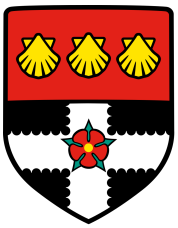
UNIVERSITY OF READING

Department of Computer Science

School of Mathematical, Physical and Computational Sciences



Sparse hierarchical models of vision

by

Radhika Nath

r.nath@pgr.reading.ac.uk

Thesis Advisor: Dr. M. Manjunathaiah

Thesis submitted for the degree of Doctor of Philosophy

Sept, 2016

Abstract:

In recent years, deep convolutional neural networks (CNNs), which are based on the hierarchical structure of the visual cortex has found remarkable accuracy rates in object classification. One of its drawbacks is the requirement of a large collection of labelled data for training.

Therefore, unsupervised hierarchical networks are more suited for a more biologically plausible model. For a classification accuracy to be closer towards CNNs, the invariance and selectivity of the extracted features need to be improved. One of the standard methods for learning invariant features is to apply non-linearity functions to the data which has been implemented in both CNNs and HMAX models. Based on these principles, an extended form of the HMAX model is proposed which applies two different types of non-linear pooling operations.

The extension is designed with the help of sparsity based algorithms such as Independent Subspace Analysis (ISA) and Topographic Independent Component Analysis(TICA). Aside from an improved classification accuracy compared to previous unsupervised hierarchical models, it also reduces the data dimensions within the layers.

Contents

List of Figures

Chapter 1

Introduction

Contents

1.1 Introduction

The brain generates an immediate response upon stimulus which stems from a rapid exchange of information across different layers of neurons. Its unique ability to process vast amounts of sensory stimuli is exemplified in the speed, robustness and generality of the primate visual system. The visual cortex has thus been a source of extensive research in the field of neuroscience.

In recent years, insights from neurophysiological findings has laid the groundwork for biologically inspired computational vision systems. These models have gained prominence due to their efficiency in task oriented processes. The motivation behind such endeavours arise from the dual purpose of designing superior artificial intelligence systems and studying the human brain. Although some large scale models has achieved almost human-level accuracies [21] or even surpassed them [22], replicating the generality and robustness of the visual cortex is still an ongoing challenge [23].

The primate vision system is characterized by an optimal balance between invariance and selectivity. For modelling such features, many modern cognitive frameworks have adopted a deep-hierarchical architecture based on the pioneering work by Hubel and Wiesel [24]. In these models, information is processed in layers of gradually increasing complexity. Each unit or receptive field within a layer is the combination of multiple afferent units in the lower layer. They also identified two different types of components that contribute towards this property in the primary visual cortex or V1: The simple cells, which selectively respond

to stimulus of a particular orientation, position and phase. And the complex cells, which are invariant to changes in position and phase, while maintaining selectivity towards orientation.

One of the most prominent models which demonstrated invariance with this type of architecture was the HMAX [8]. Its significance in the field cognitive vision is due its biological plausibility. By alternating simple and complex layer functions, it formed high level units that were selective to complex features while displaying a degree of tolerance towards translation and scale variations.

Another aspect of naturally occurring systems is unsupervised learning. The sparse coding algorithm developed by Olhausen and fields [25] demonstrated the emergence of receptive fields that resembled the simple cell properties of the V1 when applied on natural images. With this unsupervised learning technique, a second property of sparsity was also implemented. Signal sparsity has widely been supported by studies in neuroscience [26][27][10] and its application in hierarchical vision frameworks has resulted in highly efficient object recognition models [28] [13] [10].

Building upon the HMAX framework by integrating sparse, unsupervised learning techniques, new biologically inspired recognition models were developed by incorporating Independent Subspace Analysis and Topographic Independent Component Analysis. These optimizations showed an improvement in object classification compared to the current HMAX based models. These models were further extended with attentional modulation and compression techniques.

1.1.1 Sparse hierarchical models of vision

In the HMAX models, the position and scale invariance of complex cells were obtained by pooling over units of similar orientation but slightly differing locations and spatial resolutions respectively [8][29]. In other sparse unsupervised models, phase invariance was achieved by pooling over dependent units with similar orientations [13][12]. These invariant properties were generally modelled with a non-linearity function. Since learning more than one type of invariance contributes greatly to model performance, two layers of complex cell functions is implemented to improve classification accuracy as well as to reduce data dimensionality.

To implement the models, Independent Subspace Analysis (ISA) and Topographical Independent Component Analysis (TICA) was applied. These unsuper-

vised algorithms are an extension of the Independent Component Analysis (ICA) [30][31][32]. Similar to sparse coding, applying ICA on natural images also generates units resembling V1 simple cells [33]. But with ISA and TICA, the receptive fields are grouped based on their dependencies, which parallels the properties of complex cells. Multiple layers of selectivity and invariance led to the formation of high level units responsive to complex features. The activation values of these units were then evaluated on object classification and feature detection tasks, which were found to be highly competitive compared to recent biologically inspired models.

The chapters are organized as follows. In chapter 2, an overview of the established structure and functional properties of the visual cortex was given. The various stages of vision processing in the brain was described while highlighting the aspects more relevant to our vision models. The significant aspects include primary visual cortex, model hierarchy and sparsity.

In chapter 3, some of the existing biologically inspired vision models were reviewed, with a focus on unsupervised and sparse hierarchies [13], [12] [10]. HMAX models and its extensions were also investigated due to their impact on cognitive based models.

In chapter 4, two models of sparse hierarchical vision were implemented, with ISA and TICA. In these models, the learned units or filters are grouped into subspaces or neighbourhoods based on their energy correlation. The response of each subspace was determined by the L2 pooling function [13], which is the square root of its summed energies. In the first layer of our model, linear filters were applied (generated by ISA or TICA) on patches of image data. The response of these orientation, phase and frequency selective filters within a subspace or neighbourhood were L2 pooled. In the next stage, max pooling was applied over neighbouring locations on the feature maps to encode shift invariance. With each layer, receptive fields of larger size and higher complexity were learned which led to the emergence of highly invariant and selective units. After evaluating the extracted features using images of different object classes, an improvement in classification accuracy over other unsupervised hierarchical models was observed.

The different parameters of the model, such as subspace size and number of units in each layer were also investigated. The evaluations demonstrated that the models with increasing receptive field sizes and L2 pooling performed object classification with higher accuracy than the rest. Even with the absence of pooling

within multiple spatial resolutions, the new models performed better than those that modelled scale invariance.

In chapter 5, attention modulation was integrated through the use of saliency maps. The formation of saliency maps were based on similar principles as unsupervised recognition models in chapter 4. The motivation behind this optimisation was to reduce the amount of redundant data to be processed by the model. After surveying some of the biology based saliency models, a saliency map based on the feature integration theory [34] and the Itti and Koch model [17] was applied. This was based on the 'bottom-up' mechanism of attention, where the focus is directed in an involuntary process depending on the natural statistical properties of the visual scene [35]. This map was then used for directing the extraction of samples only towards the most salient regions of the visual field. Other popular saliency algorithms such as the GBVS [19] and SUN [20] were also applied to compare their performance. For a model with low number of samples, the overall classification accuracy of saliency modulated feature detectors saw a slight improvement over the models that applied randomised sampling strategy.

In Chapter 6, a compressive form of the HMAX model by applying the principles of sparsity and compressed sensing (CS) [36] was investigated. It is a powerful compression technique which is able to recover any sparse data sampled at a rate lower than the Nyquist rate. With sparse hierarchical models, applying CS allows the data to be propagated in its compressed form. Evaluation on multi-category object classification revealed that the accuracy of the model was dependent on the reconstruction error associated with the compression matrix. When compression with a low reconstruction error was applied, the performance of the model was almost on par with its uncompressed version.

In chapter 7, a summary of the insights gained in each of the chapters were provided. The limitations of some of the approaches were also discussed, after which avenues for further research were identified.

Chapter 2

Visual cortex and the mechanisms behind perception

Contents

2.1 Introduction

Understanding the mammalian visual process has been an ongoing study for over a few decades [23]. The complexity of the entire visual cortex cannot be represented by simple models since many different components continuously interact with each other. In recent years numerous biologically inspired recognition models were developed to capture the invariance and selectivity of the primate visual system. A certain degree of progress has been made in modelling these individual aspects of the cognitive process such as attention, recognition, motion detection, but a complete cognitive framework with the generality, robustness and speed of the visual cortex has not yet been achieved [23].

Since the models in the following chapters are mainly inspired by biology, some of the known structure and functional properties of the human visual system will be presented.

2.2 Hierarchical structure

A *deep hierarchical* structure was proposed as the most suitable model to reflect the various processing stages of the visual cortex [23]. This layout allows for cells in the higher levels to share the elements of lower levels which contributes towards computational efficiency and generalization. The receptive fields (RF) are described as the area of each layer that receives the input visual stimulus. The lower layers are composed of RFs of small size and low complexity.

The cells in higher levels, which receive signals directly from the lower layers, contain receptive fields that gradually increase in size sizes (figure 2.1). This increase in RF size corresponds to higher complexity of features [23]. It was also described as a mechanism of reusing the computational building blocks in each of the layers [21]. Most biologically inspired vision models such as the HMAX [8], convolutional neural networks [37] and Hierarchical Temporal Memory [14] has achieved promising results in demonstrating invariance and selective properties of the visual cortex.

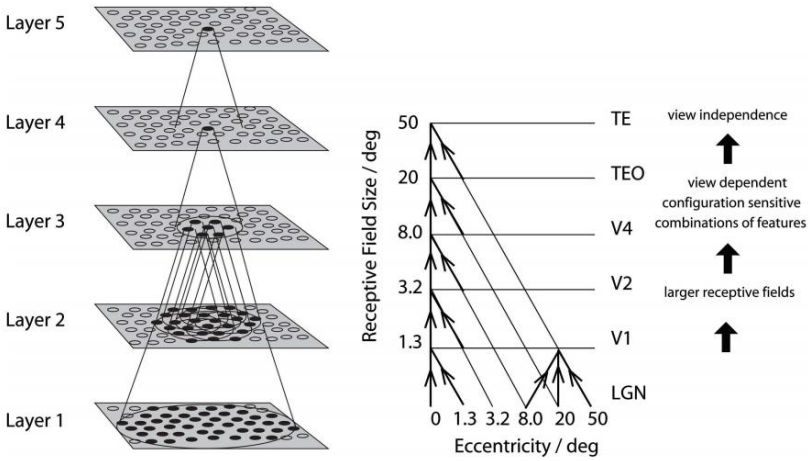


Figure 2.1: Layers of increasing receptive filed sizes (As implemented in VisNet)[1]

Due to its lack of feedback mechanisms and evidence of varying receptive field sizes and complexities in all the layers made it too simplistic to represent the visual pathways [21].

The Retina

The retina, which is considered to be apart of the central nervous system (CNS), forms the first stage of vision [38]. It consists of three layers of neurons connected by two intermediate layers of synaptic cells. Within its structure, Photoreceptors which are situated at the back of the retinal layers convert the incoming photons into electrical impulses. These photoreceptors include the low light sensitive rods and color sensitive cones. The three different types of cones correspond to the wavelengths of types red, green and blue. The different colour perception arises from the combination of these three cells which describes trichromatic vision [38][23].

The next layer that receives these converted electrical impulses are the Bipolar cells. The third layer is the Ganglion neurons which receives impulses from the Bipolar cells via the intermediate Amacrine cells. The output of the Ganglion cells transfer through the optic nerve into the Primary Visual Cortex via the LGN [38].

Lateral Geniculate Nucleus

The processed data from the retina enters the visual cortex through the Lateral Geniculate Nucleus (LGN) which acts as a form of relay as there is no spatial difference with the retinal ganglion cells [23]. Both retinal and LGN cells contain center-surround receptive fields which are linked to functions adapting to changes in luminance [23].

A computational parallel to this component was described by the Difference of Gaussian (DoG) operations, which has been applied in many of the biologically inspired attentional models [23].

2.2.1 Visual cortex

The visual cortex has a primary visual pathway, that separates into Ventral stream, which analyses object shape and Dorsal stream, which detects object motion [39].

The general structure of the Visual Cortex (VC) comprises of multiple processing layers, which starts from $V1$ or the primary visual cortex,which is the first layer that receives stimulus. $V1$ in sends data to the rest of the layers V2, V3, V4 all the way to the inferotemporal cortex IT. The IT contains the view tunes cells which are size and shift invariant [29].

Primary visual cortex

The primary visual cortex or the V1 is the most studied and modelled out of all the layers. The architecture of alternating simple and complex cells proposed by Hubel and Wiesel's study [40] has formed the foundation for many of the cognitive vision models.

The simple cells are tuned to a variation of factors such as size, rotation and position. The function of simple cells has been parametrized by two dimensional Gaussian [8] and Gabor functions [29]. The complex cells exhibit invariance properties and computationally, it has been described to have a similar function to that of a max pooling operation [8].

Colour coding cells are also present in the $V1$ area which forms $5-10\%$ of its total numbers [23]. It was found that the colour receptive fields could be learned by applying independent component analysis methods on natural colour images [32].

Retinotopy

The receptive fields of neighbouring cells in the lower layers of the visual cortex cover neighbouring areas of the visual field. Although this type of organization is said to occur in many areas, it is not guaranteed in the higher layers of the cortex [32].

2.2.2 Attention modulation

Eye movement and attention are known to form an integral component of object perception. It is described as a method for allocating resources for the visual tasks for dealing with the overwhelming amount of visual information that reaches the retina[35][41]. The brain therefore, reduces the information to gather only the most important portions of the data for further processing . An early pre-attentive mechanism is said to work in parallel to segment and categorize objects from a scene [42]. This aspect has been used to model a vision system that divides it into two processes and allows a feedback mechanism for selective spatial attention to influence the output.

The ARTSCAN, shown in figure 2.2, uses a dual structure of cortical stream such that formation of surface attention representation lead to the formation of an attention shroud in the *where* cortical stream, which in turn leads to learning

invariant object recognition in the *what* cortical stream [2] [41].

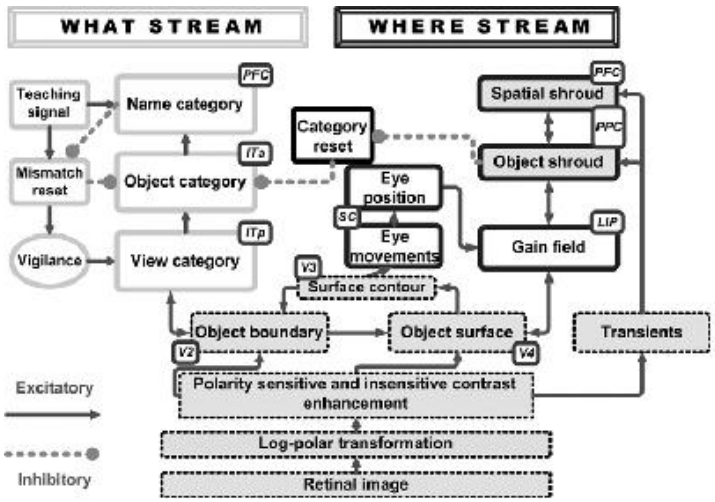


Figure 2.2: ARTSCAN system [2]

In the computational models for designing biologically inspired attention, a combination of two pathways were defined: a 'Bottom-up' mechanism that arises from the low level features such as detection of edges, colour continuity. This is occurs free viewing manner when the subject does not search for any particular object [35] [43]. The second is the 'Top-down' attention mechanism that is tied to the cognitive ability of the mammal. This directs the lower levels to adjust its response accordingly and drives the fast saccadic eye movements to focus on a particular region of interest [35].

2.3 Neuronal mechanisms underlying vision

The retinal ganglion cell's response was studied on different organisms to investigate how neurons represent and transmit data to the brain which resulted in patterns of action potentials or *spikes* [44]. The *neural code* is a sequence of such patterns that occur in the neuronal assemblies when they encounter sensory information [45]. It was found that the timing of the spikes of a single neuronal response is relevant for information transmission in the brain [44]. Sensory, cognitive and motor processes are said to result from parallel interactions among

large populations of neurons. The process responsible for linking this distributed activity in the visual system for identifying relationships among features in an image so that the object can be identified is called 'Binding' [42]. Observing the activity of clusters of neurons has shown patterns of spikes that occur repetitively. This neuronal oscillation has been suggested to play an important role in visual pattern perception.[46] According to the temporal correlation theory, an object is represented by the temporal correlation of firing activities of the distributed cells that detect different features of the object. The recording of oscillatory response of the brain by EEG has shown that a group of neurons exhibit synchronous oscillatory response when activated by visual stimulus [47]. Phase locking has been used for the detection of synchrony between different channels of visual task related EEG recordings. The channels correspond to the different areas of the cortex. Methods such as calculation of Phase Locking Value [48] have been used for finding synchronization between signals. Signal decomposition algorithms such as EMD for the analysis of brain signals has been crucial for finding more about vision related phenomenon and also for testing the response of simulated neuron models of vision.

With theories about formation of neural assemblies, it is important to take understand how spikes are generated. The Hodgkin Huxley model is the most well known model for the simulation of excitatory and inhibitory response of the neuron. It is in the form of an electrical circuit which is analogous to how the neuron fires when the membrane potential, due to stimulus, exceeds a certain threshold level [3].

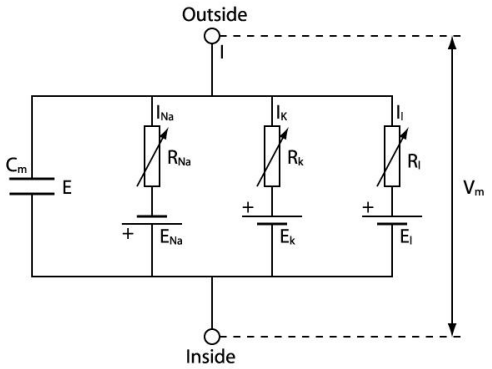


Figure 2.3: Hoddgkin-Huxley model of a Neuron [3]

The figure 1 shows the circuit model of a neuron where, following ohms law, the membrane potential V_m is given by,

$$C_m \frac{dV_m}{dt} = I - [G_k(V_m - V_k) + G_{Na}(V_m - V_{Na}) + G_l(V_m - V_l)] \tag{2.1}$$

Where, G_{Na}, G_k, and G_l are the conductances of the three ion channels (Sodium, Potassium and leakage channel respectively), I is the total current and E is the membrane potential [3]. This model was useful for simulation, but only for a limited number of neurons [49].

2.3.1 Sparsity of Neural response

Evidence of sparse activation of neurons has been supported by a wide range of experimental data [10][27][1]. Given an assembly of neurons, the number of neurons actually firing at any given instance is very low. So, when stimulus is applied, only a handful of neurons activate within particular area.

2.3.2 The neural code and perceptual grouping

In investigating the behaviour of neurons as a group, it was found that perception ties closely with the dynamic formation of neural assemblies and its synchronous activity. It was proposed that grouped features are represented (and distinguished from one another) by selective synchronisation of dynamically formed neural assemblies [50]. This was one of the possible explanations that related to the binding problem which questions how the distributed activity of the neurons leads to grouping of features. Another theory was the allocation of attention. It was argued that a pre-attentive object features could be formed according to gestalt principles with the formation of neural assemblies [50][42] .

2.3.2.1 Modelling Dual Processing Streams

It is evident that the processing of visual data in the brain occurs at low latency. Scenes and objects are perceived instantly which has been recorded to be around $100 - 150$ ms [51]. In [4], a hierarchical model of spiking neuron was described to demonstrate both the low latency and data processing in the mammalian cortex. It uses a latency encoding (with the data being sent with the low firing rate of the cortical neuron) and a temporal reference frame. The input starts a clock that generates a rhythmic oscillation. This provides a synchronisation signal to all

the cortical areas. The source of the oscillation is termed as the ILN (after the intra laminar nuclei) which is triggered with visual stimulus and the phase also changes within the different processing stages of the hierarchy. There is also a limited time constraint which does not allow too many feedback loops to occur. The model (in figure 2.4) architecture is in the form of minicolumns which contain two *cells* $A1$ and $A2$ that resemble spiking neurons. $A1$ receives only feedforward input and $A2$ receives both feedforward and feedback inputs.

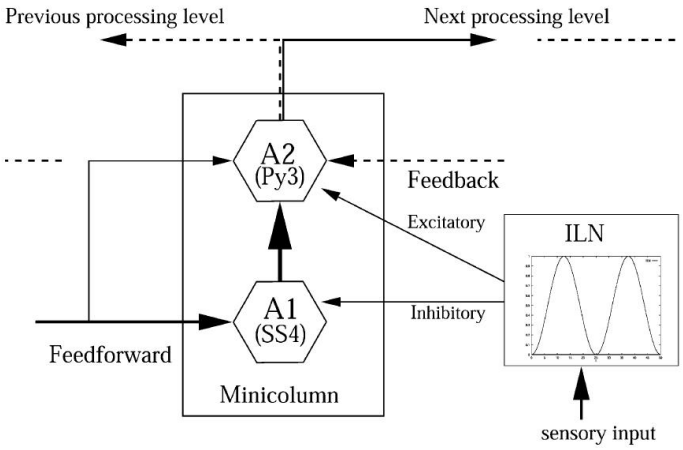


Figure 2.4: Structure of the minicolumn [4]

As seen from the figure 2.4, $A1$ only communicates with the rest of the system via $A2$. This model works corresponding to the clock input of ILN. Its implementation on invariant object recognition was demonstrated with a hierarchical vision model with each level comprising of an array of minicolumns. The first feedforward cycle was able to give correct recognition most of the time. The feedback loops were only used to suppress the error causing factors. The resulting output was a translation invariant recognition, tolerant to noise, mild changes to rotation and scaling and partial occlusion [4].

A detailed version of the neocortical architecture, which uses the columns arranged in layers, was described in [52]. The structure is defined by *minicolumns* [4], which mimics an elementary computational module of a neocortex and macrocolumns. These macrocolumns which contains the minicolumns share the same receptive field. The columns are arranged in layers similar to the visual cortex and has several processing level. The initial feedforward process forms an initial hypothesis at the highest level (which has a high probability of being

correct), which then initiates the top-down sequential refinement.

2.4 Summary

In this chapter, some of the studied mechanisms governing mammalian perception was listed along with the computational models that were adopted for simulating them. The important features include: Hierarchical structure, increasing complexity of receptive fields and receptive field sizes, sparsity of neuronal activation and bottom-up and top down attentional modulation. Based on these principles, many different versions of vision models were designed, which will be explored in detail in the next chapter.

The main drawback of current biologically inspired vision models is that they tend to highlight just one aspect of the biology. For example, the HMAX model [8] also only mimics the hierarchical structure and the simple and complex cell template. In terms of performance, deep convolutional neural networks have been able to replicate almost human level recognition, but aside from its hierarchical structure and number of neurons, there is little to no connection to biological learning process. None of these models incorporate any feedback connectivity and in terms of invariance and selectivity, they fall far behind in comparison.

To reach closer towards biological plausibility, it is essential to consider more than one aspect of the mammalian vision. The aim is to extract features with improved invariance and selectivity over previous models while also considering the properties of the visual cortex.

Chapter 3

Biologically Inspired Vision Models

Contents

3.1 Introduction

With the goal of understanding perception related mechanisms, many computational models based on the visual cortex were designed [53]. Although a universal vision framework that represents all the aspects of the visual process would be a more representative form of the cortex, most are generally geared towards a specific visual task such as object tracking, recognition or saliency. In this study, the focus is mainly towards object recognition tasks, which form the central component of all cognitive process. These models are based on the receptive field organization described by Hubel and Wiesel [40]. In their study, a hierarchical structure of alternating layer of simple and complex cells was described. Higher level cognitive functions are said to emerge from the selective and invariant properties of these cells, which increase in complexity along the layers [8].

In such types of biologically inspired models, input images are processed through these layers to generate an output feature vector. These models aim to achieve invariant response to all forms of transformations while also maintaining its selectivity towards a particular class of objects and its performance is based on its feature representation ability. The features are extracted in a hierarchical

manner, starting with low level edges and boundaries to higher levels of object descriptors. Evaluation of models such as [37], [54], [16] is usually performed by training the extracted features to classify objects into their respective categories using softmax or linear regression methods. In a different approach, the activation value of high level units were analysed for object detection in [13][10], which bears a closer resemblance to biology rather than supervised training of a classifier. The invariance and selectivity of these units determined the effectiveness of the learning algorithms.

In this chapter some of the hierarchical recognition models which were biologically motivated will be reviewed. The properties of these frameworks will form the basis for the new models in chapter 4.

3.1.1 Hierarchical models of vision

The selective firing of neurons based on the pattern of input stimulus been modelled in many artificial vision systems. The earliest such model is the *perceptron*, which has interconnected layers of neurons consisting of visible layer and hidden layer with a linear prediction function. The 'firing' of a neuron is dependant upon the comparison of the weighted sum of all the inputs to a threshold.

Similar in structure to the Perceptron, the earliest models based on Hubel and Wiesels architecture was proposed in 1980 by Fukushima, called *Neocognitron*, where its property of selective attention was applied for pattern recognition [5]. It was modelled according to the simple and complex cell layers of neurons in the form of feedforward, self-organising neural networks and demonstrated position invariance and tolerance for small amount of shape distortion.

In [55], the Neocognitron was further improved using bend detecting and line extracting cells. Although the system was robust when trained with unsupervised learning, supervised learning algorithms showed better results. A more recent modification includes a *Hypercolumn* model to increase its effectiveness for a more general set of images [56].

In conventional computer vision models, object recognition involves a process of feature extraction followed by classifier training or learning. The Scale Invariant Feature Transform or SIFT is one of such algorithms in which local feature extraction is performed [57]. Invariance to scale and rotation is achieved through a process of extrema detection, keypoint localization, orientation assignment[57]. In comparison, these neural network based models directly apply learning in which

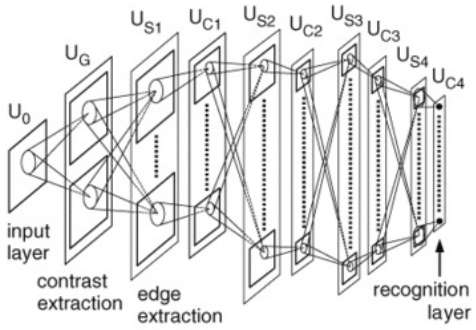


Figure 3.1: Neocognitron[5]

all the parameters of the layers are obtained with a back-propagation algorithm. In this way, biologically inspired models aim to obtain invariant descriptors without hand-crafted features which required overly complicated algorithms. The hierarchical models in this section are composed of neurons that are fully connected. Each neuron in one layer is connected to all the neurons in the next layer.

Convolutional Neural Networks and deep learning architectures

Convolutional neural networks (CNNs) are a type of hierarchical models also inspired by the Hubel and Wiesel's architecture of the visual cortex. But here, each neuron is connected only to a small number of afferent neurons in the previous layer. This type of local connectivity allows for a more efficient method for processing images as it greatly reduces the number of parameters to be learned. These models are also trained with supervised learning using backpropagation algorithms [21].

The very first convolutional neural network was described by LeCun [6], which and adopts a hierarchical structure with multiple layers. The input of each successive layer is a group of locally connected or neighbouring units of the previous layer. The weights of the units belonging to the same local group is shared which reduces the number of parameters and leads to position invariance.

A typical CNN includes a set of three fixed operations which are repeated throughout the model depending on the number of layers. The first component of the sub-layer is the convolutional layer, which spatially translates the input feature array by convolution with a linear filter. It is followed by a non-linearity

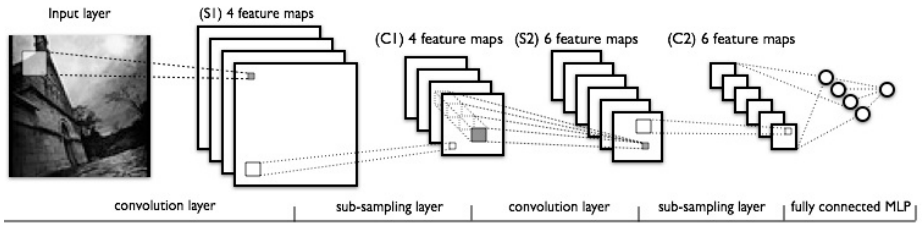


Figure 3.2: Convolutional Network model of LeNet-5l [6]

function operation. Most commonly used function is the Rectified Linear Unit or ReLU, which replaces all the negative outputs to zero. And finally, a feature pooling operation that performs max pooling or sub-sampling operation over the neighbouring units. Combination of these three steps are termed as the *convolutional* layer. The convolution step accounts for the selectivity whereas pooling gives rise to position invariance. This bears a similarity with the simple and complex cell layer like structure in the visual cortex which displays properties of selectivity and invariance respectively. In addition to encoding invariance, it also reduces the size of the image data and thereby making the operations in along the network more practical.

The LeNet-5 illustrated in figure 3.2 shows seven layers of the 3-level processing stages. Though it was initially applied for text recognition, it has been widely used in a number of applications such as face detection, video surveillance due to its computational efficiency and uniformity which allows a wide range of implementations [58]. After multiple stages of convolutional layer, the final layers are comprised of a set of fully connected neurons with multiple hidden layers and an output layer. With this, the models use a gradient based supervised learning method using backpropagation algorithm to learn all the parameters of the network.

Recent developments in this technology has displayed remarkable advancements in image recognition accuracy. Very deep architectures, with multiple convolutional layers stacked before the fully connected final layer have been found to display almost human level accuracy or even surpassing when tested with various databases [59]. These massive models contain a very large number of parameters

which results in higher performance for complex tasks and thus generally outperforms most other models [21]. Although these models are inspired by the general architecture of the brain, they do not specifically aim to mimic the structure and functional properties of the visual cortex [60].

In recent years, the benchmark for CNNs have been determined by the Imagenet Large Scale Visual Recognition Challenge (ILSVRC) [61]. It is based on the classification accuracy of the Imagenet database, which contains 1000 categories of images of over a million images [62].

Notable models include the Krizhevsky et.al.[7], which won the ILSVRC in 2012. This model was comprised of five convolutional layers and three fully connected layers trained with a stochastic gradient descent algorithm. Not all the convolutional layers applied max pooling and the total number of neurons added up to 650,000 with 60 million parameters. At the time, it achieved the highest accuracy in classification of the Imagenet dataset with an top-1 and top-5 error rate of 37.5% and 17.0% respectively on 1000 categories. Figure 3.3 shows the architecture, which was implemented using two GPUs. Each GPU was allocated to the top and bottom sections of the layers separately.

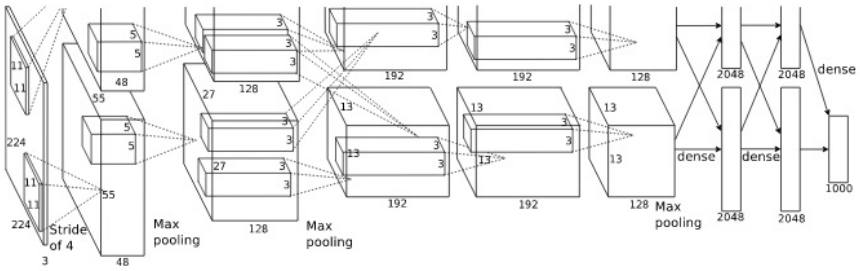


Figure 3.3: Krizhevsky,2012 model [7]

Improvements to this model was made by Zeiler and Fergus in [63], where the accuracy of image categorization was enhanced by changing the filter sizes and convolution strides within the layers. The filter sized were determined after applying an 'adaptive deconvolutional' operation which allowed visualization with the reconstruction of the input [64].

Another model similar to the Krizhevsky format is the Hybrid-CNN which was applied for scene text recognition. They used a hidden Markov model

(HMM) to form a hybrid CNN-HMM. Unlike previous techniques, where a segmentation operation around the text within a scene was performed prior to learning, this model was able to learn the text information directly [65].

The GoogLeNet [66], which won the ILSVRC 2014 had a 22 layer deep network and achieved an error rate of 6.67% on the testing and validation dataset. Efficiency was increased with a reduction in the total number of parameters by 12 times compared to the Krizhevsky model. Another difference was that it applied average pooling in place of max pooling operation within the layers. A characteristic of this model is the usage of 11 convolutions which forms a deeper network without adding more layers [66]. It also act as a dimension reduction measure. This model is an example of a class of very deep architectures where a large number of internal layers contribute to an improved accuracy levels [59].

The Deep Residual Learning model [67], which was the winner of ILSVRC 2015, had a depth of 152 layers. In [22], a Parametric ReLU method was applied to a very deep network which was one of the first models to surpass human level of accuracy in recognition.

One characteristic of these models is that it requires a large collection of training samples for the training process. Experiments in [63] showed that when a CNN trained with the Imagenet database with 1.3 million images was applied on the $Caltech-256$ dataset [68] (which contains a total of 30,607 images with 257 categories), its performance surpassed the CNN trained with the Caltech-256 dataset with a large margin. Another property is the supervised learning mechanism that requires all the training data to be pre-labelled. Although structure and performance-wise, they behave similar to the human visual cortex, the mechanism of learning spontaneously is not reflected in these models.

In [58], an unsupervised pre-training of the filter banks was proposed as a method for greatly reducing the number labelled samples required for training the network. A sparse coding algorithm was applied for learning the filter parameters. Since unsupervised learning from random input data is more in tune with the biological learning process, hierarchical models that adopt this type method serve as an important example in this field.

Self-Taught and Unsupervised learning models

Unsupervised feature extraction technique has been applied in the field of computer vision in the form of many different algorithms such as the PCA, ICA and

sparse coding techniques. One of the earlier unsupervised method implemented on a deep network scale was demonstrated by the Deep Belief Network (DBN) [69], where each layer learned features based on the statistical dependencies of the input in the previous layer. The learning process was based on maximizing the likelihood of the training data. An elementary unit of the DBN is the Restricted Boltzmann Machine (RBM) which is a type of undirected graph with binary states which contain a hidden and a visible layer. The DBNs are formed by stacking together these RBMs.

This model was later extended with the Convolutional Deep Belief Networks (CDBNs) [37], in which the weights between the layers are shared across all the locations of the input image. It applies a probabilistic max pooling method which accounts for its translation invariant response.

Adaptive deconvolutional model by Zeiler et. al. [64], was also proposed as an unsupervised feature learning model by applying a Predictive Sparse Decomposition (PSD) method. The output features were then trained with a classifier to determine its performance.

In [70], learning of high level features was demonstrated with a large collection of unlabelled data. The algorithm for this model was based on the simple and complex cell structure in which the first layer applied a linear filter bank and the subsequent layer encoded invariance by pooling operations. The simple cells in the form of sparse linear filters were learned by K-means clustering method and the complex cells were formed by agglomerative clustering which grouped the simple cells together. It was found that the simple cells in the higher layers of the model were highly selective towards human faces with this type of unsupervised training.

Similar architectures but using sparse algorithms such as ICA (Independent Component Analysis) and its extensions ISA (Independent Subspace Analysis) and TICA (Topographic ICA) were described in [12] and [13] respectively.

Among the many hierarchical biologically motivated recognition models, the HMAX model [71] also performs feature extraction in an unsupervised manner. Although hand-crafted filter designs are applied within the layers, the input data is always comprised of unlabelled data.

HMAX

A view-based hierarchical model of vision was proposed in 1999 by Poggio and Reisenhuber called HMAX. The name was coined due to max-pooling operations that occur within alternating layers to encode invariance. This feedforward model bears close resemblance to the structure of the primary visual cortex defined by Hubel and Wiesel formed with alternating layers of simple S and complex C cells [71][8].

The first layer called the $S1$ layer is based on the simple cells, which sensitive to low level features such as edges. It is comprised of an array of two dimensional Gaussian filters tuned to different orientations.

$$G_{xy} = exp(\frac{-(X^2 + \gamma_2 Y^2)}{2\sigma^2})cos(\frac{2\pi X}{\lambda} + \phi) \tag{3.1}$$

Where,

$$X = xcos\theta - ysin\theta \tag{3.2}$$

and,

$$Y = xsin\theta + ycos\theta \tag{3.3}$$

γ is the aspect ratio, σ is the effective width of the filter, λ is the wavelength and ϕ is the phase [72]. In a later modification, the Gaussian filters were replaced by Gabor filters as they allow for a more accurate orientation tuning [29].

The second layer called the $C1$ layer is modelled after the complex cells of the visual cortex. It performs a non-linear max operation on the outputs of the $S1$ layer such that the only the strongest value gets selected. The max pooling operation has been also be found to occur in the cortex [73][8]. To achieve size invariance, max pooling occurs over $S1$ cells of same orientation but varying spatial resolutions and to achieve position invariance, pooling is applied over neighbouring locations of each feature map.

After the $C1$ layer feature selectivity is performed by a simple cells of higher complexity. In the HMAX model in [8], the $S2$ units are formed by extracting patches or prototypes of $C1$ layer outputs. The $S2$ outputs are the result of template matching that is given by a Gaussian radial basis function,

$$R(X, P) = exp\left(-\frac{\|X - P\|}{2\sigma^2\alpha}\right) \tag{3.4}$$

Where X is the $C1$ layer outputs, P represents the $S2$ features or prototypes, σ is the standard deviation and α denotes the normalizing factor for the different

patch sizes [74]. The *C*2 layer encodes global invariance by max pooling over all the location and scales, forming a bag of features [74]. Due to this structure, this model was also described as spatio-temporal feature detectors of increasing complexity [39].

After *C*2, there can be a multiple number of *S* and *C* cell layers. In its implementation for object recognition in [75], four levels of simple and complex layers were applied, resulting in robust object recognition system. It was also applied for modelling a system for action recognition from a video sequence [39]. Figure 3.4 shows a common hierarchical model from [8] that illustrates how each layer contributes to the invariance to different transformations.

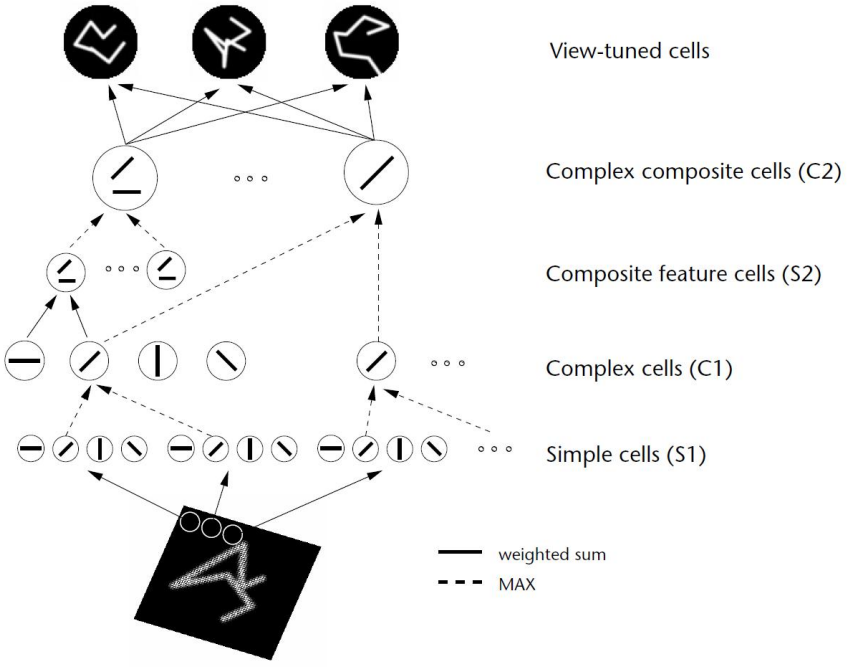


Figure 3.4: HMAX model: Hierarchy of alternating simple and complex cells forming view tuned cells [8]

The HMAX model has undergone many different modifications following its initial version in [8]. In [29], the Gaussian filters in *S*1 layer were replaced by Gabor filters. The motivation behind this adjustment was due to its similarity to physiological data and the number of free parameters that allows more accurate tuning [29]. In [54], application of this model was demonstrated on a series of recognition tasks. The extracted *C*2 features were highly robust in binary class

object classification.

