RESEARCH PROJECT IN MECHANICAL <or MECHATRONICS> ENGINEERING

A Convolutional neural network for detecting visual texture

Callan Loomes

Project Report ME110-2022

Co-worker: Conrad Scherb

Supervisor: Dr Luke Hallum

Department of Mechanical and Mechatronics Engineering
The University of Auckland

14 October 2022

ME110-2022

A CONVOLUTIONAL NEURAL NETWORK FOR DETECTING VISUAL TEXTURE

Callan Loomes

Abstract

Visual texture recognition is an integral part of day to day life; humans are able to distinguish between boundaries primarily via first order luminescence differences, but also through second order texture differences. The mechanism behind the latter is not fully understood, paving the way for computational techniques, such as machine learning, to help bridge the gap in knowledge. In this study, convolutional neural networks are used to model texture recognition in the primary visual cortex. Simple cells in the primary visual cortex have receptor fields, which take visual information from specific regions in the retina's field of view, and perform a process similar to convolution to detect texture boundaries. The parallels between this process and the computational procedure of CNNs justifies their use as a valid model.

A range of CNN architectures were trained on human texture recognition data (in the form of an array of Gabor patches) in order to determine the best performing model. This would allow biological inferences to be made about the computational make up of the primary visual cortex. The parameters which were altered in the CNN models were kernel size, and neuron number, both of which relate directly to the RFs of V1 neurones. It was found that there is an optimum receptor field size based on the boundary size between units of texture. Furthermore, it was found that models with two to three convolutional layers possessed the highest accuracy, indicating that processing in the primary visual cortex may occur in layers.

Further exploration into high performing CNN architectures involved viewing the convolutional filters and outputs generated at each layer. This exploration highlighted mechanisms within the CNNs which were very similar to existing computational models of the primary visual cortex, including the filter rectify filter model. The knowledge collated in this study will help with understanding of the computational mechanisms within the primary visual cortex, which can be used in areas of research, computer vision and medicine.

DECLARATION

Student

I hereby declare that:

1. This report is the result of the final year project work carried out by my project partner (see cover page) and I under the guidance of our supervisor (see cover page) in the 2022 academic year at the Department of Mechanical and Mechatronics Engineering, Faculty of Engineering, University of Auckland.
2. This report is not the outcome of work done previously.
3. This report is not the outcome of work done in collaboration, except that with a potential project sponsor (if any) as stated in the text.
4. This report is not the same as any report, thesis, conference article or journal paper, or any other publication or unpublished work in any format.

In the case of a continuing project, please state clearly what has been developed during the project and what was available from previous year(s):

Signature: 

Date: 10th October 2022

Supervisor

I confirm that the project work undertaken by this student in the 2022 academic year ~~is~~ / is not (strikethrough as appropriate) part of a continuing project, components of which have been completed previously. Comments, if any:

Signature: 

Date: 10th October 2022

Table of Contents

List of Figures

List of Tables

Glossary of Terms

Visual Texture	A term referring to the presence of various boundaries/contrasts in images which form the texture of the object being looked at.
Receptive Fields	The regions of vision over which V1 neurons receive sensory input from.
TensorFlow	A python package containing machine learning elements
Keras	An API built on top of TensorFlow
MATLAB	A programming language
Thalamus	A region in the brain responsible for relaying sensory information
Luminescence	The light characteristics of an object; can refer to brightness or colour
Gabor patch	A orientated and grated pattern common in vision experiments

Abbreviations

CNN	Convolutional Neural Network
fMRI	Functional magnetic resonance imaging
RF	Receptor Fields
FRF	Filter Rectify Filter
LGN	Lateral Geniculate Nucleus
LNL	Linear, Non-Linear, Linear
MEG	MagnetoEncephaloGraphy
DNN	Deep Neural Network
CR	Correct Rejection
FA	False Affirmation
SGD	Stochastic gradient descent

Acknowledgements

I would like to thank the following people for their help and support during this project:

Dr Luke Hallum for being a reliable and supportive supervisor, and always providing high quality feedback and guidance on aspects of the project.

Conrad Scherb for being my project partner and putting in a lot of effort to the project, particularly in the final weeks, even on top of other commitments.

Kevin Lee for providing a plethora of knowledge early on in the project with regards to neural networks.

1. Introduction

Visual texture discrimination in humans is a process that is seamlessly integrated into modern life; one scarcely pauses to consider the reason they are able to distinguish the boundary between their wooden cutting board and marble kitchen bench. This object recognition is based on luminescence and texture differences, the latter of which is the focus of this project [2].

1.1 Project Scope

This project aims to provide further analysis and understanding behind the mechanisms occurring in the primary visual cortex (V1) when texture boundaries are identified. This will be done through the creation of a CNN that is able to identify different components of texture in an image, distinguishing whether or not there is a boundary between two textures present. The model will be trained on behavioural data generated through human testing, which includes distinguishing whether there is a pattern of orientation in a series of Gabor patches. The scope of the project also encompasses trail and error with different CNN architectures, with the objective of drawing biological parallels between V1 neuron receptor fields and kernel filters. The understanding of texture detection is useful in scientific research regarding the V1, and has practical applications in the fields of computer vision and medicine [3].

1.2 Research Objectives

This project aims to encapsulate the following objectives:

- Generate a comprehensive range of testing data based of human responses to visual texture.
- Experiment with different CNN structures and hyper-parameters for models trained on user response data..
- Compare the trained CNNs to existing computational models of the V1, and make biological inferences based on CNN performance with different structures.

2. Literature Review

2.1 Biological Mechanisms Behind Texture Identification

Visual identification in the human brain is a complex procedure reliant on a hierarchical processing structure of neuron layers in the visual cortex. The use of fMRI has shown that a large amount of neurological processing, with regards to decoding colour and form, is done through the early areas of the visual cortex, V1 to V4 [4]. This early processing includes making simple distinctions, such as orientation or visual texture of perceived objects, which are then conjugated and processed at higher visual layers.

2.1.1 Pathway into the Primary Visual Cortex

Human identification of an object from its surroundings occurs primarily through first-order luminescence distinctions, such as colour and light level; also important however are the second order texture contrasts which include orientation and spatial frequency of a surfaces particulates [2]. The visual pathway from the retina (where light is transduced into electrical signals) to the visual cortex, where visual processing occurs,

is well known. Visual information travels from the retina to the LGN of the thalamus, where it undergoes a base level of organisation and processing before being sent to the V1 [5]. This process is summarised in Figure 1.

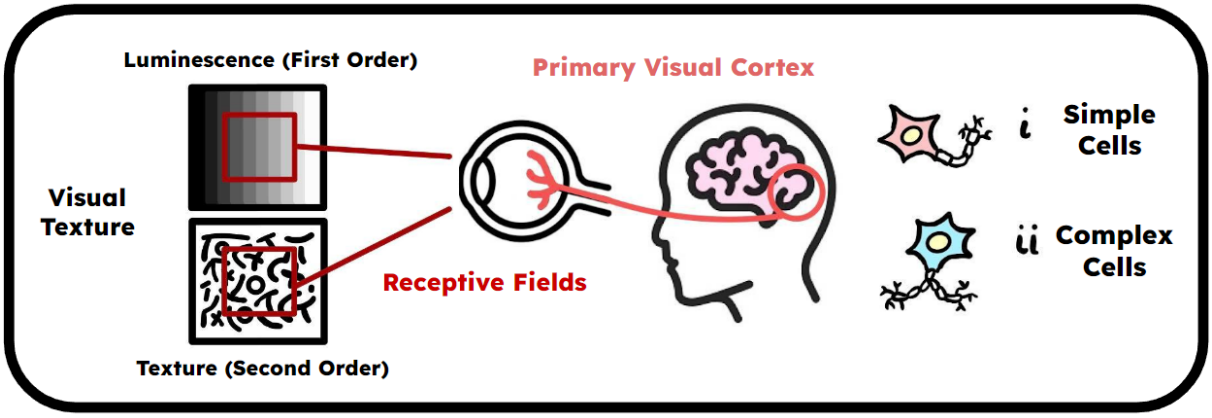


Figure 1 Pathway for visual texture processing in the human brain

It has long been known that the V1 is involved with low level texture recognition. In a 1998 study, Tootell et al were able to distinguish the V1 from other cortical areas through its response to contrasting luminance [6]. The same study utilised fMRI and grating stimuli (alternating light and dark patches) to showcase the sensitivity to orientation in human V1. These mechanisms occur due to the relationship between simple and complex cells in the V1 [7].

2.1.2 Simple and Complex Cells

There are two major neuron types in the V1 which are involved with low level texture segregation: simple cells and complex cells [7]. Both cell types operate over small regions of vision called receptive fields (RFs), and respond to contrasting regions of luminescence. The receptor fields of simple cells are spatially specific, and respond strongly to contrasting regions and orientated gratings. The spatial plane refers to the position of the stimuli in the RF of the cell. The RFs of simple cells can be modelled computationally with a linear-nonlinear model [8]. In this model, the input image is filtered (a process which can be modelled using convolution), and then passed through a static non-linear function, such as half wave rectification. Figure 2 summarises the process in which simple cells RFs extract orientation and spatial frequency information, the constituents of visual texture, from input they receive from the LGN.

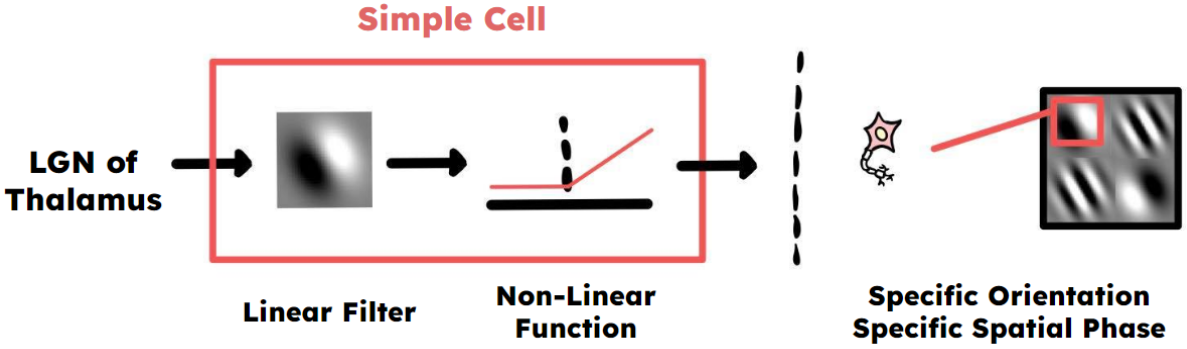


Figure 2 Computational model for simple V1 neurons

Complex cells also respond strongly to orientation, as well as motion, but are spatially

invariant and less sensitive to stimuli than simple cells [9]. There is some contention in the existing literature as to how complex cells achieve this spatial invariance [7]. The hierarchical model suggests complex cells pool information from multiple simple cells with the same orientation preferences but different spatial preferences. This results in a complex cell which is orientation specific, but spatially invariant. This model has a been around for a while, and has a significant amount of experimental evidence [10]. The discovery of direct LGN connections to a small subset of complex cells, however, introduced the parallel and recurrent models, where simple and complex cells process information from the LGN in tandem [11]. Because evidence suggests that only a small number of complex cells operate this way, this project will focus on modelling the feed-forward hierarchical model of the V1, summarised in Figure 3.

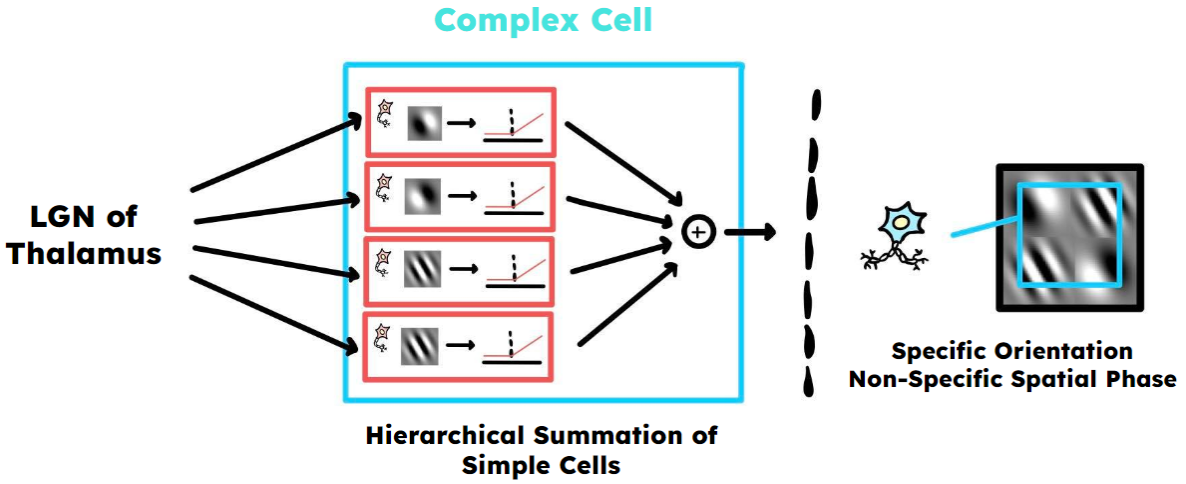


Figure 3 Hierarchical computational model for complex V1 neurons

2.2 Existing Cognitive Models of the Primary Visual Cortex

Computational models for attempting to detect visual texture are not a new concept. In 1991, Bergen and Landy explore a model consisting of subsequent filtering and pooling layers [12]. Many of these models, however, lack a fundamental biological basis for comparison, meaning they are most likely unrepresentative of the actual process which occurs in the visual cortex [13]. Two types of model which do have significant biological merit are Fourier transfer and filter rectify filter (FRF) approaches.

