



NATIONAL TECHNICAL UNIVERSITY OF ATHENS

SCHOOL OF ELECTRICAL AND COMPUTER ENGINEERING

DEPARTMENT OF INFORMATION TRANSMISSION SYSTEMS AND MATERIAL TECHNOLOGY

Functional assessment of retinal models using machine learning techniques

DIPLOMA THESIS

NIKOLAS PAPADOPOULOS

Supervisor: Konstantina S. Nikita
Professor, NTUA

Athens, January 2021



NATIONAL TECHNICAL UNIVERSITY OF ATHENS
SCHOOL OF ELECTRICAL AND COMPUTER ENGINEERING
DEPARTMENT OF INFORMATION TRANSMISSION SYSTEMS AND MATERIAL
TECHNOLOGY

Functional assessment of retinal models using machine learning techniques

DIPLOMA THESIS
NIKOLAS PAPADOPOULOS

Supervisor: Konstantina S. Nikita
Professor, NTUA

Approved by the three-member examination committee on January 18, 2021.

(Signature)

(Signature)

(Signature)

.....

Konstantina S. Nikita
Professor, NTUA

.....

Andreas-Georgios Stafylopatis
Professor, NTUA

.....

Giorgos Stamou
Associate Professor, NTUA

Athens, January 2021



NATIONAL TECHNICAL UNIVERSITY OF ATHENS
SCHOOL OF ELECTRICAL AND COMPUTER ENGINEERING
DEPARTMENT OF INFORMATION TRANSMISSION SYSTEMS AND MATERIAL
TECHNOLOGY

(Signature)

.....

Nikolas Papadopoulos

Electrical and Computer Engineering Graduate, NTUA

Copyright © Nikolas Papadopoulos, 2021- All rights reserved.

It is prohibited to copy, store and distribute this work, in whole or in part, for commercial or for-profit purposes. It is permissible to reprint, store and distribute for non-profit, educational or research purposes, provided that the source is mentioned and the present message is maintained.

Questions concerning the use of this work for commercial- profit purposes should be addressed exclusively to the authors

The views and conclusions contained in this work express the authors and it should not be construed that they represent official positions of the National Technical University of Athens, including its Schools, Departments, and Units.

Abstract

This diploma thesis proposes the functional assessment of retinal models, as an alternative to the currently common practice of comparing the similarity of model-generated and ground truth neural responses, as recorded by retinal implants. Functional assessment describes the concept of evaluating the performance of retinal models on image understanding tasks. In this work, we developed a pipeline for functional assessment using machine learning techniques, where we feed retinal models with images and we receive the neural responses of the model, with which we train classifiers on object and digit recognition tasks. In particular, we used CIFAR10, Fashion MNIST and Imagenette datasets for object recognition tasks and MNIST dataset for digit recognition tasks. We also trained common classifiers, such as Multilayer Perceptron (MLP), Support Vector Machine (SVM) and Random Forest. We examined a number of relevant questions: which classifiers perform better in such type of tasks, how the performance of a model differs depending on the task and how the number of available neurons affects the performance of the model. Given also that the Convolutional Neural Network (CNN) retinal model provides a limited number of available neurons (60 in our case), we investigated ways to artificially increase this number. For that, we split the images into different parts, we fed separately each part to the retinal model and we investigated how we can optimally combine the neural responses produced by the model for each part, in order to achieve high performance in image recognition tasks. We found that Random Forest classifier achieves the highest performance and that models perform better with datasets such as MNIST and Fashion MNIST, where we achieved up to 90% accuracy. In addition, simulations indicated that using more neurons improves the overall performance of the model, thus highlighting the need to design retinal prostheses with a larger number of available neurons. Finally, we applied functional assessment in order to compare the performance of two different retinal models. The results showed that there is an agreement between standard and functional assessment, as the two models had similar performance in both techniques. Thus, we conclude that functional assessment produces reliable results and it can be used as an alternative to the standard evaluation technique.

Keywords

retinal models, functional assessment, retinal prosthesis, machine learning, neural networks, classification, digit recognition, object recognition

"Why did we become blind, I don't know, perhaps one day we'll find out, Do you want me to tell you what I think, Yes, I do, I don't think we did go blind, I think we are blind, Blind but seeing, Blind people who can see, but do not see."

–Blindness, Jose Saramago

Acknowledgements

We gratefully acknowledge the support of NVIDIA Corporation with the donation of the Quadro P6000 GPU used for this research.

Contents

Abstract	1
Acknowledgements	5
Contents	7
List of Figures	11
List of Tables	15
1 Introduction	17
1.1 Motivation	17
1.2 Thesis Contribution	18
1.3 Chapter Outline	19
2 Physiology of the Eye	21
2.1 Structure of the Human Eye	21
2.2 The Retina	22
2.2.1 Photoreceptor Cells: Rods and Cones	24
2.2.2 Horizontal Cells	25
2.2.3 Bipolar Cells	25
2.2.4 Amacrine Cells	25
2.2.5 Retinal Ganglion Cells	26
2.3 Receptive Field of Retinal Ganglion Cells	26
3 Spike Trains and Firing Rates	29
3.1 Action Potential and Spike Trains	29
3.2 Types of Firing Rates	29
3.3 Estimating Firing Rates from Spike Trains	31
4 Retinal Models and Retinal Prosthesis	33
4.1 Retinal Models	33
4.2 Retinal Prosthesis	35

5	Machine Learning	39
5.1	Introduction	39
5.2	Types of Machine Learning	39
5.2.1	Supervised Learning	40
5.2.2	Unsupervised Learning	40
5.2.3	Reinforcement Learning	40
5.3	Machine Learning Algorithms	41
5.3.1	Decision Trees	41
5.3.2	Random Forest	42
5.3.3	Support Vector Machines (SVM)	43
5.4	Artificial Neural Networks (ANNs)	45
5.4.1	Single Neuron (Perceptron)	45
5.4.2	Multilayer Perceptrons	46
5.4.3	Activation Functions	46
5.4.4	Cost Functions	47
5.4.5	Training a Neural Network	49
5.5	Convolutional Neural Networks (CNNs)	50
6	Development of a Framework for Functional Assessment	53
6.1	Pipeline of Functional Assessment	53
6.2	Description of Software Tools	54
6.3	Description of Datasets	54
6.4	CNN Retinal Model used in Simulations	54
6.4.1	Architecture of the CNN Retinal Model	54
6.4.2	Description of the Dataset used in Retinal Model's Training	55
6.4.3	Training the CNN Retinal Model	59
6.5	Analysis of Data Preprocessing Steps	60
6.6	Design of Simulations	67
7	Results	71
7.1	Classifiers Performance in CIFAR10 Dataset	71
7.2	Classifiers Performance in MNIST Dataset	71
7.3	Classifiers Performance in Fashion MNIST Dataset	72
7.4	Classifiers Performance in Imagenette Dataset	72
7.5	Comparative Assessment of Results	79
7.5.1	Assessment of classifiers performance in terms of the number of neurons	79
7.5.2	Comparing the performance of classifiers between datasets	79
7.5.3	Comparing the performance of classifiers between simulations with "Adjust=yes" and "Adjust=no"	80

7.5.4 Comparing the performance of classifiers between retinal models	80
8 Discussion	87
8.1 Comparative Assessment of Results	87
8.1.1 Data Preprocessing Steps	87
8.1.2 Datasets	88
8.1.3 Classifiers	89
8.1.4 Retinal Models	89
8.2 Limitations of Functional Assessment	89
9 Conclusions	91
9.1 Summary	91
9.2 Future Work	91
Appendices	93
A Image repetitions in temporal dataset	95
Bibliography	104

List of Figures

2.1	Cross and horizontal section of the human eye [1], [2].	21
2.2	A drawing of a section through the human eye with a schematic enlargement of the retina [3].	23
2.3	Simplified schematic representation of the retina [4].	23
2.4	Schematic representation of the receptive field of an ON-type cell (left) and of an OFF-type cell (right) [5].	27
2.5	Receptive field of an ON-type ganglion cell represented as difference of Gaussians [4].	27
3.1	Estimate firing rate by counting spikes within time bins [6].	31
3.2	Estimate firing rate by using a square (top) or Gaussian (bottom) window. Adapted from [6].	32
3.3	Different approaches to estimate the firing rate. Figure 3.3A shows a spike train from a neuron. Figure 3.3B shows the firing rate estimation computed using bin splitting with a bin size of 100ms. Figure 3.3C shows firing rate estimation determined by sliding a square window function along the spike train with $t = 100ms$ and Figure 3.3D shows firing rate estimation computed using a Gaussian window function with $\sigma_t = 100ms$ [7].	32
4.1	The Linear - Nonlinear (LN) model [8].	34
4.2	The Generalized Linear Model (GLM) [9].	34
4.3	An overview of a retinal prosthesis system. An image is captured by a video camera and converted to an electrical signal by an imaging processing unit. The electrical signal is transmitted through radio frequency (RF) telemetry and/or optical link and processed by external and/or implanted circuitry. The circuit produces a pattern of electrical stimulus, similar to light stimulus, which is applied to the retina via a microelectrode array positioned near the retina. Adapted from [10].	36
4.4	Three types of retinal prostheses: epiretinal, subretinal and suprachoroidal. Ganglion cells (yellow) and bipolar cells (purple) are shown and damaged/eliminated photoreceptors are not shown [11].	37

5.1	Decision tree classification (left). Illustration of decision tree as segmented space (right) [12].	41
5.2	Conceptual schema of Random Forest algorithm [13].	42
5.3	An example of one linearly (left) and one non-linearly (right) separable two-class problem [12].	43
5.4	Maximum-margin hyperplane and margins for an SVM trained with samples from two classes (Wikipedia).	44
5.5	Biological (left) and mathematical (right) model of a neuron [14].	45
5.6	Structure of a Multilayer Perceptron with two hidden layers [15].	46
5.7	Different types of activation functions. Left to right: sigmoid, tanh, ReLU [14].	48
5.8	The general architecture of a CNN [16].	50
6.1	Functional assessment pipeline.	54
6.2	Architecture of the CNN retinal model used in this project [17].	57
6.3	Sample images from the dataset used in retinal model's training [18].	57
6.4	Part of a temporal dataset with <i>temporal_interval</i> = 30. The leftmost column of each row contains the actual image projected onto the retina. The rest of the columns contain the image history. Initial dark frames can also be seen.	58
6.5	Smoothed ganglion cell's firing rate.	58
6.6	An example of data augmentation.	62
6.7	An example of image splitting in nine parts ($p = 3$).	63
6.8	A clipping of the temporal dataset. Every row in the temporal dataset represents an input sample to the retinal model and it has as many frames as the variable temporal_interval (here, <i>temporal_interval</i> =30) . The frame in the leftmost column represents the image that is projected onto the retina at a specific moment and the rest of the frames represent the history of the images that the retina was exposed in previous moments. Empty frames represent darkness projected onto the retina.	64
7.1	Σενσιτιβη πλοτς υσινγ ΡετιΜοδελλι (Σεξτιον 6.4.3) φορ διφφερεντ δατασετς (Ταβλε 6.1), διφφερεντ σιμουλατιονς (Ταβλε 6.5) ανδ διφφερεντ ζομβινατιονς οφ παραμετερς (Ταβλε 6.4). (α) "ΙΦΑΡ10 δατασετ, σιμουλατιονς Α2 & Δ1 ("Αδθυστ=ψεσ"). (β) "ΙΦΑΡ10 δατασετ, σιμουλατιονς Β1 & "1 ("Αδθυστ=νο"). (γ) Ιμαγενεττε δατασετ, σιμουλατιονς Α2 & Δ1 ("Αδθυστ=ψεσ"). (δ) Ιμαγενεττε δατασετ, σιμουλατιονς Β1 & "1 ("Αδθυστ=νο").	81

7.2	Sensitivity plots using RetModel1 (Section 6.4.3) for different datasets (Table 6.1), different simulations (Table 6.5) and different combinations of parameters (Table 6.4). (a) MNIST dataset, simulations A2 & D1 ("Adjust=yes"). (b) MNIST dataset, simulations B1 & C1 ("Adjust=no"). (c) Fashion MNIST dataset, simulations A2 & D1 ("Adjust=yes"). (d) Fashion MNIST dataset, simulations B1 & C1 ("Adjust=no").	82
7.3	Comparison of the maximum performance of classifiers between different datasets (Table 6.1) for both retinal models (Section 6.4.3). (a) Comparison between CIFAR10 and Imagenette using RetModel1. (b) Comparison between CIFAR10 and Imagenette using RetModel2. (c) Comparison between MNIST and Fashion MNIST using RetModel1. (d) Comparison between MNIST and Fashion MNIST using RetModel2.	83
7.4	Comparison of the maximum performance of classifiers between simulations with "Adjust=yes" and those with "Adjust=no" (Table 6.4) for different datasets (Table 6.1) for both retinal models (Section 6.4.3). (a) Comparison for CIFAR10 using RetModel1. (b) Comparison for CIFAR10 using RetModel2. (c) Comparison for Imagenette using RetModel1. (d) Comparison for Imagenette using RetModel2.	84
7.5	Comparison of the maximum performance of classifiers between simulations with "Adjust=yes" and those with "Adjust=no" (Table 6.4) for different datasets (Table 6.1) for both retinal models (Section 6.4.3). (a) Comparison for MNIST using RetModel1. (b) Comparison for MNIST using RetModel2. (c) Comparison for Fashion MNIST using RetModel1. (d) Comparison for Fashion MNIST using RetModel2.	85
7.6	Comparison of the maximum performance of classifiers between RetModel1 and RetModel2 (Section 6.4.3) for all datasets. (a) Comparison in CIFAR10. (b) Comparison in Imagenette. (c) Comparison in MNIST. (d) Comparison in Fashion MNIST.	86
A.1	An example of a row in temporal dataset.	95
A.2	Graphs representing the firing rates of neuron_5 as a function of different numbers of images in each temporal row. For each diagram, we give different number of images as input to the model and we receive different number of outputs (i.e. firing rates). In Figure A2c, we plot the errorbar, as we have an input of five hundred images.	96

List of Tables

6.1	Description of the datasets used in simulations.	55
6.2	Summary of the CNN retinal model used in this project.	56
6.3	Steps of data preprocessing.	66
6.4	Range of values for the parameters of simulations. Datasets are described in Table 6.1. Other parameters are described in this section.	67
6.5	Five types of simulations for different preprocessing procedures. . .	68
6.6	Summary of all simulations per dataset. Datasets are described in Table 6.1. All types of simulations are described in Table 6.5.	69
7.1	Mean and standard deviation of classifiers performance in CIFAR10 dataset (Table 6.1) using RetModel1 (Section 6.4.3) for simulations A2, D1, B1, C1 (Tables 6.4, 6.5). Yellow cells in the Table below correspond to simulations with "Valid" parameter set to "yes", while the light blue cells correspond to simulations with "Valid=no".	73
7.2	Mean and standard deviation of classifiers performance in CIFAR10 dataset (Table 6.1) using RetModel2 (Section 6.4.3) for simulations A2, C1 (Tables 6.4, 6.5). Light blue cells in the Table below correspond to simulations with "Valid=no".	74
7.3	Mean and standard deviation of classifiers performance in MNIST dataset (Table 6.1) using RetModel1 (Section 6.4.3) for simulations A2, D1, B1, C1 (Tables 6.4, 6.5). Yellow cells in the Table below correspond to simulations with "Valid" parameter set to "yes", while the light blue cells correspond to simulations with "Valid=no".	75
7.4	Mean and standard deviation of classifiers performance in MNIST dataset (Table 6.1) using RetModel2 (Section 6.4.3) for simulations A2, C1 (Tables 6.4, 6.5). Light blue cells in the Table below correspond to simulations with "Valid=no".	76
7.5	Mean and standard deviation of classifiers performance in Fashion MNIST dataset (Table 6.1) using RetModel1 (Section 6.4.3) for simulations A2, C1 (Tables 6.4, 6.5). Light blue cells in the Table below correspond to simulations with "Valid=no".	77

7.6	Mean and standard deviation of classifiers performance in Fashion MNIST dataset (Table 6.1) using RetModel2 (Section 6.4.3) for simulations A2, C1 (Tables 6.4, 6.5). Light blue cells in the Table below correspond to simulations with <i>"Valid=no"</i>	77
7.7	Mean and standard deviation of classifiers performance in Imagenette dataset (Table 6.1) using RetModel1 (Section 6.4.3) for simulations A2, C1 (Tables 6.4, 6.5). Light blue cells in the Table below correspond to simulations with <i>"Valid=no"</i>	78
7.8	Mean and standard deviation of classifiers performance in Imagenette dataset (Table 6.1) using RetModel2 (Section 6.4.3) for simulations A2, C1 (Tables 6.4, 6.5). Light blue cells in the Table below correspond to simulations with <i>"Valid=no"</i>	78
A.1	Errorbars for eight different neurons.	97

Chapter **1**

Introduction

In this Diploma Thesis, we study the case of functional assessment of retinal models. Using machine learning techniques, we evaluate the performance of retinal models in image recognition tasks. Our aim is to propose an alternative to the currently common practice, as described in [18], [19], of comparing the similarity of model-generated and ground truth neural responses.

1.1 Motivation

Blindness, and vision disabilities in general, are among those problems that affect a large number of people worldwide (more than 40 million [20]) and at the same time, they greatly affect patient's daily life in various ways. Visual prostheses aim to tackle some of the most difficult of the visual disabilities and to do so, they depend on advanced retinal models, which translate the visual scene to retina neural spike patterns, mimicking the computations normally done by retina neural circuits [21]. The development of retinal models is also a great tool for elucidating mechanisms of vision in the human eye [22]. In both cases, there is imperative need to develop reliable methods for assessing the performance of different retinal models.

In this work, we propose a technique for functionally assessing retinal models. Functional assessment of retinal models describes the concept of evaluating the model's performance on image understanding tasks. This constitutes a divergence from the currently common practice of evaluating a model comparing the similarity of model-generated and ground-truth neural responses. Motivation for functional assessment stems from the observation that visual prostheses aim to restore the capacity of individuals to comprehend their visual environment, thus we should directly evaluate our models on such tasks. The need for functional assessment of prosthetic (i.e. acquired through prostheses) vision has been raised in the literature [23], and functional assessment has been applied to evaluate

vision in implantees [24] and in augmented-reality interventions in people with severe vision impairment [25]. The visual functions that seem most important to the blind are mobility, face recognition and reading [20].

A proposed way to implement functional assessment is to feed the retinal model's output to a neural network that solves image understanding tasks. In particular, we chose to implement object and digit recognition tasks using CIFAR10 [26], MNIST [27], Fashion_MNIST [28] and Imagenette [29]. To do so, we fed image data to the retinal model and then, we fed firing rates of retinal model to machine learning classifiers, such as MLP, SVM and Random Forest. Considering the limited number of available neurons provided by the retinal model, we split images in parts and then fed each part separately to the retinal model, so as to artificially increase the number of neural responses. On this basis, we designed and optimized the procedure of combining the neural responses produced by the model for each part. Furthermore, we evaluated the use of different datasets and classifiers in the performance of the functional assessment framework. In the end, we compared two different retinal models using functional assessment.