In [72][74], sparsity was introduced in the $S2$ inputs such that only the most dominant orientation of the $C1$ unit is captured. Additionally, lateral inhibition was applied which suppressed the $S1$ and $C1$ outputs. Based on the findings that $V4$ and the IT neurons are selective to a range of scales and visual field [74], the invariance property of the $C2$ units were also limited by applying a localized pooling of features. This type of pooling was also described as 'attending' towards a particular region of the visual field.

Further improvement was demonstrated in [16], where localised pooling method was adopted for the high level filter responses (in this model, the $C2$ or $L4$ outputs). The model comprised of four alternating layers of convolution and pooling. In the first layer, Gabor filters of multiple orientations and scales were convolved with the input images. This was followed by the local max pooling operation at the same orientation maps in the $L2$ layer. Adjacent scale maps were also pooled to achieve a degree of scale invariance. Patches of $L2$ were extracted to form High Level (HL) filters, which were convolved with the $L2$ outputs. In the final layer, a combination of spatial pyramid [76] and localised pooling was applied. In this method, max pooling over various locations was performed by using concentric search regions of various sizes. The location and scale of the pooling region was also encoded in the final feature vector. It was mentioned that depending on the complexity of the input images, the number of best suited required to represent the features varied. With this model, the classification accuracy on multiple categories of objects improved significantly over the previous HMAX [74]. With increase in scale resolution of the pooling, the classification accuracy was further improved in [11].

Some limitations that were identified for this model include off-line learning and lack of feedback which made it too simplistic to account for the complexities of the cortical neurons [77]. In [77], to account for feedback mechanisms, which are considered as an important aspect of the visual cortex that can modify $V1$ layer responses [78][79], a feedback mechanism for HMAX model was developed based on Bayesian networks and belief propagation. A theoretical framework for hierarchical Bayesian system was proposed in [80]. They described a feedback process where the lower layers would get updated with the influence from higher layers till the system reaches an equilibrium state. A hierarchical model for pattern recognition was demonstrated in [9] where a conditional probability

distribution matrix is calculated for the lower level modules that is updated as the learning process is repeated. Figure 3.5 shows the Hierarchical Bayesian Network in which each node contains a probability distribution [9].

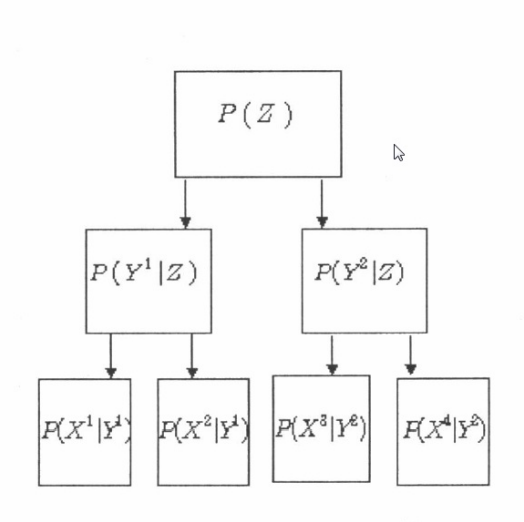


Figure 3.5: Bayesian Hierarchical model [9]

Each mid-layer node X of the model is influenced by its parent nodes and children nodes. This dependency between the nodes defines X to be the joint probability of bottom-up beliefs $\lambda(X)$ and top-down beliefs $\pi(X)$ [77].

$$Bel(X) = \alpha.\lambda(X).\pi(X) \tag{3.5}$$

Where, α is a normalizing constant. Due to the computational cost of belief propagation, a loopy belief propagation model was applied to approximate the HMAX functions. Based on these principles, a Bayesian network was designed where each node represented the features encoded at a given location and layer. The probability distribution over the possible states or orientations was calculated to determine the response of the nodes. Finally, conditional probability tables were used for linking these nodes to parent nodes in the next layer, which also provided an approximation of the max-pooling operation of the complex cell layers of the HMAX. One drawback with this model is the computation cost. Even with more optimised belief propagation, its scalability was limited and could not be applied for large datasets [77].

In all the above HMAX models, the first layer of HMAX contained hard coded

Gabor filters corresponding to realistic parameters of of the visual cortex cells [10][29]. Unsupervised learning is usually reserved for the higher levels. In [10], an unsupervised method based on the natural statistics of input images was adopted for learning all the layers of the HMAX model (figure 3.6). In addition, the applied learning techniques also modelled the sparsity of neuronal activation which is said to characterize the response of all the layers of cortical cells [81]. The learning methods included the sparse coding optimization algorithm and the equivalent independent component analysis (ICA). A similar model was also designed in [28], where two layered sparse coding model was trained for invariant and discriminative feature detection. Higher order dependencies were modelled by pooling over local regions to generate two sets of codebooks. In an expansion of this model, the sparse HMAX in [10] stacked multiple layers of feature selectivity and pooling.

Each *S* layer was followed by max pooling operation over slight variation of spatial positions for shift invariance in the *C* layer. The final feature vector was obtained by applying spatial pyramid max pooling [76] over the final *S* layer outputs. It was reported that even without pooling over different scales, this model performed improved object classification than the original HMAX and also learned object specific *neurons* similar to [13] but with much lower computational resources.

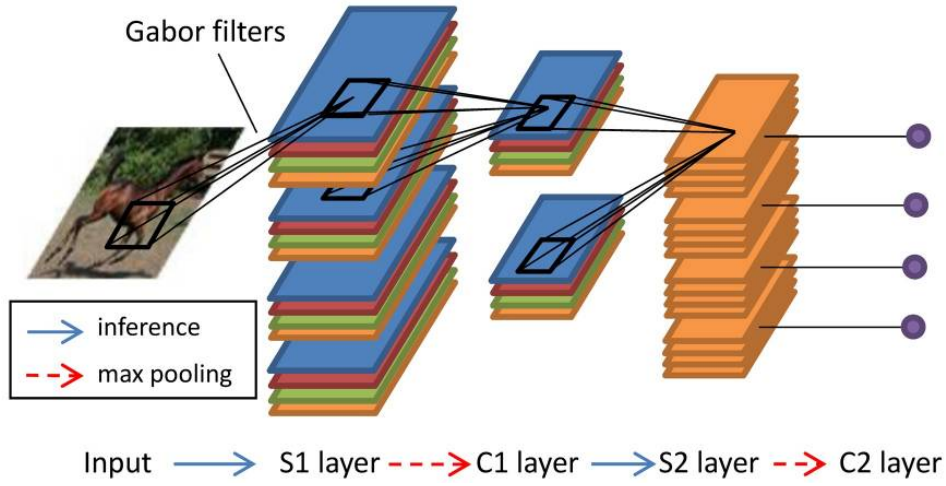


Figure 3.6: Sparsity regularised HMAX model [10]

The *S*1 layers were learned from unsupervised algorithms from random

patches of the input images. These units resembled Gabor-like edge detectors whose response within neighbouring spatial locations were pooled in the next $C1$ layer. The sampled outputs of $C1$ formed the training set for learning higher order feature detectors in $S2$, with either sparse coding or ICA. In this manner, multiple layers were formed with increasing complexity of receptive fields along the hierarchy. In he final C layer, features were extracted by applying spatial pyramid max pooling over the outputs. Even without applying scale invariance functions, this model achieved a high classification accuracy compared to most other models. When the activation values of the high level units were analysed, they were found to be highly selective to object category and also displayed a degree of invariance towards translation, rotation, scale and occlusion. Their study also demonstrated the advantage of max pooling in comparison to average pooling since it introduces linear higher-order dependencies among filter responses at different positions which leads to learning high level features (although some other methods such as square pooling are also known to share this property [13][12][31]) [10].

A new enhancement of the feedforward HMAX model was proposed in [82] where bottom-up saliency maps were integrated for learning high level prototypes. This type of attentional modulation directed the sampling of patches towards the regions of high saliency and reduced the data redundancy. The patches were then selected and classified into separate clusters based on their similarity with an unsupervised iterative algorithm. This method was adapted to the memory processing property of the $V2$ layer and the distributed regions of the IT [82]. Higher classification accuracy than the original HMAX model in [54] was reported for this model for a smaller set of training sizes, but its performance in comparison with the newer HMAX models were not determined.

All the HMAX models described in this section apply a different strategy of pooling at the final layer (figure 3.7). In [54], [75], global max pooling over all the locations was applied, in [74], localised regions were pooled, in [10] and [16], spatial pyramid pooling was implemented and in [16],[11], localised pooling with multiple resolutions were applied. Among these methods, the models with spatial pyramid pooling were found to perform with higher accuracy than the rest as it allows for more dense representation of features [76][10][83][11].

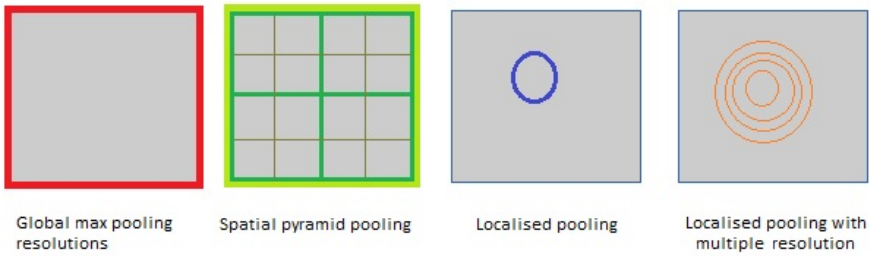


Figure 3.7: Pooling strategies the HMAX models [11]: the coloured boundaries describe the pooling areas that are applied in the above models

Sparse hierarchical models

In recent years sparse representation has been widely applied in vision related models [84][85]. Apart from the sparse-HMAX model in [10], many other models have implemented sparsity in a hierarchical manner including the two layer architecture in [28] and [86]. Models in [13], [87], [88][12] learn the simple layer units with ICA related algorithms. These models extract features from the natural statistical properties of images [32].

In [12], Independent Subspace Analysis (ISA) was applied for learning multiple S layer units for an action recognition system in a convolution and stacking architecture (figure 3.8). ISA is an extension of the ICA algorithm that classifies the units into groups or subspaces according to their dependencies [33]. In this model, phase and position invariance was achieved by square pooling over responses of simple cells within a subspace. Features invariant to local translation and phase was learned by pooling over responses within subspaces. Convolution resulted in faster computation time which was essential for processing the high dimensional data but it was described as biologically less plausible as the parameters are shared across all the locations [89][13]. However most convolutional neural networks as well as HMAX models share this type of shared parameters. The invariant spatio-temporal features achieved a high classification accuracy for action recognition datasets. A higher efficiency in training time was also achieved by preprocessing the output of each layer with PCA, which performs both compression and data whitening, before applying ISA.

The Topographical ICA (TICA) [30] was applied in a hierarchical model in [13] to build a large scale object detection system (figure 3.9), described as a sparse

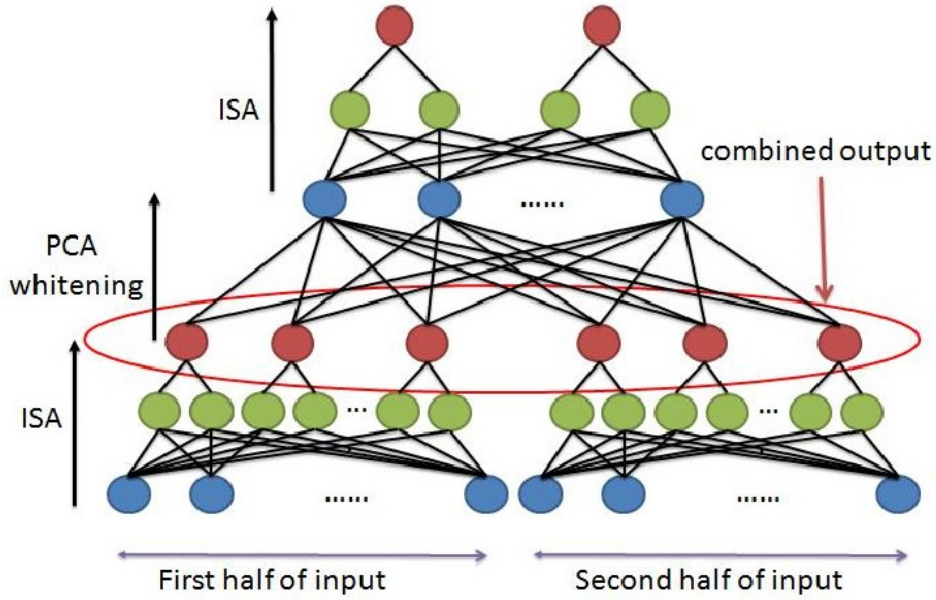


Figure 3.8: Convolutional and stacked ISA network in [12]

deep autoencoder. TICA is another extension of the ICA in which dependencies decrease with increase in spatial distance within the topography. In this unsupervised learning framework, a nine layered locally connected sparse autoencoders were trained using a reconstruction based TICA algorithm [90](for generating an overcomplete set of units). This model follows the architecture of the Tiled CNN described in [89] which differentiates it from other models. The receptive fields are localised such that each unit only observes only a portion of the input data. The parameters are not shared across all the locations of the image. In addition to biological plausibility, its claimed to be able to learn more than just translation invariances. The final layer contained class-specific neurons corresponding to the *face neurons* in the IT. The model applied a reconstruction based TICA algorithm [90] to learn overcomplete features (where number of features are greater than the dimensions of the input).

Invariance was achieved through square pooling the responses of dependent or neighbouring units. The final layer formed high level 'neurons', which were measured for object detection against a set of distractors. The best neuron for face detection achieved an accuracy of 81.7% for a dataset of 37,000 (comprised of ImageNet [61], and Labeled Faces in the Wild dataset [91]) images containing 13,026 positive samples. The neurons also exhibited robustness towards rotation

and scale variations. With these results, this model displayed high performance in object detection and invariant response however, its accuracy in classifying multiple categories of objects were not reported.

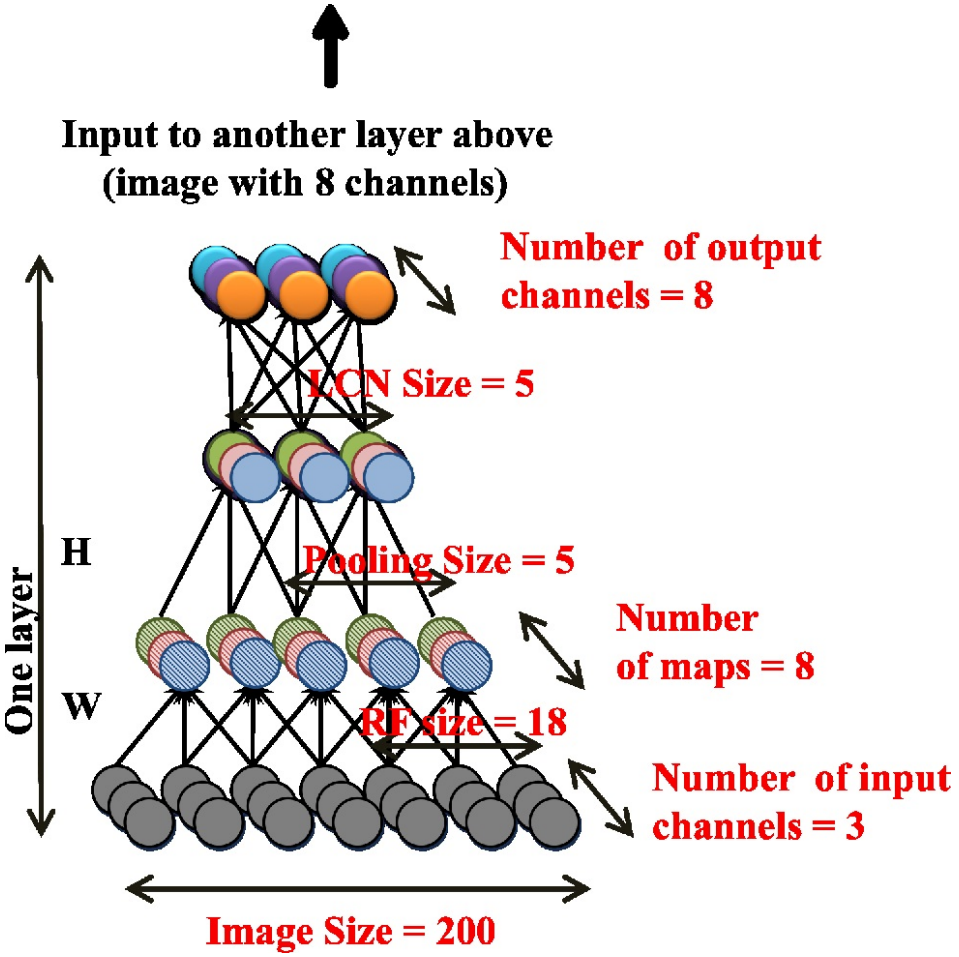


Figure 3.9: First layer of the architecture in [13], comprised of localized receptive fields

In both of these models, invariance was achieved by pooling over overlapping neighbourhood of features from the previous layer. Here, hierarchy, invariance, selectivity and sparsity is demonstrated, yet they are structurally different from the HMAX models.

Instead of localized receptive fields, as in [13], the *S* layer response in the HMAX is in the form of a hyper-column which contain feature maps for all the possible locations and scales [60]. Also, shift invariance is achieved by max pooling of neighbouring locations of the same feature map. Recently, an unsupervised

version of the sparse HMAX was developed, in which all the S layer units were learned from natural images using sparse coding and ICA [10].

The list of models that have applied sparsity based learning algorithms have displayed highly accurate results in image classification and detection. Its added advantage biological plausibility, supported by experimental data [81] makes it an important technique for designing cognitive models.

3.1.2 Temporal models

In these models, encoding of invariant features is based on the temporal as well as spatial statistics of the data. They demonstrate a self-organising learning pattern which is said to occur in the brain [1].

The Hierarchical Temporal Memory HTM [14] is a memory and time based machine learning technology which was developed according to a converging hierarchical version of the neocortex. This tree shaped network was based on the Hierarchical Bayesian model that was implemented in [9]. Inference and prediction is made by observing the temporal sequence of images that display similar patterns over a period of time. The model was extended to mimic the columnar nature of neocortical neurons in [52] where in each layer, the cells or 'neurons' were grouped into columns such that all the cells in a column would share the same receptive field. The memory allocated to each unit decides the complexity of pattern learnt by each layer. Here, learning, inference and prediction occurs in a continuous manner.

The HTM is characterised by its spatial and temporal pooling functions. Spatial pooling sends feedforward information to the next layer where patterns that are spatially similar are pooled together. The propagation of information occurs in the form of 'activation' of columns. A sparse distributed representation ensures that only a percentage of columns or units are active in a layer at a time. The neuron with stronger activation suppresses the neighbouring neurons with weaker activation [14].

Temporal pooling groups together patterns that follow each other in time. The previous input is used to form representations of the current input within a layer. It also performs prediction for the next time step which involves the formation of connections 'synapses' with neighbouring cells of the same layer with active cells form connections with previously active neighbouring cells. This is done by adjusting the weights assigned to each connection which range from 0 to 1

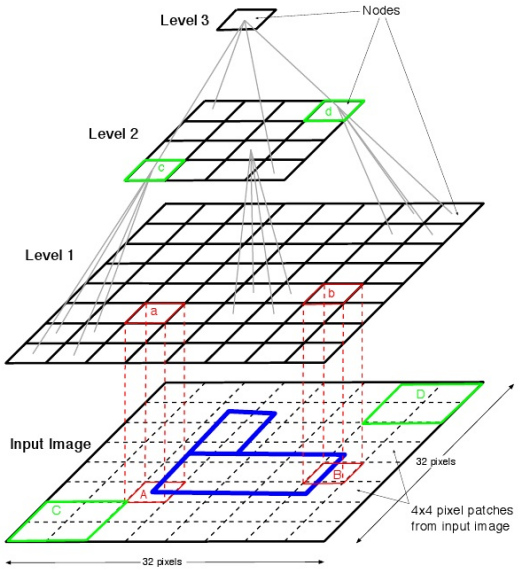


Figure 3.10: Hierarchical Temporal Memory[14]

depending on the strength of the connection [14].

Apart from recognition tasks, the HTM has been useful specially in applications that include a temporal variation of data. For static vision models, single-cell columns were found to be sufficient [14]. Higher number of cells are useful for predictions and representations based on previous input. This method has been found to be highly efficient in a vast variety of applications and closely mimics the neocortical architecture but fails to incorporate any feedback mechanism that occurs in the visual cortex [77].

Another temporal based learning mechanism was proposed in [92] called spike timing dependent plasticity (STDP). This feedforward hierarchical architecture, models the ventral stream of the visual cortex. The components here are represented in terms of neurons and its spiking response. In this model, the connections between the units or synapses are given weights depending on the frequency of the input stimulus.

This type of self-organised learning with synaptic reinforcement through temporal and spatial continuity information was also integrated into the *VisNet* framework in [1].

A similar extension to the HMAX model was proposed in [93], where connectivity between the most active units were strengthened based the temporal

sequence of images. The temporal component is not included in most of the frameworks described in this chapter. But for designing a more biologically realistic model, it would be advantageous to incorporate such mechanisms into the learning process.

3.2 Summary

In this chapter, all the hierarchical models which were described were in some manner, inspired by biological vision. Since a large part of the functional properties of the brain remain a mystery, most of these models are mainly based on the hierarchical template of the initial simple and complex layers of the visual cortex.

Since there is even less similarity between supervised learning method for the CNNs and biological vision, the focus will be on improving unsupervised hierarchical models in the next chapters. Although not all aspects of the visual cortex are considered, the aim is to improve its classification accuracy towards the range exhibited by large scale CNNs.

Chapter 4

Sparse hierarchical vision models

4.1 Introduction

In this chapter, an unsupervised hierarchical model of vision is proposed using sparsity-based algorithms: Independent Subspace Analysis (ISA) and Topographic Independent Component Analysis (TICA). The new model is inspired by the unsupervised hierarchical feature learning described in [13],[12] and the HMAX framework and its extensions [8][54][10]. It extracts high level features from a set of unlabelled data which are then classified using linear regression. Here, each layer of the hierarchy involves three stages of processing: Linear filtering, L2-pooling and max pooling.

Application of ISA and TICA on natural images has led to the emergence of complex cell properties of phase invariance [31][30]. Therefore, with its properties, features extracted from the model exhibit a high degree of invariance and selectivity which is demonstrated by an improved classification accuracy in comparison to unsupervised models such as [10] and [16]. In addition to object classification accuracy, it also reduces the dimensionality of the data at each of the layer outputs. Since the simple and complex layer template of the HMAX model is replicated in these models, they are referred to as ISA-HMAX or TICA-HMAX in the following sections.

4.2 Sparsity-based algorithms and invariant feature representation

Evidence in various studies in neuroscience suggest that sparsity of response occur in all layers of the visual cortex [81][94][1][10]. The non-Gaussianity in natural data was first represented in terms of sparse coding by Olhausen and Fields, in which an image was represented by linear combination of very small number of non-zero features [25]. The independent component analysis (ICA) generates

features similar to sparse coding but they are statistically independent [33][32]. In an extension to the ICA, the independent subspace analysis (ISA) and topographical independent component analysis (TICA) was developed in which the components were grouped according to their energy dependencies [31][30][32]. Since maximizing sparsity is equivalent to maximizing independence [31][25], ICA, ISA and TICA has been used as an alternative to sparse coding in many models [10][13][12]. Recent examples of ISA or TICA based hierarchical models include a deep learning frameworks for action recognition in [12], where a convolution and stacking method was adopted and [13], where a multi-layer model with pooling and local contrast normalization was built to simulate a large scale feature detection by training with unlabelled data.

The performance of these models greatly depends on its invariant feature representation which is generally achieved with a non-linearity function. In the convolutional neural networks, HMAX, and its sparsity regularized extension [10], translation invariance was achieved by a max pooling function over neighbouring locations on a feature map. Biological plausibility of max pooling was also supported by studies that discovered similar functions in the V4 area of primate visual cortex and complex cells in cat visual cortex [29]. In the self-taught learning models described in [13] and [12], L2-pooling function over the feature maps was applied. Additionally, the original HMAX models also encode scale invariance by max pooling over features of same orientations and positions but slightly different spatial frequency [29]. Some recent convolutional neural networks have also extended scale invariance into their model [95]. As evident from these models, there is always an aim to learn more than one type of invariances.

It has been stated that phase and position invariance are rather closely related to each other [32]. Changes in phase for a spatially localized stimulus translated into very small shifts in position (in the direction of its oscillations) such that it was termed as a special case of position invariance. Complex cell properties of the ISA and TICA therefore, displayed phase invariance and limited shift invariance [31]. To obtain high level features with improved classification accuracy, both L2-pooling and max pooling is applied in the proposed models in this chapter.

Sparse coding

The emergence of Gabor-like filters similar to the $V1$ simple cells by maximising sparsity was demonstrated by the sparse coding optimization technique [27].

With a given sample of natural image patches, these Gabor-like filters or bases are formed by solving a convex optimization problem in which the cost function is minimized with respect to an $l1$-norm regularization term as in equation 4.2.

Here, the input vector **x** is expressed in terms of a linear combination of basis functions a_i and coefficients s_i as,

$$x = \sum_{i=1}^{k} a_i s_i = AS \tag{4.1}$$

For a set of m input vectors, the optimization is carried out by the given cost function as,

$$minimize \sum_{j=1}^{m} ||x - \sum_{i=1}^{k} a_i s_i||_2 + \lambda \sum_{i=1}^{k} |s_i|_1 \tag{4.2}$$

To ensure that the s_i terms are mostly zero, the $l1$-norm regularization term is used. The term $\lambda \sum_{i=1}^{k} |s_i|_1$ imposes the sparsity penalty on the cost function, λ is a positive constant.

Independent component analysis

Like sparse coding, independent component analysis also learns filters or bases that are localized in space, frequency and orientation [32]. In ICA, the bases are constrained to be independent with respect to each other. From equation 4.1, given the matrix **S** is orthogonal,

$$A = S^{-1}X = WX \tag{4.3}$$

One of the popular methods to learn the bases and weights are is by the maximum likelihood estimation of the observed data [32]. Given $W = (w_1, ..., w_n)^T$,

$$maximize \sum_{t=1}^{T} \sum_{i=1}^{k} logp_i(w_i^{T_j} x_i) + Tlog|detW| \tag{4.4}$$

Where p_j is the probability density function and $log|detW|$ is the orthogonality constraint.

In [32], ICA estimation by maximizing sparsity (instead of independence) was termed as a special case of maximizing the non-Gaussianity of natural images.

Subspace and Topographic ICA

The independent subspace analysis (ISA) and topographic independent component analysis (TICA) are two extensions of the ICA that relaxes the independence constraint and arranges the bases according to the strength of their higher order correlations [32]. This type of grouping has led to the emergence of complex cell property of phase invariance [30][32].

From equation 4.3, if two components $s_i = z_i\sigma$ and $s_j = z_j\sigma$ are defined as uncorrelated then $cov(s1, s2) = 0$, where σ is a common variance variable and z_i, z_j are zero mean and unit variance independent components. However, uncorrelated components does not indicate independence [32] since correlation of their squares is positive.

$$cov(s_i^2, s_j^2) = E\{s_i^2 s_j^2\} - E\{s_i^2\}E\{s_j^2\} \neq 0 \tag{4.5}$$

$$cov(z_i^2\sigma^2, z_j^2\sigma^2) = E\{z_i^2 z_j^2\sigma^2\sigma^2\} - E\{z_i^2\sigma^2\}E\{z_j^2\sigma^2\} = E\{\sigma^4\} - E\{\sigma^2\}^2 \neq 0 \tag{4.6}$$

This type of dependency is also termed as energy correlations between two components [30][32].

Topographic independent component analysis (TICA)

In TICA, the arrangement of the learned units is in a way such that it reduces the distance between correlated components and thereby reducing the *wiring length* between two statistically related neurons. In neuroanatomy, is explained as the length of the axons that connects the neurons [30]. This minimization of wiring length has been described in [32] as a model for the compactness of the brain volume and speed of signal processing.

Here, the grouping of dependent components is determined by neighbourhood function that defines the topography. Proximity within the topography indicates strength of its second order correlation.

The generative model for the TICA is defined by the *joint density* of the $S \in \{s_1, ..., sn\}$ components from equation 4.3. The variances of the components of S are generated, from which the components (z_i, z_j) are derived independently.

The arrangement of the components with the associated variances σ are defined by a two dimensional neighbourhood function $h(i, j)$ with neighbourhood

area of width m, which is defined as a monotonically decreasing distance measure as defined by equation 4.7.

$$h(i,j) = \begin{cases} 1 & \text{if } |i-j| \leq m \\ 0 & \text{if } otherwise. \end{cases} \tag{4.7}$$

The variance of a component σ_i is then defined in terms of the neighbourhood function as,

$$\sigma_i = \phi(\sum_{k=1}^{n} h(i,k)u_k) \tag{4.8}$$

Where ϕ is a scalar non-linearity function and u_k are the independent components.

The components s_i can then be defined as,

$$s_i = \sigma_i z_i = \phi(\sum_{k=1}^{n} h(i,k)u_k)z_i \tag{4.9}$$

The components, s_i, s_j are uncorrelated but their energies s_i^2, s_j^2 are correlated (equation 4.6). As in the case of ICA, $W = (w_1, ..., w_n)^T = A^{-1}$ (from equation 4.3, the bases and weights can be learned by the maximum likelihood estimation of the cost function,

$$logL(W) = \sum_{t=1}^{T}\sum_{j=1}^{n} G(\sum_{i=1}^{n} h(i,j)(w_i^T x)^2) + Tlog|detW| \tag{4.10}$$

$$e_j = \sqrt{h(i,j)(w_i^T x)^2} \tag{4.11}$$

The $h(i,j)(w_i^T x)^2$ represents the energy of a neighbourhood or the complex cell output, and G is a scalar function [30].

The figure 4.1 shows the components derived from a set of natural images using TICA. The input was a set of 50000 samples of 14×14 sized square patches extracted randomly from the Kyoto dataset. It shows a total of 196 bases where a neighbouring function determines the dependence of nearby components. The bases within a neighbourhood of size 3×3 exhibit very slight variations in orientation, frequency and position, but are varied in phase. As described in [32], most of the components are of high frequency range, with some low frequency *blobs* that are grouped together. When TICA is applied to colour images (in figure 4.2), the colour components are grouped into the low frequency clusters The gray scale high frequency edge detectors surround the low frequency components in the topographic map.

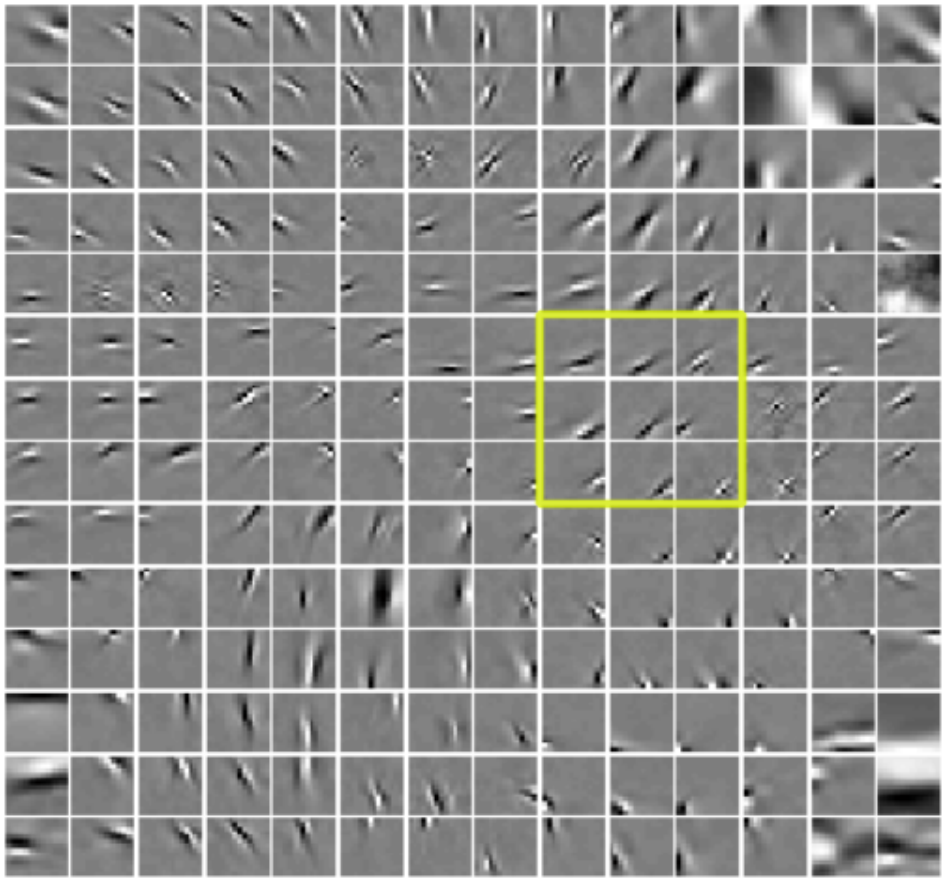


Figure 4.1: Bases learned by TICA on natural images (*the yellow box indicates the size of the neighbourhood function*)

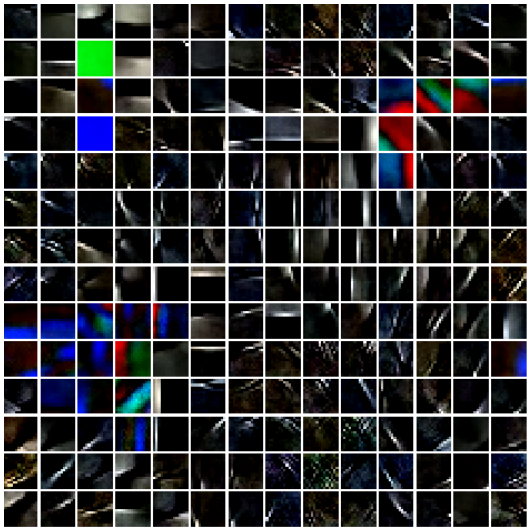


Figure 4.2: Bases learned by TICA on colour images

Independent subspace analysis

In independent subspace analysis (ISA), the dependent components are grouped into a subspaces of pre-defined size. The neighbourhood function in this case is defined as,

$$h(i,j) = \begin{cases} 1 & \text{if } \exists q : i,j \in S_q \\ 0 & \text{if } otherwise. \end{cases} \tag{4.12}$$

Where component $S \in \{s_1, ..., sn\}$ from 4.3 is divided into n-tuples such that the s_i inside a tuple are dependent on each other. $q \in \{1, ..., q\}$ is the index of the n-tuple. S_q represents the set of indices of the component s_i that exists within that tuple [31]. Equation 4.1 becomes,

With $W = (w_1, ..., w_n)^T = A^{-1}$, the cost function for maximum log likelihood estimation in this case is given by,

$$logL(W) = \sum_{t=1}^{T} \sum_{q=1}^{Q} G(\sum_{i \in S_q} (w_i^T x)^2) + Tlog|detW| \tag{4.13}$$

Where, $(w_i^T x)^2$ is the energy term, and G is a scalar function.

The total response of each subspace is the squared sum of each component,

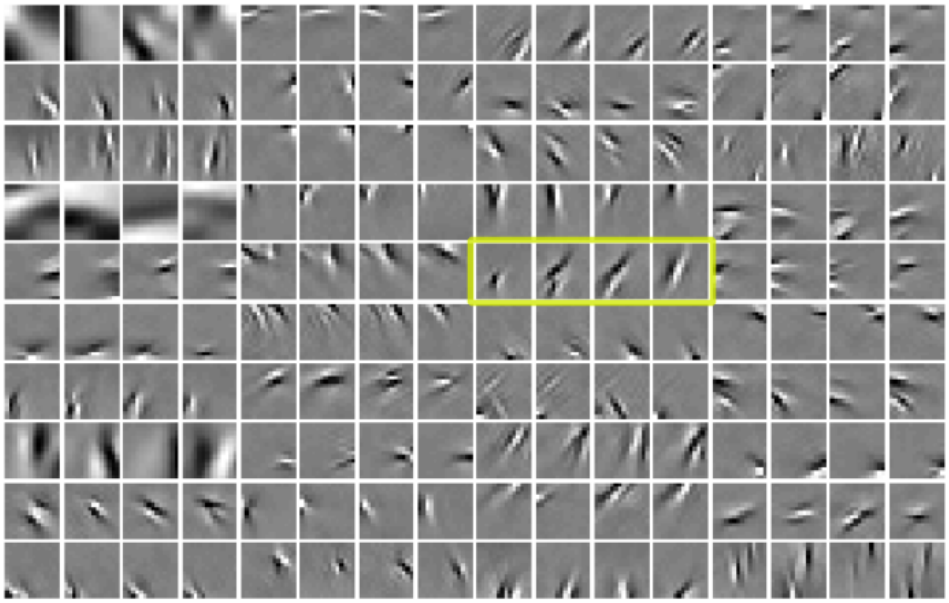


Figure 4.3: ISA on natural image samples (*the yellow box indicates the subspace of size 4*)

also termed as the L2-pooling.

$$e_q = \sqrt{\sum_{i \in S_q} s_i^2} \tag{4.14}$$

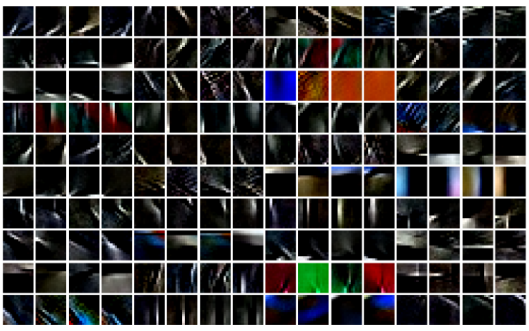


Figure 4.4: ISA on colour image samples

The figure 4.3 shows the components derived from a set of natural images from the kyoto dataset using ISA. The input was a set of 50000 samples of 14×14 sized square patches extracted randomly from the database. It shows a total of 160 bases where each subspace is comprised of 4 filters. Units within the

same subspace are localized in orientation, frequency and position (with slight variation) while having different phases. Similarly, for colour images as shown in figure 4.4, all the colour components, with different phases, are classified into a single subspace.

4.3 Image processing before unsupervised learning

The bases in figures 4.3 and 4.1 correspond to the simple cells of the V1 area in the visual cortex. Their response is highly selective towards orientation, spatial frequency and position. The complex cells pool over multiple feature selective simple cells to exhibit invariance. In the higher layers, as the complexity of features increase, there is also an increase in invariance which is attributed to the increase in receptive field size [60]. In the models based on visual cortex, to replicate this increased complexity, samples from all the features of the previous layers form the training set for the next layer. The simple and complex cell layers in the HMAX and similar models are based on the primary visual cortex, but visual signal processing in the mammalian vision starts before that in the retina and the LGN [23].

Contrast Gain Control

The contrast gain control(CGC), with its purpose to simplify the statistical structure of images by whitening and divisive normalization are operations that has been compared to functional properties of the retina and LGN based on physiological evidence [96]. With data whitening, the second-order information or correlation within the data is removed. Common method of whitening is the PCA, which also performs functions such as dimension reduction and anti-aliasing [32].

