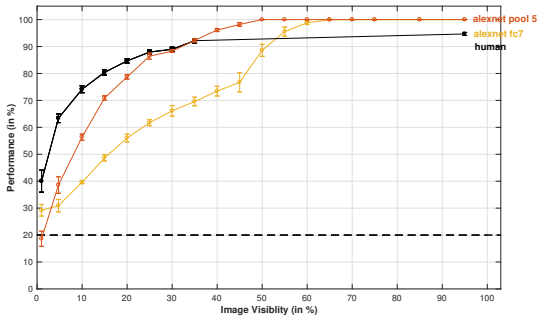


Figure 4.7: Performance of pool 5 (red) and fc7 (yellow) layers of the feed-forward computational model named AlexNet for higher visibility percentages, demonstrating the high performance of these models for fully or highly visible object classification. Results can be contrasted to chance level (dashed line) and human performance on the same dataset (black). Error bars are the s.e.m across(5-fold cross validation).

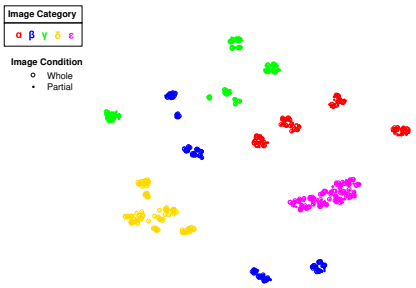


Figure 4.8: Two dimensional visualization of AlexNet Fc7 model features in the case of partial objects with high visibility (spanning the 50% to 95% range) using the t-SNE for dimensionality reduction. Notice how occluded versions converge towards the appropriate whole counterparts.

4.4 Recurrent neural networks improve performance for partial object recognition

Since backward masking was found to impair human performance for short SOAs, we hypothesize that adding recurrent connections to bottom-up models would probably achieve a more robust representation of partial object and thereby increase categorization performance. Two different recurrent neural networks (RNNs) were used for this purpose. The first one called RNNh was proposed by Hopfield[71] and relies on attractor states defined by all-to-all connections weights determined solely by whole objects. Images which are more heavily occluded are initially farther away from the attractors and would require more recurrent time steps to converge. However, this model has no training with occluded images. Therefore, another RNN, called RNN5, was implemented so as to directly minimize the distance between whole and occluded objects. It was trained on images from all five categories. The schematics of newly obtained models to AlexNet pool 5 and fc7 layers are presented in **Figure 4.9**.

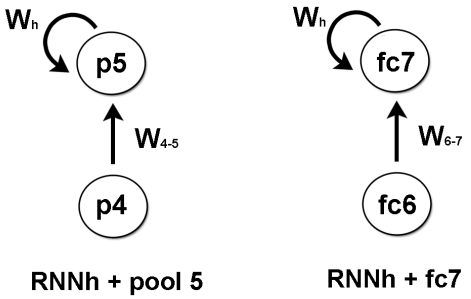


Figure 4.9: Schematic of recurrent neural network (RNN) addition on top of AlexNet pool 5 (shown left) and AlexNet Fc7 (shown right) layers. Last fully connected layer before readout Fc7 receives input from Fc6 multiplied by the characteristic weight matrix between these two layers $W_{6\to7}$. This input will be kept constant throughout the recurrent computations. The weight matrix W_h from the added RNN governs the temporal evolution of fc7 features. The definition and update of this weight matrix determines of the attractor-based Hopfield (RNNh) or RNN5 network is implemented. RNN5 is trained to minimize distance between whole and partial images.

In a first investigation, the purely feed-forward model AlexNet Fc7 is augmented with a Hopfield network just before applying the activation function and run for 256 time steps (referred to as the RNNh+fc7 model). It significantly improves initial Fc7 performance for all visibility levels below 40% as shown by comparing the red and purple curves in **Figure 4.10** ($p<3\times10^{-9}$, Chi-squared test) and even approaches human results although still being just significantly lower ($p<0.04$, chi-squared test). Similarly, a RNNh network was added after the AlexNet pool 5 feature extraction but was run only for 16 time steps because of lack of time to get all desired results (I will run it until 256 time steps later). As illustrated by the blue and yellow curves, adding recurrent connections to the pool 5 architecture also increases performance compared to the purely feed-forward base model ($p<10^{-9}$, Chi-squared test). Moreover, the RNNh augmented model of pool 5 once more outperforms the augmented fc7 one as shown by the purple and yellow curves ($p<0.002$, Chi-squared test) which is consistent with previous observations from the feed-forward model analysis. Interestingly, RNNh + pool 5 even has higher performance than human results across all presented image visibility levels ($p<0.02$, Chi-squared test).

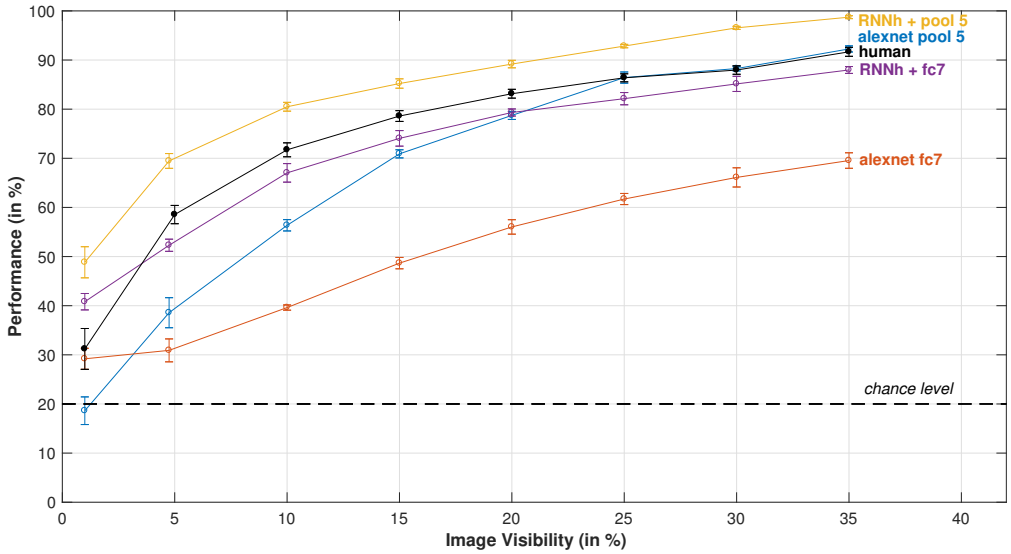
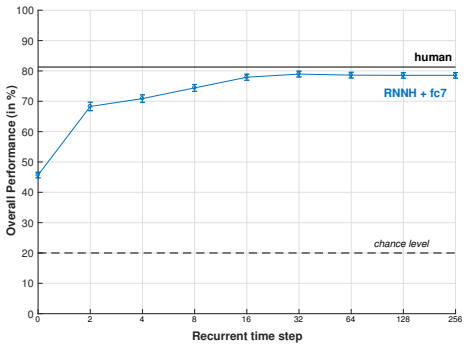
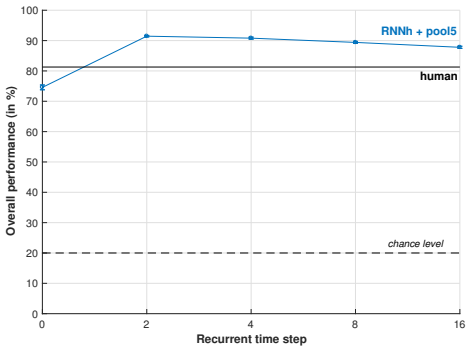


Figure 4.10: Performance of the AlexNet Pool 5 and Fc7 layers after addition of a recurrent Hopfield model as a function of image visibility. For both new models, augmenting the purely feed-forward counterpart lead to a significant increase in performance compare to step t=0. The RNNh + fc7 combination now almost reached human-like performance while RNNh+pool5 even outperforms human results for all visibility levels. Dashed line indicates chance level. Error bars denote s.e.m.