As early as 1978, V1 neurons were described by Mayhew and Frisby as cells which are responsive to different orientations and spatial frequencies [14]. In this study, it was initially theorised that the visual cortex performed a sort of Fourier based transform to discriminate textures. Following studies also suggest the visual cortex performs a Fourier approximation, with particular sensitivities for spatial frequencies [15]. Given the age of these studies and what is now know about the V1, a purely Fourier based model is unlikely.

A more recent theory for texture segregation in the V1 is the FRF model, described by Landy in 2013 [16]. This model, summarised in Figure 4, first applies a linear filter tuned to a particular orientation and spatial frequency. The resulting output is strongly positive when textures fully match the preferred parameters and strongly negative where textures fully match the opposite. As recognised by Hallum et al and Lian et al, this initial filtering stage draws very similar parallels to how simple cells in the V1 have receptor fields which respond to orientation and spacial frequency [2, 7]. In the subsequent rectify stage, a non linear threshold or point-wise function separates

the positive values (matching alignment) from the negative values (non-matching alignment), essentially highlighting regions of high variance. Following this, the second linear filter is applied which identifies the overarching texture edge in the image. Because of the sequence of filters, the FRF models is also referred to as the linear, non-linear, linear (LNL) model. Research suggests the second filtering stage has physiological ties to the V2, V3 and V4 areas of the visual cortex, with evidence that it may begin as early as the V1 layer [2, 17].

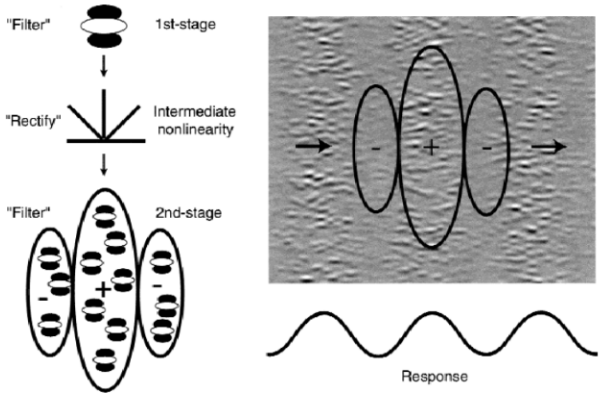


Figure 4 Computational representation of the FRF model of the Primary Visual Cortex [1]

Recent literature builds off the FRF model; the addition of a normalisation layer to form FRNF or FRNFRN models is a common addition which has filled the biological gaps presented by a base FRF model [2, 16]. While there is significant computational evidence that simple and complex cells perform the initial filter, the biological link to the subsequent rectify, normalisation and second filter stages are not confirmed [8]. If following with the hierarchical structure, it is possible that this second filter is a result of complex cells, while if following the parallel model, this second filter could occur outside of the V1; a study of texture perception with CNNs may provide insight to this challenge [7].

2.3 Existing Literature Surrounding CNNs

A convolution network is one where at least one or more of its layers involve a convolution step [18]. The benefit of these convolution steps is their ability for highly specific edge detection by breaking down images into texture components. Traditionally, neural networks were designed on the principles of biology, and hence are very suitable for the task of modelling the visual cortex [19].

The three main aspects to consider when designing neural networks are: the overall architecture, the training procedure and the task to be learned. Many studies which give methods of designing and hyper tuning CNNs to improve performance. In general, CNNs consist of a collection of layers which feed forward into each other [20]. Each layer has a different function, which can range from convolution, noise reduction, pooling, correction layers and more [21]. Schirrmeister et al explore a large range of architecture design considerations in their 2017 study regarding the decoding of EEGs [18]. Deep ConvNets were designed which extracted a large number of generic features, rather than specific ones. These deep CNNs were characterised by their large number of pooling blocks and classification layers. To contrast, shallow ConvNets were more focused on specific feature extraction, and their architecture consisted more of processing convolution and filter layers. From this study, the link between architecture and function is highlighted.

Convolutional layers provide the core computational breakdown of images; they use a kernel which passes over the input image and performs computational convolution at each position. The kernel size is a parameter which can be adjusted, with larger kernels increasing computational time for a decrease in spatial specificity [22]. An important comparison for the purposes of this paper involves the similarities between the kernel filters of a CNN and the receptor fields of a simple cells in the V1; both define a small region over which a calculation is performed [23]. This aspect of CNNs allows the shape of simple and complex cell RFs to be inferred from kernels of trained CNN models.

Another important aspect of CNNs are the activation layers, which typically follow convolutional layers to transform their outputs into meaningful patterns of edge detection. There are many different types of activation layer, such as rectified linear units (ReLU), sigmoid functions, and exponential linear units (ELU), many of which are biologically inspired" [24]. Notably, these layers draw parallels to the rectify stage in the FRF model described in section 2.2, and may be a worthwhile area of investigation to understand the visual texture processing pathway.

In addition to network architecture, the training data presented to a CNN is one of the key factors that determines its success. Training data need to be sufficiently consistent, accurate and in a form where it can be easily understood by machine learning algorithms. Additionally, there needs to be a sufficient amount of training data available to ensure memorization does not occur [25]. This over fitting, can suppress important pattern detection in CNNs resulting in a less generalized response.

2.4 Similar Visual Studies

In recent literature, the use of machine learning models has been an effective technique for mapping and understanding regions of the brain. In 2016, Cichy et al created a deep neural network consisting of eight layers, each of which performed a variety of tasks such as convolution, pooling or normalisation [21]. While previous studies relied on extrapolating data from well understood areas to generate behaviour, this neural network built these pathways from scratch against large sets of training data. When compared with real data generated in space and time via fMRI and MEG respectively, both the trained model and real data showed a hierarchy where lower level processing was conjugated into higher level processing.

Another study by Kociolek et. al. explores the secondary visual cortex (V2) with a CNN approach, very similar to the methodology used in this project [26]. In this paper, architectures of different depths (number of convolutional layers) were tested on a range of human trained texture-grating data to investigate the processing of directionality in the visual cortex. Furthermore, Yamins and DiCarlo explore the use of DNNs to model hierarchy of the layers in the sensory cortices [27]. Both studies reach the conclusion that it is more practical to model small processing regions of the sensory cortices with shallow networks, rather than trying to model entire sensory systems with deeper models. The currency and success of these studies indicate that using CNNs to analyse regions of the brain has viability and potential as a technique.

For all visual studies, a reliable means to control texture and luminescence is required. Gabor patches can be easily generated in a variety of different orientations and sizes through computational convolution, and can be easily detected and analysed by computational models. Gabor patches have been used as an experimental medium in many studies related to visual stimuli [28]Furthermore, studies have shown that Gabor patches

can be used to stimulate the V1 to configurations commonly seen in nature [29]. Another topic of relevance to the input data of this study refers to the human ability to detect and learn patterns from from noise and vice versa.. In very early studies, Burgess et al. suggests that humans have a high efficiency for the separation of signal from noise, at around 70% efficiency of detection [30]. Recent studies show that the quality of signal detection increases after an increased amount of training [31]. This may be an important factor when designing trials for the generation of training data.

3. Methods

3.1 Data Generation and Collection

Early testing with pre-existing response data quite quickly highlighted issues with a lack of training data for the CNN models; this resulted in highly over fitted models, and a lack of validation data. Hence, a pipeline for generating test data was created.

3.1.1 Generation of Gabor Patch Datasets

Test data consists of an array of Gabor patches placed on a grey background, each at a different orientation. The generated datasets are generated such that half the images contain a pure noise (a completely random orientation of patches), while the other half contains a signal in addition to this noise. The signal is such that patches on the right hand side of the image are shifted 90 degrees out of phase from those on the left side, resulting in a vertical texture down the middle of the image. Within the same side of the signal, some patches are randomly flipped 180 degrees out of phase. Notably, the average luminescence either side of the texture boundary is the same, meaning any distinctions are based solely on second order texture differences.

The sequence of patches were generated in MATLAB; for the individual Gabor patches, a Gaussian blur was applied to a ramp function at a given orientation. For the datasets with no signal, the list or orientations was simply created using a sequence of random complex numbers. For the datasets with a signal, a signal was generated using an offset angle, to which a signal with random noise was added. A randomly generated noise-gain factor was applied to the signal-noise, such that some signal data sets had very little noise, while others had more. An example of these data patches are shown in Figure 5.

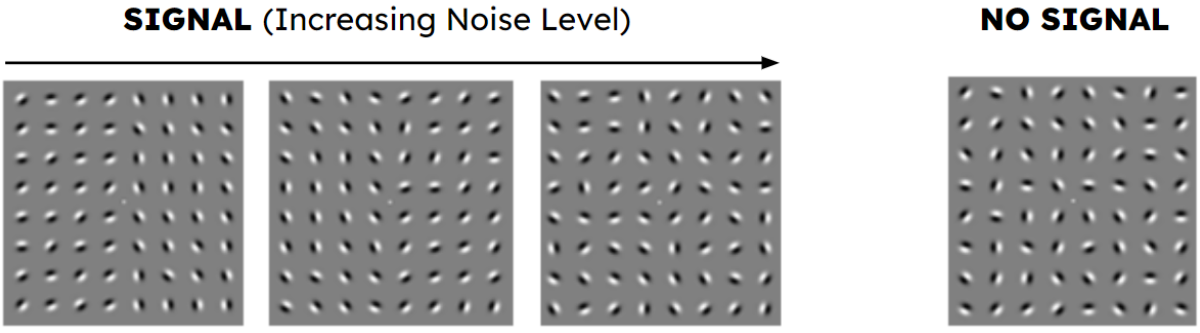


Figure 5 Example Gabor patch arrays showing increasing noise in signal and no-signal images

The noise-gain factor was manually set such that in signal images with the highest amount of noise, the signal was only just detectable by human eyes from the background noise. A total of three datasets were generated, varying in the number of Gabor patches present in the texture. Each dataset contained 2500 images in total, 1250 of

which contained a signal. The purpose of this was to experiment with different texture complexities to see if they affect architecture trends. A summary of these different datasets is shown in Figure B1 in Appendix B.

3.1.2 User Labelling of Datasets

The goal of this study is to train a model that can accurately mimic a human's V1 response to visual texture. Hence, the datasets generated in section 3.1.1 need to be manually labelled with human response data.

A MATLAB script was used to briefly flash a Gabor-patch image for 0.5 seconds to a observing user. The user would then input whether they thought the image contained a signal or not. The image was sorted as either a hit, miss, correct rejection (CR) or false affirmation (FA) based on the user response and ground truth. Flashing the image for only a short period of time ensured the human response was mediated only through the low-level V1 processing layer in the brain. Ideally, this limits any higher level decision making on the side of the user, isolating the V1. Each dataset was labelled separately by two young adult male subjects, with a sufficient amount of time between tests to avoid learning and saturation [31].

3.2 Creating and Training CNNs

Following the generation of user data, the goal was to create a base CNN structure from which an investigation into hyper-parameters could be performed. For the sake of consistency, all models were trained using the same NVIDIA GeForce GTX 1050 Ti GPU. A full list of training data is available in Appendix A.

3.2.1 Machine Learning Framework and Training Data

In order to build and test the CNNs for this project, the python package TensorFlow was used, with particular emphasis on the Keras API. In general, TensorFlow and Keras were chosen as they are more flexible and versatile than other packages such as Pytorch [32]. Furthermore, existing literature from which we based our initial structure primarily focused on these packages.

To transfer the images from MATLAB into TensorFlow , the Keras image-dataset-from-directory() package was used. This allowed greyscale images to be loaded from an organised file structure. To speed up training, preprocessing on the images reduced them to one sixth of their original size. This reduction in size reduced training time by around ˜75%, with little to no loss in accuracy of the models.

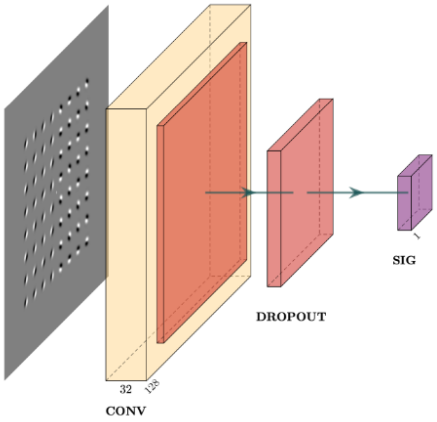
To ensure a valid comparison between the models, ten-fold cross validation was used while training the data. This process introduces reliability into the model results by splitting the training dataset into ten different segments. The model is then trained a total of ten times, with one of the segments being held out and used for validation each fold [33]. This provides a mean and standard deviation of accuracy on the validation dataset, which can be used to generate confidence intervals for each model. Each model was trained for a minimum of 350 epochs, and until the loss function on the training data set hit a plateau over the last 20 epochs (indicating the model was no longer learning new trends).