1.2 Thesis Contribution

Below, we summarize the main contributions of our work:

- We designed a novel functional assessment pipeline for retinal models.
- We applied functional assessment to compare the performance between two different retinal models. In this way, an alternative to standard assessment technique is provided.
- We fine-tuned parameters during data preprocessing in order to optimally fit a retinal model in functional tasks.
- We evaluated the performance of retinal models in object recognition tasks (CIFAR10, Fashion MNIST, Imagenette) and digit recognition tasks (MNIST). We found that models perform better with datasets such as MNIST and Fashion MNIST, where we achieved up to 90% accuracy.
- We evaluated the performance of different machine learning classifiers, when combined with retinal models. We found that Random Forest classifier achieves the highest performance.
- We found that using more neurons improves the overall performance of the model. Therefore, we highlighted the need for a new generation of improved

retinal implants, with increased number of implanted electrodes in order to achieve better performance in visual understanding tasks

1.3 Chapter Outline

In the first part, we describe the necessary theoretical background in order for the reader to understand the structure and the contribution of this thesis. In particular, in Chapter 2, we present information regarding the physiology of the human eye and mainly of the retina. In Chapter 3, we describe the mathematical background needed for understanding how firing rates are computed. In Chapter 4, we describe the concepts of retinal models and retinal prosthesis and we provide relevant examples. In Chapter 5, we present an overview of machine learning theory, focusing on the models that we used in this project.

In the second part, we describe in detail the simulations (in silico experiments) that we conducted. More specifically, in Chapter 6, we describe the methodology used and the simulations conducted in order to develop a pipeline for functional assessment of retinal models. In Chapter 7, we analyze the results of the simulations and we present comparative tables in order to illustrate the importance of the results. In Chapter 8, we further discuss the findings of the simulations.

Finally, in Chapter 9, we present a summary of this thesis, as well as future directions for improving functional assessment of retinal models.

Chapter 2

Physiology of the Eye

In this chapter, we describe the theoretical background and all necessary information in order for the reader to understand the structure and the contribution of this thesis. First, we present the physiology of the eye and the retina and we analyze the structure of artificial retinal models. Secondly, we provide a brief overview of visual prostheses, which constitute the broader research area of this work. Thirdly, we give a brief explanation of functional assessment accompanied by relevant examples. Finally, we present the concept of machine learning and neural networks, emphasizing on the models that are used in this thesis.

2.1 Structure of the Human Eye

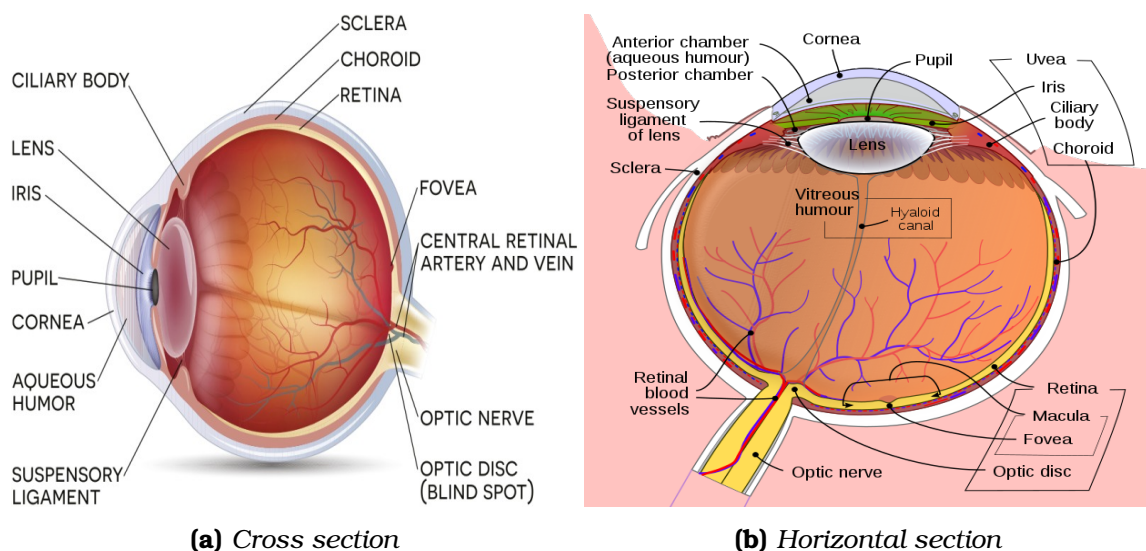


Figure 2.1. Cross and horizontal section of the human eye [1], [2].

The eye together with the brain visual centers constitute the visual system of vertebrates. The phrase “the eye sees an image” describes the procedure when the eye gathers patterns of light coming from the outside world and converts them

through retina into neural signals. The processing of these neural signals takes place in the brain's visual centers, which are responsible for extracting the perceived visual information [4].

Figure 2.1 shows a schematic representation of a cross and a horizontal section of the human eye with some of the main parts labeled. The most widely recognized parts of the human eye are iris and pupil. Pupil is the dark opening in the center of the eye. It appears to be dark due to the pigments in the rear of the retina which absorb light. Iris is the colored muscle wrapped around the pupil and controls the size of the pupil according to the light conditions. The external part of the eye is covered with a transparent membrane, called cornea. Lens, a transparent tissue behind the iris, together with cornea, are the two lenses of the eye which allow the formation and focus of a sharp image in the rear of the eye. The place on the back of the eye, where the image is focalised, is called retina. The retina is also in charge of converting the light energy into neural signals and then, convey them to the brain through the optic nerve. It is important to point out that all previously described parts of the eye are responsible only for focusing the perceived image on the retina and the actual neural processing starts at the retina. Lastly, in the center of the retina lies a region called macula and in the very center of the macula, there is another region called fovea. Macula and particularly fovea are responsible for the central, high-resolution, color vision under good light conditions and provide humans with the ability to read and see in great detail, whereas the rest of the retina provides peripheral vision [4], [2], [30]

2.2 The Retina

The retina is one of the most significant components of the eye and covers around 65% of the inside eyeball [4]. Its function is to convert light into electrical nervous pulses and transfer them from the eye to the brain. Without the retina, visual perception of the environment and visually-dependent behaviors would be impossible [30]. In this section, we briefly describe the basic components of retina, as well as retina's main functionalities and cell types.

The retina is composed of five major classes of neurons: photoreceptors (rods and cones), horizontal, bipolar, amacrine and retinal ganglion cells (RGCs). These cells are clustered in a layered structure that can be easily distinguished in Figure 2.3. The three main layers are: the outer nuclear layer (ONL), consisting of the photoreceptor cells, the inner nuclear layer (INL), containing bipolar, horizontal and amacrine cells and the ganglion cell layer (GCL), containing the ganglion cells. Between the ONL and the INL, there is the outer plexiform layer (OPL),

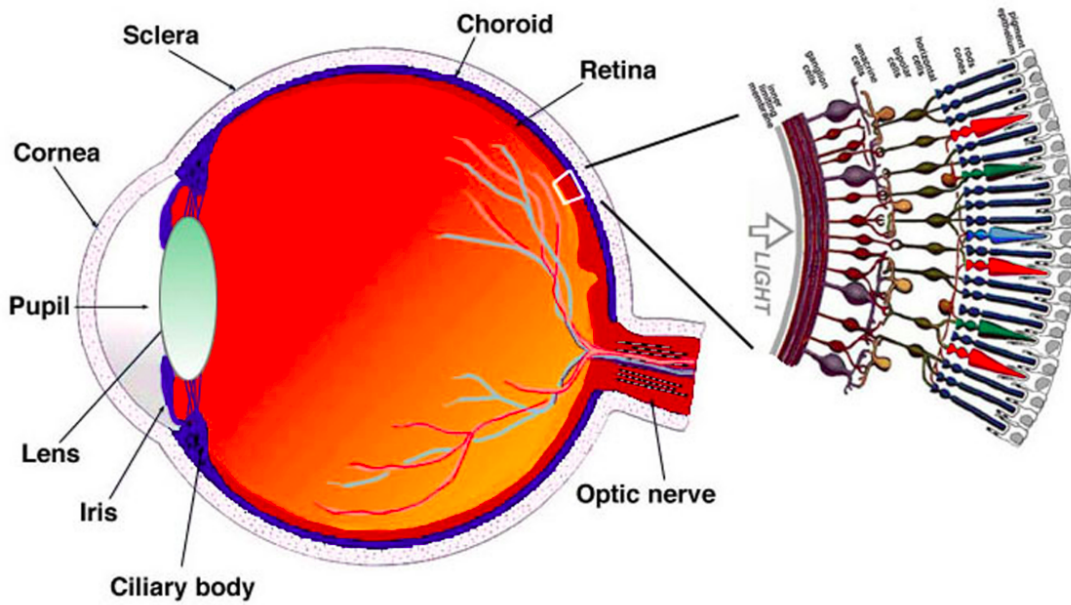


Figure 2.2. A drawing of a section through the human eye with a schematic enlargement of the retina [3].

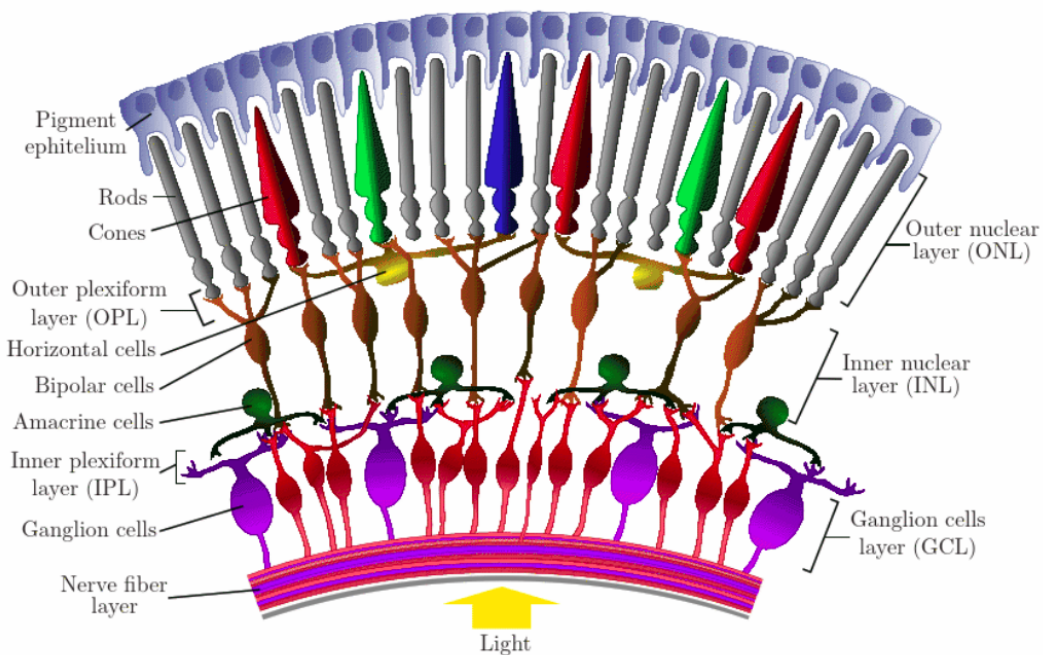


Figure 2.3. Simplified schematic representation of the retina [4].

which contains the synaptic connections between the photoreceptors, bipolar, and horizontal cells. Between the INL and the GCL, there is the inner plexiform layer (IPL), containing the synaptic connections between the ganglion cells and amacrine cells [4], [31]. The ganglion cells, whose axons form the optic nerve, are at the front of the retina; therefore the optic nerve must cross through the retina in order to connect the eye with the brain. The point where the optic nerve

traverses the retina is called the blind spot (or optic disk), since there are no photoreceptors in this region, as shown in Figure 2.1. The existence of the blind spot is not perceived, because the blind spot of one eye is compensated by the other eye, and vice-versa [4], [30]. It should also be noted that the vertebrate retina is inverted in the sense that the light has to traverse all retinal layers to be sensed by the photoreceptors, as shown in Figure 2.3. The exception to this rule occurs in the fovea, where nerves are pushed away so that the cones are directly exposed to light [4].

Before analyzing the types and functionalities of each group of retina cells, let us give a brief overview of how retina operates. At first, the photoreceptors detect changes in illumination and colour of the visual environment by absorbing photons and converting them to an electrical signal capable of stimulating the forward neuronal layers of the retina. When the electrical stimuli arrive at the GCL they are conveyed to the brain through the optic nerve in the form of action potentials (or *spikes*). The layers between photoreceptors and retinal ganglion cells, namely the bipolar, horizontal and amacrine cells, are responsible for processing the electrical stimuli coming from the different photoreceptors and creating the diverse encodings of the visual world that the retina transmits to the brain [4], [32].

For having a better understanding of the retinal procedures, we will present below information for all cell types of the retina.

2.2.1 Photoreceptor Cells: Rods and Cones

Rods and cones are the two types of photoreceptors in the human retina. They transform light into electrical signals and synapse onto bipolar and horizontal cells. The rods (around 1200 million in number) are long and thin, while the cones (around 7 million) have a shorter and thicker conical shape. The rods are very sensitive to light (they can even be excited by a single photon), they enable *scotopic vision* (vision under lowlight conditions), they do not provide color information and can be found scattered everywhere in the peripheral retina, but not in the fovea. In contrast to rods, cone photoreceptors are responsible for color perception, they respond mainly at medium to high light levels, providing the eye's *photopic vision* and they are almost exclusively concentrated in the fovea. There are three different types of cones in the human retina: the blue, green and red cones, corresponding to the visible light wavelength to which they respond. The photoreceptors get their vital energy from the pigment epithelium, which lies outside of them in the deepest layer of the retina [4], [30], [31].

2.2.2 Horizontal Cells

Most mammals have two types of horizontal cells: type A and type B cells. Type A cells are associated with the cones, receiving an excitatory synaptic input from the cones and feed an inhibitory output back to them. Type B cells have two separate dendritic arbors, one interacting with cones and the other interacting with the synaptic terminals of the rods. Horizontal cells are in charge of preserving details of different objects in an image, even if they are under different lighting conditions. Their wide lateral spread across the photoreceptors permits the detection and sharpening of the edges between different regions of the image. In this way, the horizontal cells modulate the photoreceptors' signals under different lighting conditions, allowing signaling to become less sensitive in bright light and more sensitive in dim light [4], [32], [33].

2.2.3 Bipolar Cells

Bipolar cells transmit photoreceptor signals to retinal ganglion cells. There are approximately 12 types of bipolar cells, receiving synaptic input from either cones, rods or both. Exact types of bipolar cells have not yet been decrypted by scientists. Distinctive types of bipolar cells express distinctive sets of receptors, ion channels and intracellular signaling systems and thus, each of them transmit to the inner retina a different component extracted from the output of the photoreceptor. These differences among the types of bipolar cells could be attributed to many reasons. In the case of ON-type and OFF-type bipolar cells, some cells' receptors respond to glutamate by activating an OFF pathway, so that they detect dark areas on a lighter background, while others activate an ON pathway, and the cell detects light areas on a darker background. In the case of sustained and transient bipolar cells, the distinction is caused by the expression of rapidly or slowly inactivating glutamate receptors. There is also a structural difference that defines the 12 types of bipolar cells, which is the level of the IPL at which their axons terminate. While the bipolar cells receive input from all of the cones within their reach, they terminate on very different regions [4], [32], [33].

2.2.4 Amacrine Cells

Amacrine cells lie at the bottom of the IPL, just above the GCL. They are the most diverse population of interneurons in the retina and only a small percent of all types and functionalities of amacrine cells has been identified. Among the known types, the most dominant one is Amacrine AIII, a narrow-field cell receiving information from the bipolar cells and transmitting it to ganglion, bipolar and amacrine cells. Amacrine AIII also provides a link between ON and OFF systems

of bipolar and ganglion cells. Another special type of amacrine cell is called A17. A17 is a widely spreading neuron, which improves the transmission of information by the rod bipolar cells. There are several other types of amacrine cells that spread horizontally, interacting with hundreds of bipolar cells and many ganglion cells. They can even connect to neighbouring amacrine cells, increasing the speed with which signals can be transmitted across large areas of the retina. Another role played by the amacrine cells is in modulating the retinal response for different illuminance conditions. By liberating several kinds of neurotransmitters, they inhibit or reinforce the synaptic connections between the neuronal layers in the retina [4], [32], [33].

2.2.5 Retinal Ganglion Cells

Retinal ganglion cells (RGCs) constitute the very last layer of retinal cells, which transmits through the optic nerve visual information, in the form of spatiotemporal patterns of action potentials (or spikes) to the brain's visual regions. The number of ganglion cell types is somewhere in the neighborhood of 30 [34]. They vary significantly in terms of their size, connections, and responses to visual stimulation but they all share the defining property of having a distinct dendritic structure, with a long axon that extends into the brain. There are about 0.7 to 1.5 million retinal ganglion cells in the human retina [4], [31], [32], [33].

2.3 Receptive Field of Retinal Ganglion Cells

The concept of the receptive field (RF) is considered fundamental to neuroscience and is the standard method when studying the relationship between a cell's response and the stimulus that causes it. In the visual system, a receptive field of a cell defines the area of the visual field in which, if a stimulus or a specific change is detected, the cell will be stimulated. In the case of retina, the shape and size of the receptive field differs in the different areas of the retina. Photoreceptors react to photons directly falling over them, so they have narrow receptive fields. Bipolar cells have a medium-sized receptive field as they are associated with a few cones and horizontal cells have a larger one as they are activated by many cones.

Retinal ganglion cells have a receptive field that can be characterized by the area on which light influences the neural response and is organized based on "*center-surround organization*", as defined by Kuffler [35]. More specifically, the receptive field of ganglion cells consists of two concentric elliptical structures, with the parts being either ON-center or OFF-center, and antagonistic between them (Figure 2.4). The ON-center ganglion cells become active when a spot of light

Receptive Fields

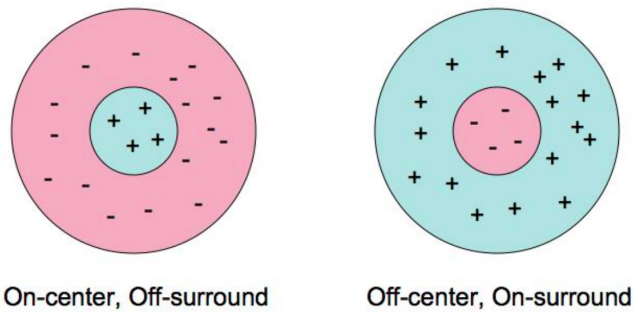


Figure 2.4. Schematic representation of the receptive field of an ON-type cell (left) and of an OFF-type cell (right) [5].

falls in the center of their receptive field and are inactivated when light falls on the field periphery. The OFF-center ganglion cells act in the opposite way; their activity increases when the periphery of their receptive field is lit and decreases when light falls on the center of the field. The receptive fields of the ganglion cells can be modeled as the difference of Gaussian functions (Figure 2.5). The receptive field of an OFF-center cell would have a symmetric shape relative to the one in Figure 2.5. The competition of the two regions allows the neurons to detect differences in brightness between points, i.e. changes in contrast. In other words, neurons are only interested in changes, a function related to edge detection in image processing [4], [7], [36].

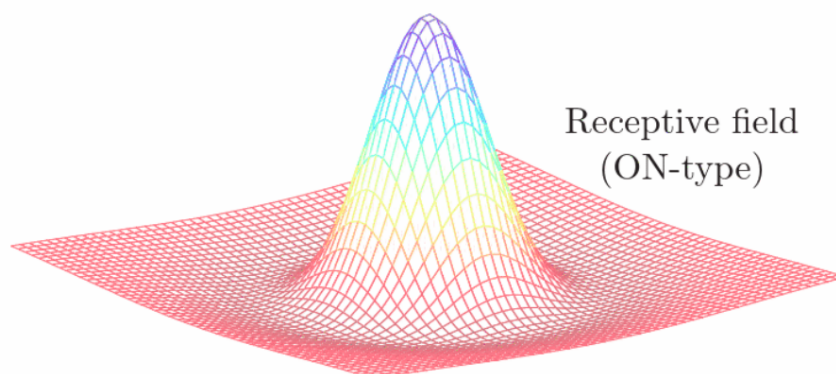


Figure 2.5. Receptive field of an ON-type ganglion cell represented as difference of Gaussians [4].