The performance of RNN models is dynamic, meaning that it initially evolves with the number of time steps during which the features are updated until it reaches convergence. Time 0 denotes the original purely feed-forward AlexNet model performance, before any recurrent computations are started. The RNNh + pool 5 model's overall performance was found to increase significantly after the first time step but did not change significantly after that, suggesting that computing more than 16 time steps might not add much to the model anyway. On the other hand, the RNNh + fc7 model's overall performance is found to initially increase with the number of time steps as shown in **Figure 4.11 b)**. Convergence happens at 32 time steps where performance reaches a plateau, meaning that computing 256 time steps was probably an overkill.



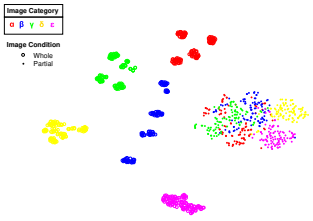
(a) RNNh + pool 5 model



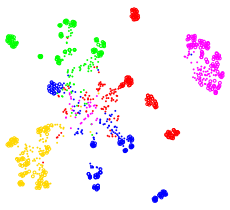
(b) RNNH + fc7 model

Figure 4.11: Performance evolution of Alexnet pool 5 (left) and fc7 (right) models augmented with RNNh (blue line) across number of recurrent time steps used. Performance is compared to chance level (dashed line) and human results (continuous black line). RNNh was only run up to 16 time steps due to lack of time before thesis submission.

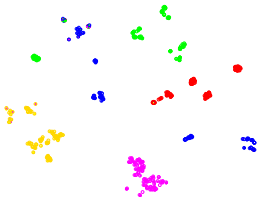
In order to get a better intuition about why performance increases compared with the corresponding feed-forward models and with the number of time steps, the dynamic trajectory of the feature representations of both augmented models is visualized using t-SNE in as shown in **Figure 4.13** and **Figure 4.12**. Before adding any recurrent connections, at time step t=0, the representations of partial object cluster together far away from their whole counterparts as previously observed. But as the number of time steps of the recurrent computation increases, the partial objects cluster is pulled apart and corresponding feature representations migrate towards the whole counterparts clusters of each category, explaining the increased performance of the RNNs.



(a) RNNh + AlexNet fc7 at timestep t=0

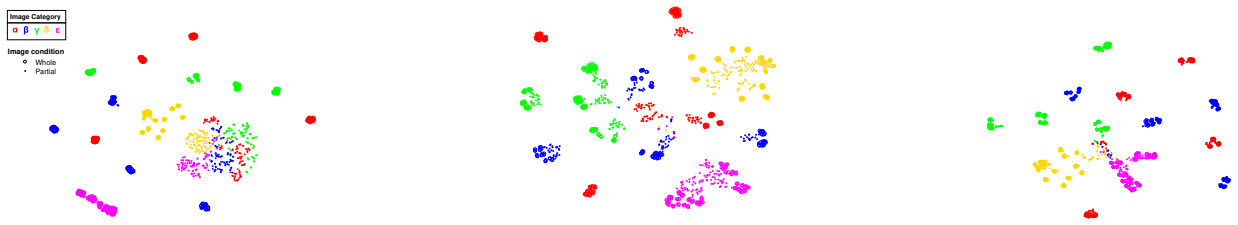


(b) RNNh + AlexNet fc7 at timestep t=16



(c) RNNh + AlexNet fc7 at timestep t=256

Figure 4.12: Temporal evolution of representations of the RNNh augmeted AlexNet Fc7 model features using t-SNE for dimensionality reduction. Whole objects (open circles) and their partial counterparts (filled circles) are colored according to their category label. Only one partial object for each object is shown so as not too create representations being too dense. As the number of recurrent time steps increases, partial renderings gradually converge to their whole counterparts, explaining the observed increase in performance from figure 4.11 b).



(a) RNNh + pool 5 at timestep t=0

(b) RNNh + pool 5 at timestep t=4

(c) RNNh + pool 5 at timestep t=16

Figure 4.13: Temporal evolution of representations of the RNNh augmeted AlexNet pool 5 model features using t-SNE for dimensionality reduction. Whole objects (open circles) and their partial counterparts (filled circles) are colored according to their category label. Only one partial object for each object is shown so as not too create representations being too dense. As the number of recurrent time steps increases, partial renderings gradually converge to their whole counterparts.

Interestingly, adding an attractor based model of recurrent connections to the top a feed forward hierarchical model was able to drastically improve classification performance while there was no exposure to partial objects during the definition of the attractor weights. Therefore, we next investigated if an additional increase of performance would be obtained by augmenting our best performing feed forward layer, AlexNet pool 5) with RNN5 which was explicitly trained to minimize distance between whole and occluded objects for all five object categories. Results are presented in **Figure 4.14**. Performance is increased with respect to the RNNh augmented model and human data for very low visibility levels. However, it drops below the other models for higher visibility levels.

In conclusion, adding attractor based recurrent networks such as the Hopfield network without making any other changes to the feed-forward models can significantly improve partial object recognition, even for very low visibilies. The combination of AlexNet pool 5 and RNNh in particular performs above humans for this task involving novel objects.

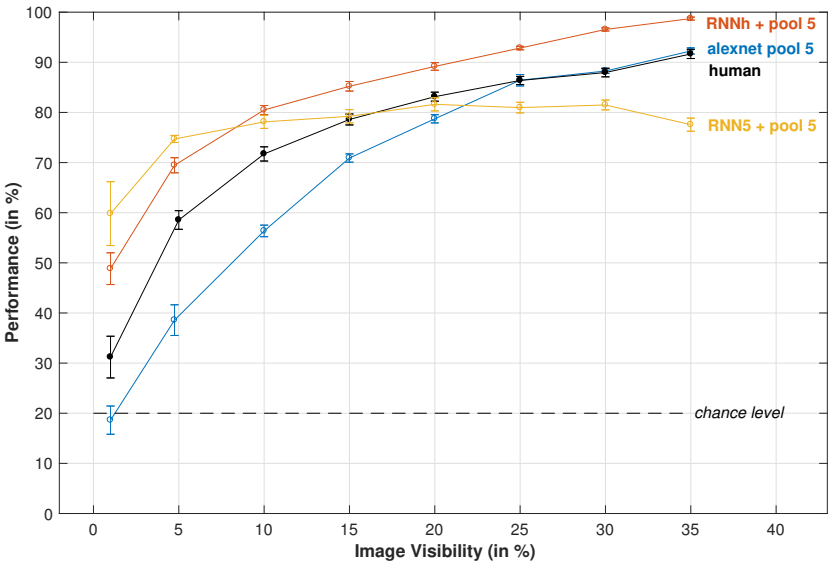


Figure 4.14: Performance of the AlexNet pool 5 after addition of a recurrent RNN5 model as a function of image visibility. Performance is increased for very low visibility levels with respect to the alexnet pool 5 augmented with RNNh but drops below other models for lower deletion percentages. Error bars denote s.e.m. (5 fold cross validation)