CGC is applied by dividing the whitened image patches with its variance. This reduces the dependencies between the components which is useful for ISA and TICA algorithms. As described in [32], for two uncorrelated (but not independent) components, defined as $s_i = z_i\sigma$ and $s_j = z_j\sigma$, where σ is a common variance variable and z_i, z_j are zero mean and unit variance independent components, equation 4.1 becomes,

$$X = \sigma \sum_{i=1}^{k} a_i z_i \tag{4.15}$$

Assuming that the variance $\bar{\sigma}$ of each patch is almost equal to the global variance

σ, normalization is carried out by dividing the image patch with is variance $\bar{\sigma}$. So, practically, it does not entirely eliminate but reduces the variance dependencies.

$$\bar{X} \leftarrow \frac{X}{\sigma + \varepsilon} \tag{4.16}$$

Where ε is a small constant for preventing division by zero [32].

The necessity of CGC was demonstrated in [96], where the subspace sizes in the ISA algorithm was estimated by maximizing the likelihood with respect to subspace size and pooling non-linearity. Without CGC, the strong dependencies increased likelihood of almost all the bases to be categorized into one large subspace. Thus, it is an important step before learning the dictionaries.

Max pooling over neighbouring positions of the same feature map introduces positive correlations between the ICA components [10]. Figure 4.5 shows the correlation coefficients of the S and C layers of an HMAX model (similar to [10]), where the bases were generated by ICA. The input was a random image, which was processed by linear filters obtained by ICA. The outputs of the ICA filters represents the S layer. Over each of the outputs, max pooling was applied over neighbouring values which represents the C layer.

The red histogram shows the coefficients between the response maps of all the filters in the S layer of the model. The S layer outputs are observed to be highly uncorrelated (with the average correlation coefficient close to zero). For the C layer outputs, the size of pooling area p on the feature maps was varied. The histograms (except red) show the effect of applying the max pooling function on the S layer outputs. The average positive correlation increases with increase in pooling area. This analysis was also presented in [10], where it was demonstrated that max pooling non-linearity produced second order linear interactions among the filters .

Figure 4.6 depicts the correlation coefficients of the first and second stages of the ISA model (in red and blue histograms respectively) after applying CGC. Its response shows the effects of applying the non-linearity function defined in equation 4.14. The histogram in red represents the simple S layer output which is highly uncorrelated and thus, its average correlation coefficient is close to zero. The responses within subspace (of size $Z = 5$) were pooled according to equation 4.14. Here, blue histogram (which is superimposed with the red histogram), shows that the remaining correlation coefficients are further reduced since ISA minimizes the dependencies between norms of projection into subspaces [32].

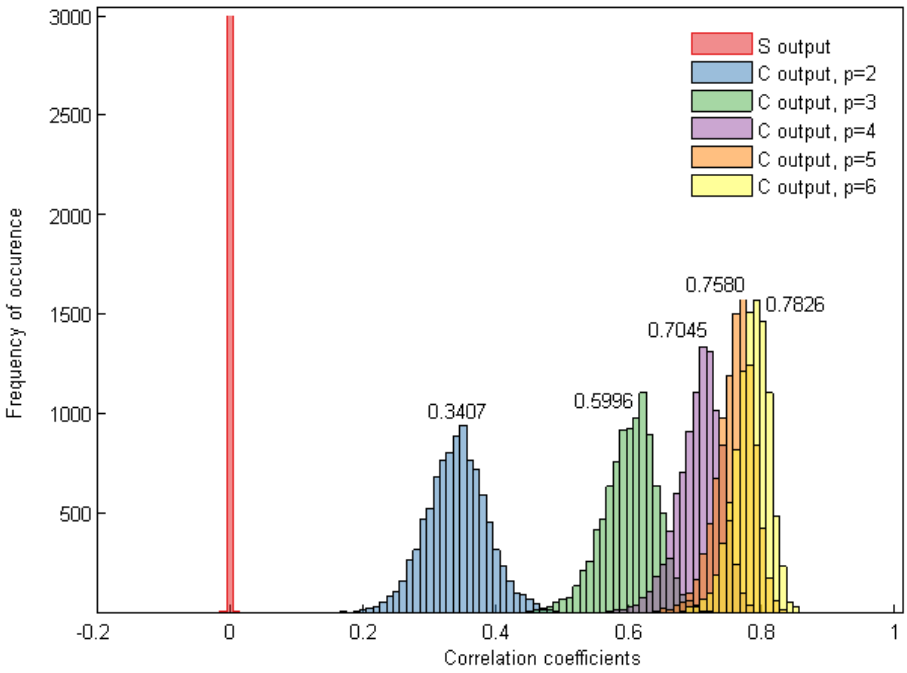


Figure 4.5: Correlation coefficients of the S and C layers of ICA HMAX in [10] (The values above the histograms indicate their average correlation coefficients.)

Applying max pooling over local area reintroduces linear dependencies in the form of positive correlation (with an average of 0.51), as observed by the histogram in green which is larger than that of figure 4.5 (as observed by the blue histogram which was 0.34).

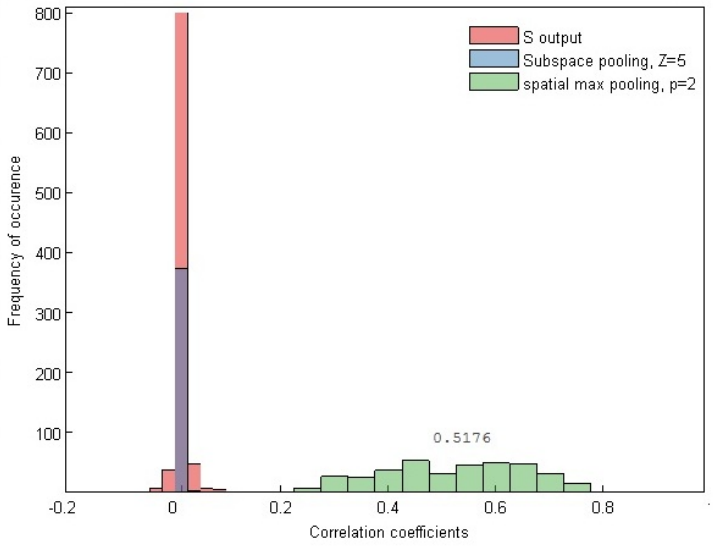


Figure 4.6: Correlation coefficients of the ISA layers (The values above the histogram indicate their average correlation coefficients)

It was observed that inhibition of activation values below zero at the S layer displayed better classification accuracy than allowing negative values (which is the same as ReLU function of the convolutional neural networks). After setting the negative values to zero, applying equation 4.14 shifted the histogram to the right, indicating positive correlation as seen in figure 4.7. The average correlation coefficients then also increased with subspace size (figure 4.7, *left*). Applying max pooling further increased the linear dependencies (as observed in figure 4.7*right*), with an average correlation coefficient slightly higher than in figure 4.6.

The histograms for the TICA outputs also displayed similar results, where increase in neighbourhood size and pooling area shifts the histogram to the further right. Setting negative values to zero before L2-pooling and then applying local max pooling further strengthens the dependencies between the filters. This indicates that the type of non-linearity function in the complex layers affects the extent of interaction between the components and thereby influencing the invariance properties of the extracted features. The study in [10] and the evaluations in

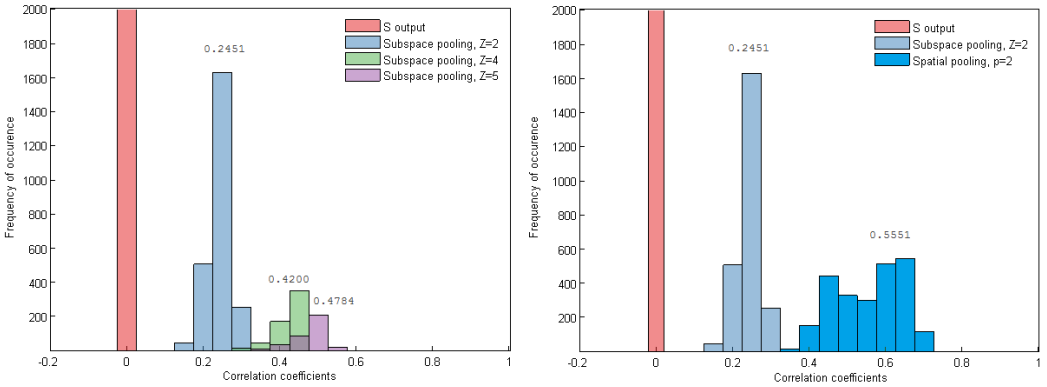


Figure 4.7: Correlation coefficients of the ISA layers after setting all the negative values to zero: *left*: Total number of filters is fixed at 100, and the subspace size Z is varied. *right*: After outputs within subspace size Z=2 is pooled, max pooling is applied over local positions of size p=2

figure 4.21 indicates that the introduction of linear dependencies in the complex layers coincides with higher classification performance of the models. Also, due to the appearance of these linear dependencies, CGC was applied on the sampled data from the complex layers before applying unsupervised learning algorithms in the next layer.

4.4 Enhanced HMAX models with phase and position invariance

In this section, implementation of the new hierarchical feature extraction model is presented. The first simple and complex cell layers are denoted by S_1 and C_1. Here, the combination of S_1 and C_1 layer functions is referred to as V_1 layer. The V_i layer of this model comprises of three sub layers: S_i is the response of orientation, spatial frequency and position selective linear filters, C_{i_a} represents the non linear L2 pooling of the S_i outputs within a subspace or topographic neighbourhood by equation 4.14, and C_{i_b} denotes max pooling output over neighbouring locations for each C_{i_b} feature. Since these non-linearities correspond to phase and position invariance respectively, they are referred to as such in the model description.

ISA HMAX model

In this model, the S layer filters were learned by applying ISA algorithm.

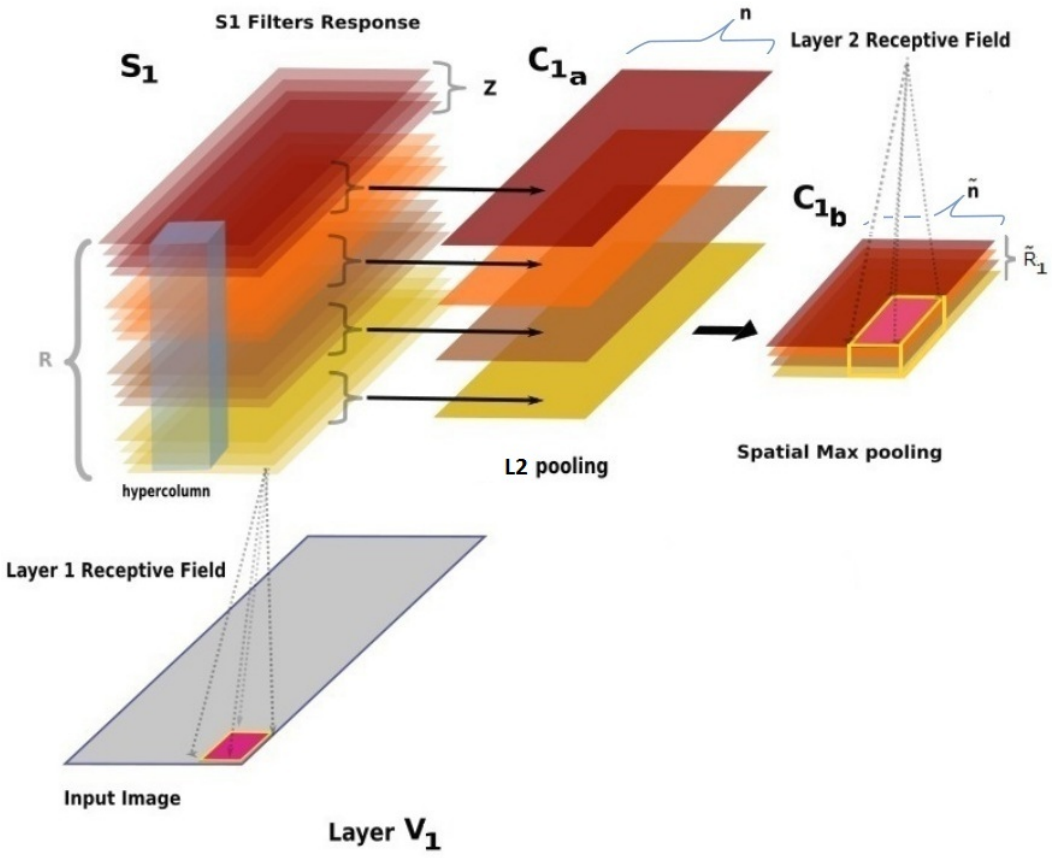


Figure 4.8: V_1 layer of the ISA HMAX model

Figure 4.8 shows the first layer V_1 of the feed-forward model. The structure of the model is in the form of *hypercolumns* (also illustrated in [60]) or feature maps, which is comprised of all the filter outputs for a spatial location . The subspace size is denoted by Z_1 and the total number of filters or bases at S_1 is R_1. Figure 4.9, shows the full model comprised of multiple V_1 layers. The receptive field size p_i of the V_i layer of the model indicates the width of the square area that is sampled from the $C_{(i-1)_b}$ layer (which is the input image for S_1).

S layer

The filters generated from sampling random patches of images were grouped into subspaces based on higher order energy correlations. The impact of subspace

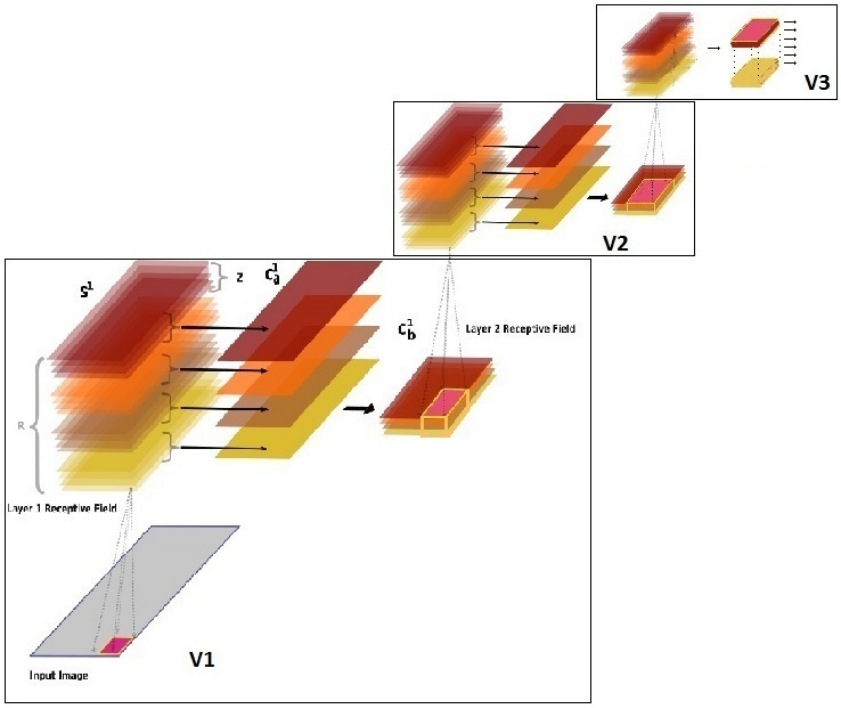


Figure 4.9: Multiple V_i layers of the ISA HMAX model

size along the different layers has an effect on object classification results. For a fixed set of S_i filters, increasing subspace size Z strengthens phase invariance but decreases the number of features.

S_i **layer:** For the first layer V_1, each S_1 layer filter is of size $p_1 \times p_1$, where p_1 is the width of the square receptive field of the first layer. The S_1 filter is applied on a patch of $p_1 \times p_1$ of the input image X which is of size $M \times N$.

If $W_i = \left\{ w_{i_1}, w_{i_2}, ..., w_{i_{R_i}} \right\}$ is the set of filters, the S_i response is of dimensions $\tilde{M} \times \tilde{N} \times R_i$, where $\tilde{M} = M - p_i + 1$ and $\tilde{N} = N - p_i + 1$.

$$S_i = \left\langle W, X_p \right\rangle \tag{4.17}$$

Where, X_p is a decorrelated and normalized set of patches extracted from the input image. Any negative output of S_i layer was set to zero before the complex layers, which resulted in a better performance in classification accuracy.

C layer

C_{i_a} **layer:** With subspace size Z_i, all the S_i values within the subspace are pooled such that the output of C_{i_a} has dimensions $\bar{M} \times \bar{N} \times \tilde{R}_i$. Where $\tilde{R}_i = R_i / Z_i$.

$$C_{i_a} = \sqrt{\sum_{j \in Z_i} {S_{ij}}^2} \tag{4.18}$$

Equation 4.18 represents the output of one feature detector at the C_{i_a} stage.

C_{i_b} **layer:** Each of the responses of the C_{i_a} are max pooled over non-overlapping areas of size $r_i \times r_i$ similar to [10].

TICA HMAX model

In this model, the S layer filters are arranged according to a topography determined by their energy correlations. The model structure is similar to the ISA version depicted in figure 4.8, but instead of equation 4.18, the components are pooled according to the neighbourhood of influence given by equation 4.8. The hierarchical layers are formed by alternating S and double C layers. The V_i layer of this model comprises of three sub layers: S_i represents the simple cell response of linear filters, C_{i_a} represents non linear pooling of S_i within a group represented by a two dimensional neighbourhood function $h(i, j)$, and C_{i_b} for the max pooling over local spatial position of the responses of C_{i_a}.

The figure 4.10, shows the V_1 layer of the HMAX model using TICA.

S layer

S_i **layer:** Input image X is of size $M \times N$. Similar to the previous model, W_i is the set of filters (which contains a total of R_i filters of size $p_i \times p_i$), the S_i output is of dimensions $\bar{M} \times \bar{N} \times R_i$, where $\bar{M} = M - p_i + 1$ and $\bar{N} = N - p_i + 1$ (given by equation 4.17).

C layer

C_{i_a} **layer:** With square neighbourhood of width h_i, all the S_i values within the area are pooled such that the response of C_{i_a} has dimensions $\bar{M} \times \bar{N} \times \tilde{R}_i$, where $\tilde{R}_i < R_i$.

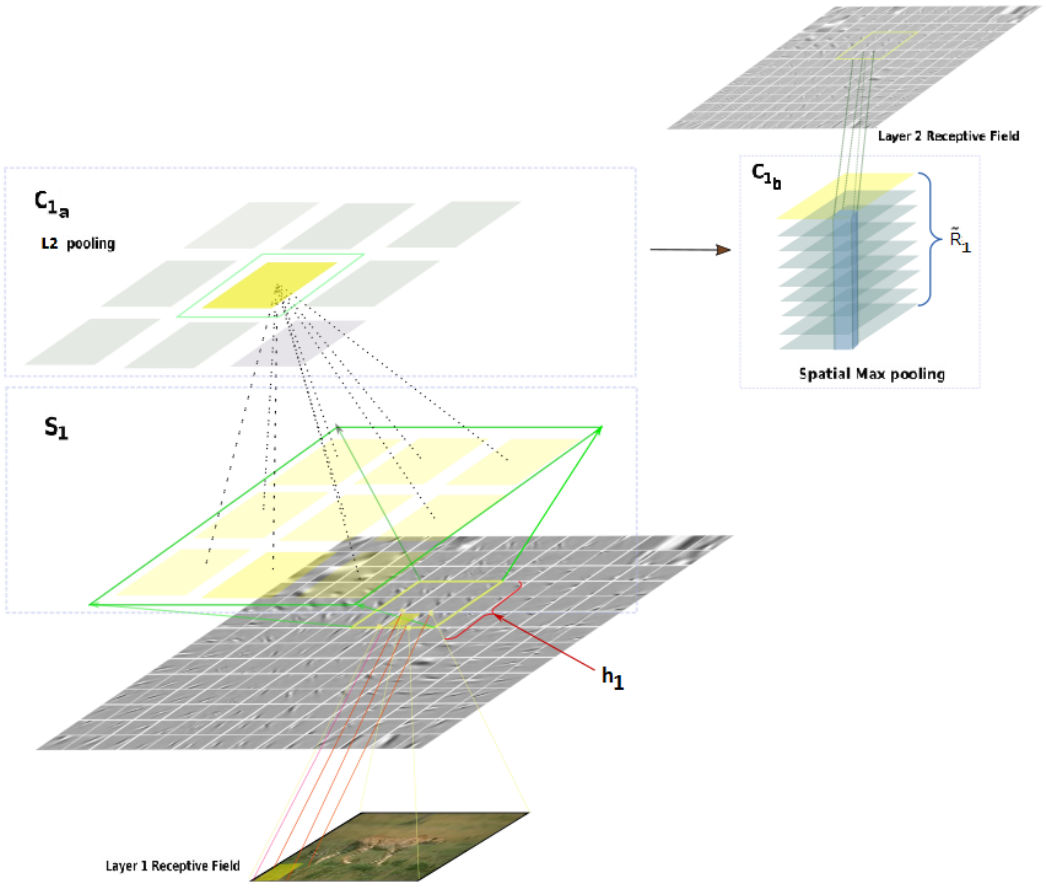


Figure 4.10: V_1 layer of the HMAX model (TICA) (image from CalTech101 dataset[15])

$$C_{i_a} = \sqrt{\sum_{j=1}^{n} h(i,j) {S_{ij}}^2} \tag{4.19}$$

Where $h(i,j)$ includes all the units that fall within a neighbourhood of size $h_i \times h_i$.

C_{i_b} **layer:** Again, each of the outputs of the C_{i_a} are max pooled over non-overlapping areas of size $r_i \times r_i$.

Final S layer

In the final S_n layer of the model, the square root of the sum of energies (or L2 pooling) of the values across all the location on each feature map is obtained as the feature vector.

$$C_n = \sqrt{\sum S_n^2} \tag{4.20}$$

The C_{i_a} step is generally not applied when obtaining the final feature vector (as illustrated in the V_3 layer of figure 4.9). The final feature vector of size $1 \times R_n$ forms the input for the classifier.

In both of these models, there is a dimensionality reduction within each of the C_{i_a} layers. This differs from the $1 * 1$ convolution method applied in the GoogleNet model [66]. It introduces non-linearity by another ReLU function immediately after each of the $1 * 1$ method. But when applied to unsupervised feature learning models such as these, the performance does not improve compared to L2-pooling. Moreover, the learning process is completely different in these models, compared to the backpropagation like methods of convolutional neural networks, so its applicability here is debatable.

4.5 Empirical evaluation

The different hierarchical models illustrated in figures 4.9 and 4.10 were tested on a database of 10 different categories [1] of objects from the *CalTech*101 dataset [15]. The initial experiments presented in this section are to evaluate between the ISA and TICA models along with different iterations of parameters such as subspace and receptive field size for which a small sample of the categories is used.

[1] categories included: airplane, bonsai, butterfly, car-side, chandelier, faces, ketch, leopards, motorbikes, watch

The models were comprised of three V_i layers which included linear filtering, non-linear L2 pooling and spatial max pooling. The S layer filters were learned from a database of the 10 categories which contained a total of 30 images per class. After learning the S layer dictionaries, features were extracted by the models from a separate testing database of 600 images from the same 10 categories. For evaluation of the models, the extracted feature vectors were classified with multi-class linear SVM (LibSVM software [97]). A sparse HMAX model using ICA as described in [10] was also trained with the same parameters and database for comparison. The model had the same number of V_i layers except that it only included linear filtering and spatial max pooling functions.

Experiment 1: Multi-class object classification: Comparison of HMAX models

In all the models, for learning the filters in each S layer, a total of $50,000$ data samples were randomly extracted from the previous layer. The samples were then whitened and normalized to reduce linear dependencies before applying the learning algorithms (ICA, ISA or TICA).

Table 4.1: Model specifications

Models	V_1, $p_1 = 11$			V_2, $p_2 = 12$			V_3, $p_3 = 13$
		C_{1_a}			C_{2_a}		
	S_1 (R_1)	Z_1/h_1	$\tilde{R}_1$	S_2 (R_2)	Z_2/h_2	$\tilde{R}_2$	S_3 (R_3)
ICA	36	-	36	64	-	64	400
ISA	144	4	36	100	4	25	400
TICA	144	2	36	100	2	25	400

Table 4.2: Number of S_1,C_1,S_2,C_2,S_3 filter outputs. The ICA model does not have a C_{i_a}, so the number of filters do not change

Similar parameters were used for all the three models, detailed in table 4.2, where R_i is the number of S_i filters, Z_i is the subspace size for ISA, h_i is the width of the neighbourhood function for TICA, p_i is the width of the square receptive field area and $\tilde{R}_i$ is the final number of V_i features after subspace or topographic pooling. Although the number of filters in the S_1 and S_2 layers for the ISA and TICA is larger than the ICA models, they are reduced in numbers after pooling in C_{i_a}. Since there is no C_{i_a} stage in the ICA model, its number is fixed at 36, which equal to the number of filter outputs obtained after C_{i_a} in the ISA and

TICA models.

The final number of filter outputs in the V_2 layer of the ICA model is kept larger than the TICA and ISA models. Greater number of features usually result in better classification accuracy, but the increased dimension of the data slows down the learning process in the next layer. So in this case, the ICA model actually has a larger number of final C_2 layer outputs than TICA and ISA models.

In the case of ISA and TICA models, the C_{i_a} stage pooling reduces the number of features such that computation for the next layer is easier. The results were also compared with the same dataset on the HMAX model (called S-HMAX) from [16][11] using the accompanying source code [2]. The size of final feature vector was 400 for all the models.

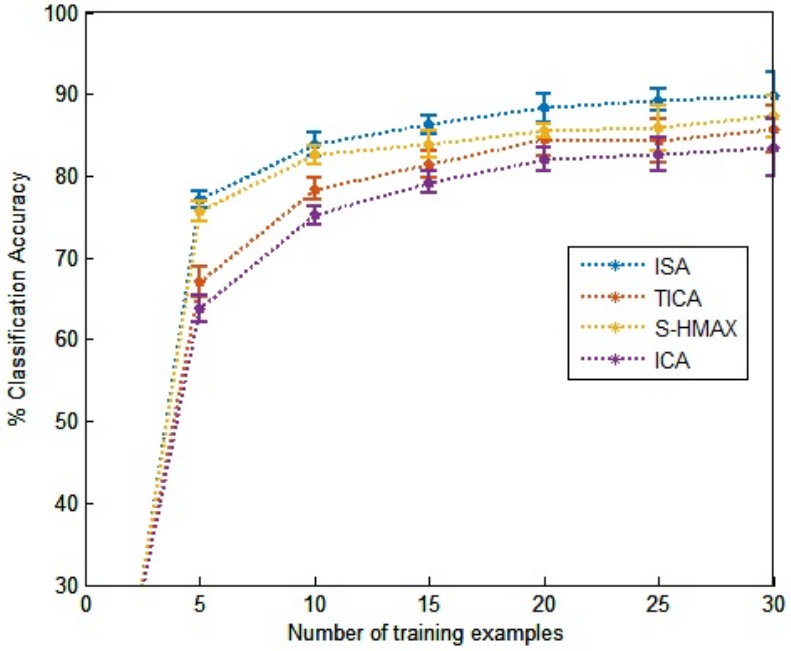


Figure 4.11: Classification accuracy for the different hierarchical models: ISA, TICA, ICA [10] and S-HMAX [16]

The figure 4.11 displays the classification accuracy for the models on ten categories of objects. The accuracy was determined from 30 individual runs of the classifier with random splits of training and testing data. It is observed that the ISA model outperforms all the other models, including TICA as well as S-HMAX

[2] http://webia.lip6.fr/ cord/BioVision/

[16]. In the S-HMAX model, the S layer outputs were pooled across multiple spatial resolutions for scale invariance, which was not included in our model. In [10], the ICA model was reported to perform better than the HMAX model in [72], and other convolutional hierarchical models such as [28][64]. But their final feature vector was obtained by spatial pyramid pooling [98] with resolutions 4, 2 and 1, rather than a global L2 pooling. The resulting feature vector was of size 43,008, which combined outputs from two different types of architectures (with 4 and 6 layers) [10]. Also, in this graph, the TICA model is observed to perform poorly in comparison to the ISA and S-HMAX with these parameters, due to which the parameters such as pooling neighbourhood size or overlap size need to be examined. For ISA and TICA models, various parameters such as number of dictionaries and receptive field sizes affect the classification result. For example, in the case of TICA, the size of neighbourhood function as well as the size of overlapping area affects the model performance. In this experiment, there was no overlap of the pooling area.

In [96], a generalized ISA model was used in for estimating the optimal subspace sizes from natural image statistics. It was discovered that models with a relative increase in subspace sizes, provided a better statistical representation of natural images than ICA. Estimation of pooling method also found squared summation to be the best form of modelling non-linearities compared to absolute values. This coincides with the observations in figure 4.11, where in a hierarchical setting, ISA demonstrably extracts more distinguishable features than ICA.

Experiment 2: Subspace size of the ISA model

Generally, large number of feature detectors are optimal for recognition models as they capture image complexity more accurately. Reducing subspace size increases the final feature size of the V_i layer, whereas increasing subspace size improves processing speed by reducing the size of C_{ia} output. To study this effect of changing feature dimensions, the same dataset of 10 object categories as experiment 1.

Subspace size of final layer

The parameters for V_1 and V_2 were fixed, while changing subspace size for the V_3 layer in the ISA model from figure 4.9. The model specifications for this experiment are described in table 4.3.

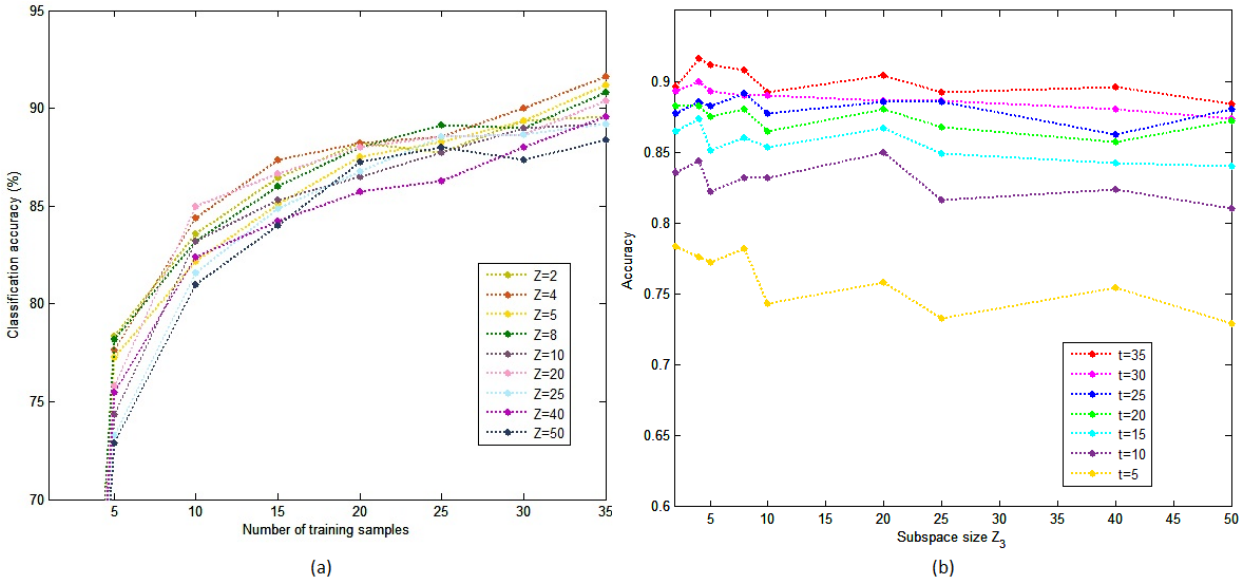


Figure 4.12: Classification accuracy for 10 classes when subspace size Z_3 is changed with fixed number of R_3. L2 pooling at V_3 is not applied so the feature vector is of the same length for all the cases: a) Accuracy with respect to number of training samples b) Accuracy with respect to subspace size, where t represents the number of training samples

Table 4.3: Model specifications: The subspace size Z_3 of V_3 is varied

Models	V_1, $p_1 = 11$			V_2, $p_2 = 12$			V_3, $p_3 = 13$
		C_{1_a}			C_{2_a}		
	S_1 (R_1)	Z_1	$\tilde{R_1}$	S_2 (R_2)	Z_2	$\tilde{R_2}$	S_3 (R_3)
ISA	100	4	25	150	5	30	400

In figure 4.12, the S_3 layer filters were formed with different subspace sizes, but were not pooled with equation 4.18. So, after applying spatial pooling over all the locations of the features, the final feature vector was of size 1×400. In figure 4.13, the values within the subspaces were pooled with equation 4.18 such that the feature vector was of variable size.

Figure 4.12a demonstrates when the overall performance of the features with smaller subspace size (with a fixed feature vector size) perform better than the subspaces of largest sizes (45, 50). However, figure 4.12b indicates that $Z_3 = 4$ and $Z_3 = 20$ classifies with better accuracy than the rest for most of the training sample sizes.

When L2 pooling (equation 4.18) was applied, the reduction in feature size showed the model with highest $\tilde{R}_3$ (which in this case is 200) to perform better object classification than the rest (figure 4.12a). Again, the models with subspaces of size 4 and 8 are on average less accurate than 5 and 10 respectively in (figure 4.12b).

Number of S2 layer filters

Here, the parameters for V_1 and V_3 were fixed, while changing subspace size for the V_2 layer in the ISA model from figure 4.9. The model specifications for this experiment are described in table 4.4.

Table 4.4: Model specifications

Models	V_1, $p_1 = 11$			V_2, $p_2 = 12$			V_3, $p_3 = 13$
		C_{1_a}			C_{2_a}		
	S_1 (R_1)	Z_1	$\tilde{R_1}$	S_2 (R_2)	Z_2	$\tilde{R_2}$	S_3 (R_3)
ISA	100	4	25	300	-	-	200

In figure 4.14a, the classification accuracy for the different subspace sizes at the V_2 layer is depicted. The model with largest $\tilde{R}$ shows highest accuracy, but

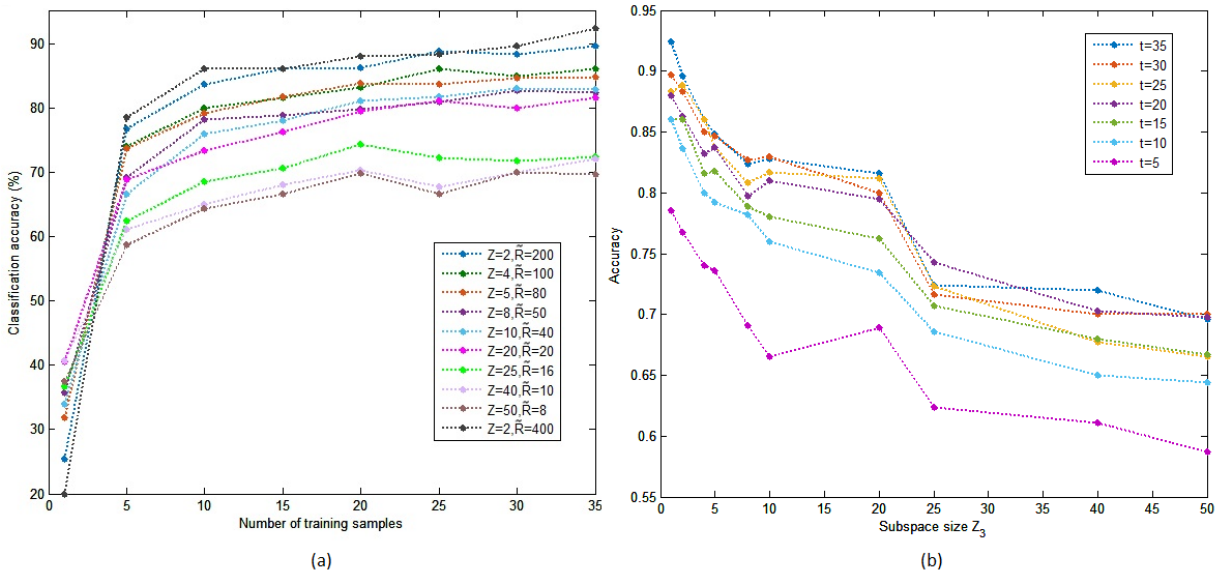


Figure 4.13: Classification accuracy for 10 classes when subspace size Z_3 is changed with fixed number of R_3. Pooling of subspace values is applied so the feature vector size $\tilde{R}_3$ changes for all the cases: a) Accuracy with respect to number of training samples b) Accuracy with respect to subspace size, where t represents the number of training samples

from figure 4.14b, it is seen that the second largest $\tilde{R}$ which is 100 for $Z_2 = 3$ does not perform better than $Z_2 = \{4, 5\}$.

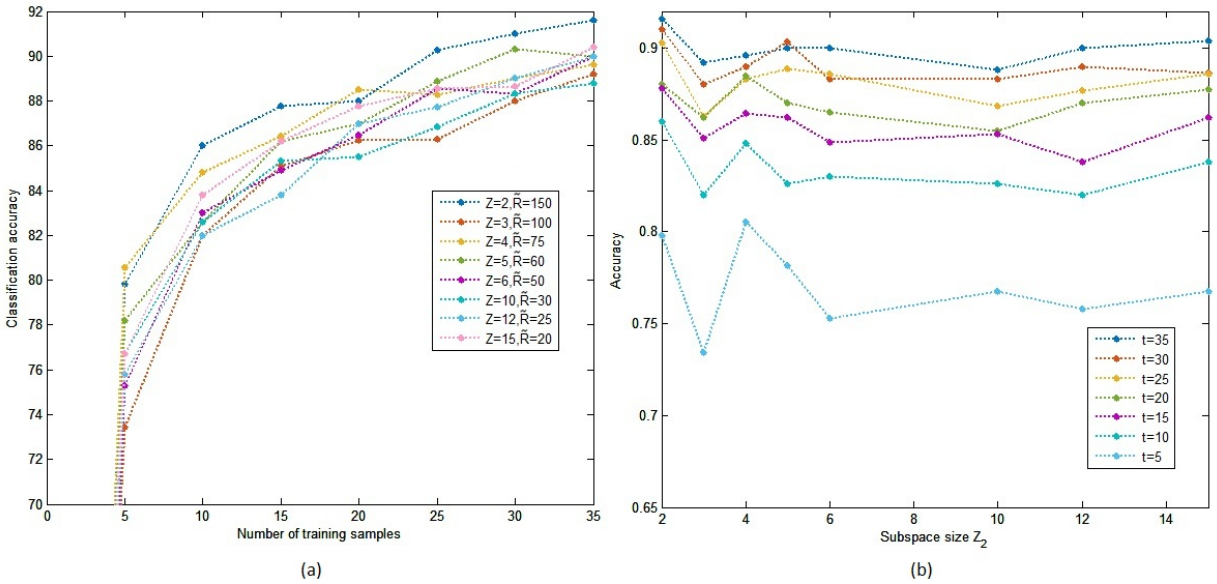


Figure 4.14: Classification accuracy for 10 classes when subspace size Z_2 is changed with fixed value of $R_2 = 300$. Pooling at C_{2_a} is applied such the feature vector size $\tilde{R}_2$ changes for all the cases: a) Accuracy with respect to number of training samples b) Accuracy with respect to subspace size, where t represents the number of training samples

Figure 4.15 shows the results when the value of $\tilde{R}_2$ is increased in a steady manner while keeping subspace size at $Z_2 = 5$. As the figure indicates, increasing the number of subspaces generally result in better performance, but $\tilde{R}_2 = 50$ and $\tilde{R}_2 = 40$ is less accurate than $\tilde{R}_2 = 45$ and $\tilde{R}_2 = 35$ respectively.