Chapter 3

Spike Trains and Firing Rates

3.1 Action Potential and Spike Trains

Action potentials are signals generated by neurons and play a central role in cell-to-cell communication, because they are the only form of membrane potential fluctuation that can convey information over large distances. Action potentials in neurons are also called *nerve impulses* or *spikes*, and the temporal sequence of action potentials is called *spike train*. The procedure when a neuron emits an action potential, is called *firing* [7], [37].

Given the small duration ($\approx 1\text{ms}$) of an action potential (or spike), we can represent the sequence of action potentials (or spike trains) as a sum of Dirac δ functions:

$$\rho(t) = \sum_{i=1}^n \delta(t - t_i), \quad (3.1)$$

where n is the total number of spikes in the spike train and t_i are the times that each spike occurred. We can $\rho(t)$ the *neural response function* and we use it to re-express sums over spikes as integrals over time. For example, we can write the spike count, the number of spikes fired between times t_1 and t_2 as the integral:

$$n = \int_{t_1}^{t_2} \rho(t) dt, \quad (3.2)$$

because each spike contributes 1 to the integral.

3.2 Types of Firing Rates

Because the sequence of action potentials generated by a given stimulus typically varies from trial to trial, neuronal responses are typically treated as random

processes, and characterized, for example, by the probability that a spike occurs at a particular time during a trial. We can write the probability that a spike occurs within a specified interval, for example the interval between times t and $t + \Delta t$ for small Δt , as $p[t]\Delta t$, where $p[t]$ is the single spike probability density. The probability density for the occurrence of a spike is, by definition, the firing rate of the cell, and we use the notation $p[t] = r(t)$ to represent this quantity. The firing rate however, is commonly used to define three different quantities: $r(t)$, r and $\langle r \rangle$. We use the terms firing rate, spike-count rate and average firing rate for $r(t)$, r and $\langle r \rangle$ respectively.

The **firing rate** at time t , $r(t)$, can be estimated by determining the fraction of trials with a given stimulus on which a spike occurred between the times t and $t + \Delta t$. For sufficiently small Δt and sufficiently large numbers of trials, this fraction provides a good estimate of $r(t)$, as guaranteed by the law of large numbers. The trial-averaged neural response function is thus denoted by $\langle \rho(t) \rangle$ and the instantaneous firing rate can now be formally defined as:

$$r(t) = \langle \rho(t) \rangle, \quad (3.3)$$

The spike-count rate is obtained simply by counting the number of action potentials that appear during a trial and dividing by the duration of the trial. Unlike $r(t)$, the spike-count rate can be determined for a single trial. We denote the spike-count rate by r where:

$$r = \frac{n}{T} = \frac{1}{T} \int_0^T \rho(t) dt, \quad (3.4)$$

The second equality follows from the fact that:

$$\int_0^T \rho(t) dt = n, \quad (3.5)$$

and indicates that the spike-count rate is the time average of the neural response function over the duration of the trial.

In the same way that the response function $\rho(t)$ can be averaged across trials to give the firing rate $r(t)$, the spike-count firing rate can be averaged over trials yielding a quantity that we refer to as the **average firing rate**. This is denoted

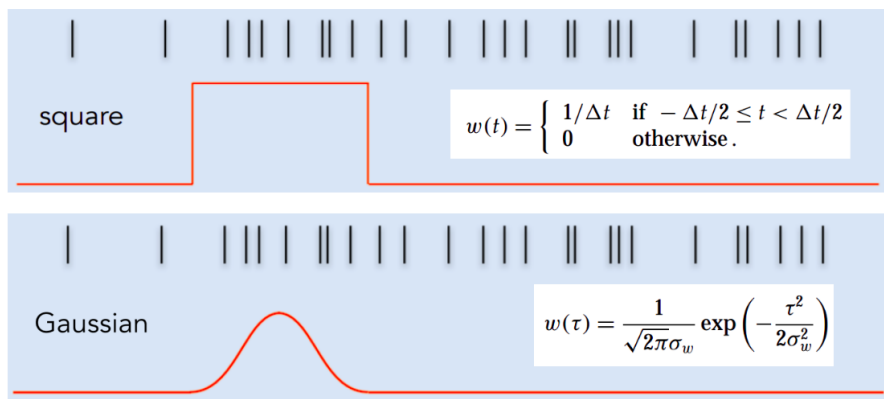


Figure 3.2. Estimate firing rate by using a square (top) or Gaussian (bottom) window. Adapted from [6].

Figure 3.3B shows the firing rate estimation computed using bin splitting with a bin size of 100ms. Figure 3.3C shows firing rate estimation determined by sliding a square window function along the spike train with $t = 100\text{ms}$ and Figure 3.3D shows firing rate estimation computed using a Gaussian window function with $\sigma_t = 100\text{ms}$.

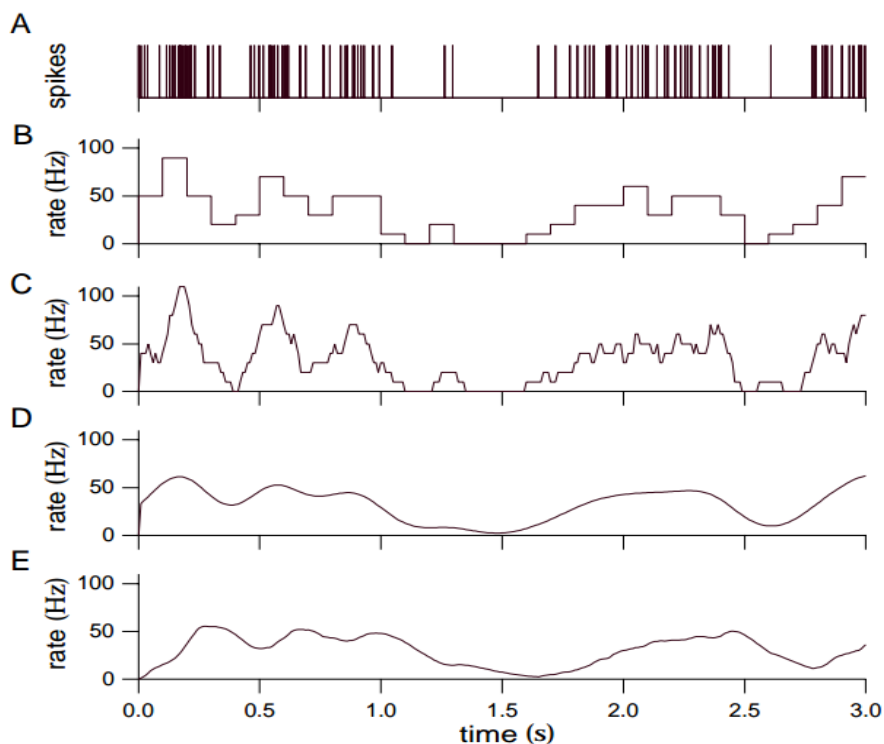


Figure 3.3. Different approaches to estimate the firing rate. Figure 3.3A shows a spike train from a neuron. Figure 3.3B shows the firing rate estimation computed using bin splitting with a bin size of 100ms. Figure 3.3C shows firing rate estimation determined by sliding a square window function along the spike train with $t = 100\text{ms}$ and Figure 3.3D shows firing rate estimation computed using a Gaussian window function with $\sigma_t = 100\text{ms}$ [7].

Chapter 4

Retinal Models and Retinal Prosthesis

In this section, we present an overview of retinal models, analyzing the different types of models and presenting their properties. Given the strong relationship between retinal models and retinal prosthesis, the concept and the types of retinal implants are also discussed.

4.1 Retinal Models

Sensory neuroscience designs neural models that predict the response of a sensory area to a stimulus in order to unveil unknown circuit computations and neural mechanisms. In the case of retina, several retinal ganglion cell spiking models have been proposed, such as Linear Nonlinear (LN) models [39], [8], Generalized Linear Models (GLMs) [40], Integrate-and-Fire models [41] and more recently Deep Learning (DL) models, using mainly Convolutional Neural Networks (CNNs) [19], [18].

LN models use a linear spatiotemporal filter, modelling the function of photoreceptors, combined with a simple static nonlinearity, describing the instantaneous relationship between the filtered stimulus and the response. As shown in Figure 4.1, the stimulus waveform $s(t)$ is passed through a linear temporal filter $F(t)$, and the result $g(t)$ is transformed by a nonlinear function $N(g)$ to the model's response $r'(t)$. LN model components could be seen as follows. The time-reverse of the filter function $F(t)$ represents the stimulus feature to which the neuron shows the highest sensitivity. The filtered stimulus $g(t)$ measures how strongly that feature is represented in the current stimulus, and the function $N(g)$ determines how $g(t)$ is transformed into a response, including threshold effects, rectification, and other distortions [19], [8].

GLM models (Figure 4.2) consist of the following components: linear spa-

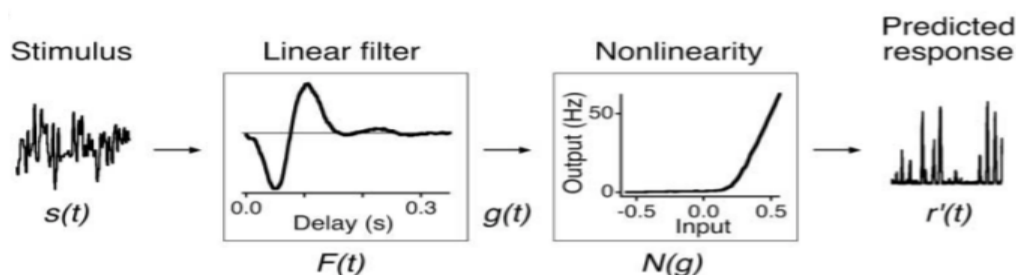


Figure 4.1. The Linear - Nonlinear (LN) model [8].

tiotemporal filtering of the visual input, exponential nonlinearity, Poisson spike generation, and a feedback waveform for each spike generated that sums with the post-filter signal. The nonlinearity after the linear filter, captures the nonlinear intensity-response relationship of the retinal ganglion cells and the post-spike feedback models the non-Poisson structure in retinal ganglion cells spike trains, such as bursting and refractoriness, that are not captured by simpler models [9].

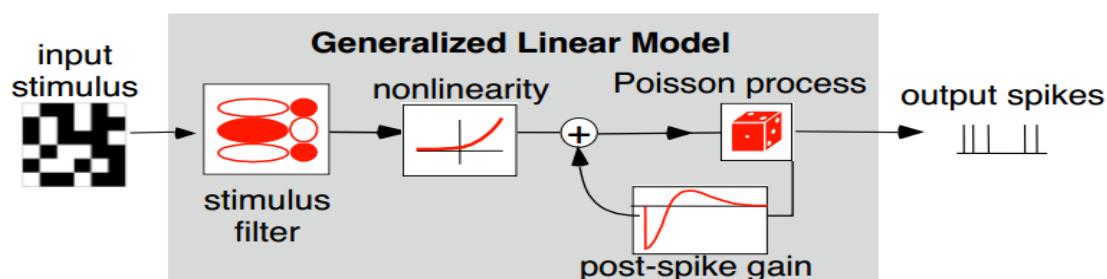


Figure 4.2. The Generalized Linear Model (GLM) [9].

Although they produce satisfying results when used with white noise stimulus, both LN models and GLMs fail to model responses to natural stimuli and thus, fail to model nonlinear behavior of the retina [19], [9]. However, retinal models using CNNs were able to overcome this problem and outperform previous models. CNNs, having a structure that imitates biological vision and being very successful in visual recognition tasks, managed to successfully model retinal responses in both artificial (such as spatiotemporal white noise) and natural (such as natural images) stimuli [18]. CNN can also model a wide range of neurons and they can do that simultaneously for many types of neurons. Furthermore, CNN retinal models present properties similar to retinal functionalities, as individual components of a CNN retinal model are highly correlated with actual retinal interneuron responses that were recorded separately and never presented to the model during training. Therefore, CNN retinal models can be used not only as accurate retinal models, but also as means of studying unknown retinal mechanisms [17], [9], [42].

4.2 Retinal Prosthesis

While it has been proved that retinal models are a great tool for exploring novel retinal functionalities, their development is closely associated with visual neuro-prostheses as well. The increasingly deeper knowledge of visual systems along with the evolving technology, give novel results in prevention, limitation or even treatment of eye diseases [19], [18]. Although some of the eye diseases can be effectively treated with surgery or medication, there are cases of severe degeneration or damage to the retina where retinal prosthesis is the only choice available [20]. Such cases are Age related Macular Degeneration (AMD) and Retinitis Pigmentosa (RP), where there are currently no treatments able to cure them. In both cases, the patient's photoreceptor cells have degenerated, but retinal nerve cells and optic nerve are still functional. Retinitis pigmentosa is a genetic disease, where patients live with a gradually shrinking visual field, then lose their eyesight, generally after the age of 50. AMD is an age-related disease, where the central visual field is affected and makes it difficult to read or recognize faces [43]. There are several devices under evaluation and some of them are commercially available, such as Argus II (Second Sight Medical Products Inc., Sylmar, CA, USA), which is the most widely used retinal prosthesis worldwide [36], IRIS II (Pixium Vision, France) and Alpha IMS (Retina Implant AG, Reutlingen, Germany) [44].

Most retinal prostheses share some basic components: an image capturing device, an image processing unit, a stimulator chip and an electrode array (Figure 4.3). The image capturing device captures images from the visual environment, which are transformed into stimulation patterns for the multi-electrode array. Visual information is either selected with a camera in a camera-electrode based system, or directly detected by the electrode-coupled photodiodes to generate electrical stimuli, in case of photodiode based systems. When necessary, radio frequency (RF) telemetry and/or optical link is used for the power and data transmission between the external part of the device and the implant. Other important components that can be found in most implants are hermetic packaging and wireless data/power transmission.

Despite the similarities in their architecture and the differences in technical details, what characterizes the retinal prostheses is the location where an electrode array can be implanted to stimulate the retina. There are three main types of retinal implants: epiretinal, subretinal and suprachoroidal (Figure 4.4) [11], [10]. **Epiretinal implants** are placed on top of the retinal surface, directly stimulating RGCs and inner retinal neurons. An array of electrodes is stabilized on the inner retina adjacent to the RGCs and the axon fibres, using micro tacks which pene-

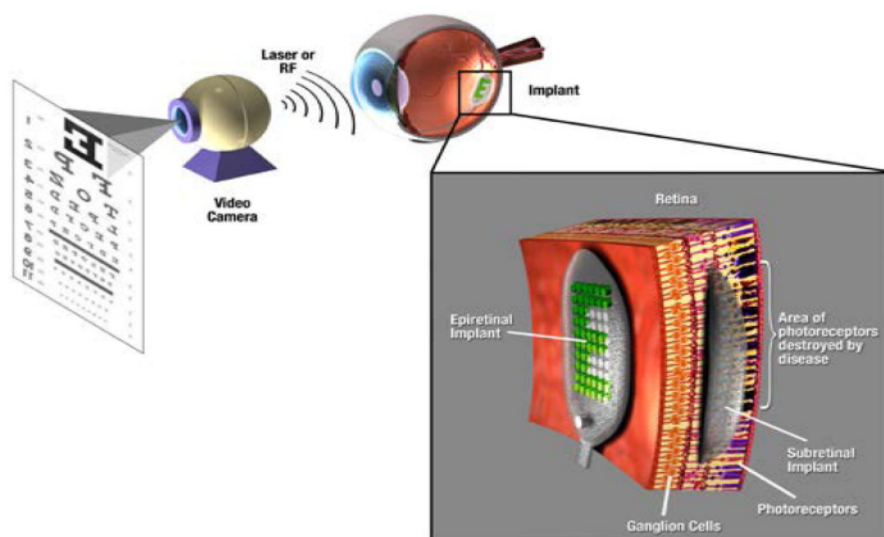


Figure 4.3. An overview of a retinal prosthesis system. An image is captured by a video camera and converted to an electrical signal by an imaging processing unit. The electrical signal is transmitted through radio frequency (RF) telemetry and/or optical link and processed by external and/or implanted circuitry. The circuit produces a pattern of electrical stimulus, similar to light stimulus, which is applied to the retina via a microelectrode array positioned near the retina. Adapted from [10].

trate into the sclera. Epiretinal prostheses are minimally invasive to the retina, as the location where the prosthesis contacts the retina is easily accessible and in this way, the risk of mechanical damage to the retina during surgery is diminished. They are also easily monitored through the lens of the patient, using a fundusopic or optical coherence tomography (OCT). However, epiretinal implants have a difficulty in fixating the electrode array in the retina and sometimes they present distorted visual perception [11], [10], [45].

Subretinal implants are placed in the subretinal space between the pigment epithelium and the degenerate photoreceptor layer. They stimulate the outer and middle sections of retina (bipolar and amacrine cells) of the blind patient, exploiting the existing neural processing and possibly avoiding the direct stimulation of RGGs, which provokes distorted visual perception. However, subretinal chips may subretinal implantation may block the fluid communication between the retina and the choroid, obstructing oxygen and nutrients diffusion to the retina [11], [45].

In **suprachoroidal prostheses**, electronics are implanted outside of the eye and electrode arrays are passed through the sclera and are fixated between the choroid and the sclera. Suprachoroidal implants are relatively distant from the

retina and this offers several advantages. Among them, minimal surgery is needed for electrode array implantation, reducing in this way the risk of retinal detachment and the risk of infection. The blockage of retinal nourishment is also reduced, as there is no barrier between the retina and the choroid, and the neural tissue is safely stimulated. On the other hand, the longer path from the electrode to the targeted tissue might cause worse spatial resolution [11], [10], [45].

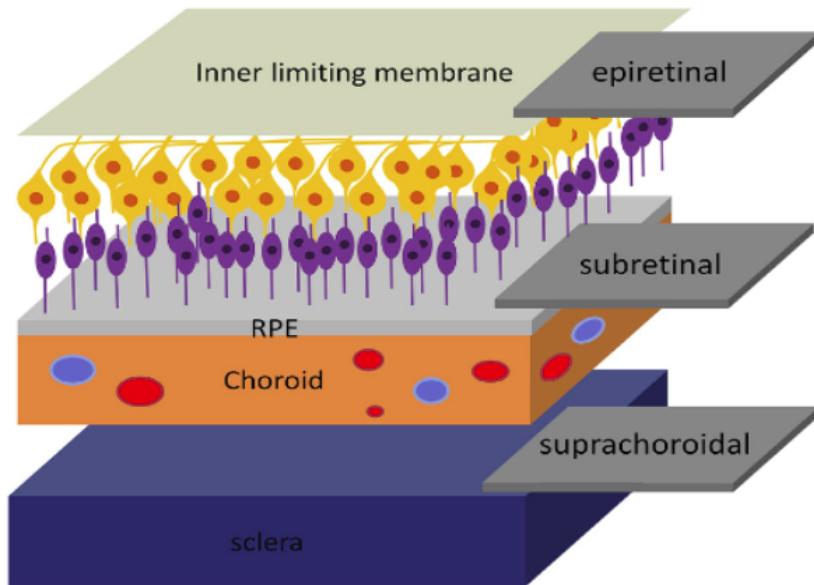


Figure 4.4. Three types of retinal prostheses: epiretinal, subretinal and suprachoroidal. Ganglion cells (yellow) and bipolar cells (purple) are shown and damaged/eliminated photoreceptors are not shown [11].