Chapter 5

Discussion

5.1 Conclusions from the psychophysics experiment

5.1.1 Impact of novel objects on human performance

Humans have to recognize partial objects on a daily basis due to presence of occluders or poor illumination for example. In a previous experiment[1] using familiar object categories (animals, faces, fruits, chairs and vehicles), the visual system was found to be capable of completing patterns and making inferences even when given only 10 to 20% of the object's pixel information. Similar results were reproduced in our behavioral experiment involving novel objects (**Figure 4.1**). In our case, performance is slightly higher as shown in **Figure 5.1** but we cannot draw direct conclusions from this comparison since very different stimuli were used. Indeed, the observed difference could be due to a number of factors: subjects implication, individual stimulus complexity, difference across categories. But the most probable explanation is the different nature of diagnostic features which we hypothesize to be focused more on general shapes than finer details in our experiment because of how our artificial novel objects were created.

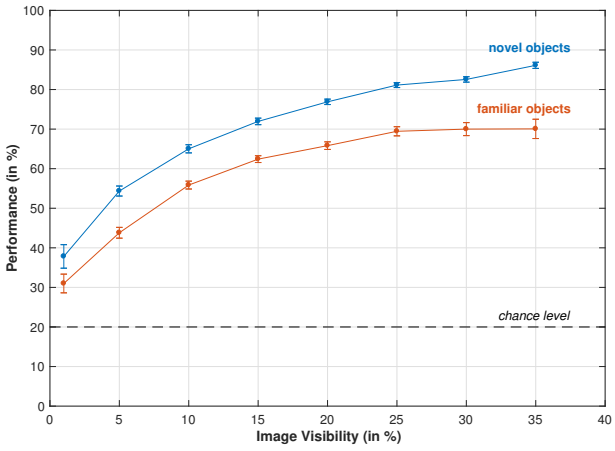


Figure 5.1: Human Performance as a function of image visibility for behavioral experiments using the same setup except for the stimuli. Our experiment using novel images (categories α,β,γ,δ or ϵ) is represented in blue while the experiment using familiar objects[1] (animals, faces, fruits, chairs or vehicles) is shown in red. Performance was averaged across SOA conditions and only data from the unmasked condition was used for the computations. Error bars denote s.e.m.

However, it is interesting that human performance is still higher than chance for low image visibilities even though subjects had only minimal categorization training with the whole objects and no prior exposure to the occluded renderings. We can conclude from these results that either humans learn to distinguish specific features of these new categories very quickly or it is not necessary to have previous experience with occluded images of a specific category at all to reach robust performance for occluded image categorization. In both cases, extensive exposure to occluded objects from multiple categories might still be needed to build a large prior library, necessary to reach the observed generalization of performance to the novel objects.

5.1.2 Minimizing prior exposure

A careful compromise had to be made in order to both minimize subject prior exposure to novel objects but also reach good performance for whole object categorization in order to truly assess generalization of an occluded recognition task rather than a simple whole object categorization learning task. During training phase, only ten different whole images were presented to the subjects (2 from each category, all coming from different families). Subjects were then asked to classify at least 8 of them correctly for a minimum of 5 consecutive repetitions. The average of exposures per image to pass this test was 8, which adds up to a total of 80 exposures. The standard deviation was half of this value, showing high variation across subjects as illustrated in **Figure 5.2**. However, each subject did eventually successfully generalize identified category specific features from those example to new unseen whole images. Indeed, 15% of images during the experiment were presented in the whole condition, serving as a positive control. Performance for these fully visible objects was above 90% in the unmasked condition as we previously exposed in **Figure 4.1**.

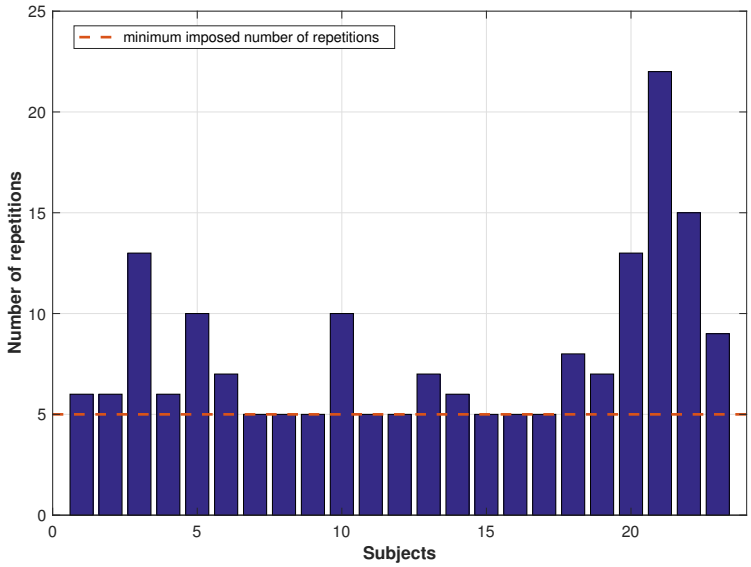


Figure 5.2: Subject specific number of required trials to pass the training test. The training test asked subjects to correctly classify 2 whole objects from each category (adding up to 10 different images, corresponding to 2% of the total number of whole images). The average number of image exposures represented only 8% of the total number of stimuli presented during the actual experiment.

5.1.3 Analysis of learning

Since humans prior knowledge of stimuli was entirely limited to a minimal exposure learning phase, we expect their performance to increase from early stimuli to later ones as a result of learning from increased stimulus exposure. Performance differences cannot be accounted for differential image visibilities because occlusion levels were randomized within the experiment as well as across subjects. To analyze this effect, the dataset was split into quarters for each subject and performance for different visibility levels was assessed separately for the first and three last remaining quarters as shown in **Figure 5.3**. Human performance was found to increase significantly over the last three quarters of the images they were presented with compared to the first quarter for image visibilities inferior to 40% ($p<10^{-3}$, Chi-squared test). Interestingly, there is also a significant increase for their performance on whole objects ($p<10^{-6}$, righ tailed two-sample t-test). However, we cannot discriminate motor learning from visual learning using the recorded data. Indeed, subjects had to press specific keys on a gamepad to enter the category they chose, which requires some familiarization. But since the subjects had no time constraint to respond, we can hypothesize that learning is mainly due to visual integration.