3.2.2 Model Optimisations and base structure

Early exploitative training with models found from current literature introduced some issues which needed to be addressed [26]. The initial model that was derived, shown in Figure B2 in Appendix B, consisted of two convolutional layers and a dense layer which fed into a binary sigmoid function for classification. The major issue encountered was to do with over fitting of the models to data, which was most likely the result of a relatively small dataset relative to the complexity of the model. Over fitting can be identified when the accuracy on the training set is significantly higher than the validation set, indicating the model was was simply "remembering" the testing images rather than learning general trends [25]. To mitigate this issue, two solutions were implemented. The first of these implementations was to include a dropout layer in the architecture of the model; this layer randomly sets a proportion of the connection weights to zero during each iteration, preventing memorisation [34]. Furthermore, data augmentation in the form of random vertical and horizontal mirrors were applied to the input data; this transformation did not affect the presence or orientation of a signal. Shown in FigureB2 in Appendix B, this effectively meant the training size was increased by a factor of four, reducing over fitting even further.

Following the implementation of anti-overfitting measures, it was found that the initial model was too complex and unrelated to existing biological models. Hence, the model was simplified down by removing the dense layer. This dense layer would also cause overhitting issues, which decrease the total possible accuracy on the validation dataset. Further to this, different activation layers were experimented with to determine which would achieve the highest accuracy. A full description of each of the activation layers tested can be found in Figure B3 in Appendix B, but the most notable nominees were ReLU layer variants, exponential and tanh functions.Medium Dataset Training

Lastly, binary cross entropy was used as the loss function; this is common practice for neural networks which output a binary state variable . Initially ADAM was used as an optimiser, however it was found that the SGD optimiser was more consistent with its results. All this initial model exploration lead to a simplified 'base model', shown in Figure 6, which acted as a starting point to experiment with tuning hyper parameters with biological relevance.



#	Layer Type	Base Network
1	Input (min size x, min size y, channels)	128,128, 1
2	Convolution (size x, size y, count)	13, 13, 32
3	Activation Layer	"ReLU"
4	Max pool	2,2
5	Flatten	
6	Dropout	0.5
7	Output (count)	1

Figure 6 Overall structure of the chosen base model

3.3 Modifying CNN Structure and Hyper-parameters

Starting from the base structure generated in Section 3.2.2, incremental modifications were made to the number of neurones in each layer, the kernel size of the convolutional

layers and the number of convolutional layers in the model. All models were trained on all three datasets, providing trends for each. Furthermore, on the medium sized dataset, additional testing was performed to combine trends across multiple parameters, generating a 2D hyperspace of parameters (eg. All neuron numbers were testing on all kernel sizes). The specific structure for each model tested can be found in C.

Modifying Neuron Count

The number of neurons in a convolutional layer refer to the amount of unique kernel filters which are applied across the image. This may relate to how many different types of filters are present in the receptor fields of V1 simple cells, and also whether larger and more complex textures require more neurons for processing. Neuron number was varied in powers of two, as is common in literature, from 128 neurons all the way down to a single neuron [20]. As shown in table C1, this was variation performed on a base model with a single convolutional layer with an 11x11 kernel size.

Modifying Kernel Size

When undergoing convolution, the computational filter, which is used to calculate the convolutional sum at each point in the input image, is called a kernel. The kernel size refers to the pixel area represented by a kernel. Kernel size was varied from 3x3 pixels to 19x19 pixels, in increments of 2 pixels as kernel size was kept at an odd number to avoid aliasing [20]. As shown in table C2, this was variation performed on a base model with a single convolutional layer with an 32 neurons. Furthermore, to contextualise the size of the kernels, Figure 7 shows the size of each kernels overlaid on the Gabor patch datasets; kernel size by itself is not a useful indicator, but its ratio to the size of the input image has biological significance.

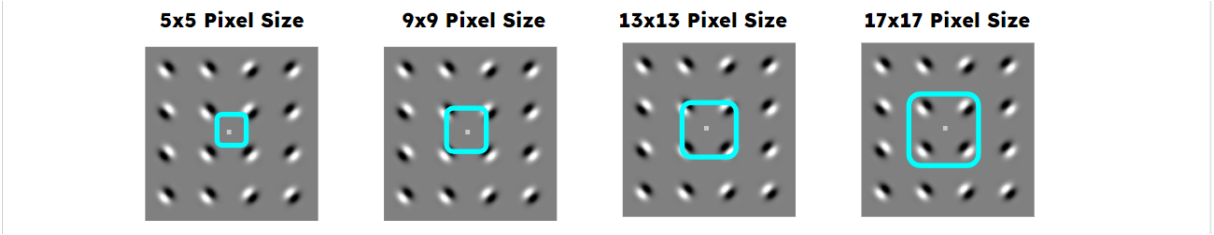


Figure 7 Size of kernels relative to Gabor patches

Further testing with kernel size was also performed on CNNs with two layers, and on a single layered convolutional model with only one neuron. This was done to follow interesting trends related to the FRF model described in section 2.2. A summary of these models is shown in table C3.

Modifying Model Depth

The final modified model parameter was the depth of the CNN. This refers to the number of convolutional layers in the model; shallow models have less convolutional layers, while deeper models have more. Model depth may provide insight as to how many filtering layers are present in the V1, and how the second filtering stage of the FRF model is achieved. Models with one to five convolutional layers were tested, and the specifics of each model is shown in table C4.

3.4 Analysis and Validation of Trained Models

Typically, the inside of CNNs are considered to be black boxes; it is very difficulty to discern what calculations are going inside the model to generate the observed outputs

[20]. Because of this reason, the primary method of inferring biological data from CNNs is to use backwards estimation to guess biological structure from CNN structure [26]. In addition to this method of inference, kernels from the trained models were visualised computationally, as well as the convolutional summation from each layer; viewing the kernel structure of initial convolutional layers could provide evidence as to what the RF of V1 simple cells look like.

Furthermore, the trained models were tested against a variety of unseen datasets, including images with boundaries that were horizontal, diagonal and offset from the centre of the page. This could provide insight as to whether the models are actually detecting texture boundaries, or whether they are just memorising the position of the vertical boundary. A summary of the alternate testing datasets is shown in Figure B4 in Appendix B.

A selection of high-performing trained models were selected for in depth analysis; these models had close ties to existing biological literature, and may provide catharsis to the actual texture segregation process in the brain.

4. Results and Exploration of Biological CNNs

4.1 User Performance on Datasets

The performance of the human response to the ground truth data is summarised in Table 1. The accuracy for each dataset are calculated with in equation (1). Subjects had an accuracy of 85.5%, 85.3% and 88.2% on the small, medium and large datasets respectively.

		Ground Truth (Small)		Ground Truth (Medium)		Ground Truth (Large)	
		Signal	No Signal	Signal	No Signal	Signal	No Signal
Prediction	Signal	1067 (41.1%)	208 (8.0%)	1026 (41.4%)	193 (7.8%)	1088 (43.6%)	130 (5.2%)
	No Signal	170 (6.5%)	1154 (44.4%)	171 (6.9%)	1091 (44.0%)	166 (6.6%)	1114 (44.6%)

Table 1 User labelling performance on small, medium and large datasets

$$User\ Accuracy = \frac{Hit\ Rate\ +\ Correct\ Rejection\ Rate}{Total\ Response\ Rate} \tag{1}$$

4.2 Initial Exploratory Models

Initial testing with exploratory models provided justification for the generated base model. Figure 8 shows the loss and accuracy over 200 epochs on a single training fold for three models; the before and after models from addition of dropout and data augmentation, shown in Figure B2, and the final base model shown in Figure 6.

Notably, the introduction of dropout layers and data augmentation provided significant mitigation to over fitting. In the original model, the training accuracy is consistently higher than the validation accuracy, while in the final base model, there is next to zero over fitting. This indicated that it was correctly identifying trends in the data rather than performing memorisation. The results from testing with different activation layers are also shown in Figure 8. From this testing, it can be seen that layers similar to

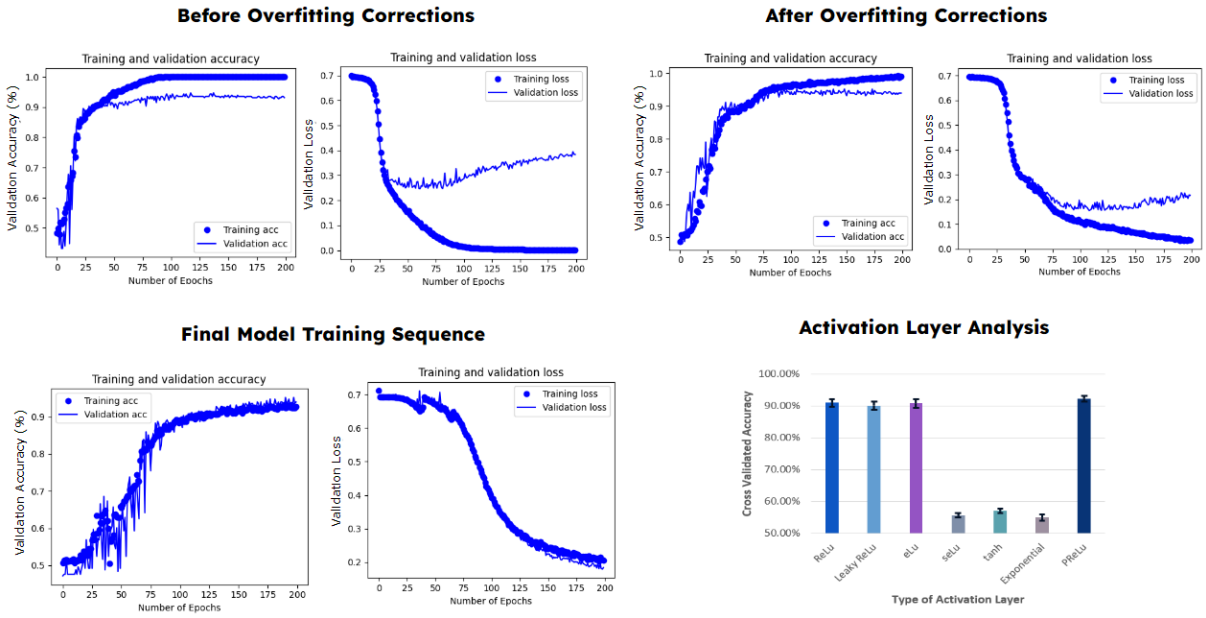


Figure 8 Accuracy and loss metrics for models before and after over fitting mitigation, and cross validated accuracies of the base model when tested with different activation layers.

half-wave rectification (eg. ReLu, eLu and PReLU) are able to achieve high accuracies, while those that don't (tanh, seLu and exponential) provide much lower accuracies.

4.3 Modifying CNN structure

Modifying Neuron Count

The results of modifying the neuron count on a single layered base model with an 11x11 kernel size is shown in Figure 9. It can be seen that with increasing neuron number, the accuracy of the model increases. At higher neuron numbers, there is a diminishing return in cross validated accuracy increase. The accuracy is highest on the large dataset, and smallest on the small dataset, however the overall trend is consistent over all three datasets; all models reach a similar validation accuracy at the highest neuron number.

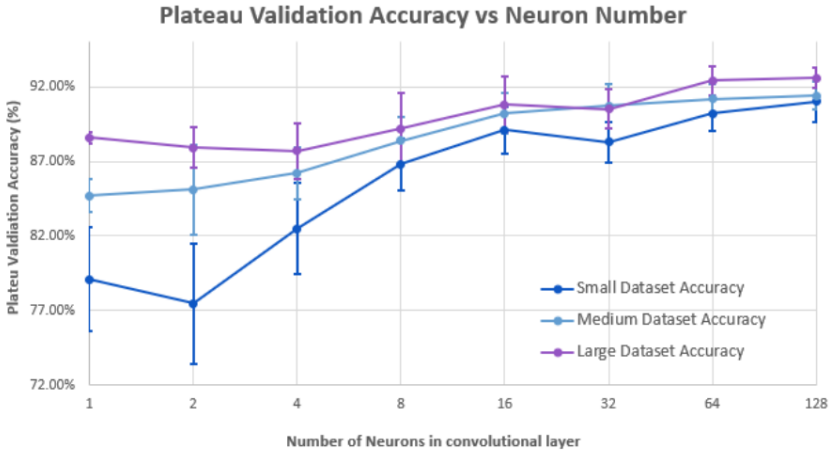


Figure 9 Cross validated accuracy vs neuron number for small, medium and large datasets (95% CI)

Modifying Kernel Size

The results of modifying the kernel size on a single layered base model with an 32 neurons is shown in Figure 10. The results show for kernel sizes below 7x7 pixels, models are not able to train properly, resulting in very low accuracies. As kernel size increases however, accuracy increases to a maximum at 13x13 pixels, before sloping off at higher kernel sizes. This trend is consistent for all datasets.