Chapter **5**

Machine Learning

5.1 Introduction

Machine Learning (ML) is the subfield of Artificial Intelligence (AI). Machine Learning describes the concept of automatically uncovering patterns in data and use them in order to extract useful knowledge from data or predict future data [46]. Another, more generic, definition of Machine Learning was given by Mitchell [47]. He defined that:

*"A computer program is said to **learn** from experience E with respect to some class of tasks T and performance measure P . If its performance at tasks in T , as measured by P , improves with experience E ."*

Based on these definitions, we can define the core components of Machine Learning [48]:

1. The *data* we can learn from.
2. A *model* of how to transform the data.
3. An *objective function* that evaluates how well (or badly) the model is doing.
4. An *algorithm* to adjust the model's parameters to optimize the objective function.

5.2 Types of Machine Learning

Machine Learning algorithms are split in three main categories: Supervised Learning, Unsupervised Learning and Reinforcement Learning.

5.2.1 Supervised Learning

Supervised Learning uses datasets containing an input set X and an output set y . X consists of a set of features for each example, while y consists of a specific label or target. The goal is to learn a function f , which maps input X to input y .

$$X \rightarrow y \text{ or } y = f(X) \quad (5.1)$$

Ultimately, the Supervised Learning model can assign the correct label to new, unlabeled input data, never experienced before. Different types of problems solved by Supervised Learning depend on the type of the output y . In Classification problems, y is a discrete variable representing a class, while in Regression problems y is a continuous variable. More specifically, classification models predict the class for each input, while regression models predict a continuous numerical value as output [48], [49]. Part of Supervised Learning algorithms are Support Vector Machines (SVM), Random Forests and Artificial Neural Networks (ANN) that we used in this work and we discuss below.

5.2.2 Unsupervised Learning

Unlike previous category, datasets in Unsupervised Learning do not contain labels for each input data. The goal is to learn the entire probability distribution of the dataset or properties and patterns that govern given data. Unsupervised Learning mainly solves Clustering problems, where the goal is to separate input data in different cluster based on a given metric [49].

5.2.3 Reinforcement Learning

Reinforcement Learning differs from previous categories, as it focuses on optimal decision making. Reinforcement Learning models contain an *agent*, which interacts with an *environment* over time. The agent receives some *observation* from the environment, performs an action and then, receives a *reward* from the environment. This process continues in a loop. The behavior of the agent is governed by a *policy*, which is a function that maps the observations of the environment to actions. The goal of reinforcement learning algorithm is to produce a good policy [48].

5.3 Machine Learning Algorithms

5.3.1 Decision Trees

Decision trees [50] belong to the family of Supervised Learning algorithms and more specifically, they are nonlinear classifiers. The main components of a decision tree model are *nodes* and *branches* and the most important elements in designing decision trees are *splitting criterion*, *stopping rule* and *pruning*. There are three types of nodes: the *root node*, which represents a choice that will result in dividing all records into two or more mutually exclusive subsets, *internal nodes*, representing one of the possible choices available at that point in the tree structure and the *leaf nodes*, which represent the final result of a combination of decisions. A decision tree model is formed using a hierarchy of branches. Each path from the root node through internal nodes to a leaf node represents a classification decision rule. A *splitting criterion* refers to the selection of the variable which best splits the data at a specific point, using some metric, such as entropy, Gini index, classification error, information gain, gain ratio, and twoing criteria [51]. Every split of a node, generates two or more descendant nodes. The splitting procedure continues until stopping criteria are met. Stopping criteria are selected in order to avoid overfitting in training data. However, there are cases where stopping rules do not work well. In such cases, we can let a decision tree grow first and then prune the nodes that provide less important information [52], [12], [46].

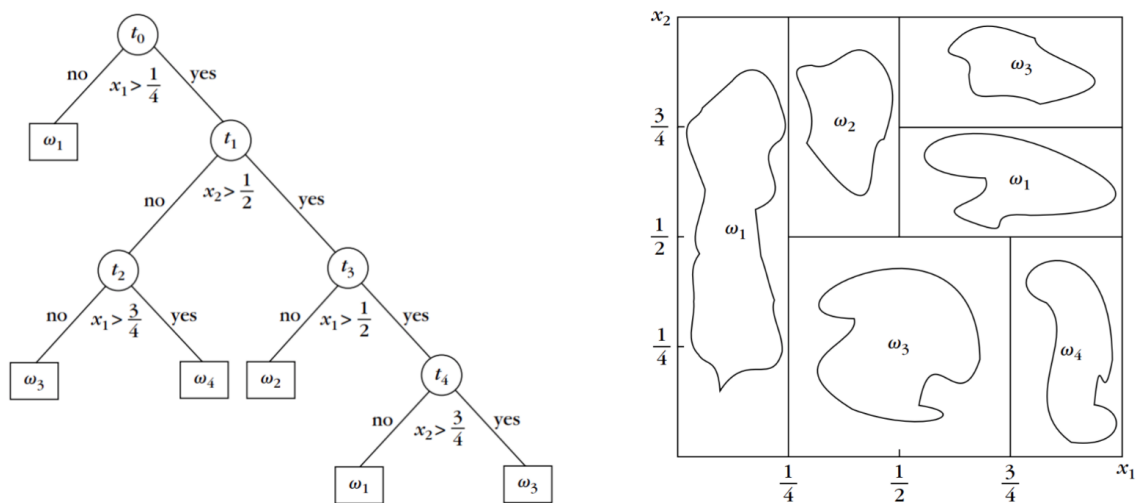


Figure 5.1. Decision tree classification (left). Illustration of decision tree as segmented space (right) [12].

5.3.2 Random Forest

A very common technique in Supervised Learning, called Ensemble Learning, is used to train multiple classifiers, independently of each other, and in the end, combine their predictions, so as to make a final prediction. This technique yields better results than using a single classifier, as it takes advantage of independent choices of different classifiers, where each of them can learn different parameters of the problem. Different predictions from different classifiers are combined using a correlation function, such as *arithmetic average rule*, *geometric average rule* and *majority voting rule* [12].

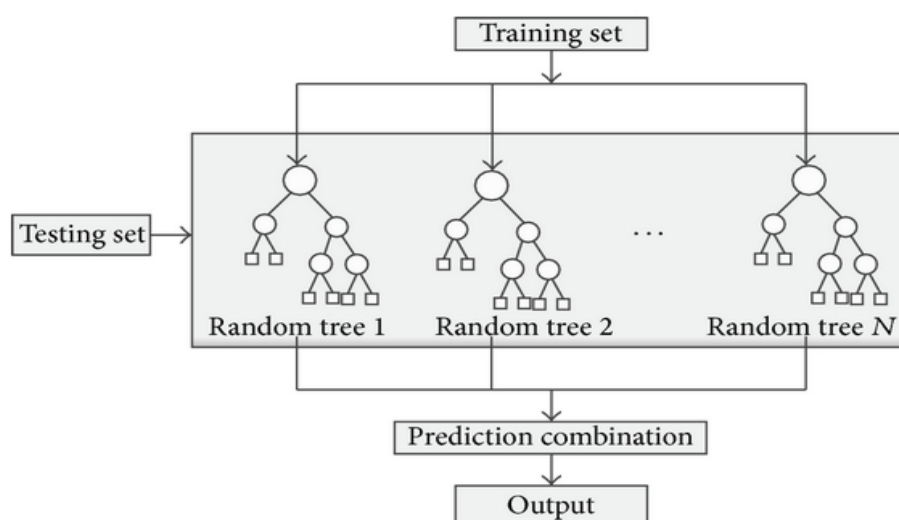


Figure 5.2. Conceptual schema of Random Forest algorithm [13].

Random Forests implement the aforementioned technique and in particular, they consist of a number of Decision Trees (or *base learners*), which are parallel and independent of each other (Figure 5.2). The idea is to train M different trees on different subsets of the data, chosen randomly with replacement, and then compute the ensemble

$$f(x) = \sum_{m=1}^M \frac{1}{M} f_m(x), \quad (5.2)$$

where f_m is the m^{th} tree. This technique is called *bagging* [53], which stands for "*bootstrap aggregating*". However, if we run many times the same learning algorithm on different subsets of the data, this results in highly correlated predictors. Random Forest, as proposed by Breiman [54], overcomes this problem by decorrelating the base learners. This happens by learning trees based on a randomly chosen subset of input variables, as well as a randomly chosen subset

of data features. Such models are widely used, as they often perform really well.

5.3.3 Support Vector Machines (SVM)

Support Vector Machines (SVM) is one of the most fundamental kernel-based algorithms of Supervised Learning. Based on the spatial representation of training samples, "the support vector machine constructs a hyperplane as the decision surface in such a way that the margin of separation between positive and negative examples is maximized" [15]. Input data are categorized in two classes: *linearly separable* or *non-linearly separable* data, as shown in Figure 5.3. Linearly separable classes can be separated by SVMs, while non-linearly separable classes require the use of nonlinear SVMs, which use techniques for transforming data coordinates into a higher dimension space and then, calculate hyperplanes that separate the data.

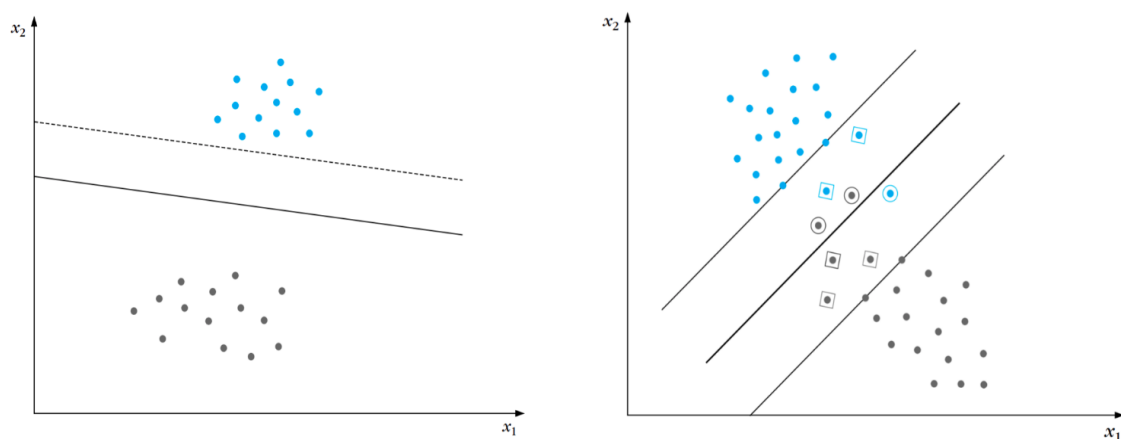


Figure 5.3. An example of one linearly (left) and one non-linearly (right) separable two-class problem [12].

Suppose we have two linearly separable classes and we want to design a linear classifier. Our goal is to design a hyperplane of the following form:

$$f(x) = w^T x + b, \quad (5.3)$$

where $x \in R^d$ represents the training data and $w, b \in R^d$ represent the parameters of the classifier. We also denote as $y \in \{-1, 1\}$ the labels of the two classes.

There is an infinite number of hyperplanes that can successfully classify all training data x . However, the selection of the hyperplane should be made in order for the classifier to generalize well in data outside the training set. The optimal hyperplane is the one that maximizes the margin between the two classes. At this point It should be also noted that the distance of a point x from the hyperplane is

given by the relation:

$$z = \frac{|f(x)|}{\|w\|}, \quad (5.4)$$

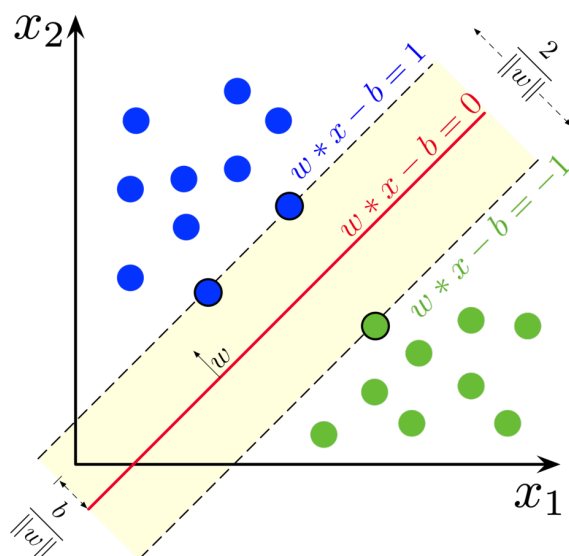


Figure 5.4. Maximum-margin hyperplane and margins for an SVM trained with samples from two classes (Wikipedia).

Given that the two classes are linearly separable, we can select parameters w and b , such that:

$$\begin{aligned} f(x) &> 0, & \text{for } y = +1 \\ f(x) &< 0, & \text{for } y = -1 \end{aligned} \quad (5.5)$$

By scaling w , b so that the value of $f(x)$ at the nearest points of the two classes is equal to 1 and -1 respectively for each class. With this requirement the maximum margin length becomes $\frac{2}{\|w\|}$ and we have:

$$\begin{aligned} f(x) &> 1, & \text{for } y = +1 \\ f(x) &< -1, & \text{for } y = -1 \end{aligned} \quad (5.6)$$

Our task now is to compute the parameters w , b of the hyperplane, so that to:

$$\text{minimize } J(w, b) = \frac{1}{2} \|w\|^2 \quad (5.7a)$$

$$\text{subject to } y_i(w^T x_i + b) \geq 1, \quad i = 1, 2, \dots, N \quad (5.7b)$$

This is a nonlinear (quadratic) optimization task subject to a set of linear inequality constraints [12], [15].

If classes are non-linearly separable the data are subject to transformations in higher dimensional spaces. The transformations are called kernels and some widely used kernels are [46]:

- **linear kernel:**

$$k(x, x') = x^T x' \quad (5.8)$$

- **polynomial kernel:**

$$k(x, x') = (\gamma x^T x' + r)^M \quad (5.9)$$

- **radial basis function (rbf) kernel:**

$$k(x, x') = \exp\left(-\frac{\|x - x'\|^2}{2\sigma^2}\right) \quad (5.10)$$

- **sigmoid kernel:**

$$k(x, x') = \tanh(\gamma x^T x' + r) \quad (5.11)$$

5.4 Artificial Neural Networks (ANNs)

Artificial Neural Networks (ANNs) or Neural Networks (NN) are a type of a Supervised Learning algorithm inspired by brain's functionality. According to a definition given by Haykin [15], "a neural network is a massively parallel distributed processor made up of simple processing units that has a natural propensity for storing experiential knowledge and making it available for use" and "it resembles the brain in two respects: (a) knowledge is acquired by the network from its environment through a learning process and (b) interneuron connection strengths, known as synaptic weights, are used to store the acquired knowledge".

5.4.1 Single Neuron (Perceptron)

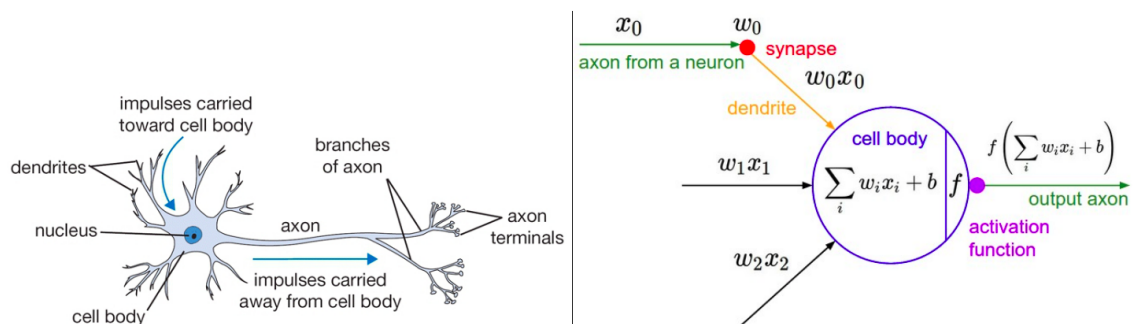


Figure 5.5. Biological (left) and mathematical (right) model of a neuron [14].

Structural component of Neural Networks is the single *artificial neuron*, also called a *node* or *unit*. It is the mathematical model of a biological neuron, based on the idea of Perceptron by Rosenblatt [55]. Both biological and artificial neurons are shown in Figure 5.5. We denote x_i as the inputs for each neuron, which are multiplied by associated weights w_i and then added to the *bias* b . The resulted sum is fed as input to an **activation function** f and given the value of the sum, f outputs 0 or 1, or as it is otherwise called the neuron is deactivated or activated respectively. The model of the neuron is given by the following mathematical equation:

$$y = f\left(\sum_{i=1}^n x_i w_i + b\right) \quad (5.12)$$

5.4.2 Multilayer Perceptrons

While a single layer perceptron can only learn linear functions, a Multilayer Perceptron (MLP) can also learn non-linear functions. A MLP contains multiple neurons (nodes) arranged in layers. Nodes from adjacent layers have connections or edges between them. All these connections have weights associated with them. In general, a MLP contains an *input* layer, one or more *hidden* layers and an *output* layer, as shown in Figure 5.6.

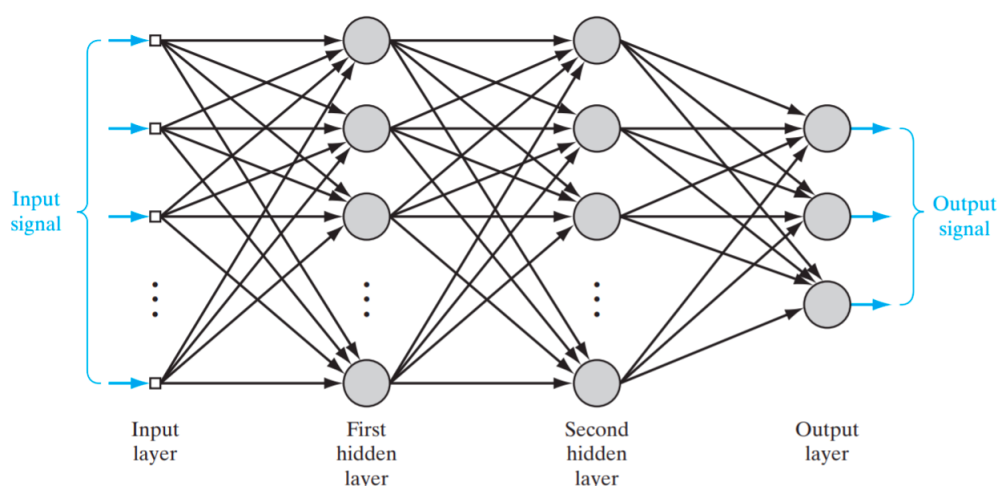


Figure 5.6. Structure of a Multilayer Perceptron with two hidden layers [15].

5.4.3 Activation Functions

The purpose of the activation function is to introduce non-linearity into the output of a neuron. This is important because most real world data is non linear and we want neurons to learn these non linear representations. Every activation

function takes a single number and performs a certain fixed mathematical operation on it [14]. Most common activation functions are:

- **Sigmoid function**, which assigns the input values to values in range between 0 and 1.

$$f(x) = \frac{1}{1 + e^{-x}} \quad (5.13)$$

Sigmoid function is widely used in cases where we want the neuron to assign a probability to an input. Such a case is the classification into two categories. However, given that the output values of sigmoid function lie between 0 and 1, there is a risk of saturating the gradients, which gradually become zero. This phenomenon is called *Vanishing Gradient*.