Learning was also found to take place especially between the first and second quarters of the data while increase was between other quarters was smaller as shown by **Figure ??**. This probably means that subject would most not improve significantly more if exposed larger amount of partial images. More extensive experience with whole objects however might still improve human performance in a more drastic way. Therefore it could be interesting to repeat this experiment with more training examples and exposures.

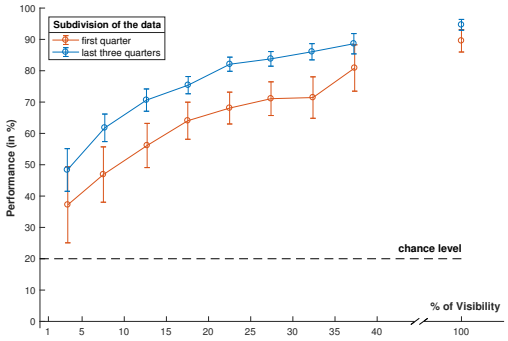


Figure 5.3: Performance as a function of image visibility for the first quarter of the data each subject was exposed to compare to the remaining portion of the experiment.

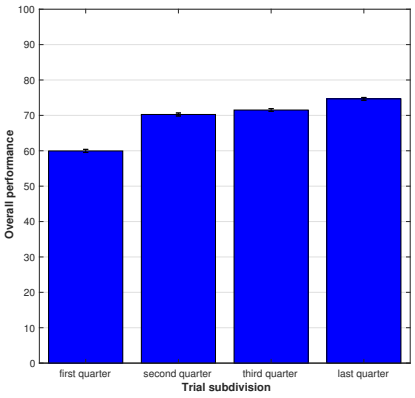


Figure 5.4: Overall performance of subject on successive quarters of the data

5.1.4 Isolating feed-forward responses

We demonstrate that backward masking significantly impairs partial object categorization when stimulus onset asynchronies (SOAs) below 100ms are used (**Figures 4.2** and **4.3**) while it only impairs whole objects for SOAs below 25ms. As exposed earlier in the **Materials and Methods** chapter, presenting a high-contrast spatial noise mask stimulus directly after an image presented with a short SOA efficiently interrupted any additional processing[61, 62, 63, 64, 65, 66]. In recent studies[29], delayed visually selective responses to partially visible images were recorded in

the human ventral cortex compared to neural responses to continuous lines or whole images. We argue that these delays are due to a difference in mathematical operation rather than weaker neural inputs. It is indeed also possible that higher brain areas simply gets weaker signals from neurons when exposed to occluded objects compared to their whole counterparts. This observation however needs to be specific to higher level neurons since no temporal delays between the whole and occluded conditions where observed in early visual processing areas such as V1. Weaker inputs can therefore not be due to low-level features such as differences in contrast, or sparser pixel information but might be caused by higher level interpretations of the data which potentially connects again with the recurrent connection explanation. The need of recurrent connections for pattern completion is however strongly supported by observed correlation between neural latencies and the effect of backward masking measured by a masking index. Another argument in favor of recurrent connections is that recurrent computational models showed correlation between the distances from partial images to their whole category center and observed physiological latencies[1]. Finally, augmenting computational models with recurrent connections significantly increased their performance for partial object categorization tasks, suggesting that these computations are indeed needed to mathematically interpret partial information. In the future, it would also be interesting to find a protocol to effeciently disambiguate horizontal connections from top-down effects. Attempts have been made in this direction by contrasting delay values of long ranging connections compared to local ones[32].

5.1.5 Limitations of chosen stimuli

While the adjectives occluded and partial were used interchangeably throughout this thesis to describe images with limited visibility, previous studies[1, 22] have reported that using an explicit occluder actually increases performance for very low visibility levels. For our behavioral experiment, deleted pixels where treated as "holes" through which the background can be seen. In real life, objects parts are rarely deleted in this way. Therefore it would be more biologically relevant to repeat our experiment using occluders to mask image information. Indeed, simply deleting information might especially favor modal completion operations within the brain rather than amodal ones.

Finally, it is well known that psychophysics investigation results depend heavily on utilized stimuli. The stimuli used in this experiment did not represent natural tasks which humans have to perform everyday. The stimuli were composed of a unique object which was well isolated from its background. It is however important to attempt to isolate different effects and this requires to initially use simple stimuli. More naturalistic images surrounded by contextual information can then be used to understand the interaction between basic principles underlying the visual cortex and influence from higher or different brain areas.

5.1.6 Differences between image identification and categorization

A recent study in the lab[1] used a similar behavioral experiment to ours but with familiar object stimuli to contrat human performance for a categorization task compared to a separate identification task. During the latter, subjects were first presented with the occluded image and subsequently asked if the presented stimulus corresponded to a whole object displayed on the right or another one displayed on the left. Stimuli were rotated randomly to prevent pixel matching behaviors. It was observed that performance for objects presented with high visibility percentages was lower in the identification task than during categorization. Indeed the task was less trivial since sometimes

subjects had to choose between images from the same categories (face versus face for example) and therefore have to pay closer attention to details. On the other hand, performance at very low visibility levels was increased in the case of the identification task. Seeing whole counterparts after image exposure can indeed help the user to identify the image if he did not do so successfully before the choice screen appeared. In this case, the subject mainly memorizes low level features and tries to find matching minimal information on the response screen. It could therefore be to see if subject identification performance also increases for low visibility levels when using novel objects to see if mainly immediate or more long term memories are involved in this process.

5.2 Conclusions from the computational models classification performance

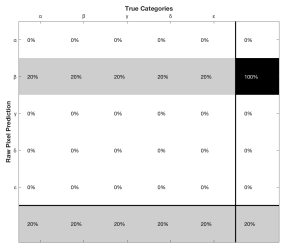
5.2.1 Impact of adding recurrent connections

Several bottom up computational architectures were tested on the same dataset of occluded novel images as the one used for the behavioral experiment and they performed below human performance on the partial image categorization task (**Figure 4.5**). While humans almost reach 60% performance for image visibilities of only $5 \pm 2.5\%$, our best feed-forward layer only reaches 38%. However, we do not imply that it is mathematically impossible for models with purely feed-forward architectures in general to be robust to very high object occlusion levels, even though the AlexNet model we investigated is widely known for its low classification error for whole object recognition in the computer science field. As a matter of fact, an analysis of feature representations of the implemented Alexnet model did indeed show easily separable clusters of whole objects, even distinguishing among families within the given categories (**Figure 4.6**). However, representations of partial objects grouped together, far away from their whole counterparts. Although any recurrent network can technically be unfolded into a feed forward architecture by adding a fully connected layer for each recurrent time step, introducing recurrent connections as inspired by the brain can be a more elegant alternative. Indeed, weight sharing over time drastically reduces the number of required neural units and weights which need to be trained. These networks are thereby less data-hungry meaning they can use smaller datasets to achieve good performance.