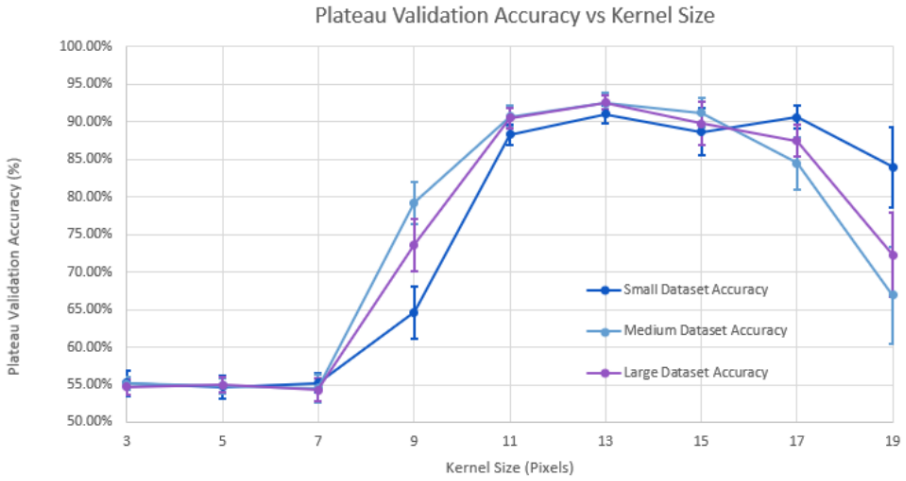


Figure 10 Cross validated accuracy vs kernel size for small, medium and large datasets (95% CI)

Similar trends with regard to kernel size are obtained when experimenting with double layered models and single neuron models, as shown in Figure 11.

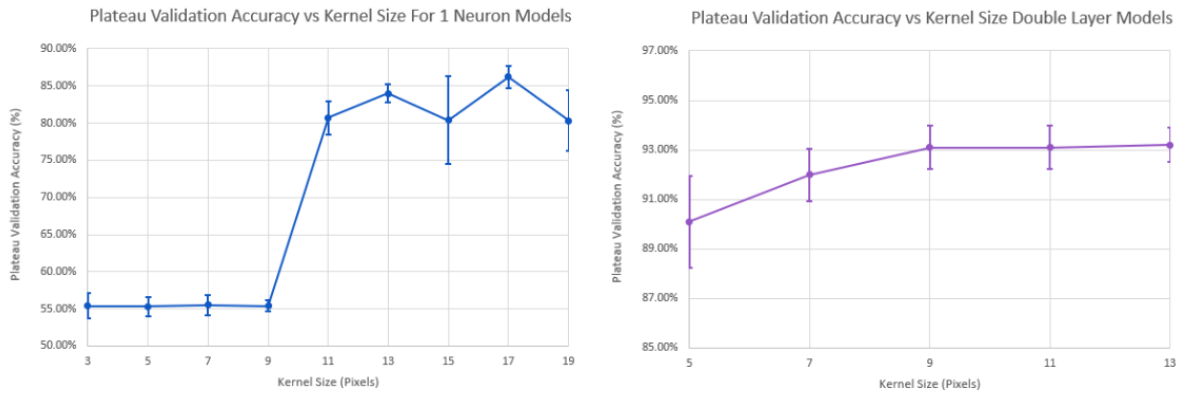


Figure 11 Cross validated accuracy vs kernel size for single neuron and dual layer CNNs (95% CI)

Modifying Model Depth

The results of modifying the depth of the base model are shown in Figure 12. For all datasets, model accuracy sees an increase from one to two convolutional layers before reaching a peak at 3-4 convolutional layers and then dropping off. Once again, the overall accuracy on the large dataset is the highest, followed by the medium dataset.

2D Hyperplane Analysis

Figure 13 describes the cross validated accuracy of models trained on a 2D hyper plane of parameters for a single layer model. This hyperplane highlights that the kernel size trend is consistent across many neuron numbers, with a size of 13x13 pixels being the most efficient across most neuron numbers. Of interest is the combination of high

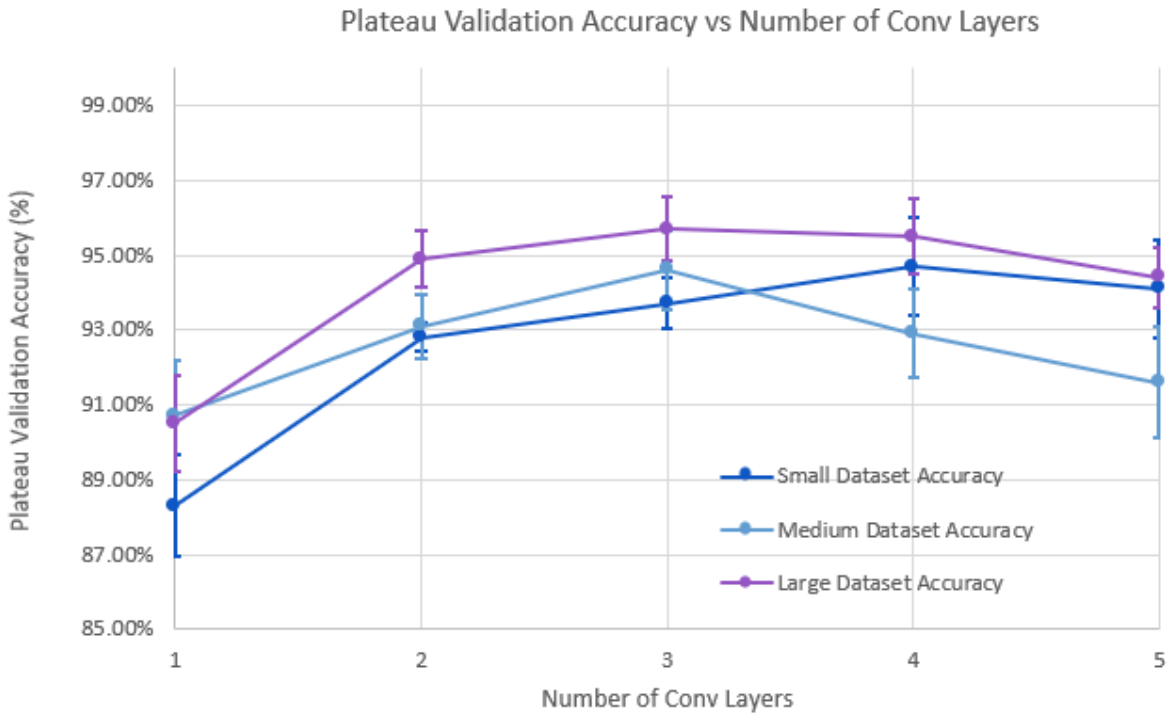


Figure 12 Cross validated accuracy vs CNN depth for small, medium and large datasets (95% CI)

neuron numbers with high kernel size, which still produces a high accuracy model. This may suggest that higher neuron numbers are able to incorporate higher kernel sizes.

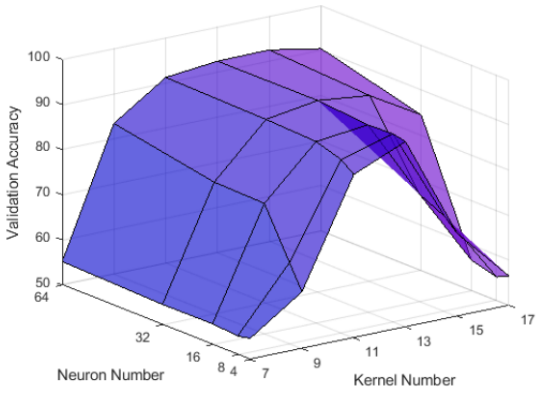


Figure 13 Surface graph showing cross validated accuracy vs neuron number and kernel size for medium dataset

4.4 Dissection of Biologically Relevant models

A selection of models were chosen for in depth analysis. These models were chosen based on the hyper parameter analysis, and their potential connection to existing biological models. Full definitions of these models are shown in Appendix C.

Low Neuron Number Single Layered Models

Single layered models consisting of a convolutional layer, half-wave rectification and classifier draw very similar parallels to a summation of simple cells in the V1. Somewhat surprisingly, these single layered models are able to perform at high accuracies of above 90%, even with a very low number of neurons. To investigate the performance of these models, the kernel structure and output of the first convolutional layer were viewed for

a 1 neuron model and a 4 neuron model in Figure (14). Both models used a kernel size of 13x13 pixels, and were convoluted against a signal dataset, a no signal dataset and an array of Gabor patches orientated at every angle.

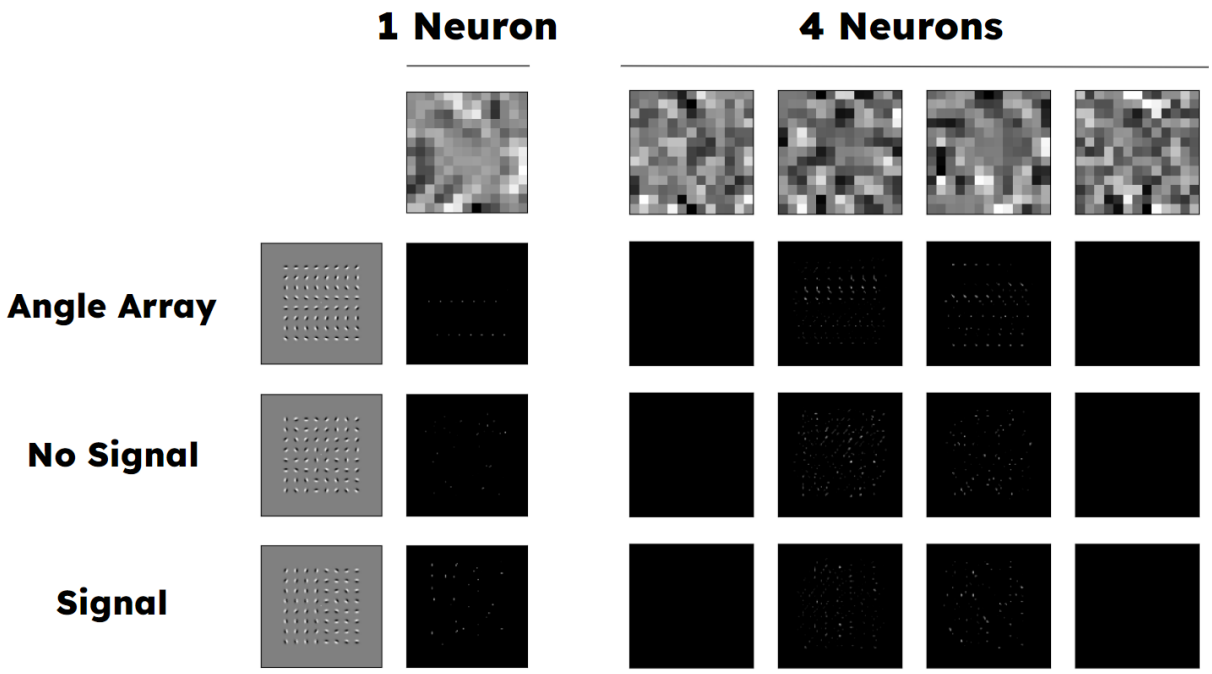


Figure 14 Kernel visualisation for 1 neuron and 4 neuron, single layered CNNs

The above figure shows an interesting property of single neuron models; the filter has a grey centre and textured edges, indicating that computation of boundaries occurs on the outskirts of the filter. Referring to Figure (7), it can be seen that the 13x13 kernel is able to capture multiple Gabor patches on the outskirts of its kernel. Henceforth, a valid theory is that the kernel spans multiple Gabor patches, and convolves the relationship between them in order to interpret texture. Further evidence to this theory is the observation that the convolution of a signal image lights up in more areas than the no signal,, and there is no clear orientation selectivity; therefore the kernel must be exploring the relationships between multiple Gabors aligned in a certain orientation with respect to one another.

In the visualisation of the 4 neuron model, it can be seen that two of the filters (leftmost and rightmost) consist of random noise, and do not produce any discernable convolutional output on any of the test images. This is consistent through all models with multiple neurons, as there are occasions where kernels are stuck in a local minimum, and are unable to train. Unlike the kernel from the single neuron model however, the two remaining kernels produce a similar output magnitude on each of the signal and non signal images, and light up different regions of the angular image. This indicates that these filters are trained to be orientation specific, and the model likely bases its texture discrimination on the presence or absence of a large quantity of similar orientated Gabor patches.

High Neuron Number Single Layered Models

Analysis of single layered models with more neurones was also performed to determine why there was an increase in accuracy as more unique filters were added. Single layered models of neuron number 8 and 32, with kernel sizes of 13x13 were examined. An annotated kernel analysis for the 8 neuron model is shown in Figure 15. A full

analysis of the 32 neuron model can be found in Figure B5 in Appendix B.