The above experiments indicate that although larger subspace sizes are preferred, simply increasing the number of filters or subspace sizes does not necessarily translate to a better model. For example, the results in figure 4.14b showing better accuracy for $Z_2 = 4, \tilde{R}_2 = 75$ than $Z_2 = 3, \tilde{R}_2 = 100$ indicate that larger subspace sizes represent the statistical properties of the data more accurately.

This highlights the drawback of applying ISA with prior assumption of pooling sizes since the probability of best data representation is not guaranteed. In [96], it was discovered that a relatively large subspace size was optimal for representation of natural image statistics, depending on the size of the input patch.

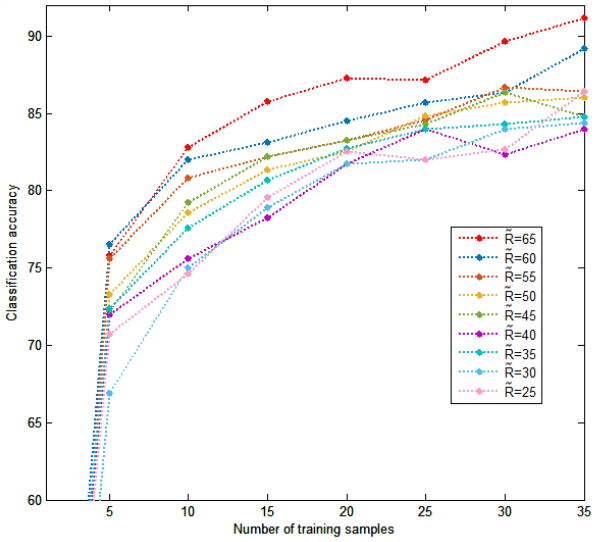


Figure 4.15: Accuracy for subspace size $Z_2 = 5$ while increasing R_2

For higher complexity data, such as the input sample to the V_2 and V_3 layers, it was found that the most optimal subspace sizes to be 2 and 5 for the V_2 layer and 4 for the V_3 layer. It is thus more beneficial for the subspace sizes to be estimated adaptively rather than fixed.

Experiment 3: Topographic ICA models

Unlike the ISA model, the S layer outputs in TICA models can be pooled with a variable size. The topographic model in experiment 1 did not show favourable result in comparison with the ISA and S-HMAX models, which had a neighbourhood function of width $h_1 = 2$ and no overlap. In this the parameters for the TICA models such as number of features and neighbourhood size which affect the overall performance of the multi-class object recognition is examined.

Overlap of pooling area

The sample of input were processed in the same manner as the previous experiments with ten categories of images from the Caltech101 database. In the first set of models, the three S layers were learned with neighbourhood function of size 3×3, 5×5 and 7×7. The specifications of the model are described in table 4.5. Here, the pooling area width is represented by h_i and the overlapped number of

units is denoted by o_i. The types of pooling types A and B are illustrated in figure 4.16. If $o_i = 0$, it follows the pooling method B and A for $o_i > 0$. ($M4$ has the same specifications as $M3$ except that in the feature learning phase, the TICA filters are formed with a neighbourhood function of size 3×3 for all the three layers).

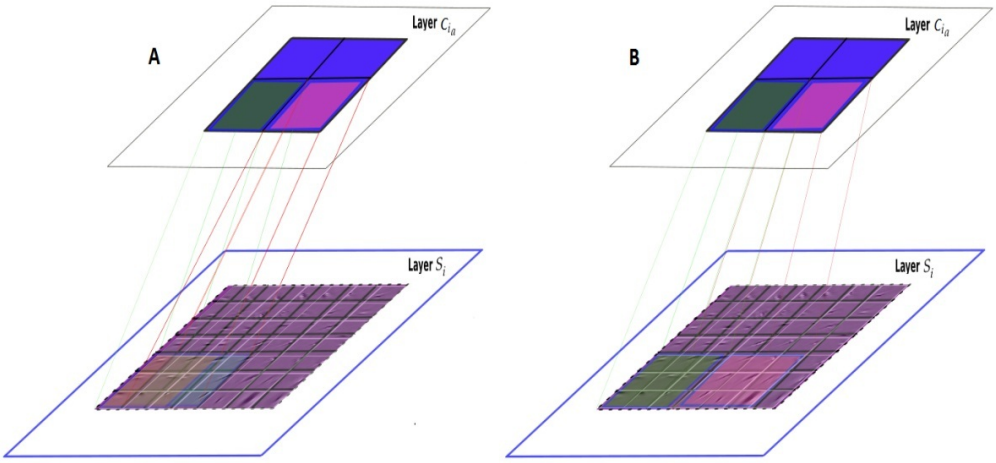


Figure 4.16: Pooling window for TICA

Table 4.5: Model specifications

Models	V_1			V_2			V_3
		C_{1_a}			C_{2_a}		
	$S_1(R_1)$	h_1	$\tilde{R}_1$	$S_2(R_2)$	h_2	$\tilde{R}_2$	$S_3(R_3)$
M1 ($o_1 = 1, o_2 = 2$)	169	3	36	196	4	36	225
M2 ($o_1 = 1, o_2 = 0$)	169	3	36	196	2	49	225
M3 ($o_1 = 0, o_2 = 0$)	144	2	36	144	2	36	225
M4 ($o_1 = 0, o_2 = 0$)	144	2	36	144	2	36	225

The figure 4.17 shows the performance of classification accuracy for the models in table 4.5. Despite having a larger number of filters for models $M1$ and $M2$, its performance is much worse than that of $M3$ and $M4$. This demonstrates that overlapped pooling does not lead to a better model even though the final number of features at the C_{i_b} is larger.

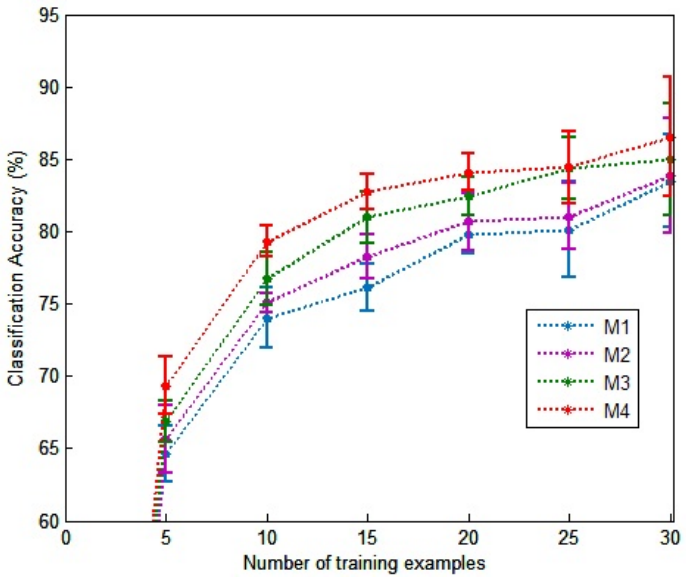


Figure 4.17: Classification of TICA models

Size of the neighbourhood function

The figure 4.18 shows the performance when the width of the pooling area h_i is varied. The model parameters for this experiment are described in table 4.6. During the learning phase, the S layer filters were formed with a topographic neighbourhood function of size 3×3 for all the layers.

Table 4.6: Model parameters

Models	V_1			V_2			V_3
		C_{1_a}			C_{2_a}		
	$S_1(R_1)$	h_1	$\tilde{R_1}$	$S_2(R_2)$	h_2	$\tilde{R_2}$	$S_3(R_3)$
M1 ($o_1 = 0, o_2 = 0$)	64	1	64	100	1	100	225
M2 ($o_1 = 0, o_2 = 0$)	16	1	16	25	1	25	225
M3 ($o_1 = 0, o_2 = 0$)	64	2	16	100	2	25	225
M4 ($o_1 = 0, o_2 = 0$)	144	3	16	255	3	25	225

From table 4.5, $M2$,$M3$ and $M4$ have the same number of C_{i_a} outputs ($\tilde{R}_i$). They are represented as the solid blue,red and black lines in the figure 4.18 respectively. The model with $h = 2$ shows an improved accuracy compared to $h = 1, 3$. The high performance of $M1$ (blue dotted line) can be attributed to the

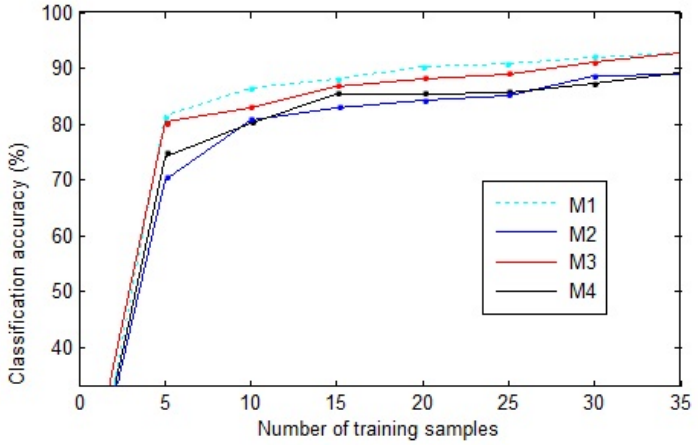


Figure 4.18: Classification of TICA models with different neighbourhood function sizes h_i

larger number of C_{i_a} outputs. The results indicate that the ratio of number of S_i filters to the pooling area (ie., R_i/h_i) should be sufficiently large for a better performance.

4.5.1 Object Classification: ISA and TICA models

Comparison of ISA and TICA models is again examined since they are similar algorithms but display widely different classification results, based on the choice of pooling parameters. From the TICA models, it was found that non-overlapped windows for pooling of neighbouring filters show better object classification than overlapping windows. While keeping the pooling area fixed, the TICA model becomes similar to the ISA models with a fixed subspace. Therefore, for comparison, similar model parameters were used, where the area of neighbourhood function for the TICA model is equal to the subspace size of the ISA model.

From the figure 4.19, it can be seen that with lower number of filters and larger subspace sizes, ISA clearly outperforms the TICA models, even for lower number of S layer filters. However, when the number of filters are high in comparison to the pooling neighbourhood size (when the ratio of R_i/h_i is larger), TICA performs on par or better than the ICA models.

In ISA models, the subspaces exhibit complex cell properties of phase invariance while the spatial frequency and orientations remain unchanged [96]. But

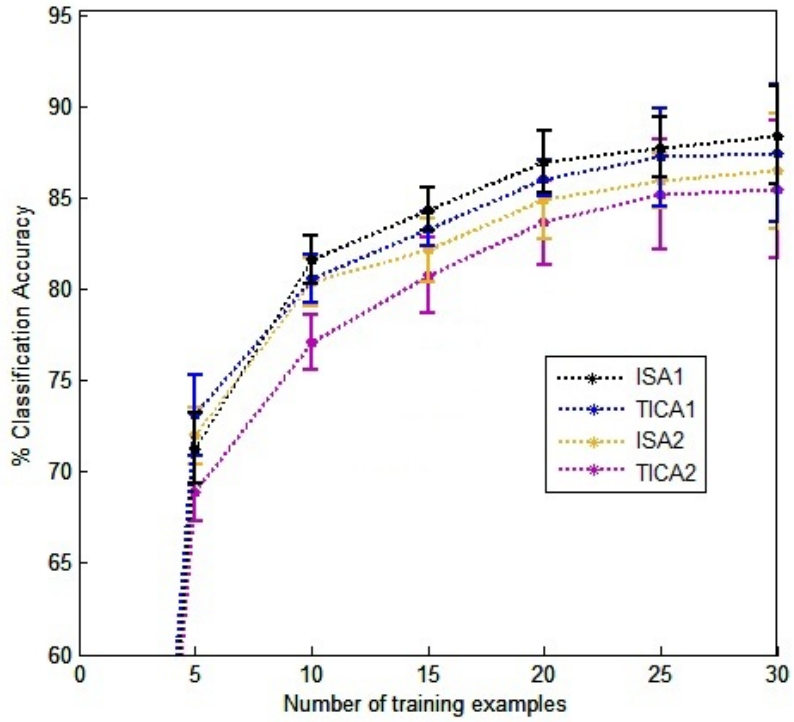


Figure 4.19: Comparison for ISA and TICA on Classification accuracy for 10 categories

Table 4.7: Model specifications

Models	V_1			V_2			V_3
		C_{1_a}			C_{2_a}		
	$S_1(R_1)$	h_1, Z_1	$\tilde{R}_1$	$S_2(R_2)$	h_2, Z_2	$\tilde{R}_2$	$S_3(R_3)$
ISA1	144	9	16	100	4	25	225
TICA1	144	3	16	25	2	25	225
ISA2	64	4	16	100	4	25	225
TICA2	64	2	16	255	2	25	225

with TICA, when the area of neighbourhood function is increased, along with phase, the orientation and frequency variation within the area is large [32]. This could explain why the models with lowest pooling area ($h_i = 1$), with respect to the number of filters show best classification accuracy. This also applies in the higher layers, where increase in data complexity and pooling area introduces even wider variation of features. Figure 4.20, displays the S_2 and S_3 units by keeping the pooling neighbourhood area $h_i = 1$ for all the layers.

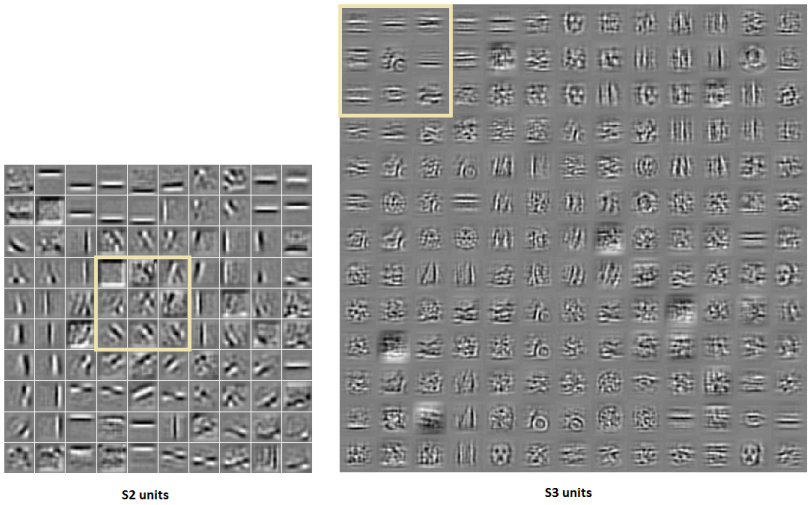


Figure 4.20: S_2 and S_3 units of the TICA model visualized using the method defined in [10]

With $h_i = 1$, it is almost similar to the ICA model but demonstrates a better classification accuracy (than ICA) [30]. It is also more biologically feasible than either ISA or ICA, as its structure resembles the retinotopic organization of cells

in the retina, LGN and V1 [32]. For a hierarchical TICA to give the best results, small neighbourhood sizes are better suited. This however does not contribute much to the dimension reduction of the data.

4.5.2 Receptive field size

Studies have shown that the receptive field size increases as we go from lower to higher levels of the VC [99], where the cells of the first layer process local stimulus within a small localized area. Figure 4.21 shows the performance of the architecture in figure 4.9 with different receptive field sizes. The results in 4.21 indicate that increase in receptive field size also improves performance. The number of filters are the same as in $ISA1$ from table 4.5 and p refers to width of the square patch.

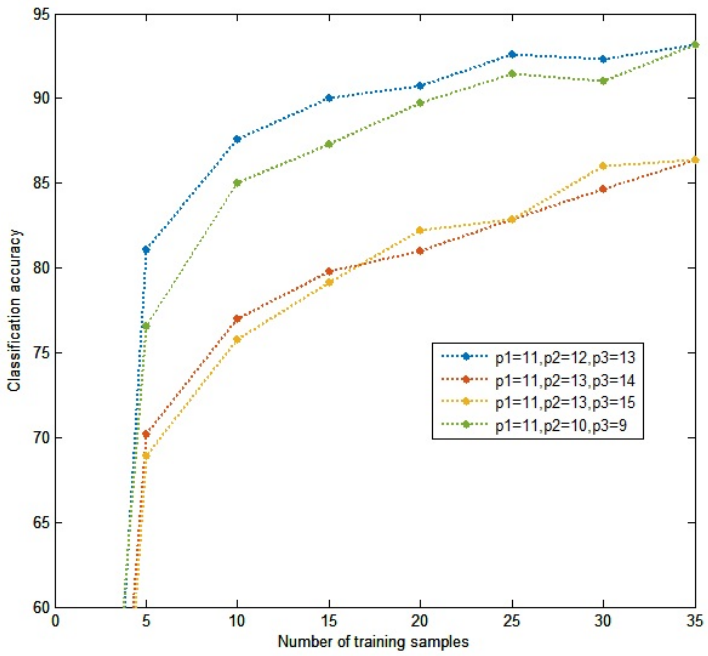


Figure 4.21: Performance for different receptive field sizes p for ISA

Increasing receptive field size improved the performance only when the ratio of increase was not too large as seen from figure 4.21. The model with decreasing RF size ($p1 = 11, p2 = 10, p3 = 9$) was also more accurate than the ones with $p1 = 11, p2 = 13, p3 = 14$ and $p1 = 11, p2 = 13, p3 = 15$.

The figure 4.22, shows more combinations of receptive field sizes. The graph

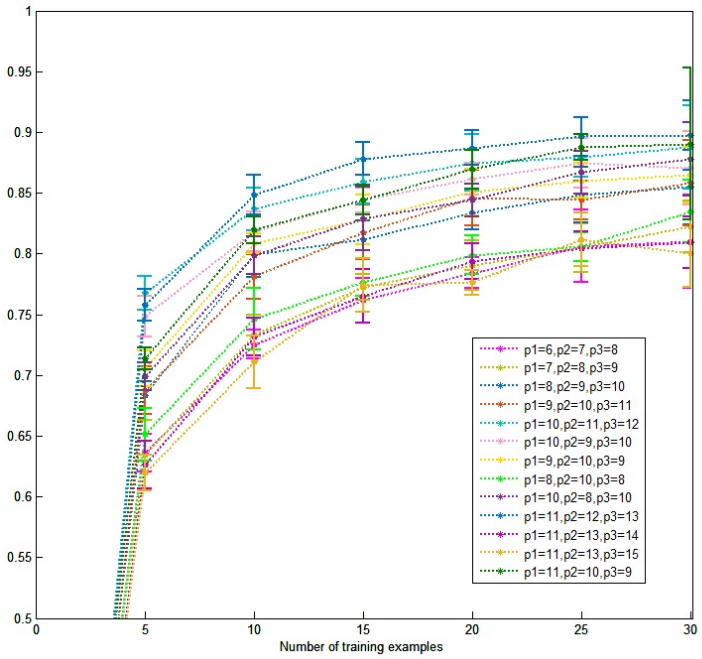


Figure 4.22: Performance for different receptive field sizes p for ISA

shows an increasing value of $p1 = 11, p2 = 12, p3 = 13$ to be the most favourable.

4.5.3 Spatial pooling at the final layer

Instead of max pooling, a global L2 pooling was applied across the values of S_3 output, which was taken as the final activation of the high level filters. Compared to max pooling method, it showed better classification accuracy. Setting any negative values to zero also further improved the model performance (figure 4.23).

4.6 Dimensionality reduction with 1*1 convolutions

The $1 * 1$ convolution method has been adopted in some convolutional neural networks such as the GoogleNet [66]. For supervised learning models, this technique has proved to improve performance by allowing deeper networks and also introducing an additional non-linearity with a ReLU operation. Since unsupervised models such as the ones described here does not rely on back-propagation methods to learn the filters and are functionally different, applying $1 * 1$ convolutions

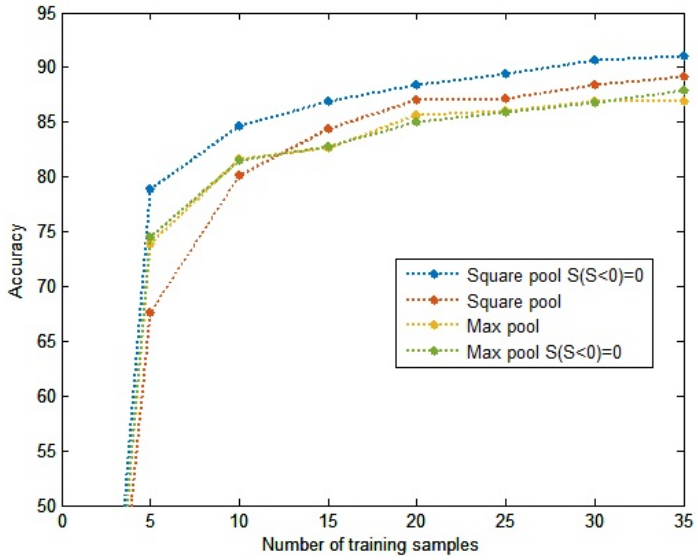


Figure 4.23: Performance for different pooling methods at the V_3 layer

Table 4.8: Model parameters

Models	V_1		V_2		$V_{2.5}$ (1*1 convolution layer)	V_3
	$S_1(Z_1)$	$\tilde{R_1}$	$S_2(Z_2)$	$\tilde{R_2}$	$1*1$ convolution	S_3
ISA (without $1*1$)	100(4)	25	150 (6)	25	-	225
ISA (with $1*1$)	100(4)	25	150 (5)	30	25	225
ICA (without $1*1$)	25	25	30	30	-	225
ICA (with $1*1$)	25	25	30	30	25	225
ICA(2) (with $1*1$)	25	25	150	150	25	225

does not result in any improvement over older ICA based models. To evaluate its effect, it was applied on an ICA and ISA HMAX model after the second layer C_2 output. The $1*1$ convolution layer (which is denoted as layer $V_{2.5}$ aims to reduce the total number of filter outputs from the C_2 without affecting the performance. The table 4.8 shows the parameters that were applied for the models. The input data used was the same as in experiment 1, with 10 categories. The values at layer $V_{2.5}$ signifies the reduction in data dimensionality with respect to the previous layer V_2. For both ISA and ICA models, the negative values were suppressed after convolution to introduce an added non-linearity.

Figure 4.24 displays the performance of the models in table 4.8. Aside from ISA performing better classification than ICA models, in both the cases, $1*1$

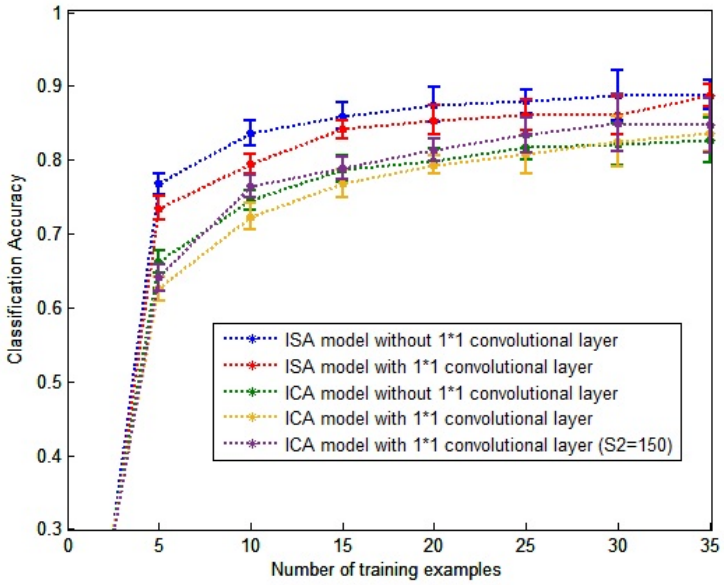


Figure 4.24: Effect of $1 * 1$ convolution on ICA and ISA HMAX models

convolution does not contribute to improving its performance. Also, even with increasing the number of filters for ICA(2), the model gives poorer results than ISA. The large scale CNNs in contrast require millions of images for training and the learning of filters does not depend on data from the previous layer. This experiment, however, represents a model with a limited amount of training data ad filter learning highly depends on the output of the previous layer. In this case, applying $1 * 1$ convolution does not seem advantageous.

4.7 Face detection

In the multi-class categorization experiments, the whole set of final layer (S_3) responses were extracted to form the feature vector, which was then used as input for an SVM classifier. Although studies in neuroscience have not yet established the mechanisms behind inference, this method for evaluation does not reflect the biological process in the visual cortex. It was studied that groups of neurons in the IT respond to a particular type of stimulus such as *face neurons* [100]. With ISA and TICA, filters or *neurons* are grouped together based on their energy correlation. These groups of neurons should be able to selectively activate depending on the stimulus.

In [13], it was discovered that unsupervised training of multi-layered model using TICA formed units (or *cells*) that were highly selective towards a category of object. Faces, in particular, were found to be highly distinguishable from other random inputs.

In this section, the S_3 layer filters, a threshold was used according to which it classified the stimulus as $face$ or $distractor$. A total of 810 images were used, out of which 405 were comprised of images of faces and the rest were random set of images. The model was trained by the 10 category dataset in the previous experiments. The threshold is changed for both ICA, ISA and TICA models. For ISA, the combined accuracy of an entire subspace is lower than that an individual unit, but all the cells within that subspace show high degree of selectivity towards the stimulus.

Figure 4.25 shows the selectivity of the best neuron for face detection using the ICA model for HMAX. After adjusting the threshold to 7.5, the unit achieved 81% accuracy in detecting faces from the dataset. (In [10], all the individual units were classified by assigning it a category label according to its activation value with respect to a fixed threshold which achieved an 84% accuracy in classifying multiple objects).

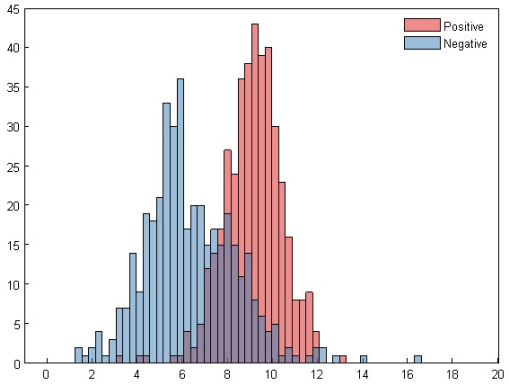


Figure 4.25: Histogram of Positive and Negative samples for a single S_3 layer unit for ICA

With ISA models, the figure 4.27 shows the histogram for the positive and negative samples for a subspace. The number of units in one subspace was 10 and the combined response of the 10 units displayed an accuracy of 88.6% for detecting faces. All the units within the subspace have demonstrated a high activation value with respect to the threshold of 34.7. (The threshold needed to

be increased with increase in subspace size). Figure 4.27 displays the histogram of the best neuron for detection of faces which was achieved with an accuracy of 92.88% accuracy. Figure 4.28 shows the histogram of all the 10 units within the highest performing subspace for detection of faces. All the individual units within the subspace in figure 4.28 had an accuracy of above 70%.

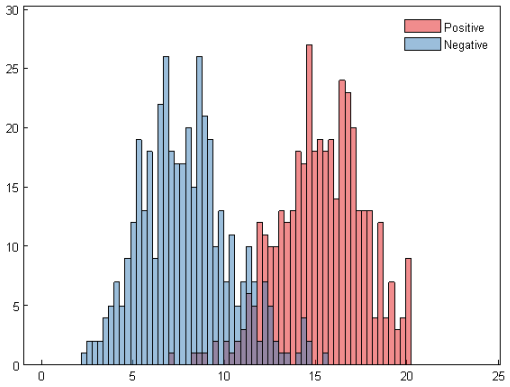


Figure 4.26: Histogram of positive and negative samples for a single S_3 layer unit for ISA

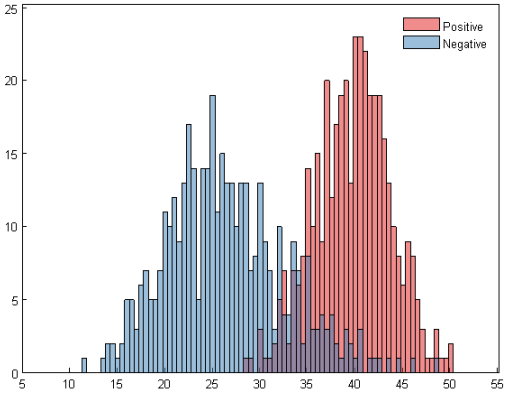


Figure 4.27: Histogram of positive and negative samples for a single S_b^3 layer subspace for ISA ($Z_3 = 10$)

Similarly, with TICA, the neighbourhood of highest activation values was analysed for feature detection. In this case, the single best unit achieved a 93.13% accuracy (figure 4.29) in distinguishing faces from random set of images. The combined output of the surrounding units displayed an accuracy of 88.96% (fig-

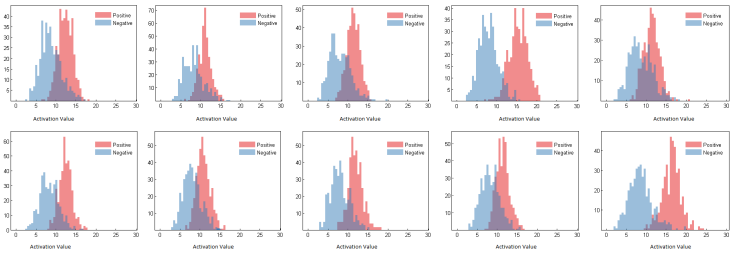


Figure 4.28: Histogram of positive and negative samples for the 10 S_3 layer units within the highest performing subspace

ure 4.30).

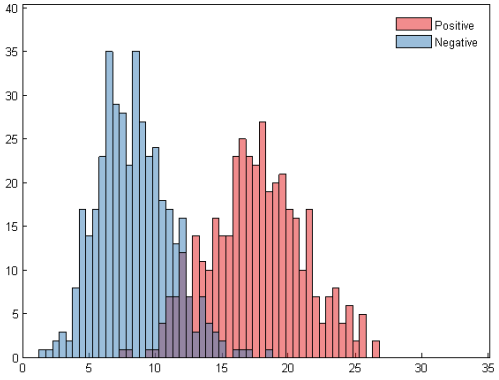


Figure 4.29: Histogram of Positive and Negative samples for a single S_3 layer unit for TICA

These graphs show that filters in the S_3 layer of the HMAX models using ISA and TICA can detect faces with high accuracy when the model is trained with unlabelled data with any random sets of images. The TICA in this case adopted the parameters with a small neighbourhood pooling size (but with 400 S_3 features), where the neighbourhood size was smaller with respect to the number of filters. The multi-object classification in this case was almost close to the ISA model (as seen in figure 4.19). Although the ISA model slightly outperformed for multi-object classification, the TICA model, with small neighbourhood size and from a random set of training images was able to learn highly distinctive face 'neurons' that are grouped together in the topography.

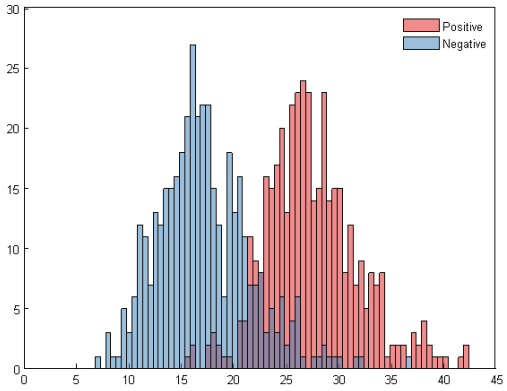


Figure 4.30: Histogram of Positive and Negative samples for a neighbourhood of S^3 layer filters ($h^3 = 2$)

4.7.1 Invariant response

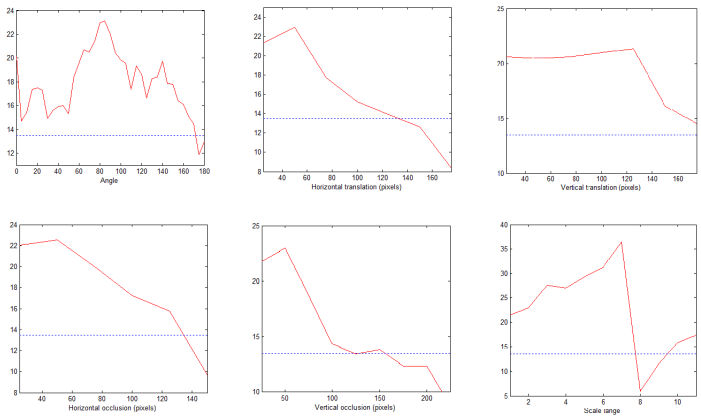


Figure 4.31: Neural activation response of the best neuron in a TICA model with respect to varying factors with threshold (blue)

The figure 4.31 shows the response of a single high level unit in the S_3 layer of the TICA model. The blue line indicates the threshold. The same threshold that was applied in figure 4.29 (13.5) for detecting faces was applied and orientation, position and scale of the images were changed. As seen from the activation values in figure 4.31, the final S_3 neuron that detects faces, does so irrespective of change in orientation, position and occlusion to a certain extent. Although pooling between multiple spatial resolutions was not applied as in [54][72][16], scale

invariance to a certain extent was also achieved.

4.7.2 Multiple scale model

In the HMAX models [72][16], Gabor filters of different sizes were applied for modelling scale invariance. The resulting features displayed a range of spatial frequencies which were sorted into scale bands and the value of adjacent scales were pooled together.

For multiple scales, learning filters of different patch sizes in an unsupervised manner generates sets that are uncorrelated with each other and thus, cannot be pooled together. An alternative method was suggested in [10], where existing filters could be resized and integrated into the model. To examine this method, filters of multiple sizes were applied on the ISA models and the outputs were resized before pooling. The performance declined considerably, which indicated that HMAX method for scale pooling is not applicable.

In [95], to address the limited scale tolerance of CNNs, a scale invariant convolutional network (SiCNN) was proposed where a multi-column approach was applied [95]. The feature vector of all the columns, which processed different scales, was concatenated before applying the fully connected final layer. This method of concatenating the output feature vector also improved accuracy for the ISA model, when feature vectors from two architectures were concatenated but this could also be attributed to the larger feature vector. Since improving invariant response is always desirable, more research into this area is needed.

4.8 Multi-class object categorization on CalTech101 dataset

The previous set of experiments were performed on a small dataset images. The purpose was mainly to observe the behaviour of the model with varying its parameters. Both the ISA and TICA models show improved classification when compared to ICA models. TICA model perform best with a small pooling neighbourhood size and therefore, the extent of possible data dimensionality reduction is lower compared to ISA model. When the training set for learning the filters was randomized, the accuracy levels were similar. The figure 4.32 display the an ISA model where filters were trained with five different randomization of the same dataset. However, there is a possibility for the outcome to be different for different datasets. Therefore, it is important for the model to be tested with a

larger dataset.

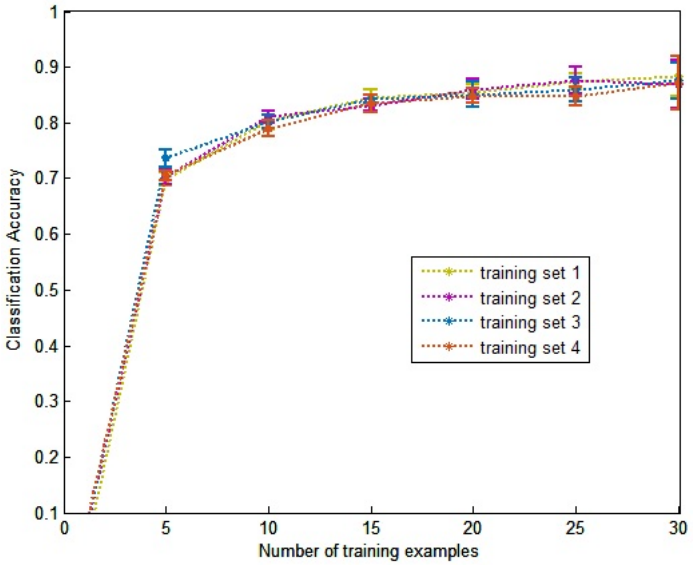


Figure 4.32: ISA HMAX model trained with different randomization of the same dataset

To compare its performance with other state of the art feature extraction models, the complete set of *CalTech*101 [15] database was used for multi-class object categorization. The standard method of splitting the training set of images into 15 and 30 images per class was applied. There are some limitations to the *CalTech*101 dataset: The uniformity of data with most of the images center aligned makes learning less challenging. With small number of images in certain categories, the largest training size is limited to 30 images. And the presence of artifacts, which appear due to image rotation or scaling [68]. Many different models, such CNNs and HMAX, has been previously evaluated on this dataset, so for comparisons the CalTech101 was used.

The ISA model was evaluated since it is comparatively faster to train than TICA. In the previous experiments, feature length of 400 was used (which refers to the number of S_3 units). But for a larger database with total number of 9144 images, the feature length was increased to 1000. The number of S_1 and S_2 layer filters was 144 and 300 respectively. The corresponding subspace sizes were $Z_1 = 9$ and $Z_2 = 5$. Although for better results, a larger number of dictionaries in each layer is more beneficial (figure 4.15), these parameters allowed for a faster computation time.

The sample size for learning dictionaries at each layer was 50,000. The dictionaries (or filters) were learned from just 10 images from each category. Learning the 1000 S_3 layer high level filters was the most time consuming part of the model. Therefore, the total learning phase of the model was approximately two hours. The average inference time per image for V_1, V_2 and V_3 layers were 1.5, 1.1, and 1.2 seconds respectively.

Current state of the art models has achieved very good results for the *CalTech*101 database. The list of models in this section are mostly biologically motivated hierarchical models based on the HMAX model. Most of the reported accuracy of these models were the result of varying length of features. For example, in the HMAX model in [54],[75], classification using a dictionary of 4075 features had an accuracy of 54% [16]. In [16], by increasing the scale depth of the $S1$ units, an accuracy of 61% with 4080 features was reported. In [10], an accuracy of 73.67% was achieved for training size of 30 for feature length of 21,504. A further increase of 76.13% was reported for a feature length of 43,008. In [28], an unsupervised two layer model with sparse coding and pooling was developed which also achieved a high classification accuracy of 74% with codebook of 4096 features.

The type of pooling techniques at the highest layers in all these models were also different. In [10], spatial pyramid pooling [76] was applied on the high level features with a grid size of 4,2 and 1. Global and localized maxima were pooled in [54] and [74] respectively. In [11], the model in [16] was extended with localized pooling at multiple resolutions that resulted in increased classification accuracy.

In the ISA-HMAX, L2 pooling of global spatial information of S_3 response forms the final feature vector. Similar to the models in [10] and [64], the larger receptive field size at S_3 covers almost the entire image such that the pooling occurs over a very small set of values. The Liblinear [101] classifier was applied at this stage due to the larger number of categories.

With a dictionary size of 1000, classification accuracy of 54.20% for training size of 15 images and 62.30% for 30 images per category was achieved for the entire data set.

Since the number of features were too small, the feature length was increased sampling patches of C_2 and C_1 responses. The position of the samples were kept constant for all the input images. This method, however, did not improve the classification accuracy. Therefore, another set of 1000 features was learned from

the C_2 layer of the model (figure 4.33). The subspace size was 10, similar to the previously trained 1000 units. The S_3 layer was trained in two separate runs. With this new dictionary size of 2000, the classification results improved considerably. The results were obtained from an average of 10 independent runs. An increase in accuracy was observed with $61.99\% \pm 0.42$ and $70.29\% \pm 0.33$ for 15 and 30 training sizes respectively.

In the experiments using 10 categories of objects, the number of images in the test set was even. With the entire *Caltech*101 dataset since there is a widely varying number of images, the average accuracy for each category was obtained. The final result was calculated as the mean of all the class specific accuracy rates. In this case, an accuracy of $52.40\% \pm 0.32$ and $60.03\% \pm 0.15$ was achieved for 15 and 30 training sizes respectively.