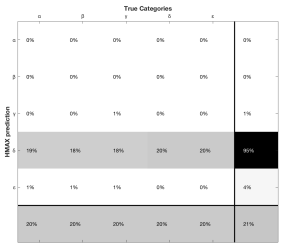
Because of the previously exposed motivations, we fine-tuned sections of the pre-trained Alexnet feature extractor by adding a recurrent layer either to the last convolutional or fully connected layer respectively. This addition to the model's architecture resulted in computational models which in some cases outperformed humans across all visibility levels in the novel image categorization task (**Figure 4.10**). Although it is unclear how to link the recurrent neural network computational time steps to physiological delays measured in milliseconds, overall performance of these networks increased with each time step(**Figure 4.11**). This effect can also be observed by observing how feature representations evolve dynamically (**Figure 4.12** and **4.13**). Indeed, feature representations of the partial objects progressively cluster with their whole counterparts. A previous study[1] has also analyzed the effect of backward masking on these neural networks. The idea was to extract features of a spatial noise mask using AlexNet up to its last layer before read-out. The resulting features were then supplied to Hopfield and RNN5 recurrent models during partial image feature extraction at different recurrent time steps. The introduction of this mask increasingly dropped the model's performance with decreasing onset time, which was consistent with the effect observed for humans. This investigation could be an interesting addition to our study.

5.2.2 Some hints about HMAX sub-chance level performance

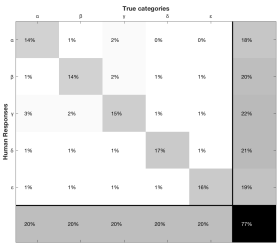
To explain the surprisingly low performance of the HMAX computational model, we plotted the confusion matrix for the pixels and HMAX features performances as presented in **Figure 5.5**. One would expect all image categories to be equally hard to classify if no bias were present in the data. This is not the case here since the pixel-based classifier decides that every image belongs to the category β while when using the HMAX features all the images are classified as category δ. This implies that some low-level feature is most probably biasing these two classifiers even if it is weird that the error type is not the same. Contrast can be ruled out because all images were throughly normalized. Image area with respect to background might explain this erratic behavior amongst other possibilities. Further investigations would be needed to determine the exact cause of this behavior.



(a) Pixel confusion matrix shows a bias for category β



(b) HMAX confusion matrix shows a bias for category δ



(c) Human confusion matrix show well balanced error distributions.

Figure 5.5: Confusion matrices for the classifier using raw pixels (a) and the HMAX model (b) as opposed to human performance results (c). Confusion matrices give a better idea of the type of errors the classifier make and are a first step to analyze potential erratic behaviors. Ideally, the images introduce no bias and errors should have a symmetric distribution across categories

To visualize the effect of this bias on the model representations, a t-SNE projection of the features extracted by HMAX on two dimensions presented in figure 5.6 provides us with additional support about why all images are classified as δ. While the representation of whole objects shows a clear separation between categories, the occluded objects are clustered together and their position is biased towards the whole δ group.

5.2.3 Observations about Alexnet features generalization and specificity

With the novel object dataset, it was observed in **Figure 4.5** that the Alexnet pool 5 model performs better than the fc7 model. Although this result contrasts previous results using Alexnet on datasets with more familiar objects[1], it can be explained by the fact that fully-connected layers extract more class-specific features. Since Alexnet was trained on everyday objects from the ImageNet database, it can be argued that the last convolutional layer (pool 5) generalizes better to the novel objects than fc7[78, 79]. It is in fact advised to use the more generalizable layers for transfer learning[80] which consists of using the extracted features of a pre-trained model and train another network which will be specific to our dataset on top of it. This approach is very popular in machine learning because it requires less time-intensive training and enables the usage of small datasets. We can thus conclude the convolutional layers of Alexnet are powerful feature

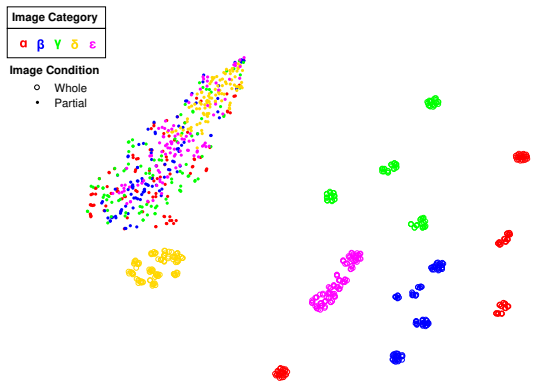


Figure 5.6: The representation of layer C2b of the HMAX computational model is visualized using the stochastic neighborhood embedding (t-SNE) dimensionality reduction technique. While whole objects on which the SVM was trained (open circles) can be easily separated into distinct categories, the partial renderings seem to be more difficult to classify. Moreover, the bias towards the δ category is very clear from this plot since it is much closer to the partial objects than any other category.

extractor tools which manage to successfully and independently identify discriminative features even for artificially created novel objects that do not resemble the data it was trained with in any way.

5.2.4 Importance of prior exposure to partial objects

As explained in the **Materials and Methods** chapter, the parameters of RNNh (Hopfield network) do not depend on the partial objects and all the weights are entirely determined the whole object features extracted by the last layer before recurrent augmentation. Even though it has no prior knowledge of occlusion, the combination of RNNh and fc7 signficantly outperforms the fc7 layer and the combination of RNNh with pool 5 even outperforms human results for all studied visibility levels. Previous experience with object occlusion therefore does not seem necessary with our stimuli. However, the images we used were very simple and artificially created using computational algorithms. These characteristics might make them easier to comprehend for computational models than natural images whose classification rules are less intuitive for a machine. Using novel stimuli created from artificial evolutionary algorithms might counteract this potential facilitation. It would also be interesting to repeat the performed behavioral experiments with more training with whole and/or occluded images for humans to see if they would eventually match the RNNh + pool 5 model's performance.

Exposing the computational models to partial objects by training them to explicitly minimize distance between whole and partial objects did only increase performance for very low visibility levels compared to Hopfield based models and human data (**Figure 4.14**). Although special care was taken to reduce overfitting by early training stopping and putting aside unseen data for testing, the model has to adjust a very large number of parameters. We cannot rule out that the model might be merely memorizing very specific image features (potentially at the single pixel level) because training that the model was shown to not generalize well when trained with only one category. Another study[1] has explored the amount of prior knowledge needed by the this type of recurrent algorithm by training it on only one category and testing it on all others. Performance was found to stay at base feed-forward level before recurrent augmentation. This implies that rich dictionaries of features are most likely needed for successful generalization which might also explain human high performance with novel objects. Indeed humans are trained to recognize whole and occluded objects since they were infants and have a very large pre-existing library of category specific features. They might unconsciously draw links between identified features of the novel objects and the most similar features they already learned.

5.3 General conclusion and further scientific questions to be addressed

In order to further understand the benefits of recurrent connections in the visual processing area of the brain and eventually use this understanding to optimize existing models in computer vision, we explored the role of prior experience with objects. Humans were found to robustly categorize a set of artificially created novel stimuli without having any prior experience with these partial objects. In addition, training with whole counterparts was also restricted to the minimal amount necessary to fully account performance to the occlusion effect. Therefore, prior knowledge about the object does not seem drastically deteriorate low visibility partial images. Whether this ability can be attributed to extensive training with whole and occluded objects throughout the person's live is a question that remains to be answered.