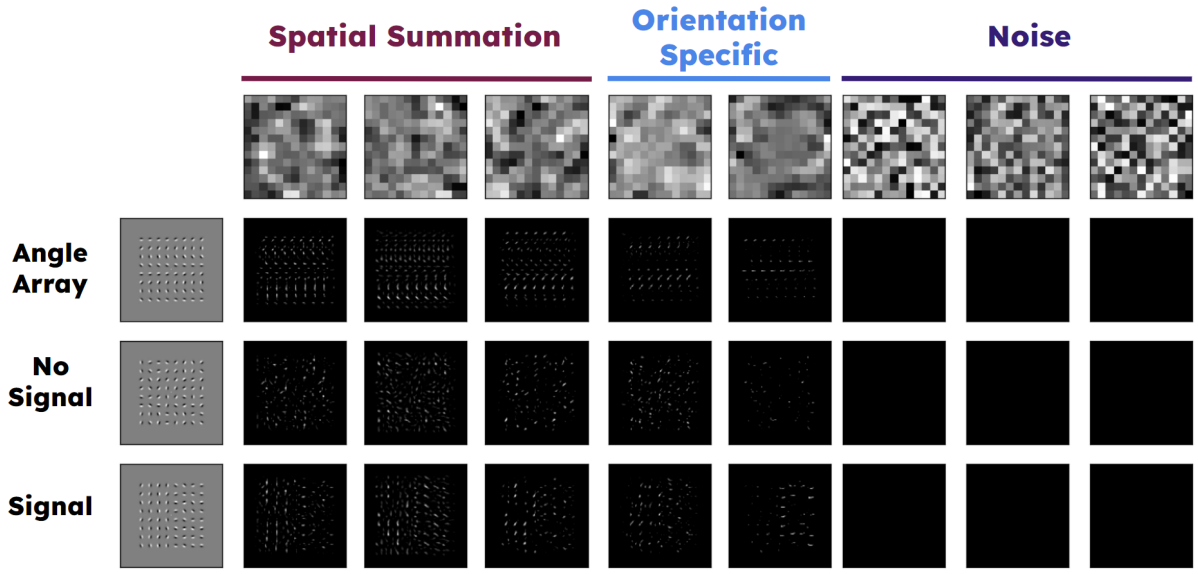


Figure 15 Kernel visualisation for 8 neuron, single layered CNNs

In the 8 neuron and 32 neuron models, kernels tend to organise themselves into the two categories described in the low neuron number models. Orientation specific kernels tend to distinguish texture through determining whether there is an overwhelming presence of a particularly orientated Gabor patch; these kernels highlight specific areas of the angle array, indicating they are specific for that given angle. On the other hand, spatial summation kernels tend to distinguish texture via the relationships of multiple Gabor in close proximity which are similarly originated. The distinguishing feature of these kernels are the higher activation on signal images, and a blanket activation on the angle array image. These two mechanisms of action are very similar to the parallel model of simple and complex cells in the V1.

Double Layered Models

Boasting much higher accuracies, models with two convolutional layers tend to be more similar to the hierarchical structure of simple and complex V1 neurons, and may reasonably model a FRF style of computation. The model that was chosen for further analysis consisted of a first convolutional layer with 32 neurons and a 13x13 kernel, and a second convolutional layer with 16 neurons and a 3x3 filter. This model achieved an accuracy of 93.2±1.1%, and a summarised kernel investigation is shown in Figure 16.

Unlike the previous single layered models, there was no presence of orientation specific kernels in the first convolutional output layer of the dual-layer model. Instead, extensive spatial summation of Gabor patches was observed, which had the effect of amplifying and highlighting the texture boundaries present in the images. Presumably, this allows the second convolutional layer (smaller kernel size) to detect the exact location texture boundaries, and hence compute the likelihood that a signal is present. This computational method, while having little relevance to a model of simple cells in the V1, is reminiscent of a reversed FRF model. A FRF model involves initially extracting spatial frequency and orientation, followed by detection/summation with a large RF to discern texture; the described model does this in the reverse order by spatial summation with a large RF, followed by orientation/spatial frequency segregation with smaller kernels.

This discovery prompted the generation of a CNN that was a reverse of the original dual

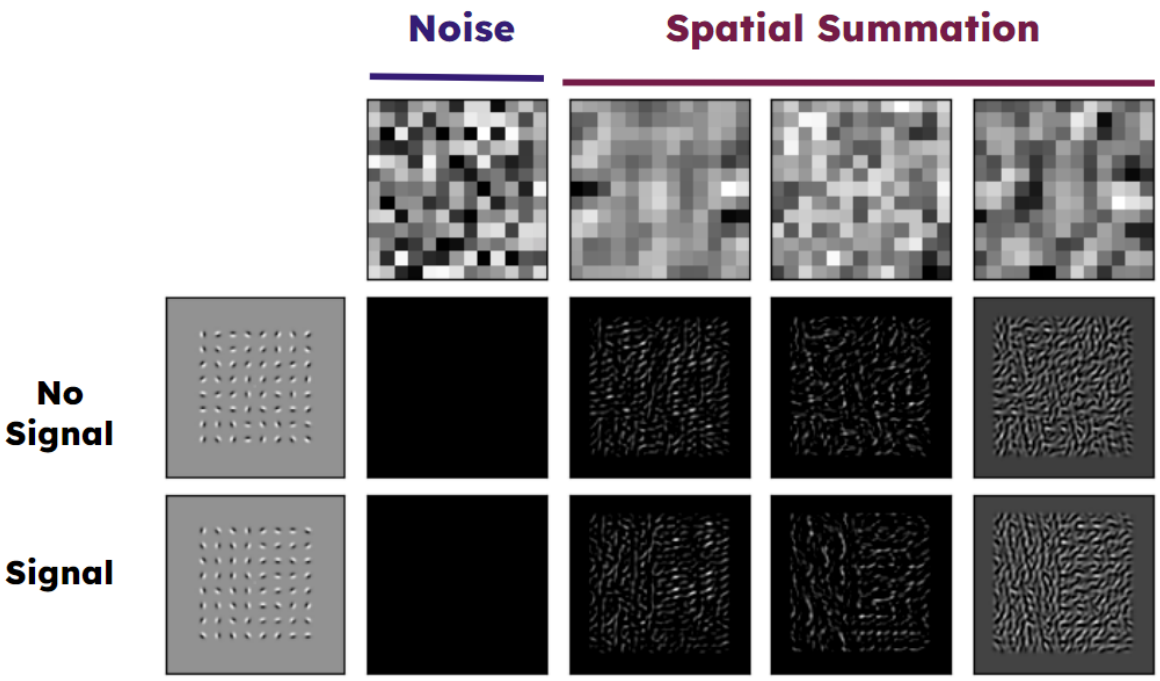


Figure 16 Kernel visualisation for dual layered CNN

layered model. This new reversed model contained a first convolutional layer with 32 neurones and a 5x5 pixel kernel size, and a second convolutional layer with 16 neurones and a 13x13 pixel kernel size. When cross validated, this model had an accuracy of 95.1±1.3%, one of the highest achieved by any model. The kernels of this model were analysed and summarised in Figure 17.

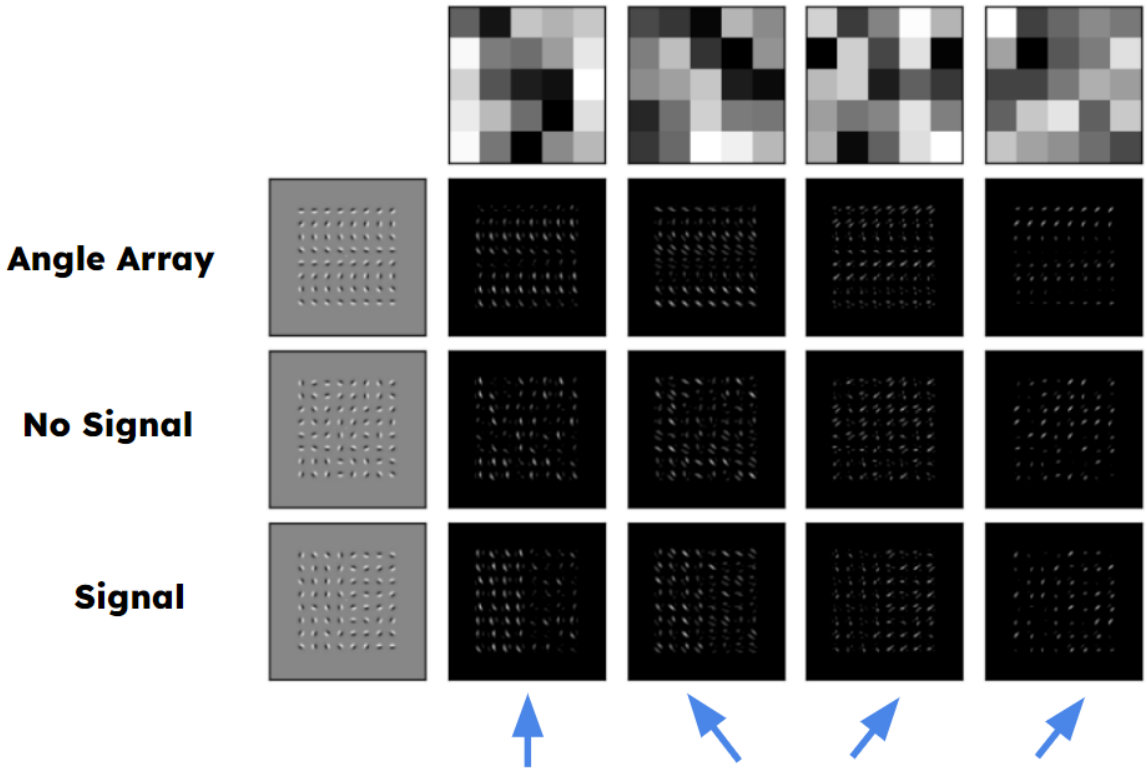


Figure 17 Kernel visualisation for reverse dual layered CNN

The kernel outputs of this model from the first convolutional layer show a clear ori-

entation selectivity; this is similar to the expected response of simple cells in the V1 which are specific to orientation, and are involved with the first filter and rectify stages of the FRF model. Furthermore, the second larger layer provides a large= sweep and detection of texture, which is similar to the second filter in the FRF model.

4.5 Validation of models on unseen Data

As an additional validation step, the models described in Section 4.4 were tested on unseen textures. These included horizontal boundaries, diagonal boundaries and also boundaries that were offset from the centre. Because the CNN models were trained exclusively on datasets with a vertical boundary, this test provided information regarding whether the models were generalised for a texture recognising visual system. The results of this training for the six evaluated models is shown in Table 2

Network	Accuracy			
	Vertical Dataset	Horizontal Dataset	Diagonal Dataset	Offset Dataset
1 Neuron, Single Layer	90.0% (+-1.9%)	84.2%	87.0%	85.3%
4 Neurons, Single Layer	91.1% (+-2.0%)	87.4%	87.8%	88.9%
8 Neurons, Single Layer	91.7% (+-1.6%)	87.2%	85.2%	88.0%
32 Neurons, Single Layer	92.5% (+-2.1%)	89.0%	87.2%	89.0%
Dual Layer CNN	93.2% (+-1.1%)	87.4%	83.8%	85.0%
Reversed Dual Layer CNN	95.1% (+-1.3%)	90.2%	93.2%	88.4%

Table 2 Accuracies of biological models for unseen datasets

Interestingly, the evaluated models all achieved accuracies of over 85% for each of the datasets; while this isn't as high as the cross validation accuracy on the trained dataset, it is still promising to see the models perform so well on unseen data. This may indicate that the chosen models are valid and can be used to infer structure of the V1. Furthermore, the reversed dual layer CNN was able to identify the unfamiliar datasets with the highest accuracy, which may suggest its biological significance.

5. Discussion of Biological Implications

Following the training and analysis of models, it was time to convert the quantitative data and trends into qualitative data regarding the processes occurring in the V1. While there is sufficient evidence to draw conclusions about the structure of the V1, it is important to remember that these inferences are speculative, and further testing would need to occur before these processes are known for sure. This section will focus on linking the structure of high performing CNNs discovered during testing to existing literature to provide evidence for or against particular models.

5.1 User Testing and Exploratory Analysis

As shown in Table 1, users had a relatively high performance of above 85% on the three datasets. Accuracy was highest on the larger datasets at 88.2%; this may highlight that texture is easier to detect when many texture units are present. This same trend of higher accuracies on the large dataset was also present during the training of CNN models. The split between hit, miss, CR and FA rates was relatively even, indicating that the noise level in the dataset was set accurately.

Analysis into activation layers highlighted that layers which were similar to half-wave reification produced models which achieved high accuracies. This is interesting as it directly models simple cells in the V1, which are believed to undergo a half-wave rectification computation after convolution of their receptor field. In contrast however, the FRF model, shown in Figure 4, uses full wave rectification. This can still be achieved with a half wave rectification however; in the biological model this is done via the hierarchical structure of simple and complex cells. When the RFs of anti phase simple cells are summed to form the output of a complex cell, it is equivalent to a full wave rectification; this is indicated in Figure 3 [36]. Henceforth, it is possible that a similar process could occur in the generated CNNs to form a FRF model.