For the ISA model, these classification results were obtained from only 2000 S_3 outputs, whereas the other models had dictionary sizes of at least 4000 high level units. Thus, with increase in the number of *S*3 filters from 1000 to 2000, a jump in classification accuracy was observed. Since most of the hierarchical models listed in table 4.8 have a feature length of at least 4000, the same process in figure 4.33 of learning extra 1000 filters was repeated.

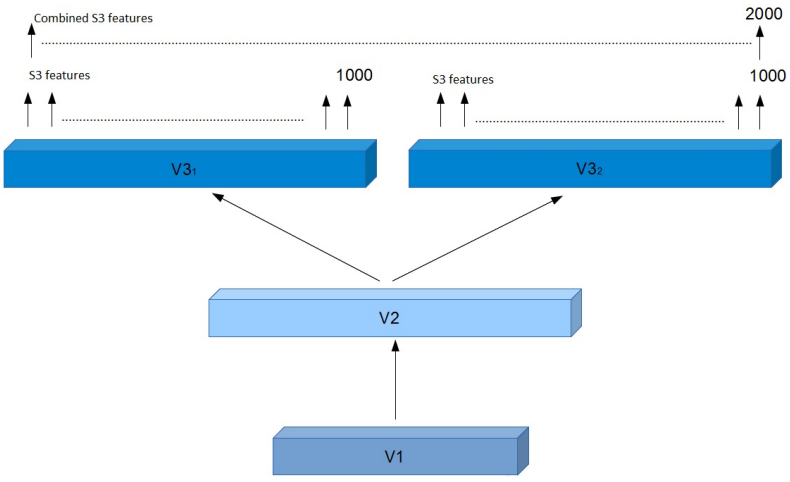


Figure 4.33: S_3 units were learned in two separate runs

The resulting accuracy is higher than the unsupervised learning models listed in the given table. Compared to the Adaptive Deconvolutional Networks [64], which uses 4 layers for feature extraction the accuracy of the ISA-HMAX model

with 4000 features is much higher.

Classification accuracy for number of 15 and 30 training images per category		
Model	15 images	30 images
Serre [54]	35	42
Mutch & Lowe [74]	48	54
HMAX-S [16]	54	61
HMAX-S (extended) [11]	68.49 $\pm$ 0.75	76.32 $\pm$ 0.97
Lee et al. [37]	57.7 $\pm$ 1.5	64.5 $\pm$ 0.5
Zeiler et al. [64]	-	71 $\pm$ 0.10
Yu et al. [28]	-	74.0
Sparsity regularised HMAX [10]	68.98 $\pm$ 0.64	76.13 $\pm$ 0.85
ISA HMAX (dictionary size 2000)	61.99% $\pm$ 0.42	70.29% $\pm$ 0.33
ISA HMAX (dictionary size 2000, average of per category classification rate)	52.40% $\pm$ 0.32	60.03% $\pm$ 0.15
ISA HMAX (dictionary size 4000)	72.65% $\pm$ 1.08	79.70% $\pm$ 0.55

The ISA-HMAX model however, lags behind most convolutional neural networks when it comes to classification accuracy. For example, the supervised CNN trained in [63] demonstrated an accuracy of 72.6% $\pm$ 0.1 for 60 training images per category on the Caltech-256, which is a much more difficult dataset to learn. For the Caltech-101, the performance accuracy for 30 training classes was 85.4% $\pm$ 0.4. Also, the model was pre-trained with the Imagenet dataset, which contributed heavily towards the improved classification result.

To examine the ISA-HMAX model for larger number of images, the Caltech-256 database was trained with the same model parameters as the Caltech-101. Aside from the complexity of images , the number of categories and images are larger. The model was thus evaluated on a feature set of 4000, but contained a lesser number of mid-layer $S1 = 100$ and $S2 = 150$ filters for a faster inference time. For learning the filters, 30 images from each category was used. The resulting accuracy for 60 training images per category only reached 35.35%, which is much lower in comparison. This could indicate that for larger datasets, the number of filters within the layers need to be higher.

4.9 Summary

In this chapter, an enhanced form of the HMAX model is presented. By applying ISA and TICA algorithms for learning both low and high level features, simple and complex cell properties in a three layer operation was designed. These layers performed linear filtering for feature selectivity, L2 pooling, which represents phase invariance and max pooling to introduce position invariance. It was demonstrated that the added non-linearities of L2 and max pooling contribute towards an improved feature learning.

Comparison of ISA and TICA regulated models highlighted certain limitations and advantages of both algorithms. The advantage of TICA stems from its biological plausibility, since its topographic arrangement closely resembles the retinotopic organization of receptive fields in the cortex. Evaluation of its S_3 layer units for feature detection also demonstrated a higher detection ability than the ISA. However, in terms of multi-class categorization, its performance lagged behind ISA and other HMAX models when the pooling neighbourhood size was too large with respect to the topography size. Compared to TICA, the ISA version of the model performed displayed a much higher accuracy. In addition, the learning speed was also much higher as it is a much faster algorithm.

One explanation behind this occurrence could be due to the widely varying range of phases values of the components within a neighbourhood, unlike the ISA, where the components within a subspace are phase-shifted [32]. Invariance properties usually arise when the features are pooled over a range of slightly shifted variations. With larger neighbourhood sizes, there is a larger variation in orientation and frequency which becomes even more evident in the high level features. It was also discovered that while pooling the filter outputs within an area, overlapping areas can cause the classification accuracy to reduce. In terms of multi-class categorization with 10 classes, the accuracy of the ISA-HMAX model was much higher than either ICA or the S-HMAX (from [16]). The highest accuracy for a training size of 30 images was found to be 94.3%. Its performance on the Caltech-101 data was also an improvement over other unsupervised feature extraction models.

Another advantage of ISA and TICA in comparison with ICA, was the dimension reduction of the data after each layer. This made the learning of higher order features much easier than the ICA models since the is sampled directly from the layer below.

One drawback with these algorithms is that training speed is quite slow in comparison with HMAX models. Learning the three S layer feature vectors took much longer than extracting prototypes with random sampling. The inference time, however, was quite fast and dependent on the high level filter sizes. The model in figure 4.33 took an average of 3 seconds per image.For smaller $S3$ size of 400, it was an average of 1.5 seconds per image. Another area improvement could be in terms of integrating faster and overcomplete learning algorithms.

Due to the variability of feature sizes with different sizes of $S1$ filters, pooling over responses of multiple spatial resolutions did not yield favourable results. Therefore, the scale invariance property was not modelled. Thus, scale invariance is an area that needs to be addressed in future works.

Even though most models with the smallest subspace size ($Z = 2$) had the best accuracy, there were many cases where a larger size performed better. Therefore, this variation in classification accuracy depending on the subspace size calls for the need of data adaptive subspaces rather than a fixed size.

The above models only represent a feedforward mechanism which only models a small fraction of the visual cortex functions. Feedback signals that modulate responses of lower layer neurons are an inherent part of the perceptual mechanisms. One explanation behind the feedback connections is the attentional modulation mechanism that eliminates redundant information by focusing on the most salient regions of a visual scene. Vision models with saliency has already been developed with high accuracy in object classification [102]. In the following chapter, the application of saliency modulation to the hierarchical model for improving efficiency of feature extraction is examined.

Chapter 5

Enhancing Object Recognition with saliency Maps

5.1 Introduction

In chapter 4, unsupervised learning algorithm was applied in each S_K layer of the model which involved sampling of patches from the C_{K-1} layer. Large number of samples ensures a better probabilistic representation of the input data, but also slows the down the learning speed. Increase in the number of categories of training datasets would thus require an even higher number of samples. For learning from a large set of images with limited amount of samples, the process can be optimized such that only the most salient part of the images are sampled. In this chapter, both the low and high level features of the images will be utilized to form a self regulated attention-recognition framework. In addition, some existing saliency maps will also be combined with the model for comparison. Efficiency of the saliency modulated HMAX will then be evaluated based on its classification accuracy performance.

5.2 Saliency models

Attentional modulation is one of the mechanisms that greatly reduces the redundancy in input data that enters the visual stream. By prioritizing, the brain is able to process the vast amount of incoming information rapidly [103]. As an integral component of the cognitive framework, it has been a topic of extensive study in neuroscience as well as psychology which has provided the foundations for current models in computer vision. The most fundamental one being the *Feature integration theory* [34], based on which, a *saliency map* generating algorithm was proposed by Koch and Ullman [104] and subsequently implemented by Itti et al. [17]. Based on the Itti model, similar feature combination methods has been

adapted in many of the saliency models that followed. In this method, generally the types of low level features, such as orientation, intensity or colour, and its integration techniques play an important role but newer advancements have also included high level features such as object shapes [105] for generating saliency maps.

The term *saliency* has been defined as the external stimulus driven bottom up component of the whole attentional process [35] [17]. Most of the current models in computer vision deal with this feedforward component of the attentional mechanism. It is characterised by involuntary response towards the statistical properties of the visual scene as opposed to the top down mechanisms which are task driven [35]. Studies have linked this process to the neural activities in the V1 layer of the visual cortex [106]. In this chapter, bottom-up saliency based attentional modulation will be integrated on the hierarchical vision models.

5.3 Background: Saliency Map Algorithms and hierarchical models

The standard model from which most current algorithms are derived from was proposed by Itti and Koch, in which salient areas were localized in a bottom up process. This model is categorised as *cognitive* type, which applies to most other models (based on the feature integration theory) to a certain degree [35]. In this model, feature maps of input images of different scales are generated by Difference of Gaussians (DoG) operations (that compares average value of center with average surrounding value) on colour, intensity and orientation channels [107][17]. After that, for each channel, the feature maps are combined across scales and normalized. The maps in the channel are then linearly summed and again normalized to form 'conspicuity' maps, which are again linearly combined to form the saliency map. This type of model generates a bottom-up type attention where the salient region emerges from the low level information of the scene [107].

Models based on Feature Integration theory

Several frameworks based on the Koch-Ullman model has been developed after the Itti model, such as the Saliency Toolbox [108], the C++ Neuromorphic Vision Toolkit, iNVT which contains ongoing improvements on the original algorithm

[17] [107]. Upgrades include contour integration [109], top-down influence by maximizing signal-to-noise ratio of target versus distractor [107]. In the most recent update, they have implemented a fusion of different types of state-of-the art saliency models, resulting in greater accuracy than individual saliency models [110]. Evaluation of the saliency models are usually carried out with ground truth maps which either eye fixation maps or manually tagged targets or a combination of both [35].

The guided search theory proposed that attention is directed towards regions of interest by varying the weights of the combination of features [111]. An implementation of weighted linear combination of different feature maps was developed in [105] where the coefficients were trained with ground truth maps of eye tracking data. The combined saliency map showed higher accuracy than the current models at the time. Similar type of supervised models such as [112] [113][114] [115] were also trained using existing databases of human eye movements or with very high accuracy in predicting human attentional behaviour. For the current vision models however, the focus was more on the unsupervised saliency models which are dependent on statistical properties of the input data.

In the designs based on the Itti model, the different feature maps are obtained independently and fused. A similar fusion of independent feature channels based on natural image statistics was implemented using a Bayesian framework in [20] called the *SUN* saliency model (Saliency Using Natural Statistics). Here, bottom-up saliency was generated from the self-information of the natural images, similar to the models in. It also included a method for merging top-down influence to predict attentional direction. Bayesian model of saliency was earlier implemented by Torralba [18], [20][35], where a joint probability of the presence of target and location (given target was present) was formulated. The difference from SUN model was that probability was estimated for the object being present in any location of the scene versus being present at each point of the visual space [20][35]. It was noted in [20] that increasing the area of search to the entire image turned the equations of the model the same as that of Oliva et. al. The advantage of the *SUN* saliency model is that the parameters of the filters could be learnt in a completely unsupervised manner from natural images. Similar type of models are thus suitable for integrating with the unsupervised HMAX models from the previous chapter.

SUN saliency model

The SUN model was implemented by using both Gabor filters or Difference of Gaussians (DoG) and ICA. The saliency at a point z is defined as the probability that the target C is salient given the features fz observed at location lz.

$$sz = p(C = 1|F = f_z; L = l_z) \tag{5.1}$$

With the assumption that feature and location are independent,

$$sz = \underbrace{\frac{1}{p(F = f_z)}}_{bottom-up\ influence} \cdot \underbrace{(F = f_z \mid C = 1) \cdot (C = 1 \mid L = l_z)}_{top-down\ influence} \tag{5.2}$$

The log probability estimation gives,

$$\log sz = \underbrace{\frac{1}{p(F = f_z)}}_{bottom-up\ influence} + \underbrace{(F = f_z \mid C = 1)}_{Log\ likelihood} + \underbrace{(C = 1 \mid L = l_z)}_{Location\ prior} \tag{5.3}$$

The bottom-up term was described as *self information*, which emerges from the unique local characteristics of the image. The areas that show the greatest variance in the orientation, intensity and colour are least common and draw the most attention. At this point there is no preconceived target and the attention is towards the general visual field.

The log-likelihood term affects the saliency after knowing the class of target which determines the features associated with it. The location prior gives the probability of the location of the target, given the class of the object is known. The latter two terms describe the functionality of the top-down influence which comes into effect after gathering information about the target properties. Without prior knowledge about the target, the saliency map is generated only from the first *self information* term, discarding the top-down influence.

The self-information term is calculated in the form of linear filter responses. With the Difference of Gaussian method, filters of different scales on intensity and colour channels are combined with the equation 5.4.

$$\log sz = -log p(F = f_z) = \sum_{i=1}^{N} \left| \frac{f_i}{\sigma_i} \right|^{\theta_i} + const \tag{5.4}$$

Where N is the total number of filters, σ and θ are the shape and scale parameters of each filter and f represents the filter response.

The Gaussian filter responses were assumed to be independent of each other and added to form the saliency map with the equation although in reality they were found to be highly correlated [20]. The same condition applies to the ICA filters, where some correlation due to natural image statistics appear in practice. The same equation 5.4 applies for the ICA filters, where the term f represents the filter response of each ICA component. Number of components are restricted to the dimensions of the colour image such that $k = p \times p \times 3 - 1$ where, p is the width of the square patch. In [20], the KL-divergence criteria for evaluation produced better results for ICA than DoG filters.

Top-down models

Studies have suggested that higher order features and top-down mechanisms form a significant part of the attention process [35]. It is also described as a slower and voluntary phenomenon [116]. The top-down saliency in many models is usually combined after there is some prior knowledge about general about the target characteristics [20][35]. Others models add context information to the bottom-up saliency maps [18]. In [18], both local and global feature maps were generated in two parallel pathways, which were then combined to form the final saliency map. For integrating attentional modulation in a hierarchical vision framework, the top-down mechanisms are an important component which can model the feedback connectivity between the different layers. Such a working model can bring new insights into the feedback mechanisms of the visual cortex as well.

Current state of the art models

The MIT saliency benchmark was developed in order to rank the saliency algorithms with respect to baseline maps of human fixation data [117][107]. According to their metrics, Judd et al. [105] and Graph-based visual saliency (GBVS) [19] models displayed high accuracy (among the Itti based models) in predicting human eye fixations whereas the SUN model falls behind in comparison. Models incorporating blurrier saliency maps and center bias have have generally shown better accuracy under their evaluation criteria [107]. Recent advancements in the field of deep convolutional networks has achieved even higher performance scores than the above two models. The new frameworks, *Deepfix* [118] and *Deep*

Gaze [119][120], which use learned high level features with a feedforward hierarchical network, have demonstrated a high accuracy in eye fixation predictions. The scope of attention models is vast with many different approaches towards generating saliency maps. But since the natural statistical approach was applied in the vision models in chapter 4, similar models will be explored that rely on feature integration method of producing saliency maps.

Besides SUN, sparse representation models such as TICA [114] and PCICA (Pairwise cumulant ICA)[121] algorithms were also applied for modelling saliency. Both models, included the L2 feature pooling operation described in equation 4.19 followed by linear combination of the resulting feature maps. In [114], the responses of the TICA filters were combined with weighted summation method where a two stage supervised learning was adopted to train weights using eye fixation maps. Although it was reported to predict attention with high accuracy, supervised training models were excluded as they are not biologically plausible. The PCICA technique in [121] generated overcomplete set of filters which was convolved with the image, pooled according to similarly classified filters, processed with center surround DoG filters and linearly combined. The generated saliency maps were applied in similar manner, but using ISA on both V_1 and V_2 layers of the model (figure 4.9).

In the recognition model of chapter 4, the first layer comprised of linear filters learned by ICA, ISA or TICA algorithms followed by non-linear pooling operations. The next layer contained higher order feature detectors that were learned from the previous layer outputs. One drawback from just using linear filters is that saliency gets shifted towards highly textured regions [20]. In [20], non linear transform of DoG filters was suggested to address this problem. It was observed that although non-linear functions (also applied in [121] in the form of square pooling) are highly effective in highlighting distinct edges, it is not sufficient for extracting global information of the images. Since the localized saliency maps obtained from linear features have this limitation, these higher order feature detectors can be incorporated for a global feature map similar to the contextual guidance model by [18].

5.4 Saliency modulated object recognition

Recent studies have demonstrated that the ventral and dorsal streams of the visual cortex are interconnected[122] which confirms that saliency and attention form an integral part of cognitive process. In the HMAX models, the units in higher layers respond to high level features such as object shapes. To learn these features in an unsupervised manner, samples of data from the lower layer are extracted, which is usually carried out in a random manner. Randomized sampling is sufficient for learning lower level features, like edge detectors, as their presence is universal. But for features of high complexity, it is important to discard redundant information. Increasing the sample size reduces probability of error, but not very practical. This would require an even larger sample size for high number of training images.

In [82], an enhanced HMAX model was proposed where the patches from the C layer were extracted with the help of saliency maps. The saliency maps, which were produced in parallel with the vision model, was inspired by the Itti and Koch model. The features from orientation, intensity and colour channels were combined together after processing them in parallel streams. The patches of multiple scales, extracted with the template, were then grouped into different clusters such that the memory for similar features were shared. This type of grouping was inspired by the memory processing ability of the $V2$ and IT [82]. To categorise the different patches, they adopted an unsupervised iterative clustering algorithm.

In the ISA and TICA models from chapter 4, similar patches are also grouped together such that the memory required to represent middle layer features is lower. However, the method of grouping is completely different to that of [82]. In the ISA model, saliency maps are applied for learning the filters at the S_2 and S_3 layers using the ISA algorithm. The S_1 filters are involved in generating bottom-up saliency maps using the feature integration methods similar to [20] and [121]. The feature maps are comprised of orientation filter responses categorised into subspaces according to their energy correlations. Applying ISA on colour images groups the colour components into separate subspaces (figure 4.4). In this way, both orientation and colour feature maps can be extracted from the image.

V1 layer saliency maps

The basic structure of the model is same as figure 4.9. Each stage of the V_1 layer, (S_1, C_{1_a}, and C_{1_b} from the previous chapter) contribute towards the formation of saliency map.

S_1 **layer**: Patches of receptive field width p from the image are multiplied with each of the set of linear filters W.

$$S_1 = \langle W, X_p \rangle \tag{5.5}$$

Where, X_p is the set of patches extracted from the image. The model in [121] applies convolution at this stage. Here, both methods are equally applicable, equation 5.5 was applied to keep it consistent with the vision model. (Figure 5.1 shows the saliency maps obtained after applying subspace pooling (in C_{i_a}) for both convolution and multiplication methods.)

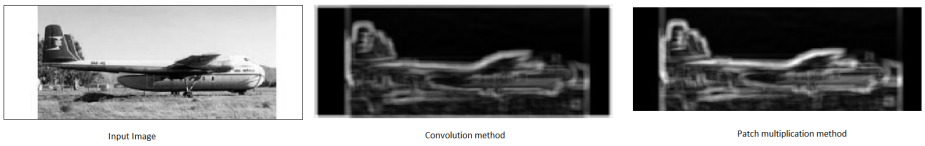


Figure 5.1: Convolution and multiplication method. (Input image from [15])

The saliency maps in figure 5.2 are obtained from linear combination of all the feature maps. Since the output of linear filters are highly sensitive towards dense textures (figure 5.2, second row), non-linear functions are applied [20][121]. The type of non-linearity plays an important role in the outcome of the saliency maps. In this model, both L2 pooling of subspace responses and max pooling of local spatial responses was applied.

C_{1_a} **layer:** All the $S_{i \geq 0}$ values within the subspaces of size Z_1 are pooled with equation 5.6.

$$C_{1_a} = \sqrt{\sum_{j \in Z_1} {S_{1_j}}^2} \tag{5.6}$$

Where the total number of feature maps is $\tilde{R}_1$. From figure 5.2 (third row, third column), it can be seen that the most prominent edges of the image are retained, while the high density textures are suppressed.

C_{1_b} **layer:** From each of the feature maps in C_{1_a}, the strongest response from within a local area is allowed, which form the conspicuity maps. Mod-

elling of this competition between neurons is usually carried out by 'Difference of Gaussian' operators at at multiple scales[121]. This type of center-surround process has been found to occur in both the LGN and V1 areas of the visual cortex [121][35][123][124]. In this model, instead of DoG, max pooling on non-overlapping local areas of size $r_i \times r_i$ was applied.

When applied directly on the S_1 layer output (figure 5.2 fourth row, first column and fourth row, second column), the resulting saliency map also suppresses high density textures, but to a lesser extent compared to that of the combined action of S_1 and C_{1_b}. The salient areas are also more scattered in comparison.

A simple linear combination of normalized feature maps was applied to form the saliency maps in figure 5.2.

$$A_b = \sum_{j=1}^{\tilde{R}_1} C_{1_b} \tag{5.7}$$

Where A_b is the bottom-up saliency map formed by the V_1 layer specifications.

Such methods are usually not very robust as they are likely to highlight the background textures and suppress the maxima of the feature maps [121]. Therefore, it is more beneficial to use weighted combinations or the iterative method from [125][121].

In this way, the properties of the existing object recognition system are utilized where the bottom-up saliency begins at the $V1$ layer of the model.

Saliency enhanced HMAX model

Figure 5.3 illustrates the V_1 layer of the HMAX model. The C layer indicates the joint operation of C_{1_a} and C_{1_b} steps. From the map obtained in the C_1 layer, which can be referred to as the V_1 saliency map, patches were sampled from the locations of highest saliency. These samples were then used for training the S_2 level filters (figure 5.3).

In the next V_2 layer, the same procedure is applied to form the saliency maps with the higher order feature maps of the S_2 layer. Again, the most salient patches of the C_2 outputs are sampled for training the S_3 filters.

The maps for some other saliency models are illustrated in figure 5.4. The ISA_1 refers to the saliency map without applying the pooling function at C_{1_a} and ISA_2 represents the saliency map applying the functions of both the stages of the C_1 layer. Out of the listed methods, the GBVS [19] was found to perform best

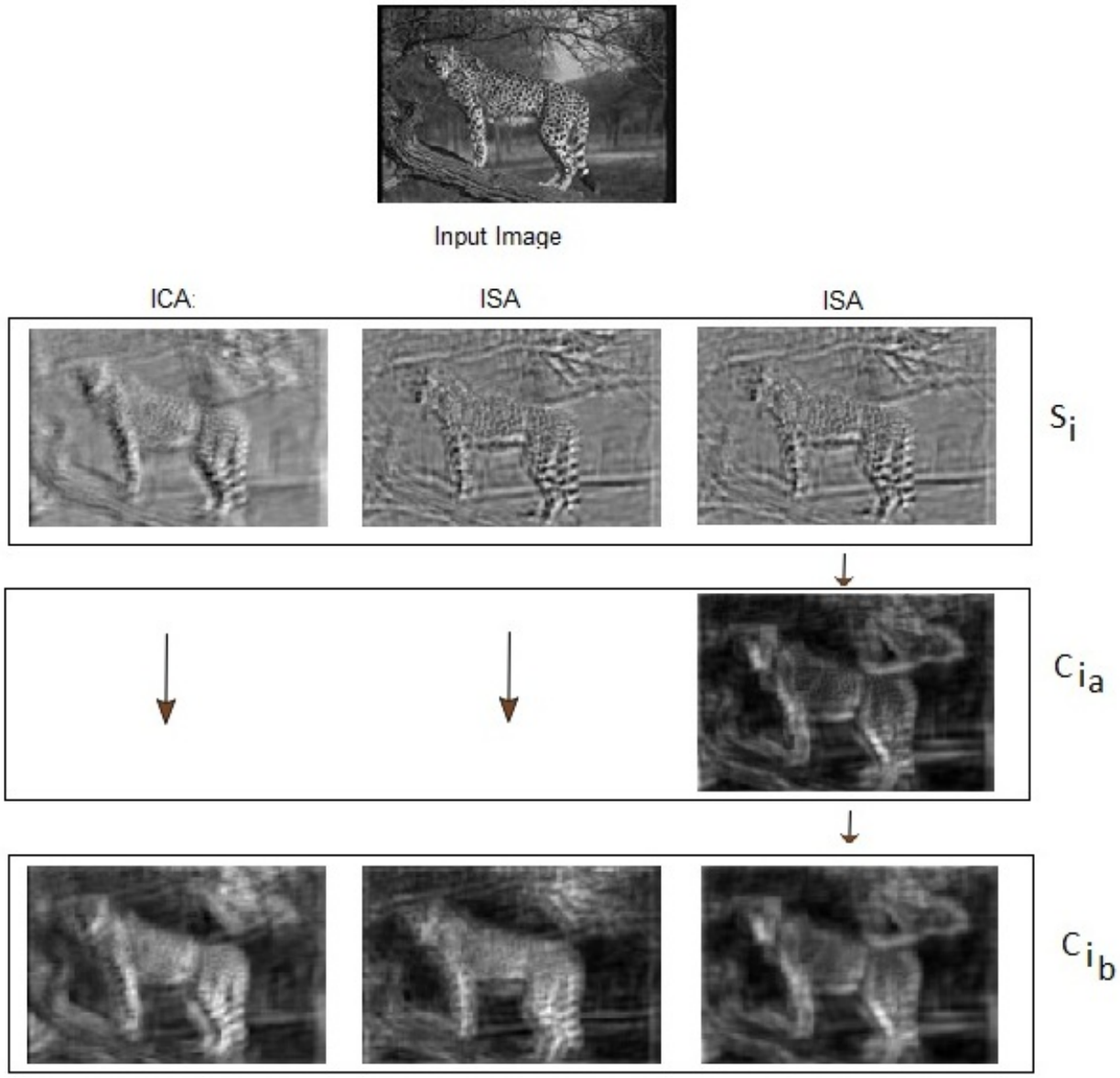


Figure 5.2: Saliency maps from S and C layers of ICA and ISA models (Input image from [15])

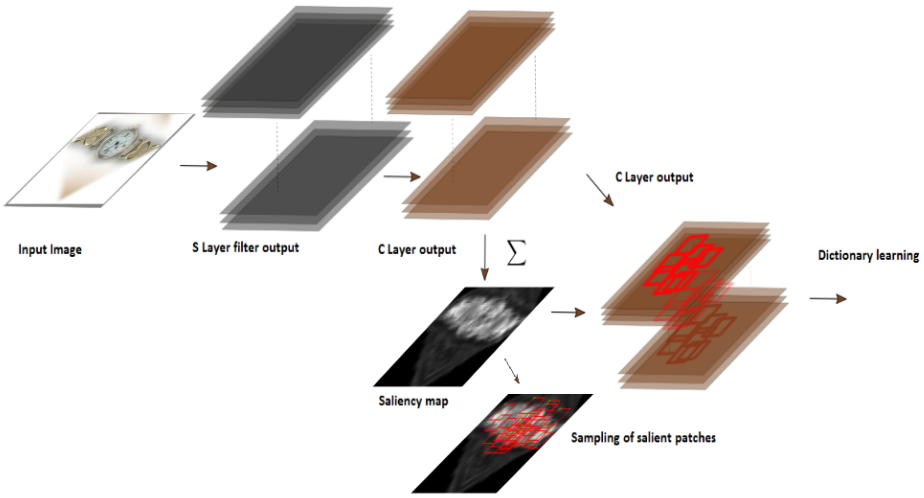


Figure 5.3: HMAX with Saliency (Input image from [15])

at predicting human attentional behaviour according to the MIT benchmark. In a lot of cases, operations such as border attenuation and center bias improves the prediction accuracy considerably since human attention tends to gravitate towards the center of any visual scene [107]. This may not hold true for the object recognition model as the target could be present at any random location.

Figure 5.5 shows the samples of data from salient regions after the C_1 and C_2 layers.

When using the saliency methods from [19],[20],[17] and [126], these maps were applied directly on the images. For example, the GBVS algorithm was used for generating a saliency map, which was then applied on the input image before the S_1 layer to suppress non salient areas and highlight the salient areas. It was observed that when these maps were used only as a template for sampling the patches, the classification accuracy did not improve. Thus, it is essentially a combination of the external saliency maps and the ISA saliency method.

To sample the patches, the method from [105] was adapted, where only the top 30% of the salient areas were extracted. To account for any discrepancies in the detection of objects, a small percentage of the samples could be reserved for either the center or any random area, but this technique was not applied in the experiments.

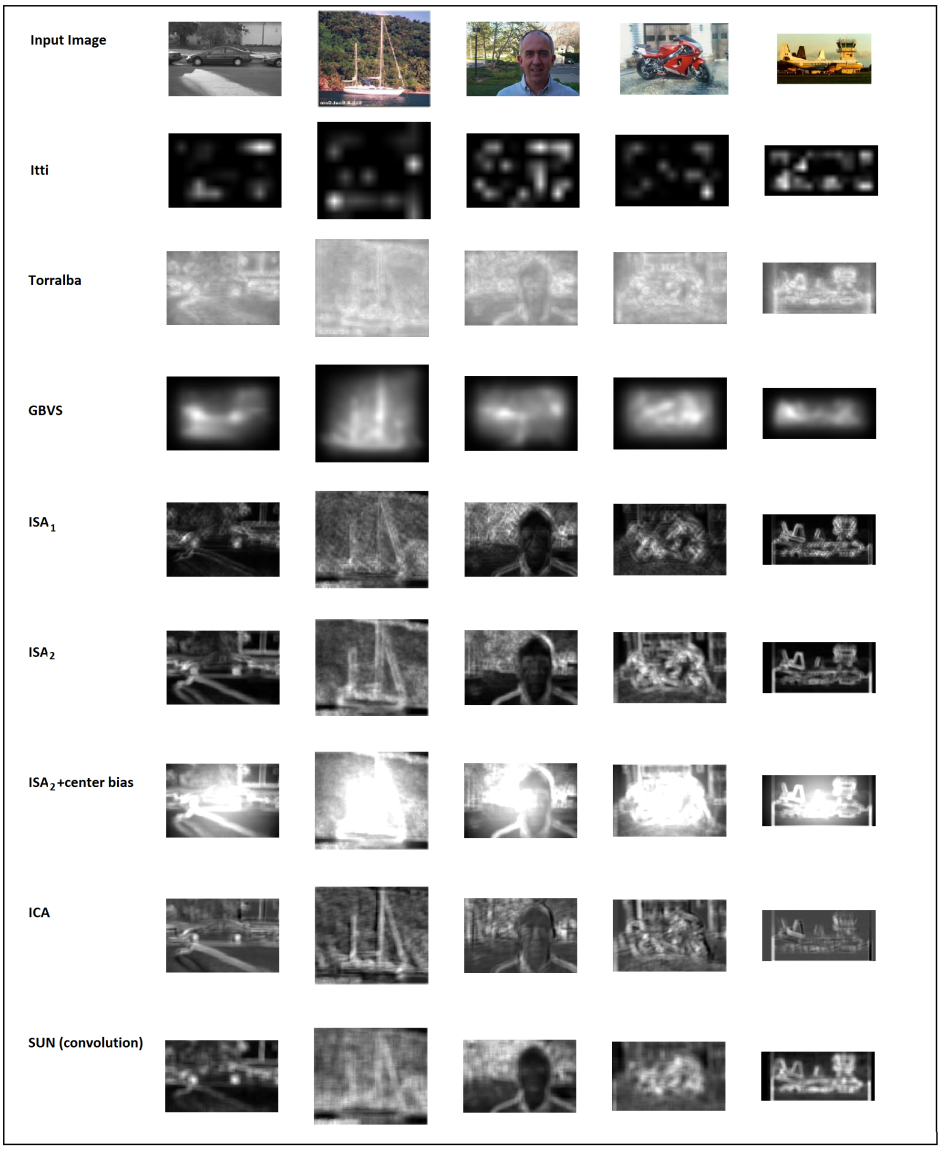


Figure 5.4: Saliency maps for different saliency algorithms, from rows 1 to 9: Input images [15], Itti and Koch [17], Torralba [18] , GBVS [19], ISA_1 without $C1_a$ non-linearity, ISA_2 including $C1_a$ non-linearity, ISA_2 with center bias filter , ICA, SUN [20]

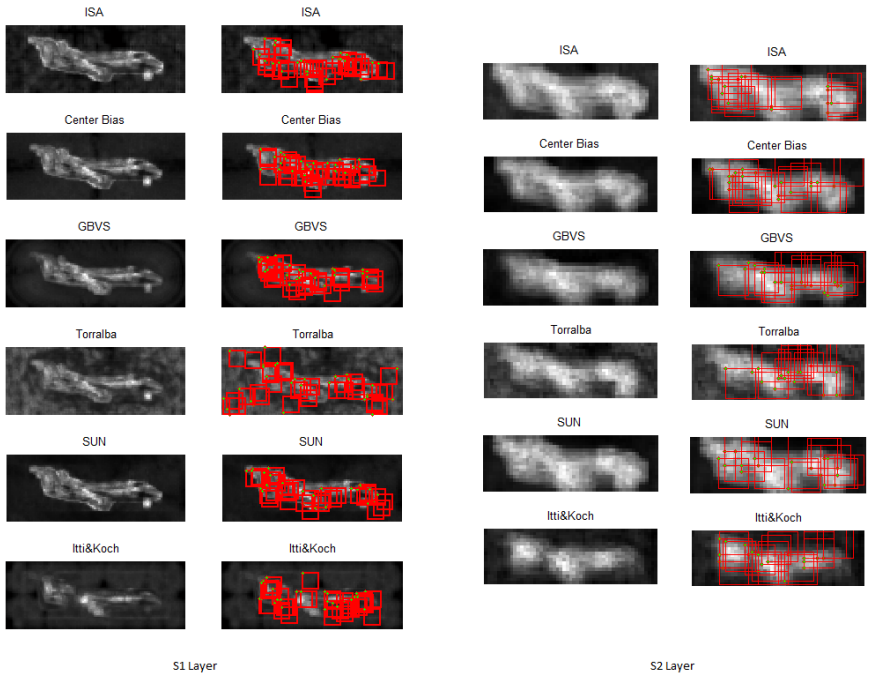


Figure 5.5: Saliency maps from V_1 and V_2 layer outputs after applying the different algorithms, (Input images from [15])

5.4.1 Evaluation

For this experiment, a set of 300 images belonging to 10 categories from the Caltech101 dataset were used for training. The receptive field sizes of the first, second and third layer were fixed at 11, 10 and 9. The specifications for the number of filters using ISA is listed in table 5.1. Z_1, Z_2 refers to the subspace size. Colour images were used after converting to LMS colour space such that the maximum number of possible S_1 filters is larger.

Table 5.1: Model specifications

Models	V_1, $p_1 = 11$			V_2, $p_2 = 10$			V_3, $p_3 = 9$
		C_{1_a}			C_{2_a}		
	S_1 (R_1)	Z_1	$\tilde{R}_1$	S_2 (R_2)	Z_2	$\tilde{R}_2$	S_3 (R_3)
ISA	144	9	16	100	4	25	400

The main difference of this model from the ones in chapter 4 is the number of sampled patches from the C_1 outputs, which is reduced to 10000 (from the earlier 50000), which greatly increased the learning speed. Sample size is further reduced to 5000 for the C_2 layer. The figure 5.6 shows the classification accuracy of the saliency enhanced model when reduced number of samples are used.

The other applied saliency maps are in combination with the already existing ISA method. Figure 5.6 shows that the saliency modulated models perform classification with better accuracy than the model with random samples. Comparing with the other saliency methods, the performance of the Itti & Koch model is much lower, whereas the SUN performs on par with the non-regulated model.

Figure 5.7 shows the accuracy of the same model, but with $50,000$ random samples per layer. The performance of saliency regulated models with much lower is much closer to the model with an increased number of random samples, than the model with no saliency (figure 5.7 , black line). Although not ideal, with improvements in saliency maps and integration techniques, higher performance can be achieved.

On of the contributing factor for lower performance is due to the limitations of the saliency map. Incorrect samples due to poor saliency maps can reduce the performance of the classification model (figure 5.8).