Some state-of-the art purely feed-forward networks were when exposed to the same dataset as the one used for the behavioral experiments to investigate the role of experience and recurrencies in computer vision. Although performance on classification of whole object was at ceiling, they were not robust to heavy image occlusion levels . When augmenting them with a Hopfield based recurrent network their performance approached or even outmatched human results even though these networks never experienced occluded objects before. Training another class of recurrent networks (RNN5) explicitely on minimizing distance between whole and partial images gave higher results only for very extremely low visibilities.

Additional performance increase would be expected by training the combination of feed-forward and recurrent networks end-to-end with a small amount of partial objects instead of just fine-tuning a pre-existing model as we did. Maybe the model will discover a pattern completion mechanism autonomously just like it was able to extract features from images without particular guidance. The presence of recurrent computations could significantly alleviate the size of the required dataset to reach this desired property. Another investigation would be to create a more brain-inspired network by adding less dense connections (not in an all-to-all fashion) between several layers of a feed forward computational model.

Finally it would be interesting to extend our investigation to other mechanisms requiring recurrent connections within the visual cortex. Context awareness for example can significantly help object recognition but so can other cues such as understanding of textures, relative positions, segmentation, movement and the source of illumination amongst others.

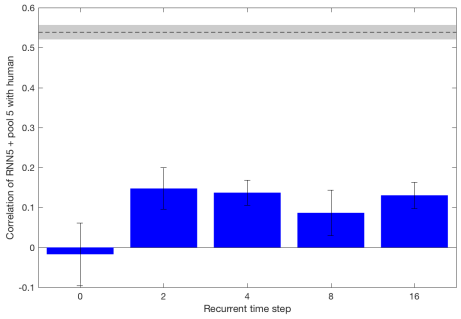
Appendix A: Examples of whole objects chosen from the novel objects repository

Figure 7: Exemples of whole pre-processed objects from categories δ and ϵ. Six images were chosen per family within these categories. Background was replaced by a neutral gray color and constrast was normalized.

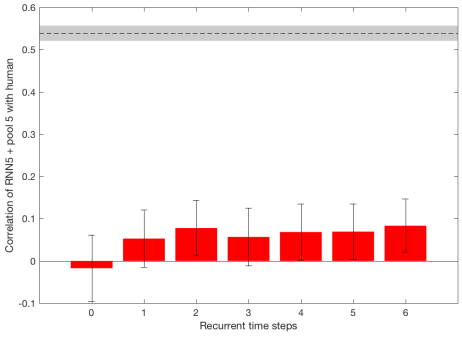
Figure 8: Exemples of whole pre-processed objects from categories α, β and γ. Six images were chosen per family within these categories. Background was replaced by a neutral gray color and constrast was normalized.

Appendix B: Only small correlation was found with human data

Although high performance is also reached for the novel object dataset with the recurrent augmented pool5 computational models, classification responses did not have very high overall correlation coefficient with human responses as presented in **Figure 9**. Moreover, correlation was not found to increase over time after the initial recurrent t=1. Correlation for augmented fc7 layer did not show any significant correlation for all timesteps. Individual object correlations between humans and the computational model presented in **Figure 10**. Small correlations values despite close performance between human and RNN networks means that the errors are not made on the same objects. However, these results contrast previous observations using familiar objects[1] were correlation was found to increase with recurrent timestep and increasing overall performance.



(a) Mean per-category correlation between the RNNh + pool 5 model and human data



(b) Mean per-category correlation between the RNN5 + pool 5 model and human data

Figure 9: Correlations in the response pattern between some recurrent networks implemented on top of pool 5 and humans. The dashed line indicates the inter-human correlation and the shaded area represents standard deviation (S.D.). Inter-human correlation was computed by correlating one half of the subjects with the other half. Correlation coefficients were computed for each time step and model separately for each category to avoid domination of correlation by category differences. Regressions were then averaged across categories. Error bars represent S.D.

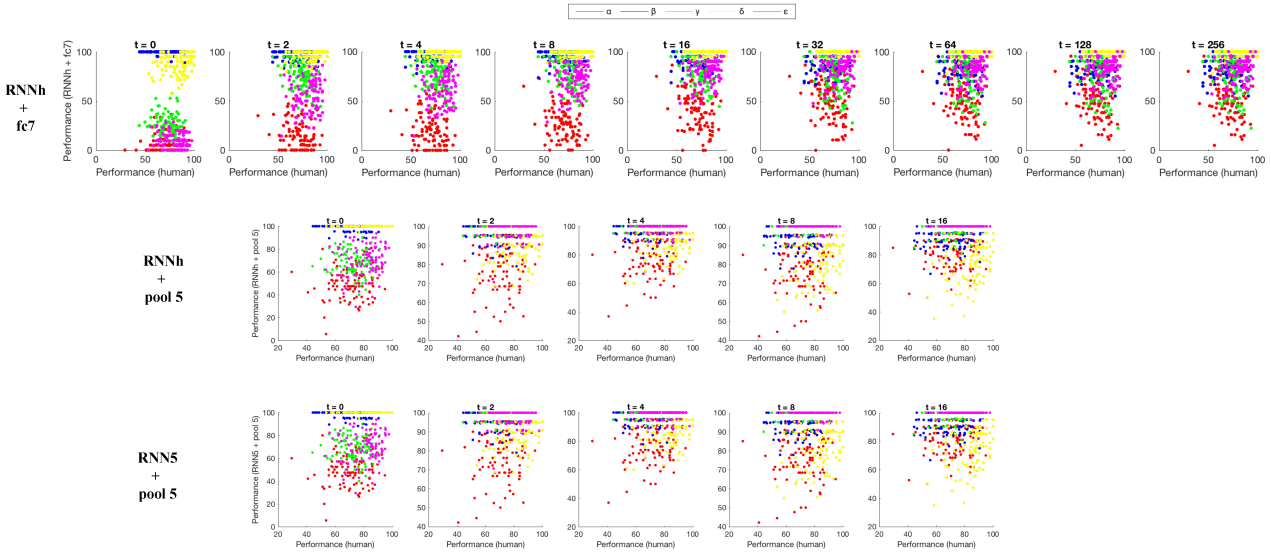


Figure 10: Correlation between RNN models and human performance at the individual object level for different recurrent time steps. The top figure shows model RNNh+fc7, the middle one RNNh + pool 5 and the bottom one RNN5 + pool 5. Each dot represent an individual object. Colors denote object categories.

Bibliography

[1] Hanlin Tang, Bill Lotter, Martin Schrimpf, Ana Paredes, Josue Ortega Caro, Walter Hardesty, David Cox, and Gabriel Kreiman. Recurrent computations for visual pattern completion.

[2] Charles E. Connor, Scott L. Brincat, and Anitha Pasupathy. Transformation of shape information in the ventral pathway. *Current Opinion in Neurobiology*, 17(2):140–147, 2007.

[3] Nikos K. Logothetis and David L. Sheinberg. Visual object recognition. *Annual Review of neuroscience*, 19(1):577–621, 1996.