5.2 Biological Implications of Structure Adjustment

Architectural modification of the CNNs produced clear trends for the effect of kernel size, neuron number and model depth. While these findings were used to create the models that were analysed in more depth, these trends also have significant biological merit on their own.

5.2.1 Modifying Neuron Number and Kernel Size

Increasing the neuron number in the models resulted in an increased performance, indicating that a larger number of unique filters is crucial for succinct texture recognition; the biological equivalent to this have many different types of simple cell in the V1, all tuned to different orientations or spatial frequencies. Interestingly, each dataset plateaued at around the same accuracy level. This differed to the hypothesis that increasing texture complexity/size would require more neurons to cover all elements of the texture. This suggests that the unique filters are not interested in particular units of texture, but rather their properties, such as spatial frequency and orientation. This may explain why diminishing returns are observed at higher neuron numbers, as the orientation resolution increases to a point that is not required for texture recognition.

Furthermore, although higher neuron numbers work best, models can still achieve high accuracies of above 90% with only a single neuron. Kernel size shows a clear maximum at around 13x13 pixels; this value in itself is only relevant to these datasets, so from a biological perspective the ratio between kernel size and the unit Gabor patches, shown in Figure 7, is more relevant. At a kernel size of 13x13 pixels, the RF of the kernel spans multiple Gabor patches. Hence it is likely that in order for simple cells to operate, the size of their RFs should encompass the boundaries of individual texture units; RFs that are too small would not be able to determine relationships between texture units, while those that are too big would lose spatial resolution and be unable to selectively view boundaries between texture units. Referring back to the low neuron number models, a possible mechanism of action is that kernels that detect Gabor patches around them, and produce a positive output if those patches indicate a texture boundary. This is observed experimentally in Section 4.4, and would explain how only a single neuron is required for signal discrimination.

5.2.2 Modifying Model Depth

When considering CNNs with two or more convolutional layers, the minimum kernel size required for valid training decreases to around 5x5 pixels, around the size of a single Gabor patch. This indicates that, in these multilayered models, the second convolutional layer can connect outputs from he first convolutional layer. This may be more indicative of a FRF model, as it allows the kernels to be selective for both

orientation and spatial frequency at the same time, rather than just one or the other; this line of enquiry is discussed more with the reversed dual layered model in Section 5.3.

As the number of convolution layers increase, typically the accuracy will peak at 2-4 layers before starting to decrease; this finding is consistent with findings in similar studies, and occurs because the model becomes too complex such that over fitting occurs, and also the original images become too convoluted to extract any meaningful information [26]. This finding however, may suggest that the V1 acts with more than one convolutional layer, such is the case with the FRF model. This result may also be modelling some of the input from higher level visual layers, such as the V2 and V4.

5.3 Analysis of Potential Biological Models

Exploration of the models in Section 4.4 highlighted two mechanisms that occur in single layered and dual layered models respectively. These models can be used to provide further evidence to two existing biological models of the V1. The reversed dual layer CNN describes a FRF model with hierarchical arrangement of simple and complex cells. This model has a lot of merit given its very high accuracy, and support from existing literature. The second biological model is that of the single layered CNN which describes the action of simple cells, with some evidence of parallel action from complex cells.

The reversed dual layer CNN has an initial convolutional layer that was shown in Figure 17to be heavily orientationally specific. This is then fed into a ReLU (equivalent to half wave rectification) and a max pooling layer. This process draws very similar parallels to the hierarchical model of complex and simple cells in the V1, shown in Figure 3. If it is assumed that half-rectified anti phase kernels are summed together to form full wave rectification, this computation stage is also equivalent to the first filter and rectify stage of the FRF model. Following this, the second filter has a much larger RF and computes the outputs from the first layers to detect texture boundaries. This is the exact same process occurring in the second filter stage of the FRF model. Lastly, the second convolutional output is fed through a ReLU activation layer, and classified by a binary sigmoid output function. It is here that the similarities between the reverse dual layer CNN model and FRF model end, however, this may classification process may be synonymous with higher level regions of the visual cortex associated with decision making.

In the 8 neuron, singled layered CNN dissected in Figure 15, kernels can be identified as orientation specific filters or spatial summation filters depending on their response to the datasets. The single layered CNN contains a convolution output followed by half wave rectification and then classification with the binary sigmoid output layer. From what is known biologically about the V1, this would be similar to the output of simple cells being classified by higher levels of the brain. However, the spatial summation filters cannot be described by the simple cells alone, which are spatially specific. Hence, a logical conclusion would be that these spatial summation filters represent the action of complex cells in parallel with simple cells.This conclusion comes from recent studies which indicate that a certain subset of complex cells have direction connections to the LGN, and may work in parallel with simple cells to provide meaningful output [7, 11]. It is important to note that this is only a representation, as it is widely accepted that the mechanisms of complex cell computation are different to convolution, and involve full wave rectification rather than half wave.

Overall, given that these models were generated separately to other techniques, but still landed upon widely accepted computational models of the V1, adds evidence to existing literature, and highlights the validity biological CNN modelling.

6. Conclusions

A wide range of CNN architectures and hyper parameters were used to model human texture visualisation in the V1. Human texture response data in the form of an array of Gabor patches was generated and used to train CNN models to mimic a human response to texture. A base model, derived from existing literature, was used to explore how kernel size and neuron number effect the accuracy of the trained CNN; these parameters relate directly to the RFs of simple V1 neurons. An ideal size of kernel was found which spanned the boundary of multiple Gabor's, and an increasing number of neurons was found to increase model accuracy, but to diminishing returns over three datasets of differing texture complexity. This suggested that the RFs of simple V1 neurons are a similar size to the boundaries between units of texture, and supported the claim that they are orientation specific.

A handful of models were chosen for further investigation, including four single layered CNNs and a two dual layered CNNs (one was a reverse of the other). The kernel structure of these models were dissected, as well as the output of their first convolutional layer on a variety of datasets. This provided insight into how the models functioned, and provided biological parallels to existing computational models of the V1. The reverse dual layer CNN obtained a very high accuracy, and followed a FRF model with hierarchical simple and complex cell components very closely [16]. Due to its high accuracy, and backing by existing literature, it is likely that this model matches the human V1 very closely. Additionally, the single layered CNNs may provide some evidence to the parallel processing of simple cells and a select subgroup of complex cells in the V1. This research provides further evidence to pre-existing models of the V1, and potential applications could include computer vision or medical diagnosis.

7. Suggestions for Future Work

An extension of this project could involve the following:

- *Improve the data collection process:* In this research report, human texture response data was only labelled by the researchers of the project. A more diverse range of participants, including Māori, could be included to create models that were more representative of the population. Furthermore, EEG could be performed during the data collection to ensure activation is only occurring in the V1.
- *Include Gabor patches of varying frequency in the test data:* Because simple cells are spatially specific while complex cells aren't, this would help to distinguish between the activity of simple and complex cells in the V1, adding more value to the biological inferences.
- *Perform more experimentation with the "reverse dual later" architecture:* This architecture was tested quite late into the project, and hence there wasn't sufficient time to fully explore it. Experimentation could include adjusting the kernel size in in the two layers, and exploring the kernel output of the second layer to a higher degree.

References

[1] F. A. KINGDOM, N. PRINS, and A. HAYES, "Mechanism independence for texture-modulation detection is consistent with a filter-rectify-filter mechanism," *Visual Neuroscience*, vol. 20, no. 1, pp. 65–76, jan 2003.

[2] L. E. Hallum, M. S. Landy, and D. J. Heeger, "Human primary visual cortex (v1) is selective for second-order spatial frequency," *Journal of Neurophysiology*, vol. 105, no. 5, pp. 2121–2131, may 2011.

[3] L. Armi and S. Fekri-Ershad, "Texture image analysis and texture classification methods - a review," 2019.

[4] J. Taylor and Y. Xu, "Representation of color, form, and their conjunction across the human ventral visual pathway," *NeuroImage*, vol. 251, p. 118941, may 2022.

[5] S. G. Solomon, "Retinal ganglion cells and the magnocellular, parvocellular, and koniocellular subcortical visual pathways from the eye to the brain," in *Handbook of Clinical Neurology*. Elsevier, 2021, pp. 31–50.

[6] R. B. H. Tootell, N. K. Hadjikhani, W. Vanduffel, A. K. Liu, J. D. Mendola, M. I. Sereno, and A. M. Dale, "Functional analysis of primary visual cortex (v1) in humans," *Proceedings of the National Academy of Sciences*, vol. 95, no. 3, pp. 811–817, feb 1998.

[7] Y. Lian, A. Almasi, D. B. Grayden, T. Kameneva, A. N. Burkitt, and H. Meffin, "Learning receptive field properties of complex cells in v1," *PLOS Computational Biology*, vol. 17, no. 3, p. e1007957, mar 2021.

[8] M. Carandini, "What simple and complex cells compute," *The Journal of Physiology*, vol. 577, no. 2, pp. 463–466, nov 2006.

[9] L. M. Martinez and J.-M. Alonso, "Complex receptive fields in primary visual cortex," *The Neuroscientist*, vol. 9, no. 5, pp. 317–331, oct 2003.

[10] J. A. Movshon, I. D. Thompson, and D. J. Tolhurst, "Receptive field organization of complex cells in the cat's striate cortex." *The Journal of Physiology*, vol. 283, no. 1, pp. 79–99, oct 1978.

[11] G. Kim, J. Jang, and S.-B. Paik, "Periodic clustering of simple and complex cells in visual cortex," *Neural Networks*, vol. 143, pp. 148–160, nov 2021.

[12] J. R. Bergen and M. S. Landy, "Computational modeling of visual texture segregation," in *Computational Models of Visual Processing*. The MIT Press, 1991.

[13] H. R. Wilson, "Non-fourier cortical processes in texture, form, and motion perception," in *Cerebral Cortex*. Springer US, 1999, pp. 445–477.

[14] J. E. W. MAYHEW and J. P. FRISBY, "Texture discrimination and fourier analysis in human vision," *Nature*, vol. 275, no. 5679, pp. 438–439, oct 1978.

[15] F. L. Royer, M. S. Rzeszotarski, and G. C. Gilmore, "Application of two-dimensional fourier transforms to problems of visual perception," *Behavior Research Methods and Instrumentation*, vol. 15, no. 2, pp. 319–326, mar 1983.

[16] M. S. Landy, *The New Visual Neurosciences*, 2013, ch. Texture analysis and perception, pp. 639–652.

[17] B. A. Wandell, S. O. Dumoulin, and A. A. Brewer, "Visual field maps in human cortex," *Neuron*, vol. 56, no. 2, pp. 366–383, oct 2007.

[18] R. T. Schirrmeister, J. T. Springenberg, L. D. J. Fiederer, M. Glasstetter, K. Eggensperger, M. Tangermann, F. Hutter, W. Burgard, and T. Ball, "Deep learning with convolutional neural networks for EEG decoding and visualization," *Human Brain Mapping*, vol. 38, no. 11, pp. 5391–5420, aug 2017.

[19] J. M. Vaz and S. Balaji, "Convolutional neural networks (CNNs): concepts and applications in pharmacogenomics," *Molecular Diversity*, vol. 25, no. 3, pp. 1569–1584, may 2021.

[20] Y. LeCun, Y. Bengio, and G. Hinton, "Deep learning," *Nature*, vol. 521, no. 7553, pp. 436–444, may 2015.

[21] R. M. Cichy, A. Khosla, D. Pantazis, A. Torralba, and A. Oliva, "Comparison of deep neural networks to spatio-temporal cortical dynamics of human visual object recognition reveals hierarchical correspondence," *Scientific Reports*, vol. 6, no. 1, jun 2016.

[22] R. Yamashita, M. Nishio, R. K. G. Do, and K. Togashi, "Convolutional neural networks: an overview and application in radiology," *Insights into Imaging*, vol. 9, no. 4, pp. 611–629, jun 2018.

[23] J. Ukita, T. Yoshida, and K. Ohki, "Characterisation of nonlinear receptive fields of visual neurons by convolutional neural network," *Scientific Reports*, vol. 9, no. 1, mar 2019.

[24] G. Bhumbra, "Deep learning improved by biological activation functions," 03 2018.

[25] D. Arpit, S. Jastrzebski, N. Ballas, D. Krueger, E. Bengio, M. S. Kanwal, T. Maharaj, A. Fischer, A. Courville, Y. Bengio, and S. Lacoste-Julien, "A closer look at memorization in deep networks," 2017.

[26] M. Kociołek, M. Kozłowski, and A. Cardone, "A convolutional neural networks-based approach for texture directionality detection," *Sensors*, vol. 22, no. 2, p. 562, jan 2022.

[27] D. L. K. Yamins and J. J. DiCarlo, "Using goal-driven deep learning models to understand sensory cortex," *Nature Neuroscience*, vol. 19, no. 3, pp. 356–365, feb 2016.

[28] P. M. Claessens and J. Wagemans, "Perceptual grouping in gabor lattices: Proximity and alignment," *Perception and Psychophysics*, vol. 67, no. 8, pp. 1446–1459, nov 2005.