- **Tanh function**, which assigns the input values to values in range between -1 and 1.

$$f(x) = \tanh(x) = \frac{e^x - e^{-x}}{e^x + e^{-x}} \quad (5.14)$$

Like the sigmoid function, its activations saturate, but unlike the sigmoid its output is zero-centered. Therefore, in practice the *tanh* activation function is always preferred to the sigmoid.

- **Rectified Linear Unit (ReLU) function**, which thresholds input values to 0.

$$f(x) = \max(0, x) \quad (5.15)$$

ReLU activation function is one of the most widely used activation functions today. Unlike previous functions, ReLU simplifies computations, as it implements only one comparison.

5.4.4 Cost Functions

In calculating the error of the Neural Network model during the optimization process, a specific function must be chosen. The function we want to minimize or maximize is called the *objective function* or *criterion*. When we are minimizing it, we may also call it the *cost function*, *loss function*, or *error function* [49]. Cost Functions are used to evaluate the performance of Neural Networks. Some common cost functions are the following:

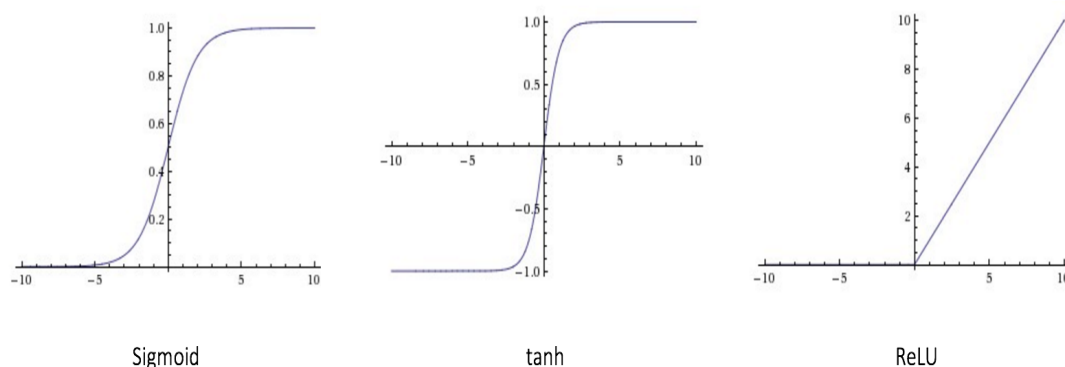


Figure 5.7. Different types of activation functions. Left to right: sigmoid, tanh, ReLU [14].

- **Mean Squared Error (MSE)**, which calculates the mean squared prediction error:

$$J(\vartheta) = \frac{1}{n} \sum_{i=1}^n (Y_i - P_i)^2, \quad (5.16)$$

where the prediction error is the difference between the true value (Y_i) and the predicted value (P_i) for an instance and ϑ is the parameter vector of the network. MSE is used with regression models [56].

- **Mean Absolute Error (MAE)**, which calculates the mean of the absolute prediction error:

$$J(\vartheta) = \frac{1}{n} \sum_{i=1}^n |Y_i - P_i|, \quad (5.17)$$

where Y_i is the true value and P_i the predicted value for an instance and ϑ is the parameter vector of the network [57].

- **Cross Entropy Loss Function**, which uses the concept of *cross-entropy*. Cross-entropy is mathematically defined as:

$$H(p, q) = - \sum_k p_k \log q_k, \quad (5.18)$$

where p and q are the true and the predicted probability distributions respectively. The more the two distributions differ, the higher the value of the cross-entropy. Cross-entropy loss function is widely used in classification problems. Based on the definition of cross-entropy, the goal of the cross-

entropy loss function is to minimize the cross-entropy between the model's distribution and the distribution of the given data [46], [49].

- **SVM Loss Function (or Hinge Loss Function)**, which aims to assign a higher score in the correct classes than the incorrect classes by some fixed margin Δ . The mathematical definition is:

$$J(\vartheta) = \sum_{j \neq y_i} \max(0, s_j - s_{y_i} + \Delta), \quad (5.19)$$

where s_j is the score for the j^{th} class, s_{y_i} is the score for the true class and ϑ is the parameter vector of the network [14].

5.4.5 Training a Neural Network

A problem often encountered in Neural Networks is the so-called *overfitting*, which occurs when the function to be optimized is so much fitted in the training data, that models even noise in them and thus, it cannot generalize well. One way to overcome this problem is through *Regularization (R)*.

$$\hat{\vartheta} = \arg \min_{\vartheta} L(\vartheta) = \arg \min_{\vartheta} - \frac{1}{N} \sum_{i=1}^N L(f(x_i; \vartheta), y_i) + \hat{\eta} R(\vartheta), \quad (5.20)$$

In Equation 5.20, hyperparameter $\hat{\eta}$ depends on the problem and the regularization term $R(\vartheta)$ calculates the norms of the parameter vector ϑ . Two common norms are $L1$ and $L2$, which are described by the following equations:

$$R_{L_1}(W) = \|W\|_1 = \sum_{ij} |W_{[ij]}| \quad (5.21)$$

$$R_{L_2}(W) = \|W\|_2^2 = \sum_{ij} (W_{[ij]})^2 \quad (5.22)$$

Alternative ways of regularization is the technique called Dropout [58], which deactivates randomly some neurons in a network or the technique Early Stopping [49], where learning stops when the Error in the test set stops decreasing.

In order to train a neural network, we have to solve the optimization problem described by Equation 5.20. Some suggested methods to do so, are Gradient Descent Algorithm, Stochastic Gradient Descent Algorithm [59], Adagrad [60] or Adam [61]. In order to minimize the Cost Function, the derivatives need to be calculated and the network to be updated. An effective method is Backpropagation [62], [63], where derivatives are calculated systematically and efficiently using the chain rule.

5.5 Convolutional Neural Networks (CNNs)

Convolutional Neural Networks (CNNs) is one of the most widely used categories of Neural Networks. CNNs "are simply neural networks that use convolution in place of general matrix multiplication in at least one of their layers" [49]. Convolution is mathematically described as:

$$s(t) = x(t) * w(t) = \int x(h)w(t - h)dh, \quad (5.23)$$

or in discrete form:

$$s(t) = x(t) * w(t) = \sum_h x(t + h)w(h), \quad (5.24)$$

where w is the *kernel* function or *filter*.

CNNs are designed to process data that come in the form of multiple arrays, as 1D arrays for signals and sequences (including language), 2D arrays representing images or 3D arrays for video or 3D images [64].

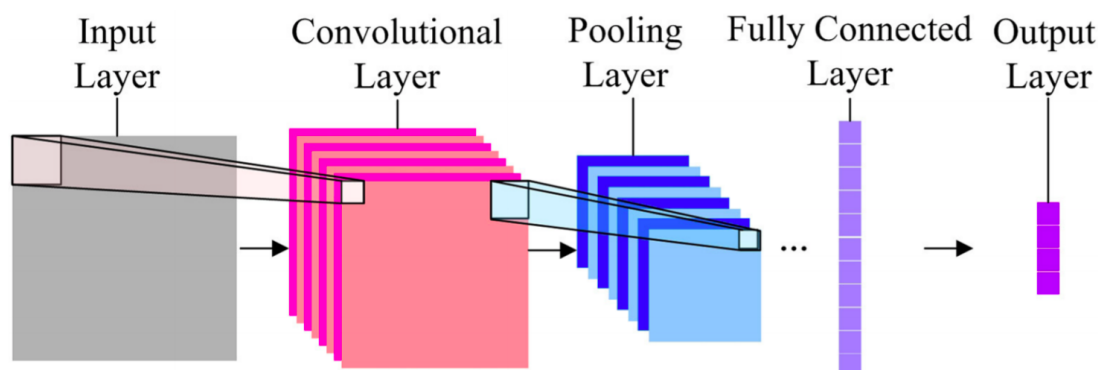


Figure 5.8. The general architecture of a CNN [16].

The architecture of a typical CNN (Figure 5.8) consists of several layers, namely *input*, *convolutional*, *pooling*, *fully connected* and *output* layer. Beyond input and output layer, convolutional layer implements a series of convolutions of the input data with specific kernels/filters, whose parameters are about to optimized through the learning process. These filters, whose size is smaller than the size of the input, are successively convolved with the entire input. An activation function (eg. ReLU) is then applied to the output of the convolutional layer and then a pooling sampling takes place. Two common pooling operators are *max pooling*, where only the maximum value within an area is selected and *average pooling*, where the value assigned to an area is the average of its values. The third layer of a CNN is a dense or fully connected layer. Each node of a fully connected layer

is connected to the outputs of all nodes of the previous level [49].

It should also be noted that the convolutional and pooling layers in CNNs are directly inspired by the classic notions of simple and complex cells in visual neuroscience and the overall architecture of the CNNs simulates the LGN-V1-V2-V4-IT hierarchy in the visual cortex ventral pathway [64].

Chapter **6**

Development of a Framework for Functional Assessment

In this chapter, we describe the method that we developed in order to functionally assess retinal models. First, we present the pipeline used in simulations. Second, we present a summary of all software tools used in this project. Third, we analyze the retinal models that we used in our simulations, the training procedure of the retinal models and the datasets used in the training. Fourth, we present the datasets used in this work and we describe the data preprocessing pipeline that we built. Finally, we analyze thoroughly the simulations that we conducted.

6.1 Pipeline of Functional Assessment

An overview of the pipeline used for functional assessment is shown in Figure 6.1. For each simulation, we used a common methodology, which includes the following steps:

1. Import dataset
2. Preprocess image data
3. Enter the preprocessed data as input to the retinal model
4. Predict firing rates
5. Enter the firing rates as input to the classifiers for image recognition and digit recognition tasks
6. Compare the performance of the different classifiers

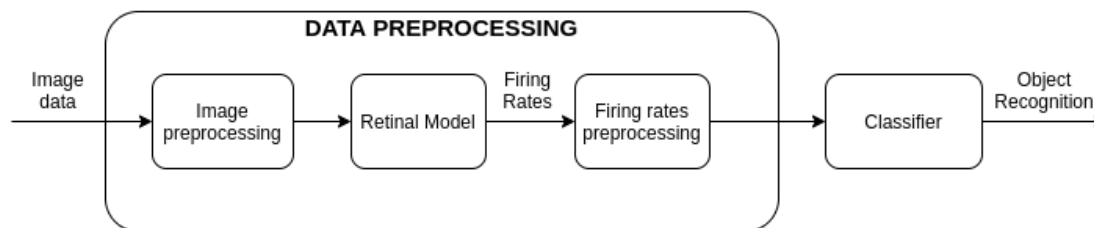


Figure 6.1. *Functional assessment pipeline.*

6.2 Description of Software Tools

The code for the retinal model that we used in this work can be found in [65]. The retinal model were trained using TensorFlow [66] and Keras [67] libraries. For the MLP classifiers, we used Keras implementation, while for Random Forest and SVM classifiers we used Scikit-learn [68] implementations. Data augmentation was applied using `imgaug` library [69]. Datasets are described in Section 6.3. All simulations were conducted with resources provided from Google Colaboratory.

6.3 Description of Datasets

For the simulations, we used the following datasets: CIFAR10 [26], MNIST [27], Fashion MNIST [28] and Imagenette [29]. A description of the datasets can be found in Table 6.1. All of these datasets are used for classification tasks with 10 different classes and that was exactly the rationale behind the choice. Our aim was to eliminate differentiating factors between the datasets and to compare equivalent classification tasks. We started with CIFAR10 and MNIST as common baseline datasets. Then, we chose to compare CIFAR10 with Imagenette and MNIST with Fashion MNIST. We chose Imagenette as a dataset containing images of much higher resolution than CIFAR10. We also chose Fashion MNIST as a more difficult and complex dataset than MNIST, as noted in [28].

6.4 CNN Retinal Model used in Simulations

6.4.1 Architecture of the CNN Retinal Model

In this work, we used the CNN retinal model designed in [19], [17], [65]. In particular, it is a 3-layer CNN that was trained to predict the firing rates of sixty simultaneously recorded retinal ganglion cells. This model was chosen, because it can effectively predict retinal responses to natural images, it has a simple and simple-to-implement architecture and, being trained with natural images, it can model a wide range of retina’s biological properties, such as rapid adaptation to

Table 6.1. *Description of the datasets used in simulations.*

Dataset	Description	Links
CIFAR10	The CIFAR10 dataset consists of 60000 32x32 colour images in 10 classes, with 6000 images per class. It is used in object recognition tasks.	[26], [70]
MNIST	The MNIST dataset consists of 70000 28x28 grayscale images of handwritten digits between 0 and 9. The digits have been size-normalized and centered in a fixed-size image. It is used in digit recognition tasks.	[27], [71]
Fashion MNIST	The Fashion MNIST dataset consists of 70000 28x28 grayscale images of 10 fashion categories. This dataset can be used as a drop-in replacement for MNIST. It is used in object recognition tasks.	[28], [72]
Imagenette	Imagenette is a subset of Imagenet. It consists of colour images of variable size representing 10 easily classified classes from Imagenet (tench, English springer, cassette player, chain saw, church, French horn, garbage truck, gas pump, golf ball, parachute). It is used in object recognition tasks.	[29], [73]

different light intensities [8]. Another advantage of this retinal model is the 3-layer architecture, which corresponds to the three main layers of cell types of the vertebrate retina (photoreceptors, bipolar and retinal ganglion cells). The first layer of the retinal model is a spatiotemporal convolution with 8 filters with (10, 10) kernel, the second is a spatial convolution with 8 filters with (5, 5) kernel, and the third is a final dense layer, as shown in Figure 6.2. Each layer consisted of a linear spatiotemporal filter, followed by a rectification using a rectified linear unit (ReLU). Batch Normalization [74] is applied to each layer as well. In Table 6.2, we show the model's summary.

6.4.2 Description of the Dataset used in Retinal Model's Training

In order to train the retinal model described above, we used a dataset used in relevant work. More specifically, in [18], they projected natural images onto the mouse retina and they recorded the response of retinal ganglion cells. Natural

Table 6.2. Summary of the CNN retinal model used in this project.

```

Model: "bn_cnn"

```

Layer (type)	Output Shape	Param #
input_1 (InputLayer)	(None, 50, 50, 40)	0
conv2d_1 (Conv2D)	(None, 41, 41, 8)	32008
flatten_1 (Flatten)	(None, 13448)	0
batch_normalization_1 (Batch Normalization)	(None, 13448)	53792
reshape_1 (Reshape)	(None, 41, 41, 8)	0
gaussian_noise_1 (GaussianNoise)	(None, 41, 41, 8)	0
activation_1 (Activation)	(None, 41, 41, 8)	0
conv2d_2 (Conv2D)	(None, 37, 37, 16)	3216
flatten_2 (Flatten)	(None, 21904)	0
batch_normalization_2 (Batch Normalization)	(None, 21904)	87616
reshape_2 (Reshape)	(None, 37, 37, 16)	0
gaussian_noise_2 (GaussianNoise)	(None, 37, 37, 16)	0
activation_2 (Activation)	(None, 37, 37, 16)	0
flatten_3 (Flatten)	(None, 21904)	0
dense_1 (Dense)	(None, 60)	1314240
batch_normalization_3 (Batch Normalization)	(None, 60)	240
activation_3 (Activation)	(None, 60)	0

```

=====
Total params: 1,491,112
Trainable params: 1,420,288
Non-trainable params: 70,824

```

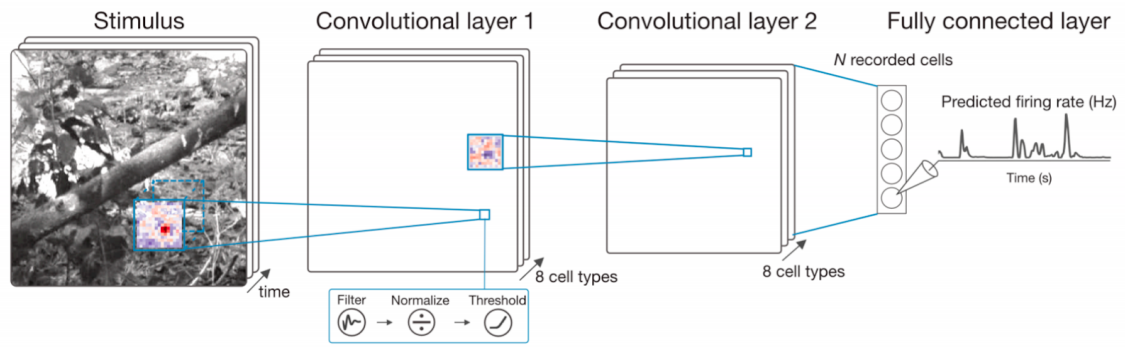


Figure 6.2. Architecture of the CNN retinal model used in this project [17].

image dataset consists of 4890 grayscale images of size 50x50 pixels, a sample of which is shown in Figure 6.3. Each frame, corresponding to 10ms of visual stimulus, was projected onto the retina for 50ms. Thus, each frame was repeated 5 times and the whole dataset consists of 24450 natural images of total duration 244.5s. Finally, each image was normalized, so that each pixel has a value between 0 and 1.

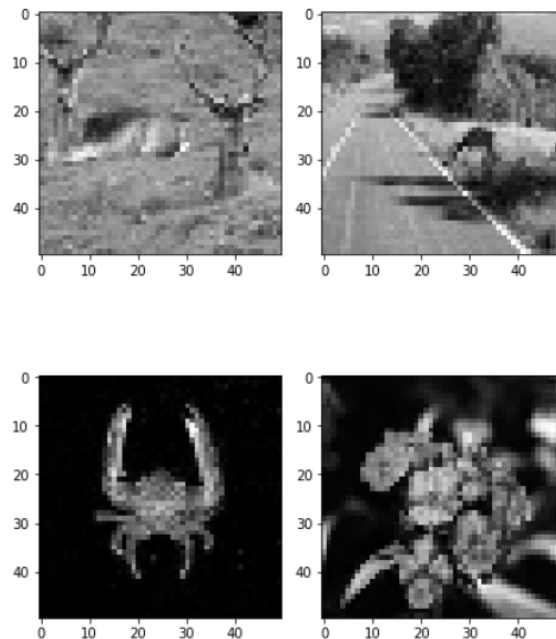


Figure 6.3. Sample images from the dataset used in retinal model's training [18].

Retinal ganglion cell's response depends not only on the current stimulus, but also depends on preceding stimulations. In order to model this temporal dependency, a temporal dataset was created before inserting images to the retinal model. This means that for each image being projected onto the retina, we kept track of the history of images projected before. The total number of images used -the actual image projected onto the retina plus additional image frames accounting for the stimulus history- is called *temporal_interval*. Therefore, each input

sample consisted of *temporal_interval* number of images. It should also be noted that the very first image's history was comprised completely of dark frames in order to model the initial dark state of the retina.