[4] Keiji Tanaka. Inferotemporal cortex and object vision. *Annual Review of Neuroscience*, 19:109–139, March 1996.

[5] Thomas Serre, Aude Oliva, and Tomaso Poggio. A feedforward architecture accounts for rapid categorization. *Proceedings of the national academy of sciences*, 104(15):6424–6429, 2007.

[6] Rodney J. Douglas and Kevan A.C. Martin. Neuronal circuits of the neocortex. *Annual Review of Neuroscience*, 27:419–451, 2004.

[7] Jonathan J. Nassi and Edward M. Callaway. Parallel processing strategies of the primate visual system. *Nature Reviews Neuroscience*, 10(5):360–372, 2009.

[8] David H. Hubel and Torsten Wiesel. Receptive fields and functional architecture of monkey striate cortex. *The Journal of physiology*, 195(1):215–243, 1968.

[9] Thomas Serre and Maximilian Riesenhuber. Realistic modeling of simple and complex cell tuning in the hmax model, and implications for invariant object recognition in cortex. *AI-MEMO-2004-017 MASSACHUSETTS INST OF TECH CAMBRIDGE COMPUTER SCIENCE AND ARTIFICIAL INTELLIGENCE LAB*, 2004.

[10] Tom Binzegger, Rodney J. Douglas, and Kevan A Martin. A quantitative map of the circuit of cat primary visual cortex. *Journal of Neuroscience*, 24(39):8441–8453, September 2004.

[11] EM Callaway. Feedforward, feedback and inhibitory connections in primate visual cortex. *Neural Networks*, 17(5):625–632, 2004.

[12] Daniel J. Felleman and David C. Van Essen. Distributed hierarchical processing in the primate cerebral cortex. *Cerebral Cortex*, 1(1):1–47, 1991.

[13] Geraint Rees, Gabriel Kreiman, and Christof Koch. Neural correlates of consciousness in humans. *Nature Reviews Neuroscience*, 3(4):261–270, 2002.

[14] Victor A. F. Lamme, Hans SupÃÍr, and Henk Spekreijse. Feedforward, horizontal, and feedback processing in the visual cortex. *Current Opinion in Neurobiology*, 8(4):529–535, 1998.

[15] Hanlin Tang and Gabriel Kreiman. Recognition of occluded objects. *Pattern Recognition*, 25(10):1107–1117, 2016.

[16] Gaetano Kanizsa. Organization in vision: essays on gestalt perception. *Praeger Publishers*, 1979.

[17] Manish Singh. Modal and amodal completion generate different shapes. *Psychological Science*, 15(7):454–459, 2004.

[18] Johan Wagemans, James H. Elder, Michael Kubovy, S. Palmer, Mary A. Peterson, and Manish Singh. A century of gestalt psychology in visual perception: I. perceptual grouping and figure-ground organisation. *Psychological bulletin*, 138(6):1172, 2012.

[19] Manish Singh and Jacqueline M. Fulvio. Bayesian contour extrapolation: geometric determinants of good continuation. *Vision Research*, 47(6):784, 2007.

[20] Ken Nakayama, Zijiang J. He, and Shinsuke Shimojo. Visual surface representation: a critical link between lower-level and higher-level vision. *Visual cognition: An invitation to cognitive science*, 2:1–70, 1995.

[21] Yoichi Sugita. Grouping of image fragments in the primary visual cortex. *Nature*, 401(6750):269–272, 1999.

[22] Jeffrey S. Johnson and Bruno A. Olshausen. The recognition of partially visible natural objects in the presence and absence of their occluders. *Vision Research*, 45(25):3262–3276, November 2005.

[23] Albert S. Bregman. Asking the "what for" question in auditory perception. *Perceptual Organization*, pages 99–118, 1981.

[24] Ken Nakayama, Shinsuke Shimojo, and Gerald H. Silverman. Stereotypic depth: Its relation to image segmentation, grouping and recognition of occluded objects. *Perception*, 18:55–68, 1989.

[25] Frank Kelly and Stephen Grossberg. Neural dynamics of 3-d surface perception: Figure-ground separation and lightness perception. *Perception and Psychophysics*, 62(8):1596–1618, December 2000.

[26] Yoshito Kosai, Yasmine El-Shamayleh, Amber M. Fyall, and Anitha Pasupathy. The role of visual area v4 in the discrimination of partially occluded shapes. *Journal of Neuroscience*, 34(25):8570–8584, June 2014.

[27] Kristina J. Nielsen, Nikos K. Logothetis, and Gregor Rainer. Discrimitation strategies of humans and rhesus monkeys for complex visual displays. *Cell*, 16(8):814–820, April 2006.

[28] Kristina J. Nielsen, Nikos K. Logothetis, and Gregor Rainer. Dissociation between local field potentials and spiking activity in macaque inferior temporal cortex reveals diagnosticity-based encoding of complex objects. *Journal of Neuroscience*, 26(38):9639–9645, September 2006.

[29] Hanlin Tang, Calin Buia, Radhika Madhavan, Nathan E. Crone, Joseph R. Madsen, William S. Anderson, and Gabriel Kreiman. Spatiotemporal dynamics underlying object completion in human ventral visual cortex. *Neuron*, 83(3):736–748, August 2014.

[30] Robert Desimone, Thomas D. Albright, Charles G. Gross, and Charles Bruce. Stimulus-selective properties of inferior temporal neurons in the macaque. *The Journal of Neuroscience*, 4(8):2051–2062, August 1984.

[31] Minami Ito, Hiroshi Tamura, Ichiro Fujita, and Keiki Tanaka. Size and position invariance of neuronal responses in monkey inferotemporal cortex. *Journal of Neurophysiology*, 73(1):218–226, 1995.

[32] Hesheng Liu, Yigal Agam, Joseph R. Madsen, and Gabriel Kreiman. Timing, timing, timing: fast decoding of object information from intracranial field potentials in humanvisual cortex. *Neuron*, 62(2):281–290, April 2009.

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

[34] Juan Chen, Bingyun Liu, Bing Chen, and Fang Fang. Time course of amodal completion in face perception. *Vision Research*, 49(7):752–758, April 2009.

[35] Timothy J. Buschman and Earl K. Miller. Top-down versus bottom-up control of attention in prefrontal and posterior parietal cortices. *Science*, 315(5820):1860–1862, 2007.

[36] Christian Keysers, D.K Xiao, P. FÃűldiÃąk, and D. I. Perret. The speed of sight. *Journal of Cognitive Neuroscience*, 13(1):90–101, 2001.

[37] Victor A.F. Lamme and Pieter R. Roelfsema. The distinct modes of vision offered by feedforward and recurrent processing. *Trends in Neuroscience*, 23(11):571–579, 2000.

[38] Matthew T. Schmolesky, Youngchang Wang, Doug P. Hanes, Kirk G. Thompson, Stefan Leutgeb, Jeffrey D. Schall, and Audie G. Leventhal. Signal timing across the macaque visual system. *Journal of Neurophysiology*, 79(6):3272–3278, 1998.