[29] U. Ernst, S. Denève, and G. Meinhardt, "Detection of gabor patch arrangements is explained by natural image statistics," *BMC Neuroscience*, vol. 8, no. S2, jul 2007.

[30] A. E. Burgess, R. F. Wagner, R. J. Jennings, and H. B. Barlow, "Efficiency of human visual signal discrimination," *Science*, vol. 214, no. 4516, pp. 93–94, oct 1981.

[31] Z. Hussain and P. J. Bennett, "Perceptual learning of detection of textures in noise," *Journal of Vision*, vol. 20, no. 7, p. 22, jul 2020.

[32] M. C. Chirodea, O. C. Novac, C. M. Novac, N. Bizon, M. Oproescu, and C. E. Gordan, "Comparison of tensorflow and PyTorch in convolutional neural network - based applications," in *2021 13th International Conference on Electronics, Computers and Artificial Intelligence (ECAI)*. IEEE, jul 2021.

[33] Y. Jung and J. Hu, "A k-fold averaging cross-validation procedure," *Journal of Nonparametric Statistics*, vol. 27, no. 2, pp. 167–179, feb 2015.

[34] N. Srivastava, G. Hinton, A. Krizhevsky, I. Sutskever, and R. Salakhutdinov, "Dropout: A simple way to prevent neural networks from overfitting," *J. Mach. Learn. Res.*, vol. 15, no. 1, jan 2014.

[35] Z. Zhang and M. R. Sabuncu, "Generalized cross entropy loss for training deep neural networks with noisy labels," 2018.

[36] J. A. Solomon and G. Sperling, "Full-wave and half-wave rectification in second-order motion perception," *Vision Research*, vol. 34, no. 17, pp. 2239–2257, sep 1994.

Appendix A Full List of Models Tested

This appendix contains a full list of models trained, including statistics regarding their final mean cross validated accuracy, mean loss and mean epoch number.

Small Dataset Training

All models trained on the small dataset used:

- A 25% dropout layer
- SGD optimiser
- Binary cross entropy loss function
- ReLu activation function (except for activation layer analysis)

No dense layers were used with on small datasets, excluding the sigmoid classification layer.

Medium Dataset Training

All models trained on the medium dataset used:

- Binary cross entropy loss function
- ReLu activation function.

Large Dataset Training

All models trained on the large dataset used:

- A 25% dropout layer
- SGD optimiser
- Binary cross entropy loss function
- ReLu activation function.

No dense layers were used with large datasets, excluding the sigmoid classification layer.

Convolutional Layers	Kernel Size	Validation Accuracy	Accuracy std.	Epoch No.
1: 32	3x3	55.2%	2.8%	102
1: 32	5x5	54.7%	2.4%	100
1: 32	7x7	55.2%	2.1%	105
1: 32	9x9	64.6%	5.7%	309.1
1: 32	11x11	88.3%	2.2%	291.2
1: 32	13x13	91.0%	2.0%	305.3
1: 32	15x15	88.7%	5.0%	336
1: 32	17x17	90.6%	2.5%	373.3
1: 32	19x19	84.0%	8.6%	384.3
1: 128	11x11	91.0%	2.2%	326.7
1: 64	11x11	90.2%	1.9%	345.4
1:16	11x11	89.1%	2.5%	326.7
1:8	11x11	86.8%	2.8%	346.8
1:4	11x11	82.5%	4.9%	330.5
1:2	11x11	77.5%	6.5%	380.2
1:1	11x11	79.1%	5.6%	371.1
1: 128	13x13	91.5%	1.8%	327.1
1: 64	13x13	92.1%	1.6%	387.3
1:16	13x13	91.2%	2.3%	373.7
1:8	13x13	89.4%	3.1%	371.4
1:4	13x13	89.5%	2.4%	400.1
1:2	13x13	86.4%	4.0%	392.9
1:1	13x13	85.6%	3.7%	476.2
2: 32, 16	13, 3	93.8%	1.1%	378.2
2: 32, 16	11, 3	92.8%	0.6%	381.9
2: 32, 16	9, 3	92.2%	1.2%	377.4
2: 32, 16	7, 3	91.3%	1.9%	404.8
2: 32, 16	5, 3	88.5%	2.4%	501.5
2: 32, 16	3, 3	56.5%	2.7%	350.7
3: 32, 16, 8	11&3&3	93.7%	1.1%	373.2
4: 32, 16, 8, 8	11&3&3&3	94.7%	2.1%	381.8
5: 32, 16, 8, 8, 8	11&3&3&3&3	94.1%	2.1%	424.6
1: 1	3x3	55.4%	2.8%	350
1: 1	5x5	55.3%	2.0%	350
1: 1	7x7	55.5%	2.1%	350
1: 1	9x9	55.4%	1.2%	361.2
1: 1	11x11	80.7%	3.6%	458.6
1: 1	13x13	84.0%	2.0%	427
1: 1	15x15	80.4%	9.5%	380
1: 1	17x17	86.2%	2.4%	428.7
1: 1	19x19	80.3%	6.6%	388.3

Convolutional Layers	Kernel Size	Activation Layer	Validation Accuracy	Accuracy std.	Epoch No.
1: 32	13x13	Leaky ReLu	90.1%	2.2%	391.4
1: 32	13x13	eLu	90.8%	2.3%	388.2
1: 32	13x13	seLu	55.7%	1.1%	350
1: 32	13x13	tanh	57.1%	1.2%	413
1: 32	13x13	exponential	55.1%	1.6%	350.4
1: 32	13x13	PReLu	92.3%	1,6%	380.6

Table A1 Total training performed on Small Dataset

Convolutional Layers	Dense Layers	Kernel Size	Optimiser	Dropout Layer	Validation Accuracy	Accuracy std.	Epoch No.
2: 64, 16	1:128	3x3	Adam	0	95.6% (1.3%)	(1.3%)	61.2 (31.4)
2: 32, 16	1:128	3x3	Adam	0	96.1% (1.2%)	(1.2%)	66.9 (27.7)
2: 16, 16	1:128	3x3	Adam	0	95.8% (1.6%)	(1.6%)	86 (27.3)
2: 8, 16	1:128	3x3	Adam	0	95.4% (1.1%)	(1.1%)	75.4 (17.5)
2. 4, 16	1:128	3x3	Adam	0	94.9% (2.3%)	(2.3%)	62.2 (35.8)
2: 32, 16	1:64	3x3	Adam	0	95.0% (1.5%)	(1.5%)	72.6 (24.1)
2: 32, 16	1:32	3x3	Adam	0	95.2% (0.7%)	(0.7%)	89.6 (27.6)
2: 32, 16	1:16	3x3	Adam	0	95.0% (1.4%)	(1.4%)	84.6 (36.5)
2: 32, 16	1:8	3x3	Adam	0	92.5% (2.1%)	(2.1%)	89.3 (20.3)
1: 64	0	3x3	Adam	0	53.5% (1.7%)	(1.7%)	25.7 (7.0)
1: 64	0	5x5	Adam	0	54.5% (2.6%)	(2.6%)	46.0 (25.0)
1: 64	0	7x7	Adam	0	53.4% (2.6%)	(2.6%)	23.8 (4.2)
1: 64	0	9x9	Adam	0	51.4% (1.6%)	(1.6%)	21.0 (5.1)
2: 64, 16	0	3x3	Adam	0	54.0% (3%)	(3%)	35.8 (16.1)

Convolutional Layers	Kernel Size	Optimiser	Dropout Layer	Validation Accuracy	Accuracy std.	Epoch No.
1: 64	3x3	SDG	0.25	54.5%	2.5%	55.7
1: 64	5x5	SDG	0.25	55.2%	1.3%	52.3
1: 64	7x7	SDG	0.25	55.2%	2.4%	55.8
1: 64	9x9	SDG	0.25	83.4%	3.9%	327.6
1: 64	11x11	SDG	0.25	91.2%	1.64%	189.5
1: 64	13x13	SDG	0.25	92.4%	1.51%	184.5
1: 64	15x15	SDG	0.25	92.5%	1.57%	195.2
1: 64	17x17	SDG	0.25	90.4$	4.00%	251.8
1: 64	19x19	SDG	0.25	75.0%	10.42%	227.4
1: 32	3x3	SDG	0.25	55.1%	1.35%	57.8
1: 32	5x5	SDG	0.25	54.9%	1.62%	60.7
1: 32	7x7	SDG	0.25	54.5%	3.0%	58.8
1: 32	9x9	SDG	0.25	79.2%	4.6%	313.1
1: 32	11x11	SDG	0.25	90.7%	2.4%	178.5
1: 32	13x13	SDG	0.25	92.5%	2.1%	193.4
1: 32	15x15	SDG	0.25	91.2%	3.2%	227.1
1: 32	17x17	SDG	0.25	84.5%	5.6%	245.3
1: 32	19x19	SDG	0.25	66.9%	10.4%	261.3
1: 16	7x7	SDG	0.25	54.6%	1.0%	203.6
1: 16	9x9	SDG	0.25	78.9%	6.6%	340.5
1: 16	11x11	SDG	0.25	90.2%	2.2%	205.5
1: 16	13x13	SDG	0.25	91.5%	2.6%	250.3
1: 16	15x15	SDG	0.25	73.7%	6.4%	365.3
1: 16	17x17	SDG	0.25	56.7%	1.2%	350
1: 8	7x7	SDG	0.25	54.1%	1.6%	210.4
1: 8	9x9	SDG	0.25	68.0%	8.3%	252.2
1: 8	11x11	SDG	0.25	88.4%	2.5%	219
1: 8	13x13	SDG	0.25	91.7%	1.6%	263.6
1: 8	15x15	SDG	0.25	71.0%	7.8%	200
1: 8	17x17	SDG	0.25	55.2%	3.2%	200
1: 4	7x7	SDG	0.25	54.7%	1.5%	200
1: 4	9x9	SDG	0.25	62.6%	2.8%	218.1
1: 4	11x11	SDG	0.25	86.2%	2.8%	218.1
1: 4	13x13	SDG	0.25	91.1%	2.0%	256.7
1: 4	15x15	SDG	0.25	68.3%	7.2%	242.3
1: 4	17x17	SDG	0.25	56.5%	1.0%	211
1:16	11x11	SDG	0.25	90.2%	2.2%	205.5
1:8	11x11	SDG	0.25	88.4%	2.5%	219
1:4	11x11	SDG	0.25	86.2%	2.8%	218.1
1:2	11x11	SDG	0.25	85.1%	4.8%	245.9
1:1	11x11	SDG	0.25	84.7%	1.8%	242.2
1:16	13x13	SDG	0.25	91.5%	2.6%	250.3
1:8	13x13	SDG	0.25	91.7%	1.6%	263.6
1:4	13x13	SDG	0.25	91.1%	2.0%	256.7
1:2	13x13	SDG	0.25	89.7%	1.7%	255
1:1	13x13	SDG	0.25	90.0%	1.9%	276.5
2: 32, 16	13, 3	SDG	0.25	93.2%	1.1%	182.3
2: 32, 16	11, 3	SDG	0.25	93.1%	1.4%	171.4
2: 32, 16	9, 3	SDG	0.25	93.1%	1.4%	215.1
2: 32, 16	7, 3	SDG	0.25	92.0%	1.7%	230.8
2: 32, 16	5, 3	SDG	0.25	90.1%	3.0%	307.8
2: 32, 16	3, 3	SDG	0.25	56.6%	2.2%	206.4
2: 64, 16	13, 3	SDG	0.25	94.0%	1.6%	214.5
2: 64, 16	11, 3	SDG	0.25	93.5%	1.2%	185.8
2: 128, 16	13, 3	SDG	0.25	94.0%	1.6%	177.4
3: 32, 16, 8	11&3&3	SDG	0.25	94.6%	1.7%	181.2
4: 32, 16, 8, 8	11&3&3&3	SDG	0.25	92.9%	1.9%	166.9
5: 32, 16, 8, 8, 8	11&3&3&3&3	SDG	0.25	91.6%	2.4%	249.4
1:16	9x9	SDG	0.25	78.9%	6.6%	340.5
1:8	9x9	SDG	0.25	68.0%	8.3%	252.2
1:4	9x9	SDG	0.25	62.6%	4.9%	237.7
1:2	9x9	SDG	0.25	60.4%	5.1%	231.4
1:1	9x9	SDG	0.25	57.2%	4.2%	202.6

Table A2 Total training performed on Medium Dataset

Convolutional Layers	Kernel Size	Validation Accuracy	Accuracy std.	Epoch No.
1: 32	3x3	54.7%	1.7%	59.3
1: 32	5x5	55.0%	1.6%	55
1: 32	7x7	54.3%	2.4%	55.5
1: 32	9x9	73.6%	5.5%	404.9
1: 32	11x11	90.5%	2.1%	336.2
1: 32	13x13	92.6%	1.5%	348.5
1: 32	15x15	89.8%	4.6%	366.2
1: 32	17x17	87.5%	3.4%	356.2
1: 32	19x19	72.3%	9.1%	306.7
1: 128	11x11	92.6%	1.1%	315.7
1: 64	11x11	92.4%	1.6%	319
1:16	11x11	90.8%	3.0%	354.9
1:8	11x11	89.2%	3.8%	330.44
1:4	11x11	87.7%	3.0%	386.7
1:2	11x11	87.9%	2.2%	379
1:1	11x11	88.6%	0.6%	356.7
2: 32, 16	13, 3	94.8%	1.3%	375.8
2: 32, 16	11, 3	94.9%	1.2%	364.1
2: 32, 16	9, 3	94.8%	0.5%	369.2
2: 32, 16	7, 3	93.9%	1.2%	388.5
2: 32, 16	5, 3	90.9%	1.3%	381.5
2: 32, 16	3, 3	57.5%	2.5%	357
3: 32, 16, 8	11&3&3	95.7%	1.4%	384.7
4: 32, 16, 8, 8	11&3&3&3	95.5%	1.6%	391.3
5: 32, 16, 8, 8, 8	11&3&3&3&3	94.4%	1.3%	356.1

Table A3 Total training performed on Large Dataset

Appendix B Supporting Figures

This appendix contains supporting figures from the methods and results sections, aimed at providing additional context to concepts mentioned in the text.