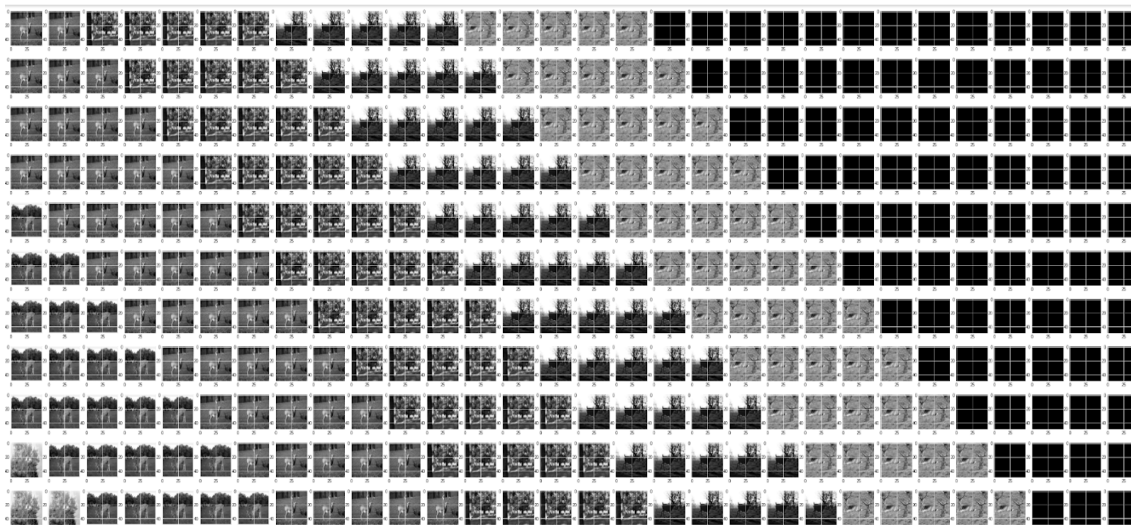


Figure 6.4. Part of a temporal dataset with *temporal_interval* = 30. The leftmost column of each row contains the actual image projected onto the retina. The rest of the columns contain the image history. Initial dark frames can also be seen.

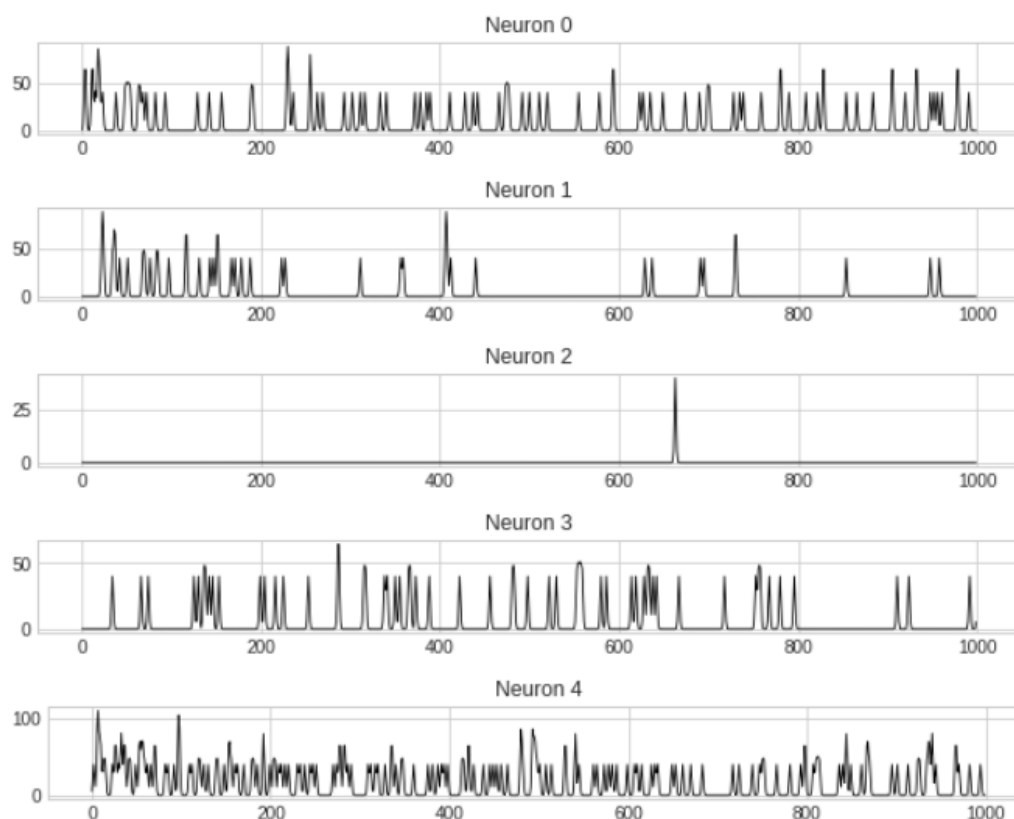


Figure 6.5. Smoothed ganglion cell's firing rate.

The target output of the model is a spiking probability function for each of

the neurons. This function comes from the convolution of spike trains with a Gaussian function, as described in Section 3.3. We used a Gaussian window with $\sigma = 10ms$, which is equal to the sampling rate of the biological recordings (each retinal response corresponds to a frame of 10ms). Convolution with a Gaussian window transforms discrete spike events into a virtually continuous target to fit, which transforms a classification problem (discrete classes corresponding to number of spikes in a bin) into a regression problem (a continuous firing rate function) [18].

6.4.3 Training the CNN Retinal Model

Taking into consideration the information described above, we trained convolutional neural network models to predict retinal ganglion cell responses to sequences of natural images, simultaneously for all 60 cells in the recorded population of the given retina. Each training sample consisted of *temporal_interval* number of frames (actual image plus image history) as input stimulus and the model's target output was the smoothed ganglion cell's firing rate corresponding to the presentation of the actual image in each time bin.

Similarly to [19] and [17], model parameters were optimized to minimize a loss function corresponding to the negative log-likelihood under Poisson spike generation,

$$L(y_t, \hat{y}_t) = \frac{1}{N} \sum_{t=0}^N \hat{y}_t - y_t \log(\hat{y}_t), \quad (6.1)$$

where y_t and \hat{y}_t are the actual and predicted firing rates of the retinal ganglion cells at time t , respectively with a batch size of N , chosen to be 50s (5000 samples). Optimization was performed using Adam [61], a variant of stochastic gradient descent. Models were trained using TensorFlow [66] and Keras [67]. The networks were regularized with an L2 weight penalty at each layer and an L1 activity penalty at the final layer, which helped maintain a baseline firing rate near 0Hz. The training dataset was divided randomly according to a 70% train and 30% validation/test split. Learning rate was 0.01. Regarding convolutional layers, as previously noted, the first layer had 8 filters with (10, 10) kernel, the second layer had 8 filters with (5, 5) kernel. Two metrics were used to measure the performance of the model: Pearson Correlation Coefficient (PCC) [75] and Mean Squared Error (MSE) [56]. Models were trained over the course of a maximum 500 epochs and the epoch with the lowest loss function was chosen. If performance was not increased over 10 epochs, learning rate was changed to 0.002 and if performance was not increased over 20 epochs, training was stopped.

In this work, we used two different models, one trained with *temporal_interval* = 30 (**RetModel1**) and the other trained with *temporal_interval* = 40 (**RetModel2**). RetModel1 performed better regarding Pearson Correlation Coefficient (PCC) metric and RetModel2 had worse PCC metrics, but produced better receptive fields (RF) ¹. More details regarding retinal models' performance can be found in [76].

6.5 Analysis of Data Preprocessing Steps

Before inserting image data as input to the retinal model, we implemented a specific preprocessing algorithm for all datasets. Main idea behind data preprocessing was to split images in parts and then feed each part separately to the retinal model. The purpose of this idea was to artificially increase the restricted number of neurons (60 in our case) provided by the retinal model, so as to increase the performance of the classifiers. Therefore, we should implement a series of preprocessing steps in order to split images, extract firing rates from the split parts and then, combine these firing rates together. To do so, we have to choose among a range of design options. All preprocessing steps together with the corresponding design decisions are analyzed below.

Step 1. (INPUT)

Import dataset and, only for CIFAR10 and Imagenette, transform each image from rgb to grayscale.

Step 2. (RESHAPE)

Reshape the size of every input image to (50, 50, 1), which is the only acceptable size to the retinal model. In the case of Imagenette, because initial images are bigger than (50, 50), reshaping takes place after splitting (in Step 5).

Step 3. (DATA AUGMENTATION)

Use data augmentation technique by choosing 2 different transformations and applying them to each image.

Data augmentation refers to applying a set of transformations to the data in order to increase the number of available data. The goal of data augmentation is

¹Receptive fields are described in Section 2.3.

to reduce overfitting and make the model perform better with unknown data. In this project, data augmentation was implemented using *imgaug* library [69] and transformations included rotation, gaussian noise, crop and translation across x and y axis. More specifically, we rotated images by 45° or -45° in order to imitate rotations that can be found in real life. Given the low resolution of the available images, we used gaussian noise of $\text{scale} = 0.1 * 255$ or $\text{scale} = 0.2 * 255$ in order to preserve the semantics of the images. We also cropped images only by 5 or 7 pixels from each side in order to keep most of the object being depicted within the frame of the image. For exactly the same reason, we translated images only by a percentage ranging from -10% to $+10\%$ of the image size. Finally, every image was augmented two times using a set of different transformations each time. In particular, the first transformation was randomly chosen among the following options:

1. Rotation (45°)
2. Gaussian noise ($\text{scale} = 0.1 * 255$)
3. Crop 5 pixel from each side of the image
4. Translation across x and y axis, ranging from -10% to $+10\%$ of the image size

and the second transformation was randomly chosen among the following options:

1. Rotation (-45°)
2. Gaussian noise ($\text{scale} = 0.2 * 255$)
3. Crop 7 pixel from each side of the image
4. Translation across x and y axis, ranging from -10% to $+10\%$ of the image size

After data augmentation, every image in the dataset has been replaced by three other images: the initial image, the one transformed using the first transformation and the one transformed using the second transformation. An example of data augmentation is shown in Figure 6.6, where we apply one of the first and one of the second transformations for each one of the four initial images.

■ **In Step 3, we should decide if we will apply or not data augmentation.**

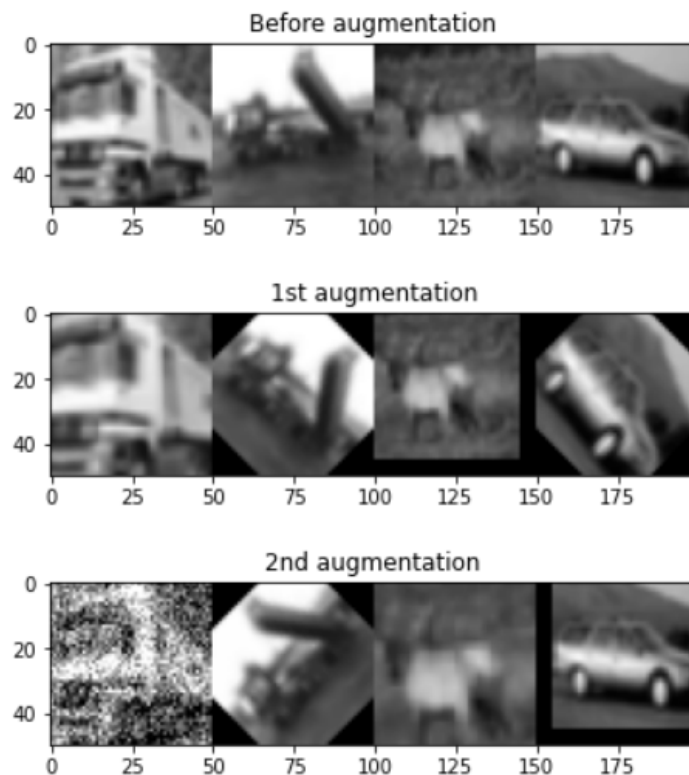


Figure 6.6. An example of data augmentation.

Step 4. (NORMALIZATION)

Normalize images, such that every pixel has value between (0, 1).

Step 5. (SPLIT)

Split image in p^2 parts, where $p = 1, 2, 3, 4$ and reshape the size of each part to (50, 50).

Image splitting is implemented by splitting each side of the image in as many parts, as the value of the variable p . In this way, smaller parts of the initial image are created. Having $p = 2$, we split the initial image in $p^2 = 4$ parts, having $p = 3$ in $p^2 = 9$ parts, etc. It should also be noted that for $p = 1$, we assume that the image is not splitted at all.

Only for Imagenette, because images are bigger than (50, 50), the size of the resulting parts after splitting is bigger than (50, 50). Therefore, as opposed to the other datasets, reshaping resulting parts to (50, 50) is a form of downscaling and not upscaling. In Figure 6.7, there is an example of image splitting in 9 parts ($p = 3$).

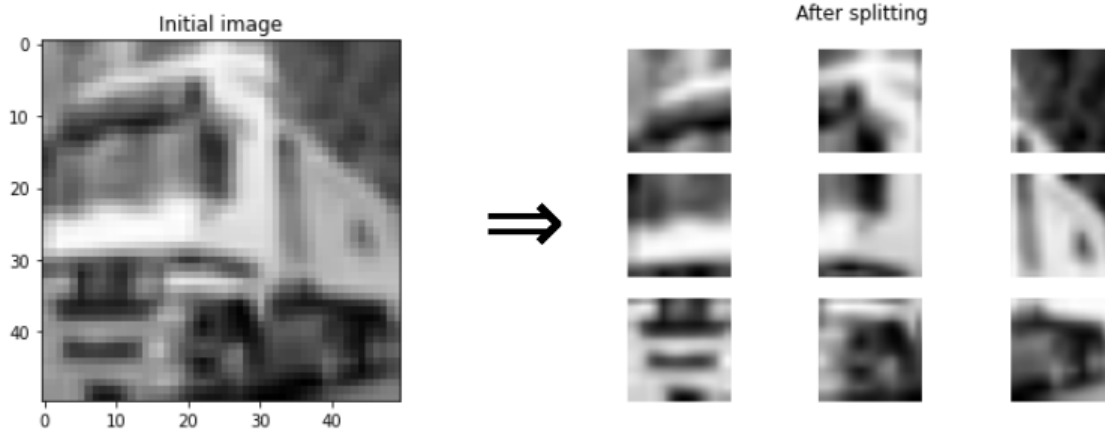


Figure 6.7. An example of image splitting in nine parts ($p = 3$).

- **In Step 5, we should decide if the image will be split and if yes, in how many parts.**

Step 6. (ADJUSTMENT)

Adjust the dataset by repeating every image five times.

The goal of Step 6 is to create a temporal dataset, similar to the one described in Section 6.4.2. For this reason, we should repeat every image five times. In order to better understand the meaning of temporal dataset, the following figure is shown:

Temporal dataset is created in order to model the duration that the retina is exposed to an image. As shown in Figure 6.8, every image can be found in 5 rows and it is repeated from 1 to 5 times in each one of the 5 rows. The rest of the frames are empty (dark) frames, representing darkness projected onto the retina. The number of image repetitions (5 in this case) represent the duration that the retina is exposed to the image. In this case, given that each frame counts for 10ms, each image is projected onto the retina for 50ms. It is important to point out that the number of image repetitions was further investigated in this thesis, as can be shown in Appendix A experiments.

It should also be noted that, if we do not implement adjustment in Step 6, the temporal dataset will consist only of one row per image with 10 image repetitions in each row (number of image repetitions was also chosen based on Appendix A

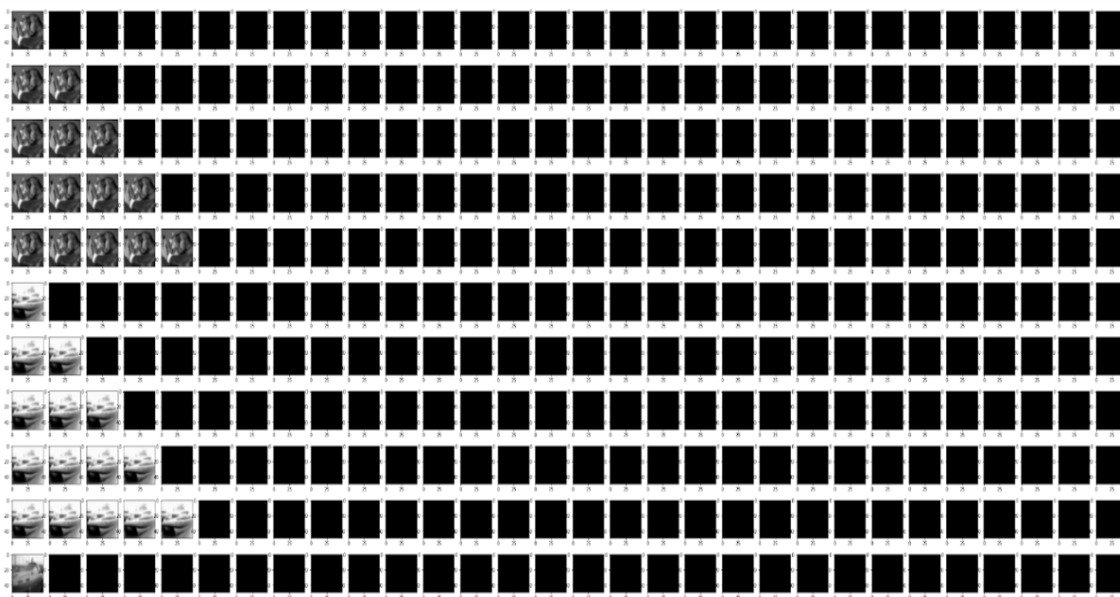


Figure 6.8. A clipping of the temporal dataset. Every row in the temporal dataset represents an input sample to the retinal model and it has as many frames as the variable **temporal_interval** (here, $\text{temporal_interval}=30$). The frame in the leftmost column represents the image that is projected onto the retina at a specific moment and the rest of the frames represent the history of the images that the retina was exposed in previous moments. Empty frames represent darkness projected onto the retina.

experiments. In the case that adjustment is implemented, there will be 5 rows per image in the temporal dataset, as previously described.

- **In Step 6, we should decide if we will implement adjustment or not.**

Step 7. (TEMPORAL DATASET)

Create the temporal dataset.

Temporal dataset is created as described in Step 6.

Step 8. (PREDICT)

Predict firing rates using the retinal model and choose (or do not choose) valid neurons.

For each one of the temporal dataset's rows, the retinal model predicts firing rates for 60 neurons. In this way, every row of the temporal dataset corresponds to a vector of 60 values, whose i -th element describes the smoothed firing rate of

the i -th neuron, where $i = 0, 1, 2, \dots, 59$. We can then identify the receptive fields of the model's neurons using STA Analysis [39]. If STA manages to compute the center of the receptive field for a neuron, then we consider this neuron as valid. For the retinal model used, there were only 12 valid neurons, in positions 2, 7, 9, 16, 19, 24, 26, 33, 41, 45, 55 of the firing rate vector, as computed in [76].

- **In Step 8, we should decide if we will choose only valid neurons or not.**

Step 9. (COMBINE)

If adjustment was implemented in Step 6, then combine all 5 vectors of firing rates corresponding to each image in one vector, using min or max transformation.

If adjustment was implemented in Step 6, this means that every image has 5 vectors with 60 firing rates (or 12 firing rates, in case of choosing only the valid neurons). These 5 vectors can be combined in one vector of size 60 (or 12), by applying elementwise min or max transformation.

- **In Step 9, we should decide if we will use min or max transformation.**

Step 10. (CONCATENATE)

If image splitting was implemented in Step 5, then concatenate in one vector all predicted firing rates corresponding to all parts of the splitted image. The resulting vector will be the input to the classifiers.

If image splitting was implemented in Step 5, there is one vector of 60 firing rates for every part of the splitted image. These vectors are then concatenated to one big vector of size $a * (p^2)$, where $a = 60$ (or $a = 12$ if valid neurons were selected) and $p = 2, 3, 4$. For example, if the image has been splitted in 9 parts ($p = 3$), the final vector has size of $60 * 9 = 540$, if $a = 60$ (or $12 * 9 = 108$, if $a = 12$). The set of final vectors after Step 10 is the training dataset for the classifiers and thus, the size of final vectors represents the number of features for the classifiers.