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

[40] Ilya Sutskever, James Martens, George Dahl, and Geoffrey Hinton. On the importance of initialization and momentum in deep learning. *Proceedings of the 30th International Conference on International Conference on Machine Learning*, pages 1139–1147, February 2013.

[41] Jia Deng, Wei Dong, Richard Socher, Li-Jia Li, Kai Li, and Li Fei-Fei. Imagenet: A large-scale hierarchical image database. *IEEE Conference on Computer Vision and Pattern Recognition*, pages 248–255, June 2009.

[42] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 770–778, 2016.

[43] Andrej Karpathy and Fei-Fei Li. Convolutional neural networks for visual recognition,, 2015.

[44] 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.

[45] David E. Rumelhart, Geoffrey E. Hinton, and Ronald J. Williams. Learning representations by backpropagating errors. *Nature*, 323(6088):533–536, 1986.

[46] JÃijrgen Schmidhuber. Deep learning in neural networks: An overview. *Neural Networks*, 61:85–117, 2015.

[47] Ning Qian. On the momentum term in gradient descent learning algorithms. *Neural Networks*, 12(1):145–151, 1998.

[48] Sebastian Ruder. An overview of gradient descent optimization algorithms. 2016.

[49] Dumitru Erhan, Yoshua Bengio, Aaron Courville, and Pascal Vincent. Visualizing high level features of a deep network. techreport 1341, University of Montreal, 2009.

[50] Nair Vinod and Geoffrey E. Hinton. Rectified linear units improve restricted boltzmann machines. *Proceedings*

of the 27th international conference on machine learning, 2010.

[51] Xavier Glorot, Antoine Bordes, and Yoshua Bengio. Deep sparse rectifier neural networks. *Aistats*, 15(106), 2011.

[52] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Delving deep into rectifiers: surpassing human-level performance on imagenet classification. *Proceedings of the IEEE international conference on computer vision*, pages 1026–1034, 2015.

[53] Sergey Ioffe and Christian Szegedy. Batch normalization: Accelerating deep network training by reducing internal covariate shift. 2015.

[54] Sepp Hochreiter. The vanishing gradient problem during learning recurrent neural nets and problem solutions. *International Journal of Uncertainty, Fuzziness and Knowledge-Based Systems*, 6(2):107–116, April 1998.

[55] Paul J. Werbos. Backpropagation through time: what it does and how to do it. *Proceedings of IEEE*, 78(10):1550–1560, Octoboer 1990.

[56] Sepp Hochreiter and Jurgen J. Schmidhuber. Long short-term memory. *Neural computation*, 9(8):1735–1780, November 1997.

[57] Merav Ahissar and Shaul Hochstein. The reverse hierarchy theory of visual perceptual learning. *Trends in Cognitive Sciences*, 8(10):457–464, October 2004.

[58] Alireza Avanaki. Exact histogram specification optimized for structural similarity. 2008.

[59] Frederic Gosselin and Philippe G.Schyns. Bubbles: a technique to reveal the use of information in recognition tasks. *Vision Research*, 41(17):2261–2271, 2001.

[60] Thomas Serre, Gabriel Kreiman, Minjoon Kouh, Charles Cadieu, Ulf Knoblich, and Tomaso Poggio. A quantitative theory of immediate visual recognition. *Progress in Brain Research*, 165:33–56, 2007.

[61] Bruce Bridgeman. Temporal response characteristics of cells in monkey striate cortex measured with metacontrast masking and brightness discrimination. *Brain Research*, 196(2):347–364, 1980.

[62] Victor A.F. Lamme, Karl Zipser, and Henk Spekreijse. Masking interrupts figure-ground signals in v1. *Journal of cognitive neuroscience*, 14(7):1044–1053, 2002.

[63] Gyula Kovacs, Rufin Vogels, and Guy A. Orban. Cortical correlate of pattern backward masking. *Proceedings of the National Academy of Sciences of the United States of America*, 92:5587–5591, June 1995.

[64] Edmund T. Rolls, Martin J. Tovee, and Stefano Panzeri. The neurophysiology of backward visual masking: information analysis. *Neurophysiology*, 11(3):300–311, May 1999.

[65] Christian Keysers and David I. Perrett. Visual masking and rsvp reveal neural competition. *Trends in cognitive sciences*, 6(3):120–125, 2002.

[66] James T. Enns and Vincent Di Lollo. WhatâĂŹs new in visual masking? *Trends in cognitive sciences*, 4(9):345–352, 2000.

[67] Leon D. Harmon and Bela Julesz. Masking in visual recognition: effects of two-dimensional filtered noise. *Science*, 180(4091):1194–1197, June 1973.

[68] Alex Krizhevsky, Ilya Sutskever, and Geoffrey E. Hinton. Imagenet classification with deep convolutional neural networks. *Advances in neural information processing systems*, pages 1–9, 2012.

[69] Nitish Srivastava, Geoffrey Hinton, Alex Krizhevsky, Ilya Sutskever, and Ruslan Salakhutdinov. Dropout: a simple way to prevent neural networks from overfitting. *Journal of Machine Learning Research*, 15(1):1929–1958, June 2014.

[70] Yangqing Jia, Evan Shelhamer, Jeff Donahue, Sergey Karayev, Jonathan Long, Ross Girshick, Sergio Guadarrama, and Trevor Darrell. Caffe. In *Proceedings of the International Conference on Multimedia*. ACM Press, 2014.

[71] John J. Hopfield. Neural networks and physical systems with emergent collective computational abilities. *Proceedings of the national academy of sciences*, 79(8):2554–2558, April 1982.

[72] Dmitry Krotov and John J. Hopfield. Dense associative memory for pattern recognition. *Advances in Neural Information Processing Systems*, 2016.

[73] J.H. Li, Anthony N. Michel, and Wolfgang Porod. Analysis and synthesis of a class of neural networks: Linear systems operating on a closed hypercube. *IEEE transactions on Circuits and Systems*, 36(11):1405–11422, 1989.

[74] Donald O. Hebb. The organization of behavior: A neuropsychological theory. *Psychology Press*, 911(1), 1949.

[75] Kendra S. Burbank and Gabriel Kreiman. *Visual Population Codes: Towards a common multivariate frame-*

work of cell recording and functional imaging, chapter 17 (Introduction to the anatomy and function of the visual cortex), pages 477–496. MIT Press, 2012.

[76] Laurens van der Maaten and Geoffrey Hinton. Visualizing data using t-sne. *Journal of Machine Learning Research*, 2008.

[77] Yoshua Bengio. Learning deep architectures for ai. *Foundation and trends in Machine Learning*, pages 1–127, 2009.

[78] Pulkit Agrawal, Ross Girshick, and Jitendra Malik. Analyzing the performance of multilayer neural networks for object recognition. 2014.

[79] Ali Sharif Razavian, Hossein Azizpour, Josephine Sullivan, and Stefan Carlsson. Cnn features off-the-shelf: an astounding baseline for recognition. 2014.

[80] Jason Yosinski, Jeff Clune, Yoshua Bengio, and Hod Lipson. How transferable are features in deep neural networks? 27.