The maps in figure 5.8 display the limitations of the linear combination method that was applied in this model. Thus, it is important to adjust the V_1

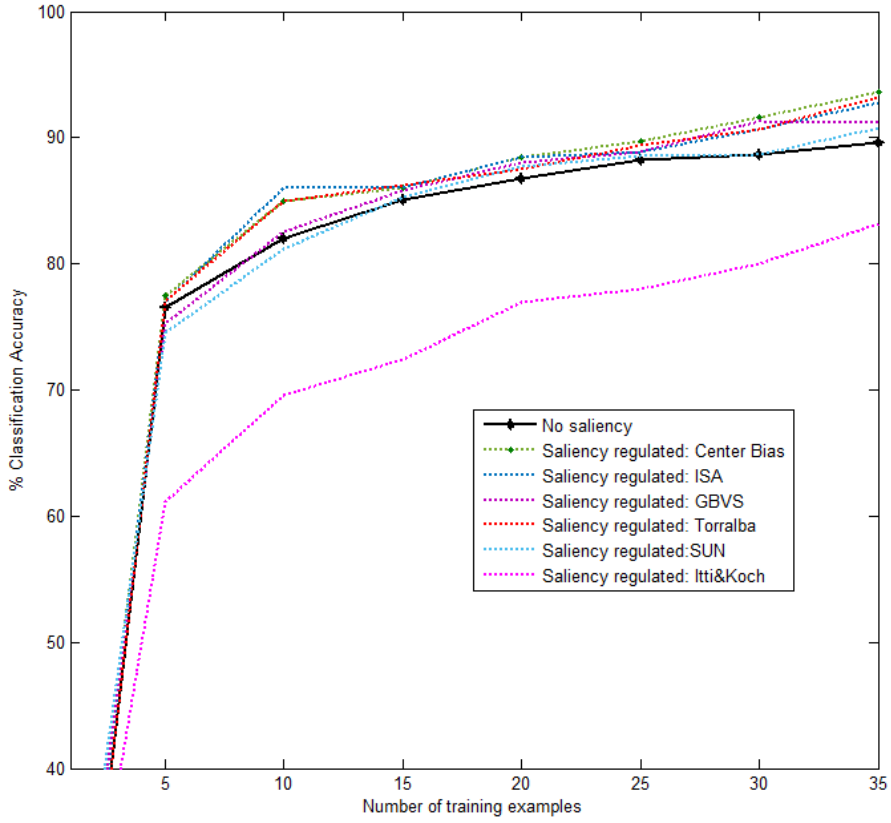


Figure 5.6: Classification Accuracy for saliency enhanced models

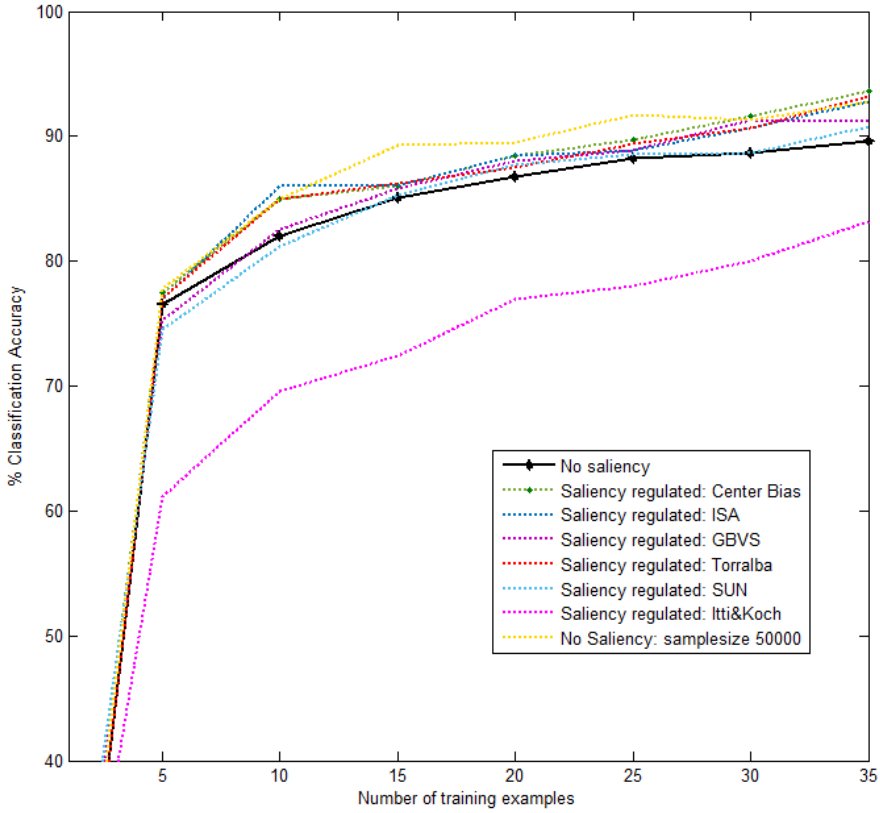


Figure 5.7: Classification Accuracy for saliency enhanced models

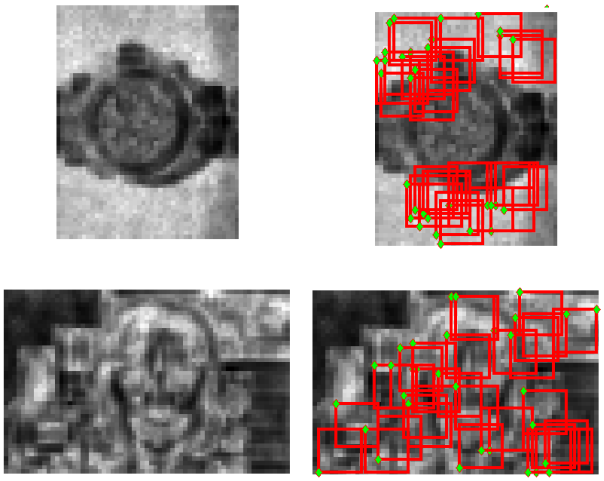


Figure 5.8: Incorrect data samples (Input image from [15])

layer outputs to prevent loss of information. The modified V_1 output would then become the new input for the V_2 layer. Instead of modulating the feature maps with iterative summation methods or supervised training of weights, another possible method of feature map combination is through the influence of top-down cues [127][128][35].

5.5 Top-down model

The feature integration theory suggests that interaction of low level features occur at the pre-attentive stage, after which focused attention follows [34]. Attention models based on this theory demonstrated that interaction between seemingly independent features highlights parts of the scene in a bottom-up mechanism. It was observed have seen how saliency maps can optimize the learning of dictionary features by adaptive sampling, but algorithms using just orientation, colour and intensity maps can be limited in its detection of high level features. In the model in the previous section, higher order feature detectors were applied in the form of S_2 filters. Its outputs were combined to form the saliency maps at V_2 but the S_2 filters were not able to detect any areas that were suppressed in the V_1 layer.

Since unsupervised and data adaptive methods are preferred, weights should be updated according to prior information from the model itself (or any external source), which would form the top-down attentional mechanism. The saliency maps can then be updated as in equation 5.2[18]. Before training the HMAX model, no such prior existed, so the initial attention was the result of the S_1 filters. As observed in 5.5, they resembled edge detectors and responded to low level properties of the image. But after obtaining the S_2 and S_3 filters, it is possible to direct a top-down mechanism to update the response of the V_1 layer. Thus, the influence of learned higher order feature detectors stored in memory would become the prior in equation 5.2. With this method however, the class of the object is not yet known so that the process is still involuntary.

In previous hierarchical models, the higher (S2 or S3) layers never interacted directly with the input image. These higher layers of HMAX models has been compared with the $V2$ or $V4$ layers of the visual cortex [8][82]. Although there is no evidence for any direct connection between $V2$ and the LGN (which relays retinal signals), feedback signals from $V2$ and higher order areas up to the IT are

said to modify the response of lower order areas such as the $V1$ [35]. Feedback connections are said to modulate the activity of lower layer neurons to distinguish object from background [129].

The role of context on the modulation of local contrast response of the $V1$ layer has been a topic of interest in the field of both neuroscience and computer vision [18] [130]. Global structure information stored in memory is one of the components that form contextual information [130]. Oliva and Torralba, defined a feedback method to integrate top-down influence for generating a saliency map. In their model, the local and global pathways of feature detection occurs in two parallel streams. The local pathway represents the local features involving bottom up saliency. The global features represents the scene as a whole and modulates the saliency at the local feature level. These were built from pooling together the low level feature detectors across multiple orientations and scales and applying PCA compression. The mean of the global features at coarse spatial resolutions were used to estimate the structure of the scene. Segmentation based on global structure of the image facilitated the saliency by suppressing the activation of locations with low probability.

Applying higher order filter on input image (top-down saliency map)

Based on the model described in [18], contextual information based on the learned S_2 and S_3 layer filters is considered. Here, the interaction of S_2 layer filter directly on the image to generate saliency maps with contextual information is described.

The dimension of each S_2 filter depends on the size of its receptive field and the number of filters or groups of filters in the previous layer. So, for a receptive field width of p_2 and number of S_1 filter responses R_1, the dimensions of each S_2 filter are of size $p_2 \times p_2 \times R_1$. Here, each square filter of size $p_2 \times p_2$ is represented as w_2. Then, for a single S_2 filter and a given input image X_p, a feature map is obtained as,

$$S_{1t} = \sum_{i=1}^{R_1} \langle w_{2i}, X_p \rangle \tag{5.8}$$

The subscript t stands for top-down interaction (since the saliency map is being calculated from pre-learned filters to be combined with bottom-up saliency maps). The S_2 filters were divided into subspaces of size Z_2, so applying the non-linearity similar to equation 5.6, is applied to reduce the number of feature maps

to $\tilde{R}_2$ with,

$$C_{1t} = \sqrt{\sum_{j \in Z_2} {S_{1t_j}}^2} \tag{5.9}$$

This step leads to blurrier saliency maps and gives a more accurate representation of the scene structure. Finally, the feature maps formed by equation 5.9 are added to form the final saliency map.

$$A_t = \sum_{j=1}^{\tilde{R}_2} C_{1t} \tag{5.10}$$

Figure 5.9 shows the maps produced by the (bottom-up) S_1 and (top-down) S_2 layer filters.

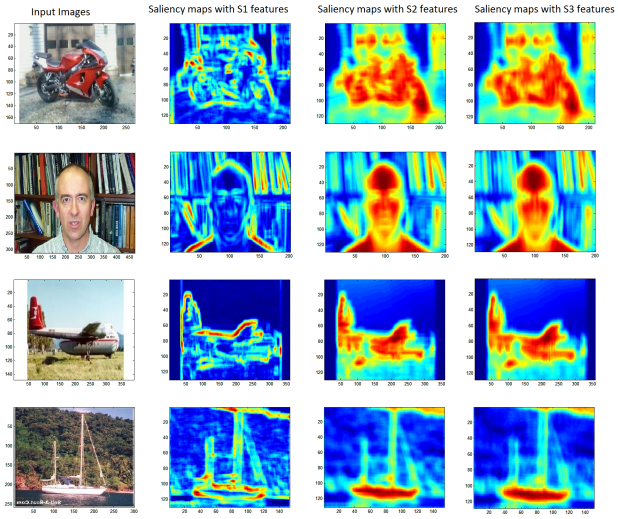


Figure 5.9: Saliency maps using local (S_1) and global (S_2) feature detectors (Input image from [15])

The first column displays the input images. The second column shows the V_1 layer saliency maps. Third column displays the global saliency maps generated by V_2 features. Fourth column shows the saliency maps formed with the same method, but using the S_3 layer filters.

Combining bottom-up saliency map with top-down attentional modulation

With the top-down saliency map is defined, it can be combined with bottom-up saliency map to direct attentional modulation. The top-down components are defined with the subscript t and bottom-up components with subscript **b**

The feedback attention and recognition model operates in the following steps (figure5.10).

1. In the first iteration of training the ISA HMAX model, the S_1 filters are learned from natural images.

2. A primary saliency map is formed with equation 5.7 A_1b. Referred to as A_1b in figure 5.10.

3. This saliency map (A_1b) directs the sampling of patches for learning S_2 layer filters.

4. The S_2 layer filters are applied on the images using equations 5.8, 5.9 and 5.10, to generate a secondary saliency map A_t.

5. The secondary saliency map A_t is combined with the primary saliency map A_1b

6. From the newly modulated A_b, patches are sampled and S_2 layer filters relearned to update the dictionary.

The connections represented by 'b' stands for the influence of the bottom-up saliency maps S_{i_b}, which directs the learning of S_{i+1} filters. The top-down influence is represented by 't', which changes the outputs of lower layers.

As observed from the figure 5.9, the S_t maps represent a global representation of the scene. The areas that share similarities are segmented. There are a few different strategies by which the maps S_t and S_{1_b} can be combined. The most straightforward method is to multiply the bottom-up and top down maps [35] or the combination technique defined in [18]. Alternatively, each of the S_1 filter responses can be modified by the S_t saliency map to form the new input for the feedforward recognition model.

At this point, due to an undefined top-down and bottom up saliency map combination method, a working model could not be demonstrated.

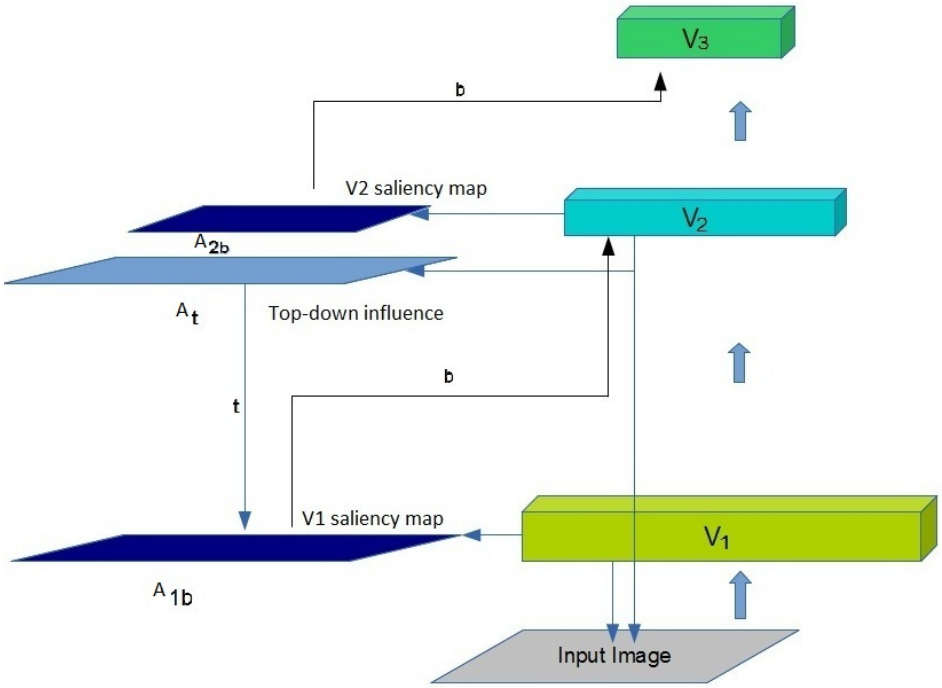


Figure 5.10: A feedback model of hierarchical vision with attentional modulation through global contextual information

5.5.1 Summary

In this chapter an integrated attention-recognition system was proposed that is unsupervised and data adaptive. The attention modulation has been demonstrated to be an integral part of cognitive function. The combination of hierarchical models and the interaction between low and high level features lead to the direction of focus towards the salient regions of a scene. The types of feature include low level edge detectors from which higher order features emerge. Although such feature detectors have demonstrated a positive impact on object recognition and saliency in many of the current vision models, there are a number of other properties that need to be included for a more robust and biologically plausible system. Factors such as depth perception, higher level contextual information such as the relationship between objects and colour are some of the areas of research that can enhance the feedback recognition model. Another area of improvement includes the typed of feedback connections between the different layers. With increase in model layers, the impact of further connectivity between the intermediate layers may bring new insights into the cognitive process.

Chapter 6

Compressed hierarchical model

6.1 Introduction

In chapter 4, hierarchical framework that comprised of feature detectors of increasing complexity was implemented. It was observed that the models with large number of middle layer units performed object classification with much higher accuracy. But increasing its numbers made the learning process for the next layer time consuming and memory intensive. Since these units were learned using ISA or TICA, the pooling of responses within subspaces or neighbourhoods somewhat reduced the data size. However, the best results were achieved when the subspace size or neighbourhood size was small in comparison with the total number of units. In this chapter, to deal with large mid layer outputs, a compressed hierarchical model which reduces the data size using the principles of compressed sensing is proposed.

6.2 Compressed Sensing

The Shannon sampling theorem states that with uniform sampling at Nyquist rate, which is twice the bandwidth of the signal, the data can be fully be recovered. Although most modern signal processing systems are built around this principle, instances where we encounter high Nyquist rates makes it difficult to implement as the required number of samples becomes too high. For data such as images, that limitation is bounded by its spatial resolution rather than Shannon theorem [131] [36]. Standard compression techniques reduce high dimensional but they are usually applied after storing the sampled data. In a ground breaking work by Candes, Romberg, Tao and Donoho [36][132], a new framework called *Compressed Sensing* or *Compressive Sampling* (CS) was proposed where compression occurs at the data acquisition stage. With this method, the signal can be sampled at a rate lower than the Nyquist frequency and efficiently reconstructed.

The sparsity of signals is the key to this type of compression. The L1-norm is commonly used as the regularization term for sparse coding cost function [25]. Given a signal that is sufficiently sparse, its compressed form can be recovered using the same L1-norm technique. In this chapter, the principles behind compressed sensing and its application towards a feedforward unsupervised HMAX model will be explored. The motivation is to investigate novel sparse models of HMAX that have 'compression' property in the learning process.

As observed from the hierarchical models in the previous chapters, object differentiability relies heavily on the number of filters in the layers. With data compression in each layer, faster and more efficient recognition models can be built. Its necessity becomes more apparent for large scale models with high number of feature detectors in each layer.

6.2.1 Compression of sparse signals

The two main principles underlying compressed sensing are sparsity of data and incoherence [36].

Sparse representation of signals

Most of the naturally occurring signals are not sparse [25], but can be expressed as a coefficient vector with small number of non zero elements with respect to a basis Ψ.

$$x = \langle s, \Psi \rangle = \sum_{i=1}^{N} s_i \Psi \tag{6.1}$$

For orthonormal basis Ψ

$$s = \Psi^T x \tag{6.2}$$

The signal that is to be 'sensed' or compressively sampled is denoted by x, where $x \in R^N$. It can be represented by sparse coefficient matrix s_i with respect to the set of bases Ψ. These basis sets can be of many different types such as Wavelets, DCT, Dirac delta functions . In the ISA-HMAX models, S layer filters form the basis vectors. Compression is applied on the coefficient matrix s_i which depends on the number of non zero values. With K non zero elements, the matrix is termed as K-sparse.

After the sparse signal $\widetilde{s}$ is recovered from the compressed version, the original signal x_s is reconstructed using equation 6.1.

$$x_s = \sum_{i=1}^{N} \widetilde{s}\Psi \tag{6.3}$$

With the reconstruction error,

$$\|x - x_s\|_{l^2} = \Psi \|s - \widetilde{s}\|_2 \tag{6.4}$$

The full recovery of data thus depends on its accuracy in sparse representation and the reconstruction technique of the compressed signal.

From equation 6.1, the sparse signal s_i can be approximated with the l1-norm,

$$min \|s\|_{l_1} \ such\ that,\ \Psi s = x \tag{6.5}$$

When a signal is compressed, a *measurement matrix* Φ is applied to its sparse vector,

$$y = \Phi s \tag{6.6}$$

From equation 6.2, sparse vector s can be represented in terms of orthonormal basis Ψ,

$$y = \Phi\Psi^T x = Ax \tag{6.7}$$

For sensing of the data, the measurement matrix extracts K samples of the sparse signal, where $K << N$. The sensed or compressed data is represented by y, which is a set of under-sampled data from the signal x,

The term A matrix in equation 6.7 is called *sensing* matrix with $A = \Phi\Psi^T$. Estimation of the coefficient matrix $\widetilde{s}$, from the sensed data y is obtained by solving the $l1$-norm minimisation equation,

$$\min_{\tilde{s}\in\mathbb{R}^N} \|\tilde{s}\|_{l_1} \ such\ that, \Phi\widetilde{s} = y$$

or from equation 6.7,

$$\min_{\tilde{s}\in\mathbb{R}^N} \|\tilde{s}\|_{l_1} \ such\ that, Ax = y$$

The measurement matrix Φ is of dimensions $M \times N$, where M is the number of samples which are randomly sensed from the N dimensional signal. The maximum number of samples M which defines the extent of compressibility of the

signal depends on the sparsity K of the basis Ψ. Which refers to the number of non-zero components associated with the basis vector.

Selecting M samples from the signal $x \in R^N$, and if the coefficient vector is K-sparse, (having K number of non-zero values), [132] If the observed samples obey,

$$M \geq CKlogN \tag{6.8}$$

Where, C is a positive constant, the l1-norm minimization equation 6.5 exact reconstruction of the signal x is possible with very high probability. It was observed that the value of M needs to be at least $KlogN$ for the signal to be recovered [132].

Incoherence

The second condition for compressed sensing is the incoherence between the basis matrix Ψ and the measurement matrix Φ. This property implies that data is spread out in the domain of its basis Ψ [36]. The minimum correlation between elements of the two matrices is ideal for giving least errors in reconstruction of the data when sampling M samples as in equation 6.8.

With a fixed orthogonal basis set Ψ, any random matrix displays incoherence to a large degree [36]. For the signal $x \in R^N$ coherence between the two matrices Ψ and measurement Φ sensing M samples is defined by,

$$\mu(\Psi, \Phi) = \sqrt{N} \max_{M>1, j \leq N} \left| \langle \Phi_M, \Psi_j \rangle \right| \tag{6.9}$$

Where, $\mu \in [1, \sqrt{N}]$, and 1 gives the maximum incoherence. It was stated that using any random matrix as Φ for a fixed orthogonal Ψ, gives sufficiently high incoherence to enable accurate reconstruction.

For $x \in R^N$ represented by an K-Sparse matrix and coherence $\mu(\Psi, \Phi)$ within $A = \Phi\Psi$ defined by equation 6.9, the number of measurements M required for signal recovery is defined becomes,

$$M \geq C\mu^2(\Psi, \Phi)KlogN \tag{6.10}$$

Where, C is a positive constant. From this equation, it becomes evident that as μ reaches towards 1, the number of measurements M required for an exact recovery also decreases. The ideal minimum number of samples is $KlogN$ when $\mu = 1$.

But for any other case with sufficient sparsity and incoherence, the minimum requirement for data recovery was found to be $4KlogN$ [132].

In some cases, signal recovery is not possible when the measurement matrix Φ only samples the zero values, where the observed $y_k = \langle x, \Phi_k \rangle = 0$. To avoid such type of measurements, another criteria called *Restricted Isometry Property* is defined.

Restricted Isometric Property (RIP)

The restricted isometry sets a rule for the distribution of the the sparse coefficients along the topography of the matrix to ensure non-zero samples necessary for reconstruction. It preserves the Euclidean length of the K-sparse signals so that it does not give 0 value in the Φ domain [36]. For a sensing matrix A of size M by N, this property states that the isometric coefficient δ_K, (for K-sparse vector) should be small enough such that,

$$(1 - \delta_K) \|s\|_{l^2} \leq \|As\|_{l^2} \leq (1 + \delta_K) \|s\|_{l^2} \tag{6.11}$$

When the condition of RIP holds true $2K$ columns of the measurement matrix Φ form a set that is linearly independent [133]. With this property, compression does not return null vector and gives a more accurate reconstruction of x with equation 6.5.

Another advantage of compressive sampling is that it is highly robust against noise. For any data that is sensed inaccurately, $x = s\Phi$, the sampled data $y = As + e$, with e as the stochastic error term with $\|e\|_{l_2} \leq \varepsilon$, the compressed sensing can be adapted for efficiently recovering the signal data.

For noisy data, the sensed data, $\tilde{y}$, is represented by l1-norm minimization of the coefficient matrix $\|\tilde{s}\|_{l^1}$ subject to, $\|A\tilde{s} - y\|_{l^2} \leq \varepsilon$.

Φ of size $M \times N$ should be constructed such that subsets of its columns are orthogonal. Among the measurement techniques, there is *Gaussian Method*, where, the values are sampled from a zero mean and $1/N$ variance Gaussian probability distribution. For a basis Ψ that does not exhibit any particular structure, random matrices were found to be sufficient for signal recovery. Other methods include Binary Measurements and Fourier Measurements [36]. The restricted isometry constant for any random matrix is $\delta_{2K} < 1$ when the number of samples is $M \geq (K \cdot (N/K))/\varepsilon)$ [133] and thus fulfils the conditions for signal recovery.

With the conditions for robust signal recovery defined by sparsity, incoherence and RIP, the convex optimization problems can be solved using many different algorithms [134]. Apart from the l1-norm minimization, there are other methods such as greedy algorithms have also been introduced for recovery of sparse signals. The CoSAMP (compressive sampling) was one of the earlier algorithms that demonstrated a highly accurate signal recovery in a greedy iterative process [133][135].

6.2.2 Compressed sensing in computer vision

Recently, CS has already found its use in many computer vision applications such as face recognition [136], object detection, MRI [36].

In most of the task specific vision applications, the CS implementation starts at the sparse representation and subsequent compression techniques applied on the whole image. In the multi stage models based on the visual cortex such as HMAX, the image data is processed in a patches or localized receptive fields (RF). The visual data from each RF is encoded as sparse representation over a basis set. The bases in the first layer function as low level edge detectors and in the higher layers, they represent more complex features. The bases or filters in each layer are modelled after the simple and complex neurons of the primary visual cortex. With sparse algorithms such as [25], [30],[33], the first condition of sparsity for the CS process is already satisfied.

The main purpose of the model is to extract the distinctive features of the images in multiple stages. Since any information that lost in the lower layers is not recovered in the next, it is important for any compression technique to preserve the data as accurately as possible. In the earlier models, compression was carried out by PCA which also served the dual purpose of minimizing linear correlations. Although these are effective methods, compression follows after data storage. The memory requirement for signals of larger dimensions makes the model impractical. CS allows for direct compression after the layer output is obtained. Effective compression thus becomes important for large scale adaptation of cognitive models.

6.3 Compression in the visual cortex

The possibility of a compressed form of data acquisition using CS has been studied in neuroscience to gain insights into signal encoding in cortex[137]. Compression is said to arise due to sparsity of neuronal activation. Down sampling of signals in the visual cortex was observed with the number of receptor cells reducing in the higher layers. As an example, signal from 150 million rods and cones transmit to only 1.5 million retinal ganglion cells [137].

6.4 Compressed Sensing in HMAX model

Compressed sensing has also been considered within the context of HMAX models by Serre in [138]. It was hypothesized that the S units gather a fraction of its afferent units in a random sampling method. It was observed that HMAX models using ISA and TICA categorises the bases such that units within a subspace or neighbourhood can be pooled. In this way, signal dimensionality is somewhat reduced while maintaining its selectivity and invariance. Although similar methods have been compared to memory processing in the V2 layer of the cortex [82], dimension reduction is limited to the number of subspaces, neighbourhoods or clusters.

In the feedforward ISA-HMAX model in chapter 4, the data from each C_{i_b} layer was sampled by the next layer units after which data whitening and normalization was applied. If the square receptive field width of a unit is p and number of filters outputs in its afferent lower layer is $\tilde{R}$, the dimension of each sampled response is $p \times p \times \tilde{R}$. The dimension of the unit is also $p \times p \times \tilde{R}$. The memory penalty is proportional to $\tilde{R}$, so for a larger scale model, increasing the number of $\tilde{R}$ makes computation difficult. Signal compression is thus an important step towards building faster and efficient vision models. (The term Ψ in equation 6.1 is represented as by R or $\tilde{R}$ for pooled units).

Since there is no concrete evidence about compression mechanisms in the visual cortex, CS can potentially be applied in any stage of the HMAX model. Efficiency of compression depends on the signal sparsity and sensing matrix. Although the type of bases $\tilde{R}$ can vary according to learning method such as ISA or TICA, the sensing matrix should be designed such that it satisfies the RIP. It was observed that whenever a sensing matrix demonstrated a high reconstruction error the accuracy of multi-class categorization declined. In the next section, the

application of CS will be demonstrated on the ISA HMAX model of chapter 4.

Compression of jointly sparse signals

The dimension of measurement matrix Φ is of size $M \times N$, where M is the number of measurements and N is the dimension of the sparse signal. For the basis vectors learned with ISA/ICA, any random measurement matrix exhibits good RIP. Since a layer is defined by a single set of basis, a single two dimensional measurement matrix is applied to compress all the data in that layer.

Since natural signals show variation in sparsity, a threshold was applied such that K highest activations are retained. This is not ideal since suppression of values leads to information loss. If the number of activations is low enough such that $K << M$, this measure would not be required. In this case, the value of K is the same both for the compressed and uncompressed model to compare its performance. For models where the set of signals have variable K, the measurement matrix Φ (or sensing matrix A) is designed for the signal with largest K for minimum information loss. If a sensing matrix A can compress $K1$-sparse signal $x1$ with low recovery error, the same matrix A can be applied to compress a $K2$-sparse signal $x2$ where $K2 < K1$, as illustrated in 6.1.

Using these principles, compressed sensing was applied at the V_2 layer of the ISA HMAX model in figure 4.9. In chapter 4, it was observed that a large number of V_2 filters contributed to a better classification performance. Since it encodes more complex variations of data, a high number of complex feature detectors are desirable. Therefore, the V_2 layer tends to be quite large, so compression at this stage is more of a requirement than V_1.

Implementation of CS in a hierarchical model

In the first model, the CS was applied after spatial pooling in the C_{i_b} layer (where, i represents the V layer from figure 4.9), as illustrated in the figure 6.2. Compression at this stage proved to be difficult since the maximum number of non zero elements increased within the afferent receptive field. This is due to pooling of neighbouring values which concentrates the number of non-zero data within a smaller area.

Because K was not significantly smaller than N, compression was not entirely efficient. For larger models, if the value K remains relatively low even after max pooling, CS can be effectively applied at this stage.

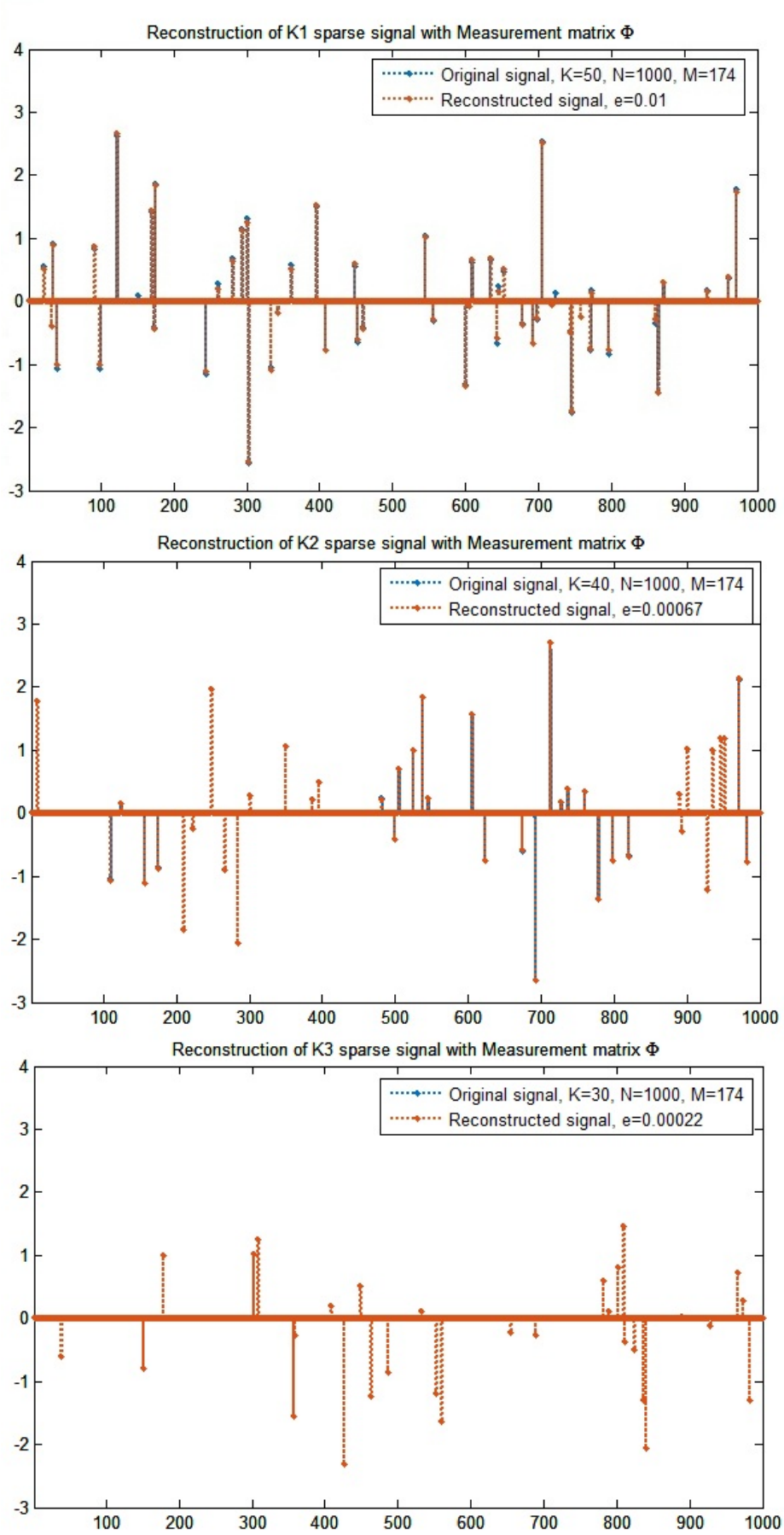


Figure 6.1: Reconstruction of three different signals with sparsity $K1 > K2 > K3$

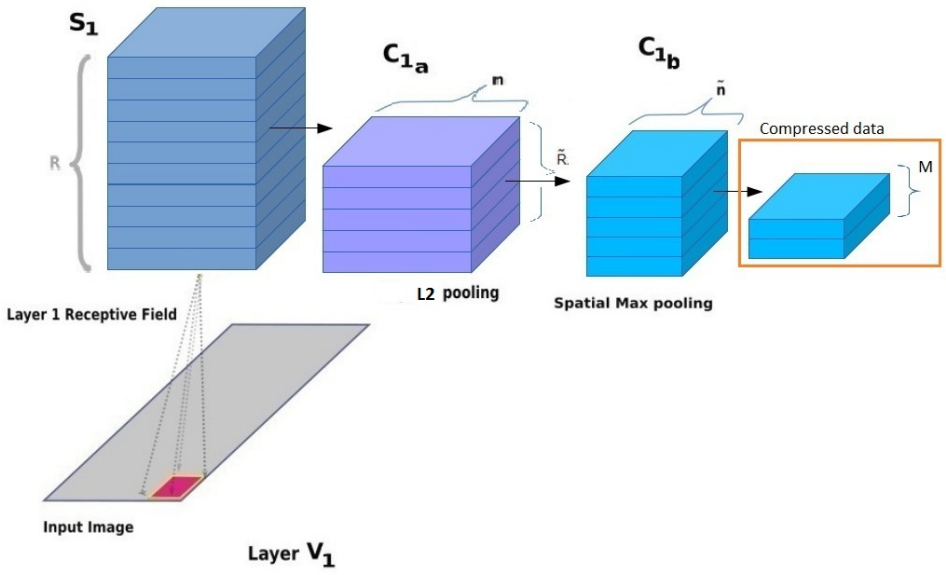


Figure 6.2: CS after C_{i_b} of the HMAX model

A more straightforward method is to apply CS right after the C_{i_a} layer as illustrated in figure 6.3. For any input image, the S_i layer output is of dimension $m \times n \times R$, where R is the number of bases. For ISA models, the output gets reduced to $m \times n \times \tilde{R}$, where $\tilde{R} = R/Z$. Although the equations 6.5 and incoherence properties refer to orthogonal bases, it has been noted in [36] that orthogonality is not completely essential for compressed sampling. The figure 6.3 shows a single layer of the HMAX model. The highlighted part represents data in its compressed form, which is then further transformed by higher layer operations. The following steps were applied to compress the data in figure 6.3.

1. A sample S of size $1 \times \tilde{R}$ size is chosen (figure 6.4A). (If the value of K across the signals are non-uniform, the sample with the largest value of K is chosen).

2. A measurement matrix Φ is constructed which such that it satisfies the conditions of incoherence and restricted isometry property. In this case, Φ was comprised of random variables.

3. To select the most optimal Φ, the value of 'M', which represents the size of compression was kept around $3K$.

4. For a given Φ, a fast reconstruction algorithm was applied and reconstruction error ε was calculated. After setting a threshold for ε, for a number of iterations, Φ was randomly generated. The one with lowest ε was then selected as the final measurement matrix.

5. The measurement matrix was then applied to the rest of the data and for every other input images as well (figure 6.4B).

$$C_{i_c} = \sum_{j=1}^{m*n} S_j * \Phi \tag{6.12}$$

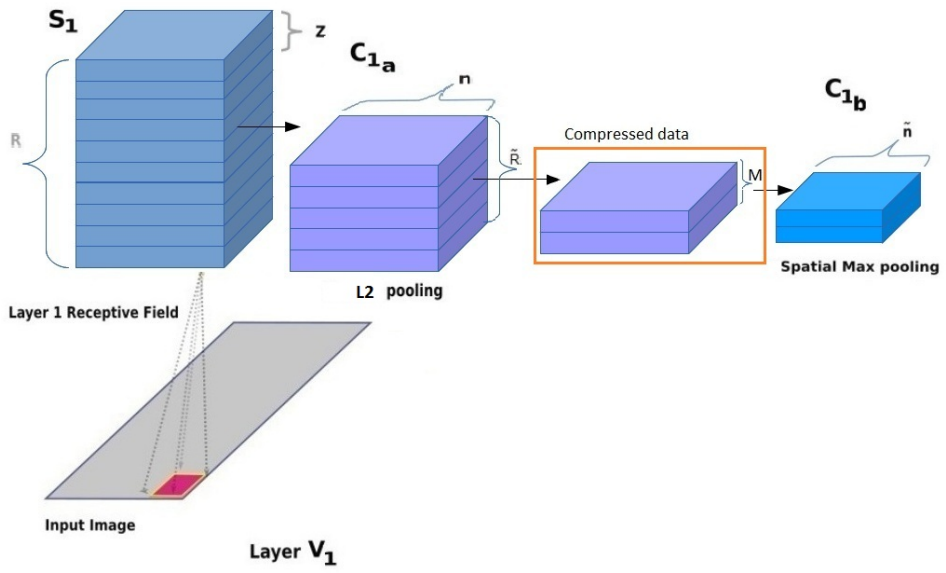


Figure 6.3: CS after C_{i_a} layer of the HMAX model

The compressed form of the input image after S layer operation followed by CS is $S_c = m \times n \times M$, where M is the number of measurements taken from the sparse signal.

6.4.0.1 Effect of compression on saliency maps

In this model, data is sent in its compressed form to the next layer. Any further operation such as max pooling at C_{i_b} should preserve the inherent features of the object. In the previous chapter, the model utilized the features of the hierarchical model to obtain saliency maps. Linear weighted summation of the C_{1_b} outputs

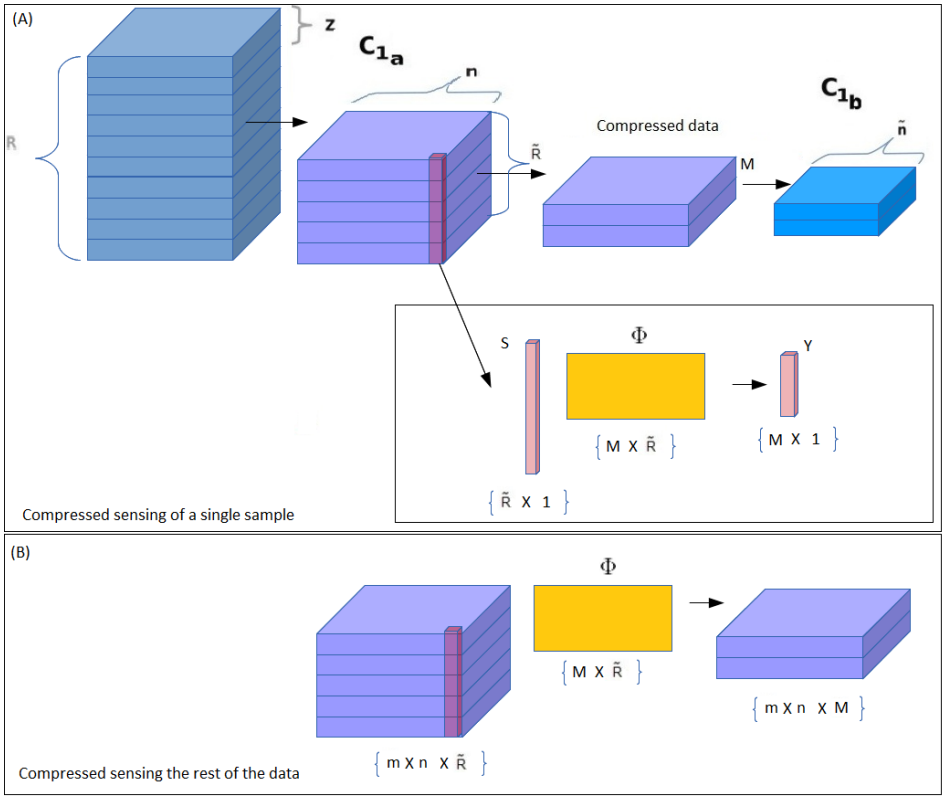


Figure 6.4: Applying CS on a 1D sample of C_{i_a} output and applying the same measurement matrix to the complete dataset

generated an initial saliency map which was then used for directing the selection of samples for learning the feature detectors in the next layer.

When data is compressed with CS, linear combination of the resulting feature maps should produce saliency maps similar to uncompressed data. For $C_{i_a} \rightarrow C_{i_c}$, from equation 5.7, saliency map is obtained by,

$$A_{b_c} = \sum_{j=1}^{M} C_{i_c} \tag{6.13}$$

Where, M is the number of measurements which is dependent on the sparsity K of the uncompressed data C_{i_a}. C_{i_c} is the compressed data of dimension $C_{i_b} = \tilde{m} \times \tilde{n} \times M$.