Everything mentioned above is summarized in the Table 6.3.

Table 6.3. Steps of data preprocessing.**DATA PREPROCESSING**

1. **(INPUT)** Import dataset and, only for CIFAR10 and Imagenette, transform each image from rgb to grayscale.
2. **(RESHAPE)** Reshape the size of every input image to (50, 50, 1), which is the only acceptable size to the retinal model. In the case of Imagenette, because initial images are bigger than (50, 50), reshaping takes place after splitting (in Step 5).
3. **(DATA AUGMENTATION)** Use data augmentation technique by choosing 2 different transformations and applying them to each image.
 - **Decide if we will apply or not data augmentation.**
4. **(NORMALIZATION)** Normalize images, such that every pixel has value between (0, 1).
5. **(SPLIT)** Split image in p^2 parts, where $p = 1, 2, 3, 4$ and reshape the size of each part to (50, 50).
 - **Decide if the image will be splitted and if yes, in how many parts.**
6. **(ADJUSTMENT)** Adjust the dataset by repeating every image five times.
 - **Decide if we will implement adjustment or not.**
7. **(TEMPORAL)** Create the temporal dataset.
8. **(PREDICT)** Predict firing rates using the retinal model and choose (or do not choose) valid neurons.
 - **Decide if we will choose only valid neurons or not.**
9. **(COMBINE)** If adjustment was implemented in Step 6, then combine all 5 vectors of firing rates corresponding to each image in one vector, using *min* or *max* transformation.
 - **Decide if we will use min or max transformation.**
10. **(CONCATENATE)** If image splitting was implemented in Step 5, then concatenate in one vector all predicted firing rates corresponding to all parts of the splitted image. The resulting vector will be the input to the classifiers.

6.6 Design of Simulations

The structure of simulations was based on the following parameters during data preprocessing: *design decisions* that should be made, the *datasets* used and the *classifiers* that were trained. In particular, design decisions were about data augmentation (**Augment**), image splitting (**Split**), combining firing rates for each image (**Combine**), adjusting the temporal dataset (**Adjust**) and choosing (or not) only the valid neurons (**Valid**). As for **Datasets**, we used CIFAR10, MNIST, Fashion_MNIST and Imagenette. As for the **Classifiers**, we trained MLP, Random Forest and SVM. Simulations showed that data augmentation consistently increases the performance of the models and restricts overfitting. Thus, data augmentation (Augment=yes) is used in all simulations. Everything mentioned above is summarized in the table below:

Table 6.4. Range of values for the parameters of simulations. Datasets are described in Table 6.1. Other parameters are described in this section.

Augment	Split	Combine	Adjust	Valid	Dataset	Classifier
Yes	None, $p = 1$	min	Yes (5 rows / image)	Yes	CIFAR10	MLP
	4 parts, $p = 2$	max	No (1 row / image)	No	MNIST	Random Forest
	9 parts, $p = 3$				Fashion MNIST	SVM
	16 parts, $p = 4$				Imagenette	

Based on the table above, we designed five types of simulations (A1, A2, B1, C1, D1) corresponding to different design decisions during data preprocessing (Table 6.5). For every type of simulation, we split images in *None*, 4, 9 and 16 parts ($p = 1, 2, 3, 4$ respectively). 4 different types of classifiers were trained: MLP_500_100 (size of input layer = 500, size of hidden layer = 100), MLP_n_n/2 (size of input layer = n, size of hidden layer = n/2, where n = number of features), SVM (kernel='rbf') and Random Forest (max_features=12). Based on initial experimentation, we chose these specific classifiers, as they better perform when

trained with firing rates. Furthermore, 10000 samples from each dataset were augmented, creating in this way 30000 samples, which were randomly divided according to 70%/30% train/test split. In total, we executed more than 200 different simulations. For each simulation, we trained each classifier 10 times and we calculated the mean and standard deviation of accuracy².

Table 6.5. Five types of simulations for different preprocessing procedures.

Five types of simulations
A1. Augment=yes, Combine=min, Adjust=yes, Valid= no
A2. Augment=yes, Combine=max, Adjust=yes, Valid= no
B1. Augment=yes, Combine=max, Adjust=no, Valid=yes
C1. Augment=yes, Combine=max, Adjust=no, Valid= no
D1. Augment=yes, Combine=max, Adjust=yes, Valid=yes

All simulations described above were conducted using RetModel1. In particular, we trained classifiers with CIFAR10 for simulation A2 (*Adjust=yes, Valid=no*), B1 (*Adjust=no, Valid=yes*), C1 (*Adjust=no, Valid=no*) and D1 (*Adjust=yes, Valid=yes*). For CIFAR10, we used 12, 48, 60, 108, 240, 540 and 960 features, each one corresponding to a different combination of *Valid* and *p* parameters. The same simulations conducted also for MNIST dataset. Then, we observed that the more features we use as input to the classifiers, the better they perform. Therefore, we trained classifiers with Fashion MNIST and Imagenette only for A2 and C1 simulations and we only used 240, 540 and 960 features.

Furthermore, the simulations that produced better results, were conducted once more using RetModel2³, in order to compare the performances between different retinal models. More specifically, we trained classifiers with all four datasets only for A2 and C1 simulation and and we only used 540 and 960 features. Results from all simulations are presented in Chapter 4. In Table 6.6, we summarize all simulations conducted.

²SVM was the only classifier trained only once (instead of ten times), because the SVM algorithm produces deterministic results.

³RetModel1 and RetModel2 are presented in Section 6.4.3

Table 6.6. Summary of all simulations per dataset. Datasets are described in Table 6.1. All types of simulations are described in Table 6.5.

RetModel1				RetModel2			
CIFAR10	MNIST	Fashion_MNIST	Imagenette	CIFAR10	MNIST	Fashion_MNIST	Imagenette
A1	A1						
A2	A2	A2	A2	A2	A2	A2	A2
B1	B1						
C1	C1	C1	C1	C1	C1	C1	C1
D1	D1						

Chapter 7

Results

In this chapter, we present tables with the accuracy of the classifiers, for each one of the simulations described in Table 6.5 and Table 6.6. There are two tables per dataset, one for simulations with RetModel1 and the other for simulations with RetModel2¹. We should mention that we present results only with "Combine" parameter set to "max", as it has consistently produced better results than "Combine=min". We also note that yellow cells in the Tables below correspond to simulations with "Adjust" parameter set to "yes", while the light blue cells correspond to simulations with "Adjust=no". Finally, in Section 4.5 we present comparative graphs.

7.1 Classifiers Performance in CIFAR10 Dataset

In this section, we present tables with the accuracy of the classifiers for the simulations A2, B1, C1 and D1, as described in Table 6.5. Classifiers were trained with CIFAR10 dataset (see Table 6.1). Table 7.1 contains simulations using Retmodel1, while Table 7.2 contains simulations using Retmodel2. Simulations with RetModel1 were conducted using samples with $n_{features} = 12, 48, 60, 108, 240, 540, 960$, while simulations with RetModel2 were conducted using samples with $n_{features} = 540, 960$.

7.2 Classifiers Performance in MNIST Dataset

In this section, we present tables with the accuracy of the classifiers for the simulations A2, B1, C1 and D1, as described in Table 6.5. Classifiers were trained with MNIST dataset (see Table 6.1). Table 7.3 contains simulations using Retmodel1, while Table 7.4 contains simulations using Retmodel2. Simulations with RetModel1 were conducted using samples with $n_{features} =$

¹RetModel1 and RetModel2 are presented in Section 6.4.3

12, 48, 60, 108, 240, 540, while simulations with RetModel2 were conducted using samples with $n_{features} = 540,960$.

7.3 Classifiers Performance in Fashion MNIST Dataset

In this section, we present tables with the accuracy of the classifiers for the simulations A2, B1, C1 and D1, as described in Table 6.5. Classifiers were trained with Fashion_MNIST dataset (see Table 6.1). Table 7.5 contains simulations using Retmodel1, while Table 7.6 contains simulations using Retmodel2. Simulations with RetModel1 were conducted using samples with $n_{features} = 240,540,960$, while simulations with RetModel2 were conducted using samples with $n_{features} = 540,960$.

7.4 Classifiers Performance in Imagenette Dataset

In this section, we present tables with the accuracy of the classifiers for the simulations A2, B1, C1 and D1, as described in Table 6.5. Classifiers were trained with Imagenette dataset (see Table 6.1). Table 7.7 contains simulations using Retmodel1, while Table 7.8 contains simulations using Retmodel2. Simulations with RetModel1 were conducted using samples with $n_{features} = 240,540,960$, while simulations with RetModel2 were conducted using samples with $n_{features} = 540,960$.

Table 7.1. Mean and standard deviation of classifiers performance in CIFAR10 dataset (Table 6.1) using RetModel1 (Section 6.4.3) for simulations A2, D1, B1, C1 (Tables 6.4, 6.5). Yellow cells in the Table below correspond to simulations with "Valid" parameter set to "yes", while the light blue cells correspond to simulations with "Valid=no".

CIFAR10					
Simulations A2 & D1 (Adjust=yes, Combine=max)					
Model1	n_features (n)	Classifiers			
		MLP_500_100	MLP_n_n/2	SVM	Random Forest
	12 (Valid=yes, p=1)	13.179% (+/-4.173)	15.687% (+/-0.856)	19.567% (+/-0.000)	30.890% (+/-0.241)
	48 (Valid=yes, p=2)	11.590% (+/-3.605)	15.378% (+/-4.673)	26.667% (+/-0.000)	39.208% (+/-0.300)
	60 (Valid=no, p=1)	21.912% (+/-0.365)	20.537% (+/-0.730)	24.850% (+/-0.000)	41.365% (+/-0.219)
	108 (Valid=yes, p=3)	11.147% (+/-2.972)	19.046% (+/-4.575)	31.550% (+/-0.000)	40.968% (+/-0.460)
	240 (Valid=no, p=2)	24.567% (+/-0.380)	25.283% (+/-0.872)	33.333% (+/-0.000)	40.740% (+/-0.585)
	540 (Valid=no, p=3)	26.288% (+/-1.264)	27.186% (+/-1.108)	41.200% (+/-0.000)	40.945% (+/-0.383)
	960 (Valid=no, p=4)	26.895% (+/-0.744)	27.771% (+/-0.896)	47.083% (+/-0.000)	40.990% (+/-0.320)
	Simulations B1 & C1 (Adjust=no, Combine=max)				
	n_features (n)	Classifiers			
		MLP_500_100	MLP_n_n/2	SVM	Random Forest
	12 (Valid=yes, p=1)	18.027% (+/-0.422)	21.258% (+/-0.204)	19.150% (+/-0.000)	28.152% (+/-0.219)
	48 (Valid=yes, p=2)	28.501% (+/-0.304)	26.259% (+/-0.318)	30.700% (+/-0.000)	34.548% (+/-0.330)
	60 (Valid=no, p=1)	28.138% (+/-0.547)	25.254% (+/-0.171)	32.333% (+/-0.000)	38.685% (+/-0.518)
	108 (Valid=yes, p=3)	31.460% (+/-0.387)	30.054% (+/-0.422)	33.333% (+/-0.000)	35.667% (+/-0.420)
	240 (Valid=no, p=2)	32.063% (+/-0.526)	31.562% (+/-0.515)	30.167% (+/-0.000)	35.225% (+/-0.369)
	540 (Valid=no, p=3)	35.121% (+/-0.277)	35.685% (+/-0.517)	19.417% (+/-0.000)	36.255% (+/-0.458)

Table 7.2. Mean and standard deviation of classifiers performance in CIFAR10 dataset (Table 6.1) using RetModel2 (Section 6.4.3) for simulations A2, C1 (Tables 6.4, 6.5). Light blue cells in the Table below correspond to simulations with "Valid=no".

CIFAR10					
Model2	Simulation A2 (<i>Adjust=yes, Combine=max</i>)				
	<i>n_features (n)</i>	Classifiers			
		MLP_500_100	MLP_n_n/2	SVM	Random Forest
	540 (Valid=no, p=3)	12.321% (+/-3.423)	10.117% (+/-0.151)	31.650% (+/-0.000)	40.212% (+/-0.389)
	960 (Valid=no, p=4)	11.172% (+/-2.628)	10.288% (+/-0.109)	34.900% (+/-0.000)	38.220% (+/-0.388)
	Simulation C1 (<i>Adjust=no, Combine=max</i>)				
	<i>n_features (n)</i>	Classifiers			
		MLP_500_100	MLP_n_n/2	SVM	Random Forest
	540 (Valid=no, p=3)	31.217% (+/-1.081)	31.893% (+/-0.703)	44.183% (+/-0.000)	39.302% (+/-0.455)
	960 (Valid=no, p=4)	30.903% (+/-0.646)	31.271% (+/-0.685)	36.300% (+/-0.000)	37.190% (+/-0.312)

Table 7.3. Mean and standard deviation of classifiers performance in MNIST dataset (Table 6.1) using RetModel1 (Section 6.4.3) for simulations A2, D1, B1, C1 (Tables 6.4, 6.5). Yellow cells in the Table below correspond to simulations with "Valid" parameter set to "yes", while the light blue cells correspond to simulations with "Valid=no".

MNIST					
Model1	Simulations A2 & D1 (Adjust=yes, Combine=max)				
	n_features (n)	Classifiers			
		MLP_500_100	MLP_n_n/2	SVM	Random Forest
	12 (Valid=yes, p=1)	44.269% (+/-0.955)	27.558% (+/-5.680)	47.167% (+/-0.000)	65.377% (+/-0.208)
	48 (Valid=yes, p=2)	56.251% (+/-1.502)	48.671% (+/-1.433)	72.967% (+/-0.000)	82.377% (+/-0.385)
	60 (Valid=no, p=1)	68.492% (+/-1.268)	58.146% (+/-1.185)	72.033% (+/-0.000)	82.723% (+/-0.262)
	108 (Valid=yes, p=3)	59.221% (+/-2.234)	59.208% (+/-2.279)	83.467% (+/-0.000)	87.838% (+/-0.160)
	240 (Valid=no, p=2)	75.112% (+/-1.885)	75.376% (+/-1.019)	87.717% (+/-0.000)	87.975% (+/-0.241)
	540 (Valid=no, p=3)	77.737% (+/-1.146)	79.301% (+/-1.964)	91.233% (+/-0.000)	89.300% (+/-0.223)
	Simulations B1 & C1 (Adjust=no, Combine=max)				
n_features (n)	Classifiers				
	MLP_500_100	MLP_n_n/2	SVM	Random Forest	
12 (Valid=yes, p=1)	48.507% (+/-0.352)	34.019% (+/-0.640)	44.667% (+/-0.000)	55.225% (+/-0.285)	
48 (Valid=yes, p=2)	72.159% (+/-0.412)	59.804% (+/-0.253)	58.950% (+/-0.000)	74.032% (+/-0.374)	
60 (Valid=no, p=1)	75.445% (+/-0.402)	65.940% (+/-0.486)	57.883% (+/-0.000)	76.327% (+/-0.280)	
108 (Valid=yes, p=3)	78.693% (+/-0.501)	74.670% (+/-0.403)	44.500% (+/-0.000)	81.632% (+/-0.200)	
240 (Valid=no, p=2)	84.258% (+/-0.390)	83.844% (+/-0.495)	27.500% (+/-0.000)	82.165% (+/-0.369)	
540 (Valid=no, p=3)	87.844% (+/-0.716)	88.648% (+/-0.794)	16.317% (+/-0.000)	85.115% (+/-0.221)	

Table 7.4. Mean and standard deviation of classifiers performance in MNIST dataset (Table 6.1) using RetModel2 (Section 6.4.3) for simulations A2, C1 (Tables 6.4, 6.5). Light blue cells in the Table below correspond to simulations with "Valid=no".

MNIST					
Model2	Simulation A2 (Adjust=yes, Combine=max)				
	$n_features (n)$	Classifiers			
		MLP_500_100	MLP_n_n/2	SVM	Random Forest
	540 (Valid=no, p=3)	24.389% (+/-16.507)	12.740% (+/-4.736)	81.700% (+/-0.000)	88.417% (+/-0.292)
	960 (Valid=no, p=4)	29.564% (+/-16.920)	11.384% (+/-0.000)	87.617% (+/-0.000)	89.442% (+/-0.111)
	Simulation C1 (Adjust=no, Combine=max)				
	$n_features (n)$	Classifiers			
		MLP_500_100	MLP_n_n/2	SVM	Random Forest
	540 (Valid=no, p=3)	79.795% (+/-1.116)	81.186% (+/-1.013)	74.467% (+/-0.000)	83.668% (+/-0.302)
	960 (Valid=no, p=4)	80.874% (+/-2.188)	84.104% (+/-1.648)	64.117% (+/-0.000)	85.195% (+/-0.214)

Table 7.5. Mean and standard deviation of classifiers performance in Fashion MNIST dataset (Table 6.1) using RetModel1 (Section 6.4.3) for simulations A2, C1 (Tables 6.4, 6.5). Light blue cells in the Table below correspond to simulations with "Valid=no".

Fashion MNIST					
Model1	Simulation A2 (Adjust=yes, Combine=max)				
	n_features (n)	Classifiers			
		MLP_500_100	MLP_n_n/2	SVM	Random Forest
	240 (Valid=no, p=2)	66.742% (+/-1.356)	67.505% (+/-1.392)	76.687% (+/-0.000)	78.255% (+/-0.238)
	540 (Valid=no, p=3)	70.440% (+/-0.941)	70.948% (+/-1.841)	80.657% (+/-0.000)	79.226% (+/-0.100)
	960 (Valid=no, p=4)	69.933% (+/-1.406)	71.167% (+/-1.271)	79.616% (+/-0.000)	79.359% (+/-0.126)
	Simulation C1 (Adjust=no, Combine=max)				
	n_features (n)	Classifiers			
		MLP_500_100	MLP_n_n/2	SVM	Random Forest
	240 (Valid=no, p=2)	76.695% (+/-1.006)	75.911% (+/-0.586)	44.606% (+/-0.000)	75.684% (+/-0.314)
540 (Valid=no, p=3)	78.853% (+/-0.781)	79.254% (+/-0.704)	19.323% (+/-0.000)	77.525% (+/-0.197)	
960 (Valid=no, p=4)	79.544% (+/-0.609)	79.591% (+/-0.694)	13.586% (+/-0.000)	77.718% (+/-0.180)	

Table 7.6. Mean and standard deviation of classifiers performance in Fashion MNIST dataset (Table 6.1) using RetModel2 (Section 6.4.3) for simulations A2, C1 (Tables 6.4, 6.5). Light blue cells in the Table below correspond to simulations with "Valid=no".

Fashion MNIST					
Model12	Simulation A2 (Adjust=yes, Combine=max)				
	n_features (n)	Classifiers			
		MLP_500_100	MLP_n_n/2	SVM	Random Forest
	540 (Valid=no, p=3)	57.700% (+/-2.267)	59.808% (+/-1.096)	75.444% (+/-0.000)	79.125% (+/-0.185)
	960 (Valid=no, p=4)	56.576% (+/-3.655)	56.978% (+/-2.118)	77.707% (+/-0.000)	79.544% (+/-0.190)
	Simulation C1 (Adjust=no, Combine=max)				
	n_features (n)	Classifiers			
		MLP_500_100	MLP_n_n/2	SVM	Random Forest
	540 (Valid=no, p=3)	72.422% (+/-1.582)	74.012% (+/-0.962)	77.545% (+/-0.000)	76.776% (+/-0.136)
	960 (Valid=no, p=4)	73.192% (+/-0.605)	74.542% (+/-0.771)	64.434% (+/-0.000)	77.268% (+/-0.159)

Table 7.7. Mean and standard deviation of classifiers performance in Imagenette dataset (Table 6.1) using RetModel1 (Section 6.4.3) for simulations A2, C1 (Tables 6.4, 6.5). Light blue cells in the Table below correspond to simulations with "Valid=no".