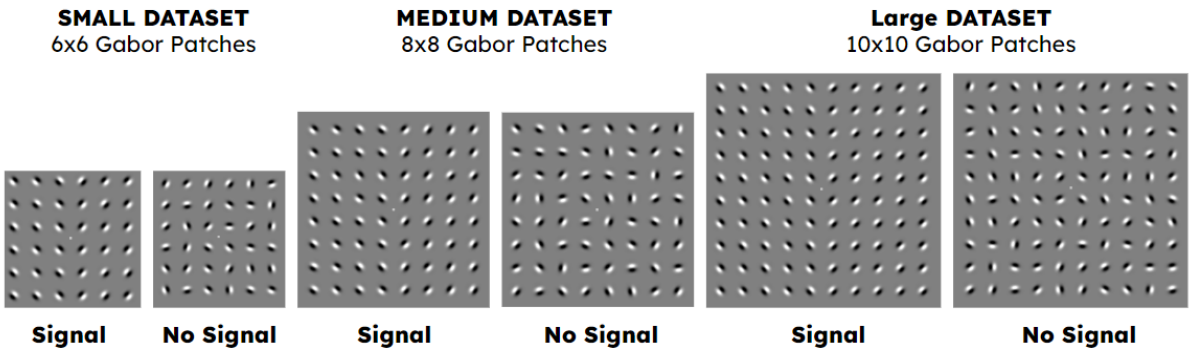


Figure B1 Comparison of small, medium and large datasets used for testing

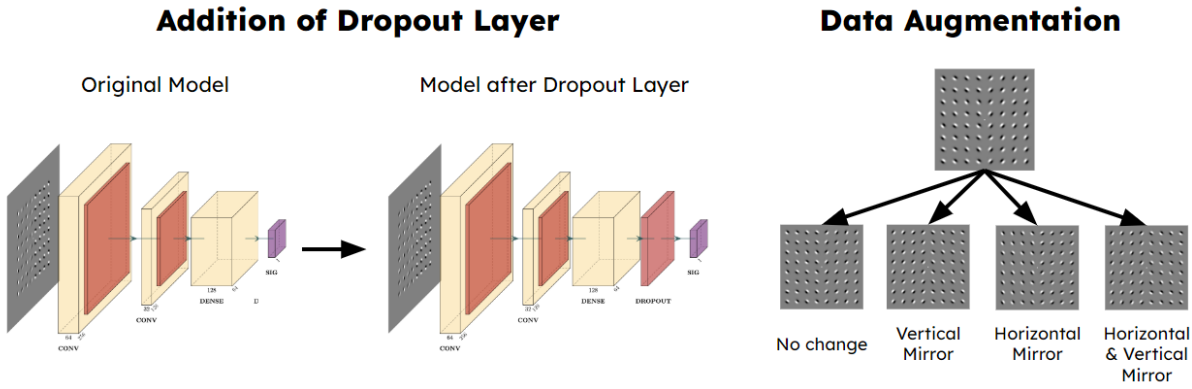


Figure B2 Addition of anti-overfitting measures in base model

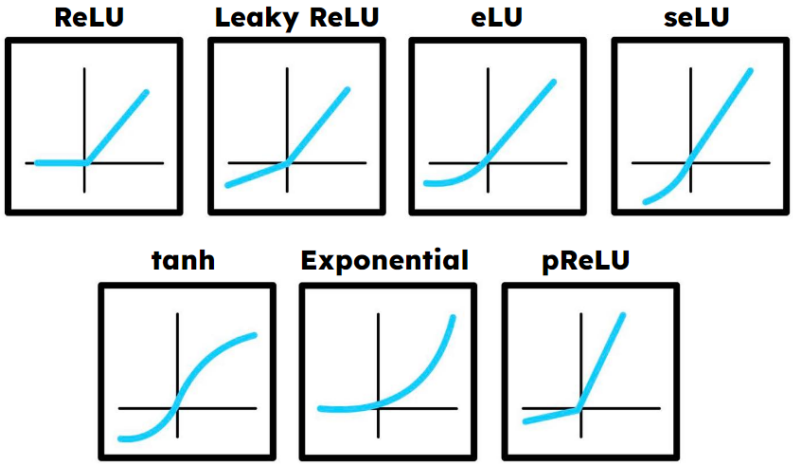


Figure B3 Sketches of tested activation layer functions.

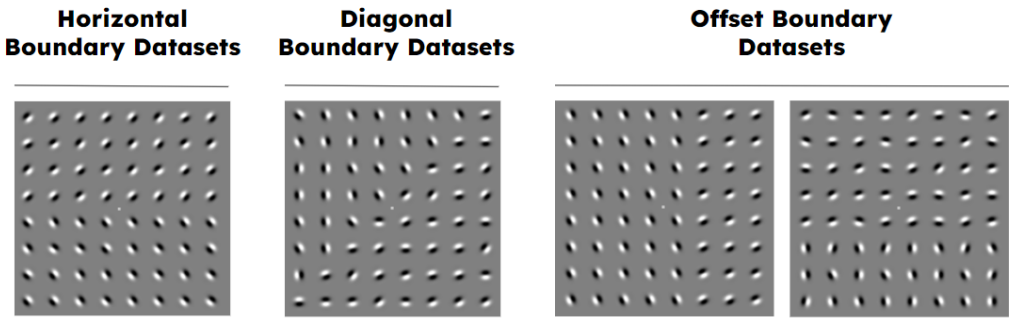


Figure B4 Unseen datasets for validating models

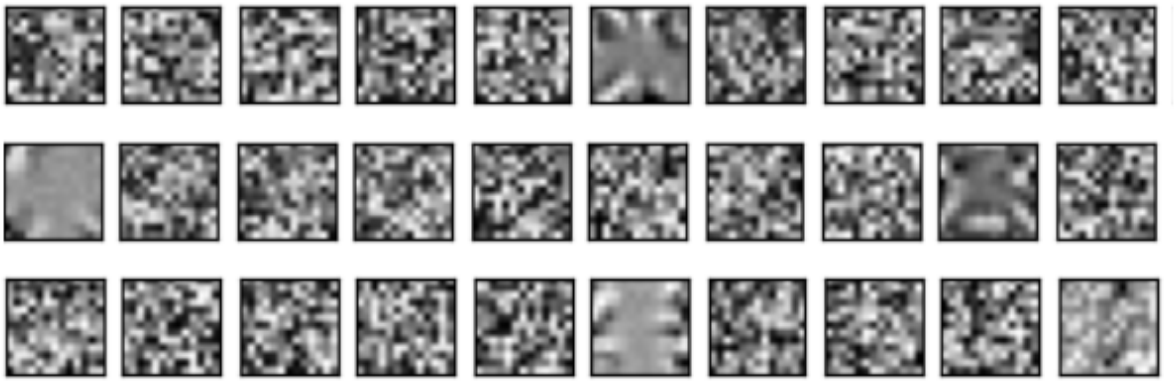


Figure B5 Kernel Analysis of 32 neuron single layer model

Appendix C Specific Model Architecture

This appendix provides additional information about the exact models used in Sections 3.3 and 4.4.

#	Layer Type	Network (1)	Network (2)	Network (4)	Network (8)	...	Network (128)
1	Input (min size x, min size y, channels)	128,128, 1	128,128, 1	128,128, 1	128,128, 1		128,128, 1
2	Convolution (size x, size y, count)	11,11,1	11,11,2	11,11,4	11,11,8		11,11,128
3	Activation Layer	"ReLU"	"ReLU"	"ReLU"	"ReLU"		"ReLU"
4	Max pool	2,2	2,2	2,2	2,2		2,2
5	Flatten						
6	Dropout	0.5	0.5	0.5	0.5		0.5
7	Output (count)	1	1	1	1		1

Table C1 CNN structure for models with different neuron numbers

#	Layer Type	Network (3x3)	Network (5x5)	Network (7x7)	Network (9x9)	...	Network (19x19)
1	Input (min size x, min size y, channels)	128,128, 1	128,128, 1	128,128, 1	128,128, 1		128,128, 1
2	Convolution (size x, size y, count)	3,3,32	5,5,32	7,7,32	9,9,32		19,19,32
3	Activation Layer	"ReLU"	"ReLU"	"ReLU"	"ReLU"		"ReLU"
4	Max pool	2,2	2,2	2,2	2,2		2,2
5	Flatten						
6	Dropout	0.5	0.5	0.5	0.5		0.5
7	Output (count)	1	1	1	1		1

Table C2 CNN structure for models with different kernel size

Single Neuron Model

#	Layer Type	Network (1, 3x3)	Network (1,5x5)	Network (1,7x7)	...	Network (1,13x13)
1	Input (min size x, min size y, channels)	128,128, 1	128,128, 1	128,128, 1		128,128, 1
2	Convolution (size x, size y, count)	3,3,1	5,5,1	7,7,1		13,13,1
3	Activation Layer	"ReLU"	"ReLU"	"ReLU"		"ReLU"
4	Max pool	2,2	2,2	2,2		2,2
5	Flatten					
6	Dropout	0.5	0.5	0.5		0.5
7	Output (count)	1	1	1		1

Dual Layer Models

#	Layer Type	Network (32,16, 3x3)	Network (32,16,5x5)	Network (32,16,7x7)	...	Network (32,16,13x13)
1	Input (min size x, min size y, channels)	128,128, 1	128,128, 1	128,128, 1		128,128, 1
2	Convolution (size x, size y, count)	3,3,32	5,5,32	7,7,32		13,13,32
3	Activation Layer	"ReLU"	"ReLU"	"ReLU"		"ReLU"
4	Max pool	2,2	2,2	2,2		2,2
5	Convolution (size x, size y, count)	3,3,16	3,3,16	3,3,16		3,3,16
6	Activation Layer	"ReLU"	"ReLU"	"ReLU"		"ReLU"
7	Max pool	2,2	2,2	2,2		2,2
8	Flatten					
9	Dropout	0.5	0.5	0.5		0.5
10	Output (count)	1	1	1		1

Table C3 CNN structure for single neuron and dual layer models

Note: All convolutional layers followed by a 'ReLU' activation function and a max pooling layer

#	Layer Type	Network (1)	Network (2)	Network (3)	Network (4)	Network (5)
1	Input (min size x, min size y, channels)	128,128, 1	128,128, 1	128,128, 1	128,128, 1	128,128, 1
2	Convolution (size x, size y, count)	11,11,32	11,11,32	11,11,32	11,11,32	11,11,32
3	Convolution (size x, size y, count)	-	3,3,16	3,3,16	3,3,16	3,3,16
4	Convolution (size x, size y, count)	-	-	3,3,8	3,3,8	3,3,8
5	Convolution (size x, size y, count)	-	-	-	3,3,8	3,3,8
6	Convolution (size x, size y, count)	-	-	-	-	3,3,8
7	Flatten					
8	Dropout	0.5	0.5	0.5	0.5	0.5
9	Output (count)	1	1	1	1	1

Table C4 CNN structure for models with different depth

#	Layer Type	Network (1)	Network (4)	Network (8)	Network (32)
1	Input (min size x, min size y, channels)	128,128, 1	128,128, 1	128,128, 1	128,128, 1
2	Convolution (size x, size y, count)	13,13,1	13,13,4	13,13,8	13,13,32
3	Activation Layer	“ReLU”	“ReLU”	“ReLU”	“ReLU”
4	Max pool	2,2	2,2	2,2	2,2
5	Flatten				
6	Dropout	0.5	0.5	0.5	0.5
7	Output (count)	1	1	1	1

#	Layer Type	Dual Layer	Reverse Dual Layer
1	Input (min size x, min size y, channels)	128,128, 1	128,128, 1
2	Convolution (size x, size y, count)	13,13,32	5,5,32
3	Activation Layer	“ReLU”	“ReLU”
4	Max pool	2,2	2,2
5	Convolution (size x, size y, count)	3,3,16	13,13,16
6	Activation Layer	“ReLU”	“ReLU”
7	Max pool	2,2	2,2
8	Flatten		
9	Dropout	0.5	0.5
10	Output (count)	1	1

Table C5 Full specification of models used in biological analysis