In the following example, saliency map is demonstrated for an uncompressed and compressed first layer of an HMAX model. The number of bases R_1 is 300 (generated using ISA). The original uncompressed model had an S_1 layer output of dimension $\tilde{m} \times \tilde{n} \times 300$. The subspace size is 2, which makes the dimension of the output at $C_{i_a} = \tilde{m} \times \tilde{n} \times 150$. The number of measurements M for the compressed model is 30. This makes the compressed layer output $C_{i_c} = \tilde{m} \times \tilde{n} \times 30$.

Figure 6.5, top row, shows the saliency map obtained by linear combination of the C_{1_a} features. After applying compressed sensing to each value of C_{1_a} with a measurement matrix (equation 6.12), its dimensions get reduced. It becomes similar to a weighted summation of features. The bottom row displays the saliency maps formed by combining the compressed features with equation 6.13. Figure 6.6 is the result of C_{1_b} features after compression. The saliency maps are unaltered when a measurement matrix with low reconstruction error is applied on the C_{1_a} values.

6.4.1 Implementation for object recognition model

For evaluating multi-class object recognition, compressed sensing was applied in the V_2 layer of the model in figure 4.9. Due to higher complexity of features, larger number of S_2 units are essential for encoding image information. With large dictionary size, its compressibility plays an important role for reducing the dimensions of the output.

The specifications of the model is given in table 6.1. M_2 represents number of features after compression, e_2 is the reconstruction error for the selected mea-

Figure 6.5: Compressive HMAX saliency maps (Input image from [15])

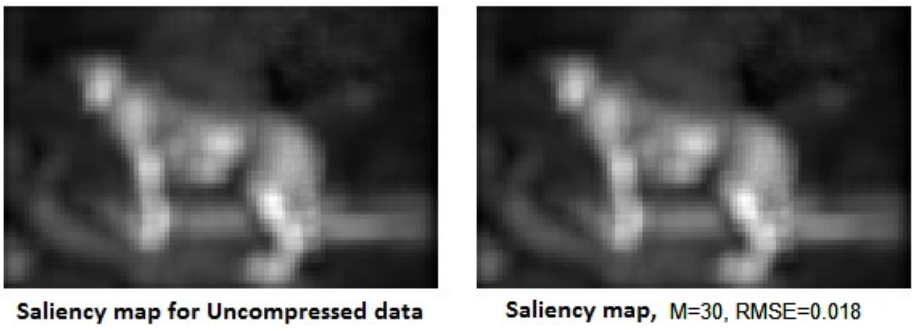


Figure 6.6: Compressive HMAX saliency maps (Input image from [15])

surement matrix and compression size. In this model, only the $K = 30$ highest activation values were allowed for the S_2 layer outputs which resulted in an overall reduction in classification accuracy. In model $1a$, CS is applied directly on the S_2 features, and in model $2a$, CS is applied on the C_{i_a} features.

Table 6.1: Model specifications

Models	V_1, $p_1 = 11$			V_2, $p_2 = 10$					V_3, $p_3 = 9$
		C_{1_a}			C_{2_a}				
	S_1 (R_1)	Z_1	$\tilde{R}_1$	S_2 (R_2)	Z_2	$\tilde{R}_2$	M_2	e_2	S_3 (R_3)
Uncompressed	144	9	16	300	2	150	-	-	100
Model 1a	144	9	16	300	-	-	100	0.007	100
Model 2a	144	9	16	300	2	150	90	0.006	100

Ten categories of Caltech101 images [15] were used for multi-category object classification. Since the purpose is to examine the applicability of compressed sensing, the entire dataset was not used.

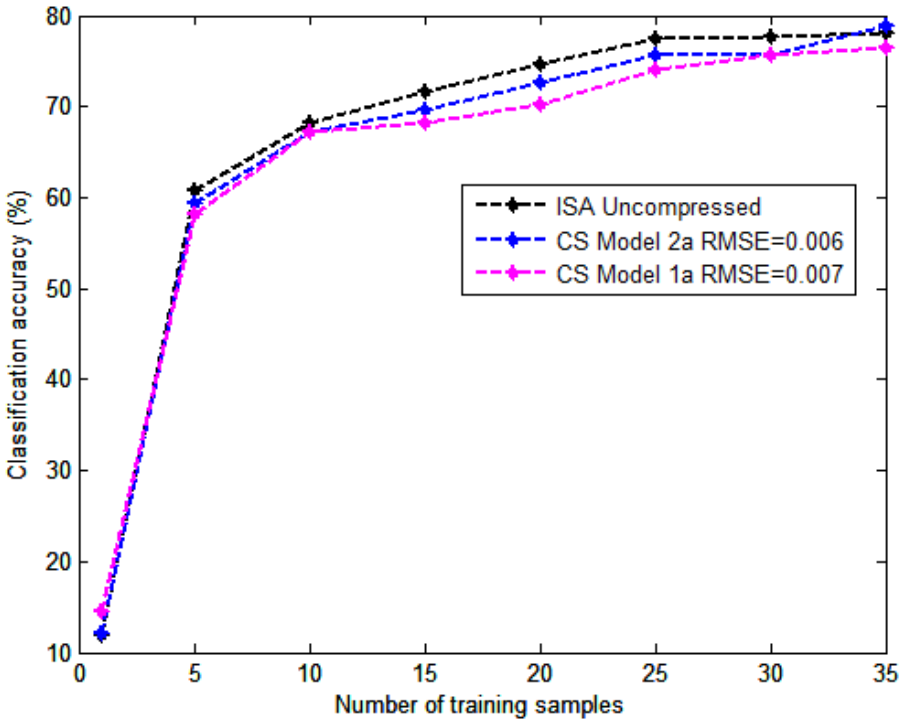


Figure 6.7: Classification accuracy of compressed models

The figure 6.7 shows the performance of models 1a and 2a for multi-class ob-

ject categorization. Model 1a shows displays less accuracy than 2a even though the number of measurements M_2 is larger. This could have occurred due to the higher reconstruction error for the sensing matrix in comparison to 2a, but another contributing factor could also be the result of bypassing the C_{2_a} step, which is connected to the phase invariance property of the model.

To test the influence of reconstruction error of the compressed representation on classification accuracy, another set of models in table 6.2 was evaluated. Here, the number of S_2 bases was increased to 400. With the larger dictionary size, the value of K was reduced to 12. C_{i_a} size is compressed from 200 to 100. All models have the same parameters, with the exception of measurement matrix M_2 and its associated recovery error.

Table 6.2: Model specifications

Models	V_1, $p_1 = 11$			V_2, $p_2 = 10$					V_3, $p_3 = 9$
		C_{1_a}			C_{2_a}				
	S_1 (R_1)	Z_1	$\tilde{R_1}$	S_2 (R_2)	Z_2	$\tilde{R_2}$	M_2	e_2	S_3 (R_3)
Uncompressed	144	9	16	400	2	200	-	-	200
CS Model 1b	144	9	16	400	2	200	100	0.00140	200
CS Model 2b	144	9	16	400	2	200	100	0.00095	200
CS Model 3b	144	9	16	400	2	200	100	0.00070	200

The baseline in this case is the uncompressed model, which was very time consuming and memory intensive. With half the number of V_2 features, the compressed models in figure 6.8 display similar results but with a reduced data dimension. Also, from figure 6.8, it can be observed that the model 1b, which has the highest recovery error shows a lower classification performance than the rest. Although the difference in value of e_2 for 2b and 3b is smaller, the model 3b with smaller e_2 displays a slightly higher accuracy. From these results, it can be inferred that the accuracy of the compressed models depend on measurement matrix that recovers the signal with lowest error using equation 6.5 or any other standard recovery algorithm.

This can be compared to the $1 * 1$ convolution method in chapter 4 where data size reduction was applied. For the ISA-HMAX models, it led to a reduction in performance accuracy. Also, using a random matrix to compress a model does not provide any method for preserving (or improving) the models classification accuracy. But with a random matrix with low reconstruction error, with the CS method can compression of the data with least amount of information loss can be

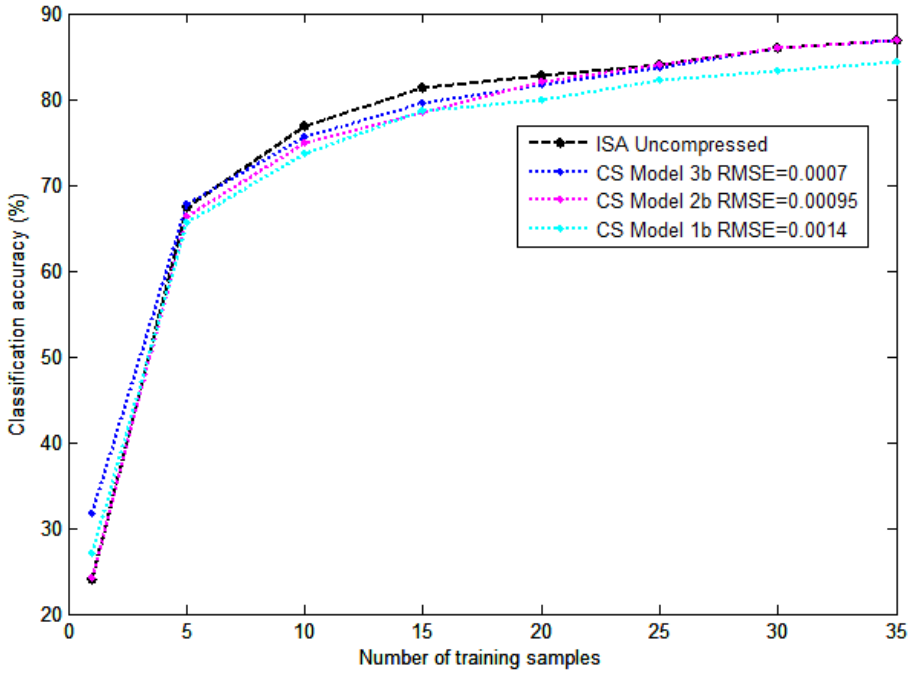


Figure 6.8: Classification accuracy of compressed models

achieved.

6.5 Summary

In this chapter, the application of compressed sensing on multi layered vision models was explored. If the sparsity of response in each S layer of the model is determined, CS can be applied to reduce its dimension without a significant loss of information. A common measurement matrix generated at random, with the smallest recovery error for a signal with largest number of non zero elements was applied on all the data. The model results in terms of saliency map generation and classification accuracy performs on par with its uncompressed counterpart. For a object robust classification, the sparse representation of image data should be as accurate as possible. With CS modulated models, the only possible drawbacks would stem from poor transformation in its sparse form. Additionally, the value of K needs to be reasonably small for higher compression.

CS modulated vision framework can be seen as a viable solution for data processing in bigger models which contain large number of S units in multiple layers.

With this method, the dimensions can be reduced to a fraction of its original size. The size of the higher layer S units are also reduced without compromising its functional properties.

With better measurement matrix design, further improvements can be made to the existing models. For now, random matrices have proved to be highly efficient sensing sparse data. In the future, design of deterministic sensing matrices based on the receptive field (or basis vector) characteristics of model can be considered as a direction of research for enhanced compression in biologically plausible vision models.

Chapter 7

Conclusion and Future research

7.1 Introduction

Models based on the visual cortex are important step towards understanding vision. Biological vision system is still a vastly complex field of study that has been explored from different viewpoints such as neuroscience and cognitive psychology.

The most fundamental and well established trait is the hierarchy of information processing, which is adopted in all the biologically inspired vision models. Each layer comprises of cells that have a unique characteristic and receptive field (RF). The increase in receptive field sizes along the layers correspond to the increase in the complexity of the cells. All these contribute to invariant response high selectivity which computer vision aims to replicate. The second integral property is the the attentional modulation which serves to optimize processing of large amount of visual data that enters the cortex. The direction of focus towards salient regions emerges from interaction between external stimulus driven activation and internal knowledge based response. And lastly, the sparse firing of neurons, that ensures that only a fraction of neurons remain active at a time. By incorporating these properties, ways to develop an enhanced object detection model was explored.

7.1.1 Key contributions

The main contribution of this thesis was to demonstrate that the combined effect of the L2-pooling and max pooling non-linearities has a positive impact on the classification performance of self-taught hierarchical learning models. It implementation was carried out by sparsity based ISA and TICA algorithms which grouped them into subspaces or neighbourhoods according to the correlation of their energies. The resulting feature vectors were highly tolerant towards scale, position, and rotation variations while maintaining high degree of selectivity for

an object class. Its classification accuracy on the Caltech-101 dataset was higher than most of the current unsupervised models. Along with high accuracy, the non-linear functions also resulted in reducing the dimensions of the data.

Comparison of object classification performance for both the ISA and TICA models with similar parameters showed ISA to perform better with than TICA when large subspace sizes or pooling neighbourhood sizes were used. TICA models performs best when the pooling neighbourhood size is small with respect to the topography size.

Building on the ISA models, the framework was then augmented with a feed-forward saliency mechanism. Existing saliency methods involving bottom-up saliency such as the GBVS [19] and SUN [20] were also integrated with the vision model. Evaluation on a limited dataset showed slight improvement in classification accuracy when the number of sampled patches were reduced. But in comparison with models with large sample sizes, the performance lagged behind considerably. This highlighted the need for more accurate saliency maps for an unsupervised attention-recognition model.

Since sparsity based algorithms were applied for the models, the data dimensions were further reduced using compressed sensing. For a small dataset of 10 categories, it was found that compression can be achieved without drastically affecting the classification accuracy, if the reconstruction error of the compression matrix is low enough.

7.1.2 Future work

Convolutional Neural Networks

As seen from the classification accuracy for CNNs, for both the Caltech-101 and Caltech-256 the model from [63] outperformed the ISA-HMAX described in chapter 4. However, the number of parameters of the model and training images that were used was much higher than the ISA-HMAX. This shows that the ISA model has the potential to improve its feature extraction ability with a wider variety of mid-layer filters.

Also, there is the possibility of applying a fully connected layer on top the final S_3 layer to build a convolutional neural network. The unsupervised learning to initialize parameters could be applied similar to [58], where a pre-training stage is used for reducing the number of required training images.

Scale invariance

Although the models listed above displayed a high degree of invariant response and selectivity, pooling between different spatial resolutions was not applied. This has been addressed in chapter 4, but a scale invariant model without increasing the final layer parameters was not successful. The SiCNN [95] model applies the multiple architecture and feature concatenation method, but since CNNs are functionally different, equivalent comparisons cannot be made. The only models that have incorporated this property do fas has been the HMAX models where Gabor filters of multiple scales are used. But since its final multi-category classification performance is lower, the same architecture may not be suitable for the ISA-HMAX models.

Attentional modulation

As observed in chapter 5, a fully integrated attention-recognition model has not reached its potential yet. The initial feed-forward model only applied attention for the learning stage and not the inference stage. When the saliency maps were applied to suppress information directly on the image, the performance declined drastically. Due to incorrect saliency maps, the possibility of information loss is high.

An attention based feedback model based on learned filters is also one of the directions where these models can be extended. The purpose is to use the already learnt low and high level filters to direct attention towards salient regions of an image. This could model the dorsal and ventral streams within the same hierarchical model without the need of separate architectures.

Sparsity and compression

In chapter 6, the sparsity of response was used to compress the data within the layers. To apply this property, the number of non-zero responses had to be relatively low. So there is a need for better representation of data with sparse algorithms to be able to allow a higher degree of compressibility.

Biological plausibility

The activation patterns of the final layer of the models resemble a pattern similar to neurons firing. In the face detection experiment from chapter 4, it was discov-

ered that these hierarchical models learn high level 'face neurons' with completely unlabelled data. This was also previously demonstrated in [13]. In terms of the visual cortex, each of the final S layer filters could resemble the different channels of signals. As mentioned in chapter 2, it has been theorized that a synchronous oscillation occurs within the various channels when perception occurs [50]. But due to the lack of experimental data, it is difficult to determine if the phase synchronous activity of channels contributes towards object perception.

7.2 Conclusion

Modelling biological vision remains a daunting task mainly due to the numerous factors that come into play during perception. Most of the models that are currently in use in computer science are an oversimplification of a very complex model. The models presented here only cover a few common features of the visual cortex. The next step towards building a cognitive model should be to integrate additional properties into the model. Another important step is to develop larger scale of unsupervised models to be able to learn features from larger databases.

Although a lot of progress has been made within the field of neuroscience and computer vision, there is not much access for validating these models with respect to empirical data. Therefore, there is an increasing need for bringing together research from both fields to understand the mechanisms behind perception.

Appendices

Appendix A

Appendices

NOTE: Sources edited manually for line-break adjustment. This usually follow some logic. In the case of matlab scripts, in a matlabic way (any [()]).

A.1 Matlab codes

```
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% Set parameter values for hierarchical ISA/TICA models
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

alg = 'isa'; %select algorithm isa or tica%
inh=0;  %threshold below which filter response is set to zerp
flg1=1; % for ISA or TICA flg=1 activates pooling within subspaces or
        %neighborhood
flg2=1;
flg3=0;
Image_size=200;

nscales=1;      % number of scales
tica_type=1;    % tica_type=1: neighborhood elements do not overlap.
learn_dir = ''; % sample dataset for learning S layer filters

data_path= strcat(datadir,category);
C1layer = ['C1_train'];
C2layer = ['C2_train'];


p_1= 11;     %RF size of layer V1
R_1 = 100;   %Total number of S1 filters
samplesize1 = 50000;
ra_1 = 3;


p_2 = 12;    %RF size of layer V2
R_2 = 300;   %Total number of S2 filters
```

```
samplesize2 = 50000;
ra_2 = 2;


p_3 =13;       %RF size of layer V3
R_3 = 1000;    %Total number of S3 filters
samplesize3 = 50000;


%%%%%%%%%%%%Parameters for ISA and TICA%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%functions adapted from https://research.ics.aalto.fi/ica/imageica/
%isaparam.groups: number of subspaces
%isaparam.groupsize: subspace size
%epsi: positive constant



isaparam1=icaparam1;
isaparam1.groups=25;
isaparam1.groupsize=4;
isaparam1.epsi=0.5;

isaparam2=isaparam1;
isaparam2.groups=60;
isaparam2.groupsize=5;

isaparam3=isaparam1;
isaparam3.groups=100;
isaparam3.groupsize=10;

ticaparam1.model='tica';
ticaparam1.algorithm='gradient';
ticaparam1.xdim=12;
ticaparam1.ydim=12;
ticaparam1.maptype='torus';
ticaparam1.neighborhood='ones3by3';
ticaparam1.stepsize=1;
ticaparam1.epsi=0.5;
ticaparam1.nb=2;
ticaparam1.ol=0;

ticaparam2=ticaparam1;
ticaparam2.xdim=10;
```

```
ticaparam2.ydim=10;
ticaparam2.nb=2;
ticaparam2.ol=0;

ticaparam3=ticaparam1;
ticaparam3.xdim=20;
ticaparam3.ydim=20;
ticaparam3.nb=1;
ticaparam3.ol=0;
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
S1filters=['ISAS1_cal101' '.mat'];
S2filters=['ISAS2_cal101' '.mat'];
S3filters=['ISAS3_cal101' '.mat'];
```

```
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%Training hierarchical ISA/TICA vision models
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

     param;   %set the model parameters

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% Learn S1 filters
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%


    fprintf('Sampling data...\n')
    %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
    % Download from :
    X = sampleimages(data_path ,samplesize1, p_1,ra_1,Image_size,1);
    X = patch_normalize(X);
    %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

   if strcmp(alg,'isa')
       [V1,~,~]=pca(X,R_1);
       Z=V1*X; % Whiten to remove linear dependencies
       fname='tempS1';
       estimateISA(Z,V1, fname, isaparam1); % from:
       load('tempS1.mat','isa')
       base.S1=isa{1}.A;
       base.S1w=(isa{1}.B)';
       base.V1=V;
       save(S1filters,'base');

   elseif strcmp(alg,'tica')
```

```
        [V1,~,~]=pca(X,R_1);
        Z=V1*X; % Whiten to remove linear dependencies
        fname='tempS1';
        estimateTICA(Z,V1, fname, ticaparam1); % from:
        load('tempS1.mat','tica')
        base.S1=tica{1}.A;
        base.S1w=(tica{1}.B)';
        base.V1=V;
        save(S1filters,'base');
    end


%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% C1 layer: L2 pooling + max pooling
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

    if strcmp(alg,'isa')
        load(S1filters,'base');
        layerC1(data_path,C1layer,base,ra_1,Image_size,alg,isaparam1,flg1);

    elseif strcmp(alg,'tica')
        load(S1filters,'base');
        layerC1(data_path,C1layer,base,ra_1,Image_size,alg,ticaparam1,flg1);

    end

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% Learn S2 (and S3) filters
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

   database=CFileStat(C1layer); % downloaded from:

    if (strcmp(alg,'isa'))
       nR_1=R_1/isaparam1.groupsize;
    elseif strcmp(alg,'tica')&& flg1==1
       nR_1=(ticaparam1.xdim/ticaparam1.nb)*(ticaparam1.ydim/ticaparam1.nb);
    end

    if (saliency==0)
    X=sample3D(database,C1layer,samplesize2,p_2,nR_1); %from:
    else
    X=sample3D_sal(database,C1layer,samplesize2,p_2,nR_1,per_sal);
    end
```

```
    [mr,~]=size(X);
    fun = @(block_struct) patch_normalize(block_struct.data);  %  from :
    X = blockproc(X,[mr 100],fun);
    fprintf('Doing PCA and whitening data...\n');

    if strcmp(alg,'isa')
        [V2,~,~]=pca(X,rdim2);
        Z=V2*X;
        clear X
       fname='tempS2';
       estimateISA(Z,V2, fname, isaparam2);
       load('tempS2.mat','isanetwork')
       S2=isanetwork{1}.A;
       S2w=(isanetwork{1}.B)';
       save(S2filters,'S2w','S2','V2');
    elseif strcmp(alg,'tica')
        [V2,~,~]=pca(X,rdim2);
        Z=V2*X;
        clear X
        fname='tempS2';
        estica( Z,V2, [],fname, ticaparam2 )
        load('tempS2.mat','isanetwork')
        S2=isanetwork{1}.A;
        S2w=(isanetwork{1}.B)';
        save(S2filters,'S2w','S2','V2');
    end

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% C1 layer: L2 pooling + max pooling
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

    if strcmp(alg,'isa')
        load(S2filters,'S2w','S2','V2');
        layerCS2_multi2(C1layer,C2layer,S2w,V2,ra_2,alg,...
            isaparam2,flg2,nscales,inh,ticatype)
    elseif strcmp(alg,'tica')
        load(S2filters,'S2w','S2','V2');
        layerCS2_multi2(C1layer,C2layer,S2w,V2,ra_2,alg,...
            ticaparam2,flg2,nscales,inh,ticatype)
    end


%%%%%%%%%%%%%%%%%%%% V1 layer response %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% Smap_pooling.m from:sparseHMAX-v1.2
```

```
% calculates V1 layer response
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
function layerC1(datadir,C1layer,base,ratio,imsiz,alg,param,flg,inh)
if ~exist(C1layer,'dir'), mkdir(C1layer); end

catlist=dir(datadir); %list of object categories
catlist(2)=[];
catlist(1)=[];
catnum=size(catlist,1);
filepaths=cell(1,catnum);


%% %%%%%%%%For different categories of data %%%%%%%%%%%%%%%%%%
files=cell(1,catnum);
for i=1:catnum
    filepaths{i}=fullfile(datadir,catlist(i).name);
    files{i}=dir( fullfile(filepaths{i},'*.jpg') );
end


for i=1:(length(r))
filestruct=files{r(i)};
  for j=1:(size(filestruct,1))
    filename=filestruct(j).name;
    X=imread(fullfile(filepaths{i},filename));
    %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
    [sy, sx, ~] = size(X);
    rescale_factor = imsiz / min(sy, sx);
    X = imresize(X, round([sy, sx]*rescale_factor));
    %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
    str=strcat('_',int2str(i));
    filename=regexprep(filename, '_',str);
    matfile= regexprep(filename, '.jpg', '.mat');
    Vresp_name=fullfile(C1layer,matfile);
    filter=base.A1pca;
    w_bases=base.V1;
    %patch=sqrt((size(w_bases,2)/3));  %% for colour
    patch=sqrt(size(w_bases,2));
    Ca=layer_SCa(X,w_bases,filter,patch,alg,param,flg,inh);

    C=Smap_pooling(Ca, ratio, ratio, ...
        mod(size(S1map,1),ratio),mod(size(S1map,2),ratio),'max');

```

```
    save(Vresp_name,C)
  end
end
```

```
%%%%%%%%%%%%%%%%%%%%%%%%%%V2layer response%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%  Smap_pooling from: sparseHMAX-v1.2
%  layerC2: calculates V2 layer response
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
function layerC2(C1layer,C2layer,sfilt,s_wh,p_2,ratio,alg,param,flg,inh)
if ~exist(C2layer,'dir'), mkdir(C2layer); end
filelist=dir( fullfile(C1layer,'*.mat') );
for i=1:length(filelist)
    filename=filelist(i).name;
    fullname=fullfile(C2respdir,filename);
    fullname1=fullfile(C1respdir,filename);
    resp=load(fullname1, 'C');
    resp=resp.C;  % 3D response vector from the previous layer

     C2a=layer_SCa(resp,s_wh,sfilt,p_2,alg,param,flg,inh);
     C=Smap_pooling(C2a, ratio, ratio, ...
        mod(size(S2map,1),ratio),mod(size(S2map,2),ratio), 'max');

    savefile(fullname,C);
end


%% Matlab code for simple S + complex Ca layer of the HMAX model
%  patch_normalize from : sparseHMAX-v1.2
function Sout=layer_SCa(X,w_filt,sfilt,rf,alg,param,flg,inh)

[row,col,nfilters]=size(X);
if nfilters==1
X_patches=im2col(X,[rf rf],'sliding');
else
X_patches=im3col(X,rf,rf,nfilters);
end
X_patches=single(X_patches);

%% Normalize the features
X_patches=patch_normalize(X_patches);
X_patches=w_filt*X_patches;

if strcmp(alg,'isa')
    S=X_patches'*sfilt;
```

```
    if flg==1
    ssize=param.groupsize;  % subspace size
    S= resp_Ca_isa(S,ssize,sfilt); % L2 subspace pooling
    S=S';
    end

elseif strcmp(alg,'tica')
    S=X_patches'*sfilt;
    if flg==1
    %ssize=param.nb;
    S= resp_Ca_tica(S',param); % L2 neighbourhood pooling
    S=S';
    end
end
newsz=size(S,2);
Sout=reshape(single(full(S)),row-rf+1,col-rf+1,newsz);

%%%%%%%%L2 pooling for ISA%%%%%%%%%%%%%%%%%%%%
function Sresp=resp_Ca_isa(S,ssize,sfilt)
    Sn=S';
    Sn(Sn<0)=0;
    Sresp=[];
    i=1;
  while i<size(sfilt,2)
      temp=Sn(:,:,(i:i+(ssize-1)));
      Smax=(sqrt(temp.^2));
      Sresp=[Sresp;Smax];
      i=i+ssize;
  end
%%%%%%%%L2 pooling for TICA%%%%%%%%%%%%%%%%%%%%
  function Sresp= resp_Ca_tica(Sn,param)
     Sn(Sn<0)=0;
     Sresp=[];
     Ind=1:param.xdim*param.ydim;
     In=reshape(Ind,param.xdim,param.ydim)';
     Indices=im2col(In,[nb,nb],'distinct');
      for i=1:size(Indices,2)
       temp=Sn(Indices(:,i),:);
       Smax=(sqrt(temp.^2));
       Sresp=[Sresp;Smax];
     end

function savefile(fullname,C)
```

```
save(fullname,'C');

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% FUNCTION: infer_S3resp
% Extracts highlevel feature response
% saves in resultsdir directory.
% OUTPUT: fvector: feature vector of size M X N; M=total number of images
% N=number of S3 filters
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
function fvector=infer_S3resp(datadir,resultsdir,base,~,S2w,V2,S3w,V3,
alg,param1,param2,ra_1,ra_2,flg1,flg2,imsiz,inh,nimages,R_3)
if ~exist(resultsdir,'dir'), mkdir(resultsdir); end
catlist=dir(datadir); %list of object categories
catlist(2)=[];
catlist(1)=[];
catnum=size(catlist,1);
filepaths=cell(1,catnum);

filelist=cell(1,catnum);
for i=1:catnum
    filepaths{i}=fullfile(datadir,catlist(i).name);
    filelist{i}=dir( fullfile(filepaths{i},'*.jpg') );
end

r=1:length(filelist);

rcount=0;
fvector=zeros(nimages,R_3); %Size of the feature vector
for i=1:(length(r));
filestruct=filelist{r(i)};
  for j=1:(size(filestruct,1))
      filename=filestruct(j).name;
      Xi=imread(fullfile(filepaths{i},filename));
    [sy, sx, ~] = size(Xi);
    rescale_factor = imsiz / min(sy, sx);
    Xn = imresize(Xi, round([sy, sx]*rescale_factor));
    X=single(mean(Xn,3));


    %%%%%%%%%%%%%%%%%%%%%%%%% LAYER 1%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
    f=base;
    filter=f.S1;
    S1w=f.S1w;
   % patch=sqrt((size(pmat,2))/3); %%for colour
```

```
    patch=sqrt(size(S1w,2));

    tic;
    C1a=layer_SCa(X,S1w,filter,patch,alg,param1,flg1,inh);
    C=Smap_pooling(C1a, ra_1, ra_1, ...
        mod(size(C1a,1),ra_1),mod(size(C1a,2),ra_1),'max');


   %%%%%%%%%%%%%%%%%%%%%%%% LAYER 2%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

   patch=sqrt(size(V2,2)/size(C,3));
   C2a=layer_SCa(C,V2,S2w,patch,alg,param2,flg2,inh);

   C=Smap_pooling(C2a, ra_2, ra_2, ...
        mod(size(C2a,1),ra_2),mod(size(C2a,2),ra_2), 'max');
   toc;

   %%%%%%%%%%%%%%%%%%%%%%%% LAYER 3%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
   sz=sqrt(size(V3,2)/size(C,3));
   d=layer_Sfinal(C,V3,S3w,sz,alg);
   d(d<0)=0;
   fd=sqrt(sum(sum(d.^2)));
   fvector(rcount,:)=fd;
   rcount=rcount+1;
   C=reshape(fd,1,size(C,3));

    str=strcat('_',int2str(i));
    filename=regexprep(filename, '_',str);
    matfile= regexprep(filename, '.jpg', '.mat');
    Sfile=fullfile(resultsdir,matfile);
    save(Sfile,C)

  end
end
```

```
%%%%%%%%%%%%%%%%%%%%Sample patches based on saliency maps%%%%%%%%%%%%%%%%%%
%Calculates saliency map based on feature integration theory
%Identifies random points on the salient regions of the map
%Extracts patches on the corresponding image based on the points
function X = sample3D_sal(database,layer,samples, winsize,numbases,perc)

% Number of patches per map
num_files = length(database);
```

```
getsample = round(samples/num_files);
samples = getsample * num_files;

% Initialize the matrix to hold the patches
X = zeros(winsize^2*(numbases),samples, 'single');

for i=(1:num_files)

  % Load the map.
  load(fullfile(layer,database(i).Cfile),'C');
  % extract patches at random from C map to make data vector X
  [rowsz,colsz]=size(C(:,:,1));
   smap1=sum(C,3); % primary saliency map
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
  [allPosIndices] = selectSal(smap1, perc, getsample, [rowsz,colsz]);
  [xp, yp]=ind2sub([rowsz,colsz], allPosIndices);

  imshow(mat2gray(smap1))
  hold on
  for j=1:getsample
    r=xp(j);
    c=yp(j);
    if r>=(rowsz-winsize+1) && c>=(colsz-winsize+1)
        rd=r-winsize+1;
        cd=c-winsize+1;
         X(:,(i-1)*getsample+j) = reshape(C(rd:r,cd:c,(1:numbases)),...
        (numbases)*winsize^2,1);
    rectangle('Position', [c-winsize, r-winsize,winsize, winsize],...
        'EdgeColor','r', 'LineWidth', 3);
    plot(c, r, 'rd', 'MarkerFaceColor','g','MarkerSize',8 )
    elseif r>=(rowsz-winsize+1) && c<(colsz-winsize+1)
        rd=r-winsize+1;
        cd=c+winsize-1;
        X(:,(i-1)*getsample+j) = reshape(C(rd:r,c:cd,(1:numbases)),...
        (numbases)*winsize^2,1);
    rectangle('Position', [c, r-winsize,winsize, winsize],...
        'EdgeColor','r', 'LineWidth', 3);
    plot(c, r, 'rd', 'MarkerFaceColor','g','MarkerSize',8 )
    elseif r<(rowsz-winsize+1) && c>=(colsz-winsize+1)
        rd=r+winsize-1;
        cd=c-winsize+1;
        X(:,(i-1)*getsample+j) = reshape(C(r:rd,cd:c,(1:numbases)),...
        (numbases)*winsize^2,1);
```

```
    rectangle('Position', [c-winsize, r,winsize, winsize],...
        'EdgeColor','r', 'LineWidth', 3);
    plot(c, r, 'rd', 'MarkerFaceColor','g','MarkerSize',8 )
    else
    rd=r+winsize-1;
    cd=c+winsize-1;
    X(:,(i-1)*getsample+j) = reshape(C(r:rd,c:cd,(1:numbases)),...
        (numbases)*winsize^2,1);
    rectangle('Position', [c, r,winsize, winsize],...
        'EdgeColor','r', 'LineWidth', 3);
    plot(c, r, 'rd', 'MarkerFaceColor','g','MarkerSize',8 )
    end
  end

end

% Adapted from: selectSamplesPerImg
% Matlab tools for "Learning to Predict Where Humans Look" ICCV 2009
% Tilke Judd, Kristen Ehinger, Fredo Durand, Antonio Torralba
function [salient_points] = selectSal(C, p, num_salPoints, dims)

   % select samples examples randomly from top p salient

    pIndx = [];
    C=reshape(C, [dims(1)*dims(2), 1]);
    [~, X] = sort(C, 'descend');

    % Find the positive examples in the top p percent
    i = ceil((p/100)*length(C)*rand([num_salPoints, 1]));
    pos_indx = X(i);
    salient_points = [pIndx, pos_indx'];
```

```
%%%%%%%%%%%%%%Compress sparse 3D data using compressed sensing%%%%%%%%%%%
%%Requires Model-CS Toolbox v1.1

function [rec,Yc]=compress(X,M,rmslim,i)

% compresses a single sparse vector with measurement matrix Phi
% and then applies it to the rest of the data

 s=sum(X~=0);  %number of non zero entries in each patch
 sz=size(X,1);

[test,idx]=max(s);
```

```
N=size(X,1);
Bm=X(:,idx); %Reference data
K=test;      %number of non zero elements of signal Bm
iter=10;
rms=0;
maxiter=0;

temp=cell(1,2);
rmsct=cell(1,2);
maxvalue=5000;
while maxiter<maxvalue
    fprintf('cosamp rms %6.4f\n',rms);
   Phi = ((1/sqrt(M))*abs(randn(M,N)));
    y = Phi*Bm;

     [xhat,~]=cosamp(y,Phi,K,iter);  % from: Model-CS Toolbox v1.1 .
     %    [xhat,~] = cosamp_fun(y, Phi_f, PhiT_f, N, K, iter);

    rms = sqrt(mean((Bm(:)-xhat(:)).^2));
    if rms < rmslim
      Phi_s=Phi;
      rms_s=rmsct{1};
      rec.iter=iter;
      rec.N=N;
      rec.M=M;
      rec.K=K;
      rec.Phi=Phi;
%       rmsval=checkfullcompress(Phi,X,K,iter);
%       rms_full=rmsval;
      break;
    end

    if maxiter==0
        temp{1}=Phi;
        rmsct{1}=rms;
    else
        temp{2}=Phi;
        rmsct{2}=rms;
        if rmsct{1}>rmsct{2}
        temp{1}=temp{2};
        rmsct{1}=rmsct{2};
        end

```

```
    end

    maxiter=maxiter+1;
    if maxiter==maxvalue-1;
        fprintf('max reached: No further compression');
        options.Interpreter = 'tex';
        qstring = ['rms=',num2str(rmsct{1}),':','Is rms less than',num2str(rmslim),'?'];
        choice = questdlg(qstring,'Boundary Condition',...
    'Yes','No',options);
        if strcmp(choice,'No')
            error('rms standard not met: K to large');
        else
            Phi_s=temp{1};
            rms_s=rmsct{1};
            rec.iter=iter;
            rec.N=N;
            rec.M=M;
            rec.K=K;
            rec.Phi=Phi;
        end
    end
end

nameCS=['CS_',num2str(sz),'_to_',num2str(M),'_',num2str(i)];
save(nameCS,'Phi_s','rms_s') % save measurement matrix
Yc=zeros(M,size(X,2));

for i=1:size(X,2)    % compress rest of the data
    Yc(:,i)=rec.Phi*(X(:,i));
end
```

Bibliography

[1] E. T. Rolls, "Invariant visual object and face recognition: Neural and computational bases, and a model, VisNet," *Frontiers in computational neuroscience*, vol. 6, p. 35, 2012, PMID: 22723777. v, 6, 11, 30, 31, 33

[2] S. Grossberg, "View-invariant object category learning, attention, recognition, search, and scene understanding," in *International Joint Conference on Neural Networks, 2009. IJCNN 2009*, 2009, pp. 1002–1004. v, 9

[3] A. L. Hodgkin and A. F. Huxley, "A quantitative description of membrane current and its application to conduction and excitation in nerve," *The Journal of Physiology*, vol. 117, no. 4, pp. 500–544, Aug. 1952, PMID: 12991237 PMCID: PMC1392413. [Online]. Available: http://www.ncbi.nlm.nih.gov/pmc/articles/PMC1392413/ [Accessed: 2013-07-06] v, 10, 11

[4] T. Rodemann and E. Krner, "Two separate processing streams in a cortical-type architecture," *Neurocomputing*, vol. 3840, pp. 1541–1547, Jun. 2001. [Online]. Available: http://www.sciencedirect.com/science/article/pii/S0925231201005549 [Accessed: 2013-07-05] v, 11, 12

[5] K. Fukushima, "Neocognitron: A self-organizing neural network model for a mechanism of pattern recognition unaffected by shift in position," *Biological Cybernetics*, vol. 36, no. 4, pp. 193–202, Apr. 1980. [Online]. Available: http://link.springer.com/article/10.1007/BF00344251 [Accessed: 2013-07-05] v, 15, 16

[6] Y. Lecun, L. Bottou, Y. Bengio, and P. Haffner, "Gradient-based learning applied to document recognition," *Proceedings of the IEEE*, vol. 86, no. 11, pp. 2278–2324, Nov. 1998. v, 16, 17

[7] A. Krizhevsky, I. Sutskever, and G. E. Hinton, "ImageNet classification with deep convolutional neural networks," in *Advances in Neural Information Processing Systems 25*, F. Pereira, C. J. C. Burges, L. Bottou, and K. Q. Weinberger, Eds. Curran Associates, Inc., pp. 1097–1105. [Online]. Available: http://papers.nips.cc/paper/

4824-imagenet-classification-with-deep-convolutional-neural-networks.pdf [Accessed: 2017-04-26] v, 18

[8] M. Riesenhuber and T. Poggio, "Hierarchical models of object recognition in cortex," *Nature neuroscience*, vol. 2, no. 11, pp. 1019–1025, Nov. 1999, PMID: 10526343. v, 2, 6, 8, 13, 14, 21, 22, 33, 95

[9] D. George and J. Hawkins, "A hierarchical bayesian model of invariant pattern recognition in the visual cortex," in *2005 IEEE International Joint Conference on Neural Networks, 2005. IJCNN '05. Proceedings*, vol. 3, 2005, pp. 1812–1817 vol. 3. v, 23, 24, 30

[10] X. Hu, J. Zhang, J. Li, and B. Zhang, "Sparsity-Regularized HMAX for Visual Recognition," *PLOS ONE*, vol. 9, no. 1, p. e81813, Jan. 2014. [Online]. Available: http://journals.plos.org/plosone/article?id=10.1371/journal.pone.0081813 [Accessed: 2016-09-06] v, vi, vii, 2, 3, 11, 15, 25, 26, 27, 30, 33, 34, 42, 43, 44, 48, 51, 52, 53, 63, 68, 72, 74, 76

[11] C. Theriault, N. Thome, and M. Cord, "Extended Coding and Pooling in the HMAX Model," *IEEE Transactions on Image Processing*, vol. 22, no. 2, pp. 764–777, Feb. 2013. v, 23, 26, 27, 52, 74, 76

[12] Q. V. Le, W. Y. Zou, S. Y. Yeung, and A. Y. Ng, "Learning Hierarchical Invariant Spatio-temporal Features for Action Recognition with Independent Subspace Analysis," in *Proceedings of the 2011 IEEE Conference on Computer Vision and Pattern Recognition*, ser. CVPR '11. Washington, DC, USA: IEEE Computer Society, 2011, pp. 3361–3368. [Online]. Available: http://dx.doi.org/10.1109/CVPR.2011.5995496 [Accessed: 2016-09-06] v, 2, 3, 20, 26, 27, 28, 33, 34

[13] Q. Le, M. Ranzato, R. Monga, M. Devin, K. Chen, G. Corrado, J. Dean, and A. Ng, "Building high-level features using large scale unsupervised learning," 2012. [Online]. Available: http://research.google.com/pubs/pub38115.html [Accessed: 2016-09-06] v, 2, 3, 15, 20, 25, 26, 27, 29, 33, 34, 68, 122

[14] D. George, "How the Brain Might Work: A Hierarchical and Temporal Model for Learning and Recognition," Ph.D. dissertation, Stanford University, Stanford, CA, USA, 2008, aAI3313576. v, 6, 30, 31

[15] L. Fei-Fei, R. Fergus, and P. Perona, "Learning Generative Visual Models from Few Training Examples: An Incremental Bayesian Approach Tested on 101 Object Categories," *Comput. Vis. Image Underst.*, vol. 106, no. 1, pp. 59–70, Apr. 2007. [Online]. Available: http://dx.doi.org/10.1016/j.cviu.2005.09.012 [Accessed: 2016-09-28] vi, vii, viii, 49, 50, 73, 86, 88, 89, 90, 91, 94, 97, 114, 115

[16] C. Theriault, N. Thome, and M. Cord, "HMAX-S: Deep scale representation for biologically inspired image categorization," in *2011 18th IEEE International Conference on Image Processing*, Sep. 2011, pp. 1261–1264. vi, 15, 23, 26, 33, 52, 53, 71, 72, 74, 76, 77

[17] L. Itti, C. Koch, and E. Niebur, "A Model of Saliency-Based Visual Attention for Rapid Scene Analysis," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 20, no. 11, pp. 1254–1259, Nov. 1998. [Online]. Available: http://dx.doi.org/10.1109/34.730558 [Accessed: 2016-09-15] vii, 4, 79, 80, 81, 89, 90

[18] A. Torralba, A. Oliva, M. S. Castelhano, and J. M. Henderson, "Contextual guidance of eye movements and attention in real-world scenes: the role of global features in object search," *Psychological Review*, vol. 113, no. 4, pp. 766–786, Oct. 2006. vii, 81, 83, 84, 90, 95, 96, 98

[19] J. Harel, C. Koch, and P. Perona, "Graph-Based Visual Saliency," 2006, pp. 545–552. [Online]. Available: http://machinelearning.wustl.edu/mlpapers/papers/NIPS2006_897 [Accessed: 2016-09-17] vii, 4, 83, 87, 89, 90, 120

[20] L. Zhang, M. H. Tong, T. K. Marks, H. Shan, and G. W. Cottrell, "SUN: A Bayesian framework for saliency using natural statistics," *Journal of Vision*, vol. 8, no. 7, pp. 32.1–20, 2008. vii, 4, 81, 83, 84, 85, 86, 89, 90, 120

[21] "Hierarchical Models of the Visual System Detailed Description - Semantic Scholar." [Online]. Available: https://www.semanticscholar.org/paper/Hierarchical-Models-of-the-Visual-System-Detailed-Serre/35a4bcace1e2d47803dc14305abfeae4823e54f1 [Accessed: 2016-09-06] 1, 6, 16, 18

[22] K. He, X. Zhang, S. Ren, and J. Sun, "Delving Deep into Rectifiers: Surpassing Human-Level Performance on ImageNet Classification," *arXiv:1502.01852 [cs]*, Feb. 2015, arXiv: 1502.01852. [Online]. Available: http://arxiv.org/abs/1502.01852 [Accessed: 2016-09-26] 1, 19

[23] N. Kruger, P. Janssen, S. Kalkan, M. Lappe, A. Leonardis, J. Piater, A. J. Rodriguez-Sanchez, and L. Wiskott, "Deep Hierarchies in the Primate Visual Cortex: What Can We Learn for Computer Vision?" *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 35, no. 8, pp. 1847–1871, Aug. 2013. 1, 5, 6, 7, 8, 41

[24] D. H. Hubel and T. N. Wiesel, "Receptive fields, binocular interaction and functional architecture in the cat's visual cortex," *The Journal of Physiology*, vol. 160, no. 1, pp. 106–154.2, Jan. 1962. [Online]. Available: http://www.ncbi.nlm.nih.gov/pmc/articles/PMC1359523/ [Accessed: 2016-09-06] 1

[25] B. A. Olshausen and D. J. Field, "Sparse coding with an overcomplete basis set: a strategy employed by V1?" *Vision Research*, vol. 37, no. 23, pp. 3311–3325, Dec. 1997. 2, 33, 34, 102, 106

[26] P. Földiák and M. P. Young, "The Handbook of Brain Theory and Neural Networks," M. A. Arbib, Ed. Cambridge, MA, USA: MIT Press, 1998, pp. 895–898. [Online]. Available: http://dl.acm.org/citation.cfm?id=303568.303958 [Accessed: 2016-09-27] 2

[27] B. A. Olshausen and D. J. Field, "Emergence of simple-cell receptive field properties by learning a sparse code for natural images," *Nature*, vol. 381, no. 6583, pp. 607–609, Jun. 1996. 2, 11, 34

[28] K. Yu, Y. Lin, and J. Lafferty, "Learning image representations from the pixel level via hierarchical sparse coding," in *2011 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, Jun. 2011, pp. 1713–1720. 2, 25, 27, 53, 74, 76

[29] T. Serre and M. Riesenhuber, *Realistic modeling of simple and complex cell tuning in the hmax model, and implications for invariant object recognition in cortex*, 2004. 2, 7, 8, 21, 22, 25, 34

[30] A. Hyvärinen, P. O. Hoyer, and M. Inki, "Topographic independent component analysis," *Neural Computation*, vol. 13, no. 7, pp. 1527–1558, Jul. 2001. 3, 27, 33, 34, 36, 37, 63, 106

[31] A. Hyvärinen and P. Hoyer, "Emergence of phase- and shift-invariant features by decomposition of natural images into independent feature subspaces," *Neural Computation*, vol. 12, no. 7, pp. 1705–1720, Jul. 2000. 3, 26, 33, 34, 39

[32] A. Hyvärinen, J. Hurri, and P. O. Hoyer, *Natural Image Statistics: A Probabilistic Approach to Early Computational Vision.* Springer Science & Business Media, Apr. 2009, google-Books-ID: pq_Fr1eYr7cC. 3, 8, 27, 34, 35, 36, 37, 41, 42, 63, 64, 77

[33] A. Hyvärinen and E. Oja, "Independent component analysis: algorithms and applications," *Neural Networks: The Official Journal of the International Neural Network Society*, vol. 13, no. 4-5, pp. 411–430, Jun. 2000. 3, 27, 34, 106

[34] A. M. Treisman and G. Gelade, "A feature-integration theory of attention," *Cognitive Psychology*, vol. 12, no. 1, pp. 97–136, Jan. 1980. [Online]. Available: http://www.sciencedirect.com/science/article/pii/0010028580900055 [Accessed: 2016-09-15] 4, 79, 95

[35] A. Borji and L. Itti, "State-of-the-Art in Visual Attention Modeling," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 35, no. 1, pp. 185–207, Jan. 2013. [Online]. Available: http://dx.doi.org/10.1109/TPAMI.2012.89 [Accessed: 2016-09-15] 4, 8, 9, 80, 81, 83, 87, 95, 96, 98

[36] E. Candes and M. Wakin, "An Introduction To Compressive Sampling," *IEEE Signal Processing Magazine*, vol. 25, no. 2, pp. 21–30, Mar. 2008. [Online]. Available: http://ieeexplore.ieee.org/document/4472240/ [Accessed: 2016-09-28] 4, 101, 102, 104, 105, 106, 110

[37] H. Lee, R. Grosse, R. Ranganath, and A. Y. Ng, "Convolutional Deep Belief Networks for Scalable Unsupervised Learning of Hierarchical Representations," in *Proceedings of the 26th Annual International Conference on Machine Learning*, ser. ICML '09. New York, NY, USA: ACM, 2009, pp. 609–616. [Online]. Available: http://doi.acm.org/10.1145/1553374.1553453 [Accessed: 2016-09-13] 6, 15, 20, 76

[38] A. London, I. Benhar, and M. Schwartz, "The retina as a window to the brainŮfrom eye research to CNS disorders," *Nature Reviews Neurology*, vol. 9, no. 1, pp. 44–53, Nov. 2012. [Online]. Available: http://www.nature.com/doifinder/10.1038/nrneurol.2012.227 [Accessed: 2016-09-26] 7

[39] H. Jhuang, T. Serre, L. Wolf, and T. Poggio, "A biologically inspired system for action recognition," in *In ICCV*, 2007, p. 18. 7, 22

[40] D. H. Hubel and T. N. Wiesel, "Republication of the journal of physiology (1959) 148, 574-591: Receptive fields of single neurones in the cat's striate cortex. 1959," *The Journal of physiology*, vol. 587, no. Pt 12, pp. 2721–2732, Jun. 2009, PMID: 19525558. 8, 14

[41] N. C. Foley, S. Grossberg, and E. Mingolla, "Neural dynamics of object-based multifocal visual spatial attention and priming: Object cueing, useful-field-of-view, and crowding," *Cognitive Psychology*, vol. 65, no. 1, pp. 77–117, Aug. 2012. [Online]. Available: http://www.sciencedirect.com/science/article/pii/S0010028512000138 [Accessed: 2013-07-05] 8, 9

[42] C. M. Gray, *The Temporal Correlation Hypothesis Review of Visual Feature Integration: Still Alive and Well*. 8, 10, 11

[43] L. Zhang, M. H. Tong, T. K. Marks, H. Shan, and G. W. Cottrell, "SUN: A Bayesian framework for saliency using natural statistics," *Journal of Vision*, vol. 8, no. 7, pp. 32.1–20, 2008. 9

[44] T. Gollisch, "Throwing a glance at the neural code: Rapid information transmission in the visual system," *HFSP Journal*, vol. 3, no. 1, pp. 36–46, 2009, PMID: 19649155. [Online]. Available: http://www.tandfonline.com/doi/abs/10.2976/1.3027089 [Accessed: 2013-07-05] 9

[45] I. Nemenman, G. D. Lewen, W. Bialek, and R. R. de Ruyter van Steveninck, "Neural coding of natural stimuli: Information at sub-millisecond resolution," *PLoS Comput Biol*, vol. 4, no. 3, p. e1000025, Mar. 2008. [Online]. Available: http://dx.plos.org/10.1371/journal.pcbi.1000025 [Accessed: 2013-07-05] 9

[46] D. S. Levine, *Introduction to Neural and Cognitive Modeling*. Routledge, 2000. 10

[47] L. M. Ward, "Synchronous neural oscillations and cognitive processes," *Trends in Cognitive Sciences*, vol. 7, no. 12, pp. 553–559, Dec. 2003. [Online]. Available: http://www.cell.com/trends/cognitive-sciences/abstract/S1364-6613(03)00289-4 [Accessed: 2013-07-05] 10

[48] J. P. Lachaux, E. Rodriguez, J. Martinerie, and F. J. Varela, "Measuring phase synchrony in brain signals," *Human Brain Mapping*, vol. 8, no. 4, pp. 194–208, 1999. 10

[49] E. Izhikevich, "Which model to use for cortical spiking neurons?" *IEEE Transactions on Neural Networks*, vol. 15, no. 5, pp. 1063–1070, 2004. 11

[50] C. v. d. Malsburg, "Binding in models of perception and brain function," *Current Opinion in Neurobiology*, vol. 5, no. 4, pp. 520–526, Aug. 1995. [Online]. Available: http://www.sciencedirect.com/science/article/pii/095943889580014X [Accessed: 2013-07-07] 11, 122

[51] S. Thorpe, D. Fize, and C. Marlot, "Speed of processing in the human visual system," *Nature*, vol. 381, no. 6582, pp. 520–522, Jun. 1996, PMID: 8632824. 11

[52] E. Körner, M.-O. Gewaltig, U. Körner, A. Richter, and T. Rodemann, "A model of computation in neocortical architecture," *Neural Networks: The Official Journal of the International Neural Network Society*, vol. 12, no. 7-8, pp. 989–1005, Oct. 1999. 12, 30

[53] D. Marr, T. A. Poggio, and S. Ullman, *Vision: A Computational Investigation into the Human Representation and Processing of Visual Information*. Cambridge, Mass.: The MIT Press, Jul. 2010. 14

[54] T. Serre, L. Wolf, S. Bileschi, M. Riesenhuber, and T. Poggio, "Robust object recognition with cortex-like mechanisms," *IEEE transactions on pattern analysis and machine intelligence*, vol. 29, no. 3, pp. 411–426, Mar. 2007. 15, 22, 26, 33, 71, 74, 76

[55] K. Fukushima and N. Wake, "Improved neocognitron with bend-detecting cells," vol. 4. IEEE, 1992, pp. 190–195. [Online]. Available: http://ieeexplore.ieee.org/document/227343/ [Accessed: 2016-09-24] 15

[56] R.-i. T. Naoyuki Tsuruta, "Hypercolumn model: A modified model of neocognitron using hierarchical self-organizing maps," pp. 840–849, 2006. 15

[57] D. G. Lowe, "Distinctive image features from scale-invariant keypoints," *Int. J. Comput. Vision*, vol. 60, no. 2, p. 91110, Nov. 2004. [Online]. Available: http://dx.doi.org/10.1023/B:VISI.0000029664.99615.94 [Accessed: 2013-07-06] 15

[58] Y. LeCun, K. Kavukcuoglu, and C. Farabet, "Convolutional networks and applications in vision." IEEE, May 2010, pp. 253–256. [Online]. Available: http://ieeexplore.ieee.org/document/5537907/ [Accessed: 2016-09-25] 17, 19, 120

[59] S. R. Kheradpisheh, M. Ghodrati, M. Ganjtabesh, and T. Masquelier, "Deep networks can resemble human feed-forward vision in invariant object recognition," vol. 6, p. 32672. [Online]. Available: http://www.nature.com/srep/2016/160907/srep32672/full/srep32672.html [Accessed: 2017-04-26] 17, 19

[60] T. Serre, "Hierarchical Models of the Visual System," in *Encyclopedia of Computational Neuroscience*, D. Jaeger and R. Jung, Eds. New York, NY: Springer New York, 2014, pp. 1–12. [Online]. Available: http://link.springer.com/10.1007/978-1-4614-7320-6_345-1 [Accessed: 2016-09-06] 18, 29, 41, 46

[61] O. Russakovsky, J. Deng, H. Su, J. Krause, S. Satheesh, S. Ma, Z. Huang, A. Karpathy, A. Khosla, M. Bernstein, A. C. Berg, and L. Fei-Fei, "ImageNet Large Scale Visual Recognition Challenge," *arXiv:1409.0575 [cs]*, Sep. 2014, arXiv: 1409.0575. [Online]. Available: http://arxiv.org/abs/1409.0575 [Accessed: 2016-09-28] 18, 28

[62] J. Deng, W. Dong, R. Socher, L.-J. Li, K. Li, and L. Fei-Fei, "Imagenet: A large-scale hierarchical image database," in *Computer Vision and Pattern Recognition, 2009. CVPR 2009. IEEE Conference on*. Ieee, pp. 248–255. [Online]. Available: http://ieeexplore.ieee.org/abstract/document/5206848/ [Accessed: 2017-04-26] 18

[63] M. D. Zeiler and R. Fergus, "Visualizing and understanding convolutional networks." [Online]. Available: http://arxiv.org/abs/1311.2901 [Accessed: 2017-04-26] 18, 19, 76, 120

[64] M. D. Zeiler, G. W. Taylor, and R. Fergus, "Adaptive Deconvolutional Networks for Mid and High Level Feature Learning," in *Proceedings of the 2011 International Conference on Computer Vision*, ser. ICCV '11. Washington, DC, USA: IEEE Computer Society, 2011, pp. 2018–2025. [Online]. Available: http://dx.doi.org/10.1109/ICCV.2011.6126474 [Accessed: 2016-09-13] 18, 20, 53, 74, 75, 76

[65] Q. Guo, F. Wang, J. Lei, D. Tu, and G. Li, "Convolutional feature learning and hybrid CNN-HMM for scene number recognition," vol. 184, pp. 78–90. [Online]. Available: http://www.sciencedirect.com/science/article/pii/S0925231215018950 [Accessed: 2017-04-26] 19

[66] C. Szegedy, W. Liu, Y. Jia, P. Sermanet, S. Reed, D. Anguelov, D. Erhan, V. Vanhoucke, and A. Rabinovich, "Going deeper with convolutions." [Online]. Available: http://arxiv.org/abs/1409.4842 [Accessed: 2017-04-26] 19, 50, 65

[67] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition." [Online]. Available: http://arxiv.org/abs/1512.03385 [Accessed: 2017-04-26] 19

[68] Caltech-256 object category dataset (PDF download available). [Online]. Available: https://www.researchgate.net/publication/30766223_Caltech-256_Object_Category_Dataset [Accessed: 2017-04-26] 19, 73

[69] G. E. Hinton, S. Osindero, and Y.-W. Teh, "A fast learning algorithm for deep belief nets," vol. 18, no. 7, pp. 1527–1554. [Online]. Available: http://dx.doi.org/10.1162/neco.2006.18.7.1527 [Accessed: 2017-04-26] 20

[70] A. Coates, A. Karpathy, and A. Y. Ng, "Emergence of object-selective features in unsupervised feature learning," in *Advances in Neural Information Processing Systems 25*, F. Pereira, C. J. C. Burges, L. Bottou, and K. Q. Weinberger, Eds. Curran Associates, Inc., pp. 2681–2689. [Online]. Available: http://papers.nips.cc/paper/

4497-emergence-of-object-selective-features-in-unsupervised-feature-learning.pdf [Accessed: 2017-04-26] 20

[71] M. Riesenhuber and T. Poggio, "Models of object recognition," *Nature Neuroscience*, vol. 3, pp. 1199–1204, 2000. [Online]. Available: http://www.nature.com/neuro/journal/v3/n11s/full/nn1100_1199.html [Accessed: 2013-07-05] 20, 21

[72] J. Mutch and D. Lowe, "Multiclass object recognition with sparse, localized features," in *2006 IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, vol. 1, 2006, pp. 11–18. 21, 23, 53, 71, 72

[73] A. J. Yu, M. A. Giese, and T. A. Poggio, "Biophysiologically plausible implementations of the maximum operation," *Neural computation*, vol. 14, no. 12, pp. 2857–2881, Dec. 2002, PMID: 12487795. 21

[74] J. Mutch and D. G. Lowe, "Object Class Recognition and Localization Using Sparse Features with Limited Receptive Fields," *International Journal of Computer Vision*, vol. 80, no. 1, pp. 45–57, Jan. 2008. [Online]. Available: http://link.springer.com/article/10.1007/s11263-007-0118-0 [Accessed: 2016-09-13] 22, 23, 26, 74, 76

[75] T. Serre, L. Wolf, and T. Poggio, "Object recognition with features inspired by visual cortex," in *IEEE Computer Society Conference on Computer Vision and Pattern Recognition, 2005. CVPR 2005*, vol. 2, 2005, pp. 994–1000 vol. 2. 22, 26, 74

[76] S. Lazebnik, C. Schmid, and J. Ponce, "Beyond Bags of Features: Spatial Pyramid Matching for Recognizing Natural Scene Categories," in *Proceedings of the 2006 IEEE Computer Society Conference on Computer Vision and Pattern Recognition - Volume 2*, ser. CVPR '06. Washington, DC, USA: IEEE Computer Society, 2006, pp. 2169–2178. [Online]. Available: http://dx.doi.org/10.1109/CVPR.2006.68 [Accessed: 2016-09-15] 23, 25, 26, 74

[77] S. Dura-Bernal, T. Wennekers, and S. L. Denham, "Top-Down Feedback in an HMAX-Like Cortical Model of Object Perception Based on Hierarchical Bayesian Networks and Belief Propagation," *PLOS ONE*, vol. 7, no. 11, p.

e48216, Nov. 2012. [Online]. Available: http://journals.plos.org/plosone/article?id=10.1371/journal.pone.0048216 [Accessed: 2016-09-24] 23, 24, 31

[78] S. O. Murray, P. Schrater, and D. Kersten, "Perceptual grouping and the interactions between visual cortical areas," *Neural Networks: The Official Journal of the International Neural Network Society*, vol. 17, no. 5-6, pp. 695–705, Jul. 2004. 23

[79] M. A. Williams, C. I. Baker, H. P. Op de Beeck, W. Mok Shim, S. Dang, C. Triantafyllou, and N. Kanwisher, "Feedback of visual object information to foveal retinotopic cortex," *Nature Neuroscience*, vol. 11, no. 12, pp. 1439–1445, Dec. 2008. [Online]. Available: http://www.nature.com/neuro/journal/v11/n12/abs/nn.2218.html [Accessed: 2016-09-25] 23

[80] T. S. Lee and D. Mumford, "Hierarchical bayesian inference in the visual cortex," *Journal of the Optical Society of America. A, Optics, image science, and vision*, vol. 20, no. 7, pp. 1434–1448, Jul. 2003, PMID: 12868647. 23

[81] R. Baddeley, L. F. Abbott, M. C. A. Booth, F. Sengpiel, T. Freeman, E. A. Wakeman, and E. T. Rolls, "Responses of Neurons in Primary and Inferior Temporal Visual Cortices to Natural Scenes," *Proceedings: Biological Sciences*, vol. 264, no. 1389, pp. 1775–1783, 1997. [Online]. Available: http://www.jstor.org/stable/51114 [Accessed: 2016-09-06] 25, 30, 33

[82] Y. Li, W. Wu, B. Zhang, and F. Li, "Enhanced HMAX model with feedforward feature learning for multiclass categorization," *Frontiers in Computational Neuroscience*, vol. 9, p. 123, 2015. 26, 85, 95, 107

[83] J. Yang, K. Yu, Y. Gong, and T. Huang, "Linear spatial pyramid matching using sparse coding for image classification," in *IEEE Conference on Computer Vision and Pattern Recognition, 2009. CVPR 2009*, Jun. 2009, pp. 1794–1801. 26

[84] J. Wright, Y. Ma, J. Mairal, G. Sapiro, T. S. Huang, and S. Yan, "Sparse Representation for Computer Vision and Pattern Recognition," *Proceedings of the IEEE*, vol. 98, no. 6, pp. 1031–1044, Jun. 2010. 27

[85] J. Han, S. He, X. Qian, D. Wang, L. Guo, and T. Liu, "An Object-Oriented Visual Saliency Detection Framework Based on Sparse Coding Representa-

tions," *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 23, no. 12, pp. 2009–2021, Dec. 2013. 27

[86] H. Li, H. Li, Y. Wei, Y. Tang, and Q. Wang, "Sparse-based neural response for image classification," *Neurocomputing*, vol. 144, pp. 198–207, Nov. 2014. [Online]. Available: http://www.sciencedirect.com/science/article/pii/S0925231214006778 [Accessed: 2016-09-06] 27

[87] M. U. Gutmann and A. Hyvärinen, "A three-layer model of natural image statistics," *Journal of Physiology-Paris*, vol. 107, no. 5, pp. 369–398, Nov. 2013. [Online]. Available: http://www.sciencedirect.com/science/article/pii/S0928425713000028 [Accessed: 2016-09-06] 27

[88] Z. Wang, Y. Huang, S. Luo, and L. Wang, "A biologically inspired system for fast handwritten digit recognition," in *2011 18th IEEE International Conference on Image Processing*, Sep. 2011, pp. 1749–1752. 27

[89] Q. V. Le, J. Ngiam, Z. Chen, D. Chia, P. W. Koh, and A. Y. Ng, "Tiled convolutional neural networks," 2010. [Online]. Available: https://core.ac.uk/display/21403339 [Accessed: 2016-09-06] 27, 28

[90] Q. V. Le, A. Karpenko, J. Ngiam, and A. Y. Ng, "ICA with Reconstruction Cost for Efficient Overcomplete Feature Learning," *ResearchGate*, vol. 24, Jul. 2015. [Online]. Available: https://www.researchgate.net/publication/268265707_ICA_with_Reconstruction_Cost_for_Efficient_Overcomplete_Feature_Learning [Accessed: 2016-09-06] 28

[91] G. Huang, M. Mattar, H. Lee, and E. G. Learned-miller, "Learning to Align from Scratch," in *Advances in Neural Information Processing Systems 25*, F. Pereira, C. J. C. Burges, L. Bottou, and K. Q. Weinberger, Eds. Curran Associates, Inc., 2012, pp. 764–772. [Online]. Available: http://papers.nips.cc/paper/4769-learning-to-align-from-scratch.pdf [Accessed: 2016-09-28] 28

[92] T. Masquelier and S. J. Thorpe, "Unsupervised Learning of Visual Features through Spike Timing Dependent Plasticity," *PLOS Comput Biol*, vol. 3, no. 2, p. e31, Feb. 2007. [Online]. Available: http://journals.

plos.org/ploscompbiol/article?id=10.1371/journal.pcbi.0030031 [Accessed: 2016-09-25] 31

[93] T. Masquelier, T. Serre, S. Thorpe, and T. Poggio, "Learning complex cell invariance from natural videos: A plausibility proof," Dec. 2007. [Online]. Available: http://dspace.mit.edu/handle/1721.1/39833 [Accessed: 2016-09-25] 31

[94] E. T. Rolls and A. Treves, "The neuronal encoding of information in the brain," *Progress in Neurobiology*, vol. 95, no. 3, pp. 448–490, Nov. 2011. [Online]. Available: http://www.sciencedirect.com/science/article/pii/S030100821100147X [Accessed: 2016-09-06] 33

[95] Y. Xu, T. Xiao, J. Zhang, K. Yang, and Z. Zhang, "Scale-invariant convolutional neural networks." [Online]. Available: http://arxiv.org/abs/1411.6369 [Accessed: 2017-04-26] 34, 72, 121

[96] A. Hyvärinen and U. Köster, "Complex cell pooling and the statistics of natural images," *Network: Computation in Neural Systems*, vol. 18, no. 2, pp. 81–100, Jan. 2007. [Online]. Available: http://dx.doi.org/10.1080/09548980701418942 [Accessed: 2016-09-07] 41, 42, 53, 57, 61

[97] C.-C. Chang and C.-J. Lin, "LIBSVM: A Library for Support Vector Machines," *ACM Trans. Intell. Syst. Technol.*, vol. 2, no. 3, pp. 27:1–27:27, May 2011. [Online]. Available: http://doi.acm.org/10.1145/1961189.1961199 [Accessed: 2016-09-28] 51

[98] J. Yang, K. Yu, Y. Gong, and T. Huang, "Linear spatial pyramid matching using sparse coding for image classification," in *Computer Vision and Pattern Recognition, 2009. CVPR 2009. IEEE Conference on*. IEEE, 2009, pp. 1794–1801. 53

[99] T. Serre, A. Oliva, and T. Poggio, "A feedforward architecture accounts for rapid categorization," *Proceedings of the National Academy of Sciences*, vol. 104, no. 15, pp. 6424–6429, Apr. 2007. [Online]. Available: http://www.pnas.org/content/104/15/6424 [Accessed: 2016-09-28] 64

[100] G. Rhodes, A. Calder, M. Johnson, and J. V. Haxby, Eds., *Oxford Handbook of Face Perception*, 1st ed. Oxford University Press, Jul. 2011. [Online]. Available: http://www.oxfordhandbooks.com/view/10.1093/oxfordhb/

9780199559053.001.0001/oxfordhb-9780199559053 [Accessed: 2016-09-14] 67

[101] R.-E. Fan, K.-W. Chang, C.-J. Hsieh, X.-R. Wang, and C.-J. Lin, "LIBLINEAR: A Library for Large Linear Classification," *J. Mach. Learn. Res.*, vol. 9, pp. 1871–1874, Jun. 2008. [Online]. Available: http://dl.acm.org/citation.cfm?id=1390681.1442794 [Accessed: 2016-09-23] 74

[102] C. Kanan, C. Kanan, and G. Cottrell, "Robust Classification of Objects, Faces, and Flowers Using Natural Image Statistics." [Online]. Available: http://citeseerx.ist.psu.edu/viewdoc/summary?doi=10.1.1.163.8947 78

[103] C. Summerfield and T. Egner, "Expectation (and attention) in visual cognition," *Trends in Cognitive Sciences*, vol. 13, no. 9, pp. 403–409, Sep. 2009. 79

[104] C. Koch and S. Ullman, "Shifts in selective visual attention: towards the underlying neural circuitry," *Human Neurobiology*, vol. 4, no. 4, pp. 219–227, 1985. 79

[105] T. Judd, K. Ehinger, F. Durand, and A. Torralba, "Learning to predict where humans look." IEEE, Sep. 2009, pp. 2106–2113. [Online]. Available: http://ieeexplore.ieee.org/lpdocs/epic03/wrapper.htm?arnumber=5459462 [Accessed: 2016-09-15] 80, 81, 83, 89

[106] X. Zhang, L. Zhaoping, T. Zhou, and F. Fang, "Neural activities in v1 create a bottom-up saliency map," *Neuron*, vol. 73, no. 1, pp. 183–192, Jan. 2012. 80

[107] T. Judd, F. Durand, and A. Torralba, "A Benchmark of Computational Models of Saliency to Predict Human Fixations," Jan. 2012. [Online]. Available: http://dspace.mit.edu/handle/1721.1/68590 [Accessed: 2016-09-15] 80, 81, 83, 89

[108] D. Walther and C. Koch, "Modeling attention to salient proto-objects," *Neural Networks: The Official Journal of the International Neural Network Society*, vol. 19, no. 9, pp. 1395–1407, Nov. 2006. 80

[109] T. N. Mundhenk and L. Itti, "Computational modeling and exploration of contour integration for visual saliency," *Biological Cybernetics*, vol. 93, no. 3, pp. 188–212, Sep. 2005. 81

[110] J. Wang, A. Borji, C.-C. Jay Kuo, and L. Itti, "Learning a Combined Model of Visual Saliency for Fixation Prediction," *IEEE transactions on image processing: a publication of the IEEE Signal Processing Society*, vol. 25, no. 4, pp. 1566–1579, Apr. 2016. 81

[111] J. M. Wolfe, "Guided Search 4.0," in *Integrated Models of Cognitive Systems*, W. D. Gray, Ed. Oxford University Press, May 2007, pp. 99–119. [Online]. Available: http://www.oxfordscholarship.com/view/10.1093/acprof:oso/9780195189193.001.0001/acprof-9780195189193-chapter-8 [Accessed: 2016-09-17] 81

[112] Y. Kavak, E. Erdem, and A. Erdem, "Visual saliency estimation by integrating features using multiple kernel learning," *arXiv:1307.5693 [cs]*, Jul. 2013, arXiv: 1307.5693. [Online]. Available: http://arxiv.org/abs/1307.5693 [Accessed: 2016-09-16] 81

[113] J. Pan, K. McGuinness, E. Sayrol, N. O'Connor, and X. Giro-i Nieto, "Shallow and Deep Convolutional Networks for Saliency Prediction," *arXiv:1603.00845 [cs]*, Mar. 2016, arXiv: 1603.00845. [Online]. Available: http://arxiv.org/abs/1603.00845 [Accessed: 2016-09-16] 81

[114] D. Stefic and I. Patras, "Learning visual saliency using topographic independent component analysis." IEEE, Oct. 2014, pp. 1130–1134. [Online]. Available: http://ieeexplore.ieee.org/lpdocs/epic03/wrapper.htm?arnumber=7025225 [Accessed: 2016-09-15] 81, 84

[115] R. Margolin, A. Tal, and L. Zelnik-Manor, "What Makes a Patch Distinct?" IEEE, Jun. 2013, pp. 1139–1146. [Online]. Available: http://ieeexplore.ieee.org/lpdocs/epic03/wrapper.htm?arnumber=6618995 [Accessed: 2016-09-17] 81

[116] L. Itti and C. Koch, "Computational modelling of visual attention," *Nature Reviews. Neuroscience*, vol. 2, no. 3, pp. 194–203, Mar. 2001. 83

[117] Z. Bylinskii, T. Judd, A. Borji, L. Itti, F. Durand, A. Oliva, and A. Torralba, "Mit saliency benchmark." 83

[118] S. S. S. Kruthiventi, K. Ayush, and R. V. Babu, "DeepFix: A Fully Convolutional Neural Network for predicting Human Eye Fixations," *arXiv:1510.02927 [cs]*, Oct. 2015, arXiv: 1510.02927. [Online]. Available: http://arxiv.org/abs/1510.02927 [Accessed: 2016-09-17] 83

[119] M. Kümmerer, L. Theis, and M. Bethge, "Deep Gaze I: Boosting Saliency Prediction with Feature Maps Trained on ImageNet," *arXiv:1411.1045 [cs, q-bio, stat]*, Nov. 2014, arXiv: 1411.1045. [Online]. Available: http://arxiv.org/abs/1411.1045 [Accessed: 2016-09-17] 84

[120] Z. Bylinskii, T. Judd, A. Borji, L. Itti, F. Durand, A. Oliva, and A. Torralba, "Mit saliency benchmark." 84

[121] Z. Qi, Z. Songnian, W. Zhe, and H. Yaping, "A neural computational model for bottom-up attention with invariant and overcomplete representation," *BMC Neuroscience*, vol. 13, p. 145, 2012. [Online]. Available: http://dx.doi.org/10.1186/1471-2202-13-145 [Accessed: 2016-09-17] 84, 85, 86, 87

[122] L. L. Cloutman, "Interaction between dorsal and ventral processing streams: Where, when and how?" *Brain and Language*, vol. 127, no. 2, pp. 251–263, Nov. 2013. [Online]. Available: http://www.sciencedirect.com/science/article/pii/S0093934X12001459 [Accessed: 2016-08-18] 85

[123] H. C. Nothdurft, J. L. Gallant, and D. C. Van Essen, "Response modulation by texture surround in primate area V1: correlates of "popout" under anesthesia," *Visual Neuroscience*, vol. 16, no. 1, pp. 15–34, Feb. 1999. 87

[124] R. Desimone and J. Duncan, "Neural mechanisms of selective visual attention," *Annual Review of Neuroscience*, vol. 18, pp. 193–222, 1995. 87

[125] L. Itti and C. Koch, "<title>Comparison of feature combination strategies for saliency-based visual attention systems</title>," B. E. Rogowitz and T. N. Pappas, Eds., May 1999, pp. 473–482. [Online]. Available: http://proceedings.spiedigitallibrary.org/proceeding.aspx?articleid=979054 [Accessed: 2016-09-18] 87

[126] A. Torralba, "Modeling global scene factors in attention," *Journal of the Optical Society of America. A, Optics, Image Science, and Vision*, vol. 20, no. 7, pp. 1407–1418, Jul. 2003. 89

[127] L. Elazary and L. Itti, "A Bayesian model for efficient visual search and recognition," *Vision Research*, vol. 50, no. 14, pp. 1338–1352, Jun. 2010. 95

[128] J. Li, Y. Tian, T. Huang, and W. Gao, "Probabilistic Multi-Task Learning for Visual Saliency Estimation in Video," *International Journal of Computer Vision*, vol. 90, no. 2, pp. 150–165, May 2010. [Online]. Available: http://link.springer.com/article/10.1007/s11263-010-0354-6 [Accessed: 2016-09-18] 95

[129] V. A. Lamme, H. Supèr, and H. Spekreijse, "Feedforward, horizontal, and feedback processing in the visual cortex," *Current Opinion in Neurobiology*, vol. 8, no. 4, pp. 529–535, Aug. 1998. 96

[130] A. Oliva and A. Torralba, "The role of context in object recognition," *Trends in Cognitive Sciences*, vol. 11, no. 12, pp. 520–527, Dec. 2007. 96

[131] "An Introduction to Compressive Sensing - OpenStax CNX." [Online]. Available: http://cnx.org/contents/f70b6ba0-b9f0-460f-8828-e8fc6179e65f@5.12 [Accessed: 2016-09-28] 101

[132] D. L. Donoho, "Compressed sensing," *IEEE Trans. Inform. Theory*, vol. 52, pp. 1289–1306, 2006. 101, 104, 105

[133] D. Needell and J. A. Tropp, "CoSaMP: Iterative Signal Recovery from Incomplete and Inaccurate Samples," *Commun. ACM*, vol. 53, no. 12, pp. 93–100, Dec. 2010. [Online]. Available: http://doi.acm.org/10.1145/1859204.1859229 [Accessed: 2016-09-28] 105, 106

[134] E. Candes and J. Romberg, "l1-MAGIC: Recovery of Sparse Signals via Convex Programming," *ResearchGate*, pp. 1–19, Jan. 2005. [Online]. Available: https://www.researchgate.net/publication/269634028_l1-MAGIC_Recovery_of_Sparse_Signals_via_Convex_Programming [Accessed: 2016-09-28] 106

[135] R. G. Baraniuk, V. Cevher, M. F. Duarte, and C. Hegde, "Model-Based Compressive Sensing," *arXiv:0808.3572 [cs, math]*, Aug. 2008, arXiv: 0808.3572. [Online]. Available: http://arxiv.org/abs/0808.3572 [Accessed: 2016-09-28] 106

[136] S. Zhang, X. Zhao, and B. Lei, "Robust Facial Expression Recognition via Compressive Sensing," *Sensors (Basel, Switzerland)*, vol. 12, no. 3, pp. 3747–3761, Mar. 2012. [Online]. Available: http://www.ncbi.nlm.nih.gov/pmc/articles/PMC3376615/ [Accessed: 2016-09-28] 106

[137] V. J. Barranca, G. Kovacic, D. Zhou, and D. Cai, "Sparsity and Compressed Coding in Sensory Systems," *PLOS Comput Biol*, vol. 10, no. 8, p. e1003793, Aug. 2014. [Online]. Available: http://journals.plos.org/ploscompbiol/article?id=10.1371/journal.pcbi.1003793 [Accessed: 2016-09-28] 107

[138] T. Serre, "Learning a Dictionary of Shape-Components in Visual Cortex: Comparison with Neurons, Humans and Machines," Apr. 2006. [Online]. Available: http://dspace.mit.edu/handle/1721.1/32544 [Accessed: 2016-09-28] 107