Imagenette					
Model1	Simulation A2 (Adjust=yes, Combine=max)				
	n_features (n)	Classifiers			
		MLP_500_100	MLP_n_n/2	SVM	Random Forest
	240 (Valid=no, p=2)	25.160% (+/-0.629)	25.648% (+/-1.375)	34.576% (+/-0.000)	44.905% (+/-0.200)
	540 (Valid=no, p=3)	26.944% (+/-0.673)	27.403% (+/-0.722)	43.384% (+/-0.000)	45.964% (+/-0.238)
	960 (Valid=no, p=4)	28.900% (+/-0.592)	28.589% (+/-1.479)	44.414% (+/-0.000)	46.364% (+/-0.195)
	Simulation C1 (Adjust=no, Combine=max)				
	n_features (n)	Classifiers			
		MLP_500_100	MLP_n_n/2	SVM	Random Forest
	240 (Valid=no, p=2)	36.070% (+/-0.423)	34.220% (+/-0.491)	32.687% (+/-0.000)	43.665% (+/-0.188)
540 (Valid=no, p=3)	39.416% (+/-0.616)	40.588% (+/-0.765)	29.030% (+/-0.000)	45.108% (+/-0.329)	
960 (Valid=no, p=4)	40.488% (+/-0.464)	41.401% (+/-0.650)	21.929% (+/-0.000)	45.087% (+/-0.444)	

Table 7.8. Mean and standard deviation of classifiers performance in Imagenette dataset (Table 6.1) using RetModel2 (Section 6.4.3) for simulations A2, C1 (Tables 6.4, 6.5). Light blue cells in the Table below correspond to simulations with "Valid=no".

Imagenette					
Model2	Simulation A2 (Adjust=yes, Combine=max)				
	n_features (n)	Classifiers			
		MLP_500_100	MLP_n_n/2	SVM	Random Forest
	540 (Valid=no, p=3)	10.917% (+/-2.451)	10.036% (+/-0.119)	32.667% (+/-0.000)	45.933% (+/-0.160)
	960 (Valid=no, p=4)	10.158% (+/-0.073)	10.109% (+/-0.111)	37.444% (+/-0.000)	46.239% (+/-0.311)
	Simulation C1 (Adjust=no, Combine=max)				
	n_features (n)	Classifiers			
		MLP_500_100	MLP_n_n/2	SVM	Random Forest
	540 (Valid=no, p=3)	30.923% (+/-0.840)	31.276% (+/-0.693)	42.121% (+/-0.000)	45.598% (+/-0.174)
	960 (Valid=no, p=4)	32.227% (+/-0.569)	32.447% (+/-0.906)	45.213% (+/-0.316)	35.152% (+/-0.000)

7.5 Comparative Assessment of Results

In this section, we present and discuss several comparative diagrams regarding different retinal models, datasets and classifiers used in simulations. At first, we present sensitivity plots, which correlate the percentage of each classifier's accuracy with the number of neurons ($n_features$). Secondly, we present bar graphs comparing the performance of classifiers among all four datasets (CIFAR10, Imagenette, MNIST, Fashion MNIST). Then, we present bar graphs comparing the performance of classifiers between the simulations with "*Adjust=yes*" and those with "*Adjust=no*". Finally, we present bar graphs comparing the performance of classifiers between RetModel1 and RetModel2 for all datasets.

7.5.1 Assessment of classifiers performance in terms of the number of neurons

Based on Figure 7.1 and Figure 7.2, we can make some useful observations. First and foremost, we note that the more neurons (or $n_features$) we use, the better performance we get. In most of the cases, we attain the highest performance for 540 neurons, while no further improvement is observed for more than 240 neurons and especially for more than 540 neurons, as 7.1c,d and Figure 7.2c,d show. Secondly, Random Forest performs consistently better than the rest of the classifiers, followed by SVM and in the end, there is MLP classifiers with the lowest accuracy across all simulations. However, in simulations B1 and C1 (where "*Adjust=no*"), SVM's performance seems to collapse after 240 neurons. Thirdly, we observe that classifiers perform significantly better in MNIST and Fashion MNIST datasets, attaining around 80% accuracy or more. On the other hand, classifiers attain half of the previous accuracy (around 40%) in CIFAR10 and Imagenette. These points and some more are going to be further discussed based on the figures presented below.

7.5.2 Comparing the performance of classifiers between datasets

In Figure 7.3, we plot bar graphs in order to compare the maximum accuracy of different classifiers between CIFAR10 and Imagenette and between MNIST and Fashion MNIST. As said in Section 6.3, we chose Imagenette as an improved (images with higher resolution) dataset than CIFAR10, and Fashion MNIST as a more difficult and complex dataset than MNIST, and that's why we compare them in pairs. MNIST performs around 10% better than Fashion MNIST in all simulations, a difference which is consistent with the results of the literature [28]. Regarding the comparison between CIFAR10 and Imagenette, we observe different

outcomes according to the classifier used and the retinal model. In particular, both MLPs and Random Forest perform better with Imagenette. SVM performs better with CIFAR10 using RetModel1, while it performs better with Imagenette using RetModel2.

7.5.3 Comparing the performance of classifiers between simulations with "Adjust=yes" and "Adjust=no"

In Figure 7.4, we compare the maximum accuracy of different classifiers between simulations conducted with *"Adjust=yes"* and those conducted with *"Adjust=no"*, for CIFAR10 and Imagenette. Both simulations have very similar performance. Random Forest performs almost the same for both *"Adjust=yes"* and *"Adjust=no"*. SVM with *"Adjust=yes"* performs better for RetModel1, while SVM with *"Adjust=no"* gives better results for RetModel2. Regarding MLPs, they consistently give better results with *"Adjust=no"* for both retinal models and they totally fail with *"Adjust=yes"* and RetModel2.

In Figure 7.5, we compare the maximum accuracy of different classifiers between simulations conducted with *"Adjust=yes"* and those conducted with *"Adjust=no"*, for MNIST and Fashion MNIST. For both datasets, SVM and Random Forest performs better with *"Adjust=yes"*. MLPs give better results with *"Adjust=no"* for both retinal models and in the case of MNIST, they totally fail with *"Adjust=yes"* and RetModel2.

7.5.4 Comparing the performance of classifiers between retinal models

Finally, we compare, for all datasets, the performance of different classifiers between RetModel1 and RetModel2 as shown in Figure 7.6. The main and direct observation is that RetModel1 is superior (although slightly in some cases) to RetModel2.

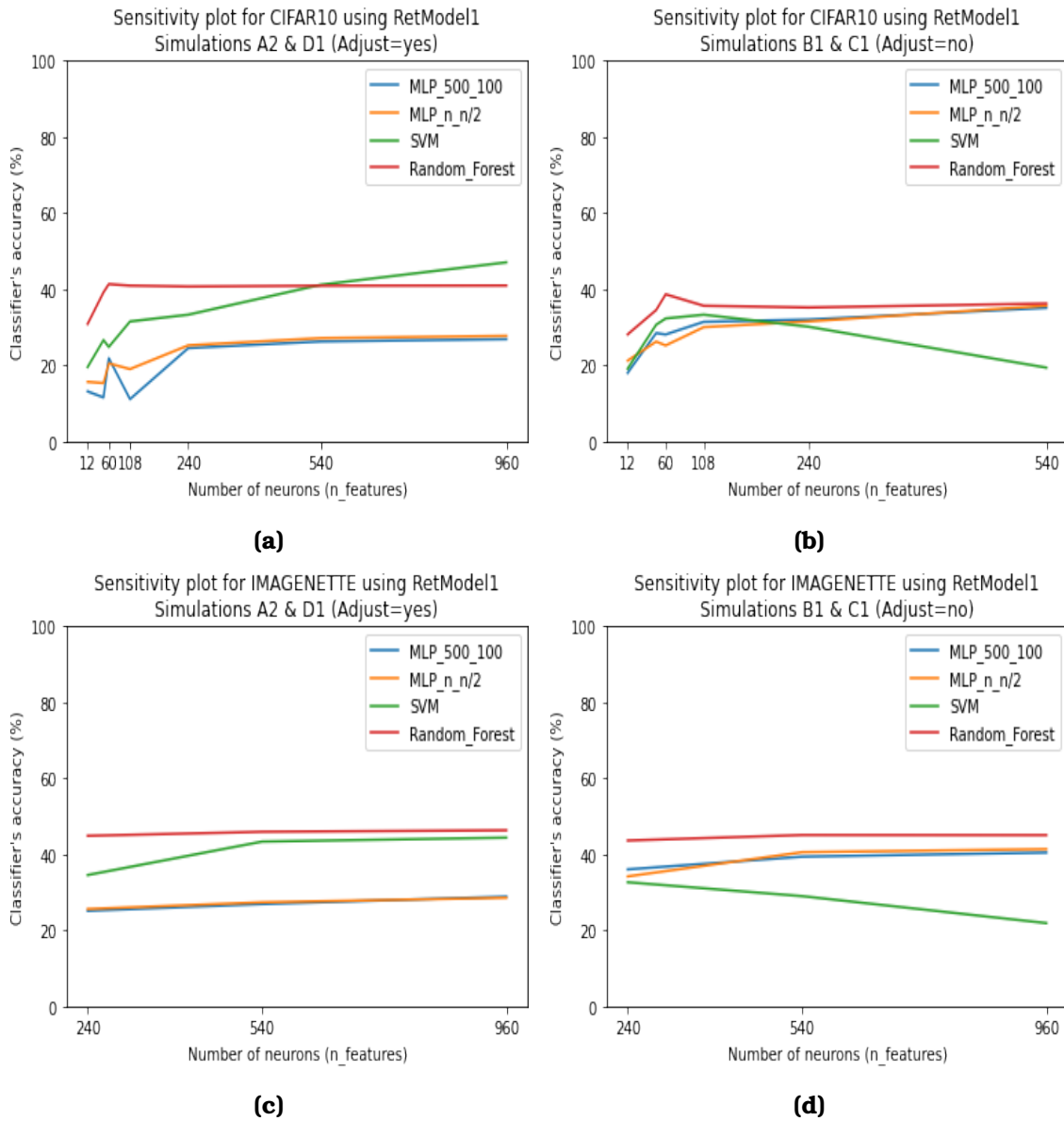


Figure 7.1. Sensitivity plots using RetModel1 (Section 6.4.3) for different datasets (Table 6.1), different simulations (Table 6.5) and different combinations of parameters (Table 6.4). (a) CIFAR10 dataset, simulations A2 & D1 ("Adjust=yes"). (b) CIFAR10 dataset, simulations B1 & C1 ("Adjust=no"). (c) Imagenette dataset, simulations A2 & D1 ("Adjust=yes"). (d) Imagenette dataset, simulations B1 & C1 ("Adjust=no").

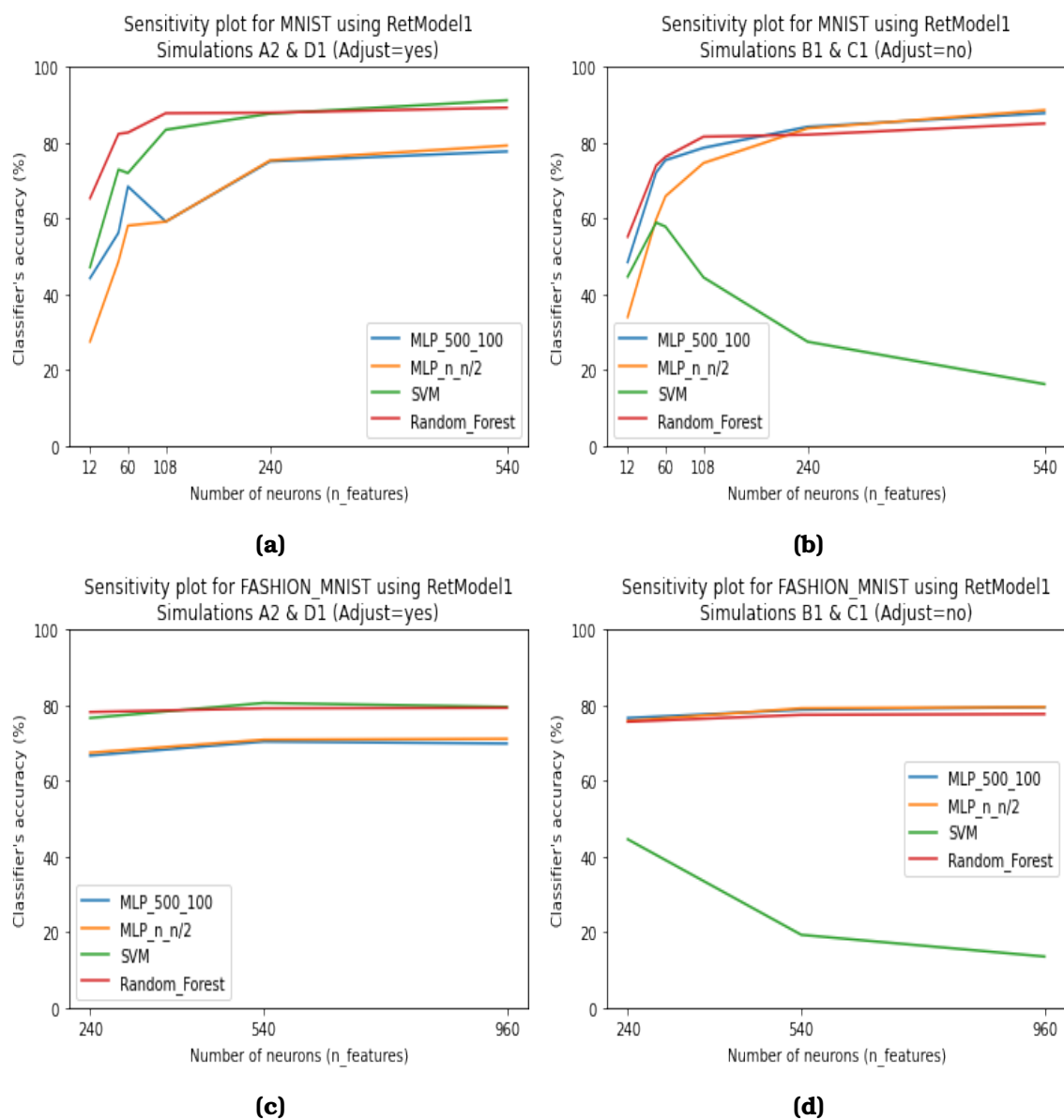


Figure 7.2. Sensitivity plots using RetModel1 (Section 6.4.3) for different datasets (Table 6.1), different simulations (Table 6.5) and different combinations of parameters (Table 6.4). (a) MNIST dataset, simulations A2 & D1 ("Adjust=yes"). (b) MNIST dataset, simulations B1 & C1 ("Adjust=no"). (c) Fashion MNIST dataset, simulations A2 & D1 ("Adjust=yes"). (d) Fashion MNIST dataset, simulations B1 & C1 ("Adjust=no").

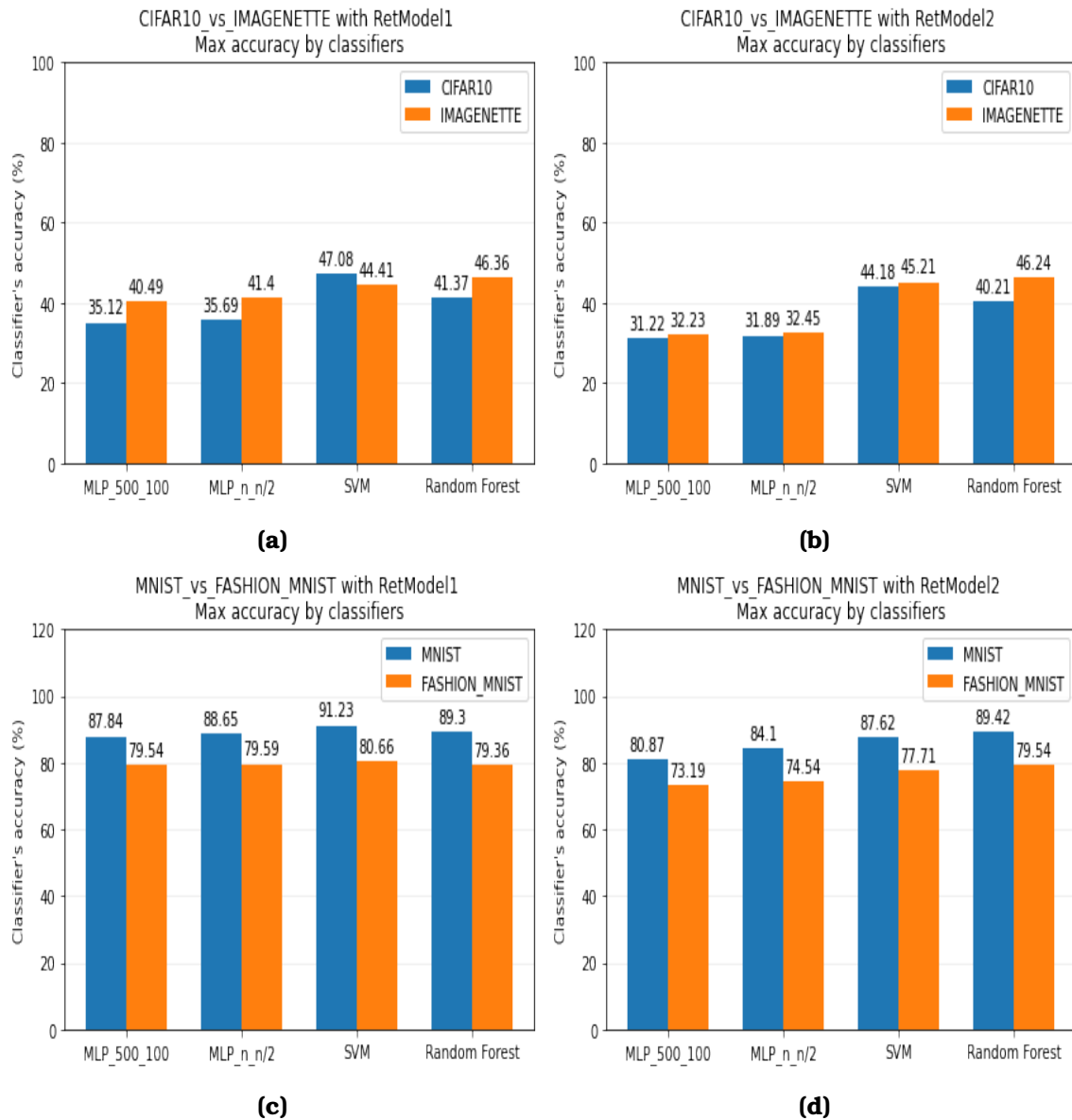


Figure 7.3. Comparison of the maximum performance of classifiers between different datasets (Table 6.1) for both retinal models (Section 6.4.3). (a) Comparison between CIFAR10 and Imagenette using RetModel1. (b) Comparison between CIFAR10 and Imagenette using RetModel2. (c) Comparison between MNIST and Fashion MNIST using RetModel1. (d) Comparison between MNIST and Fashion MNIST using RetModel2.

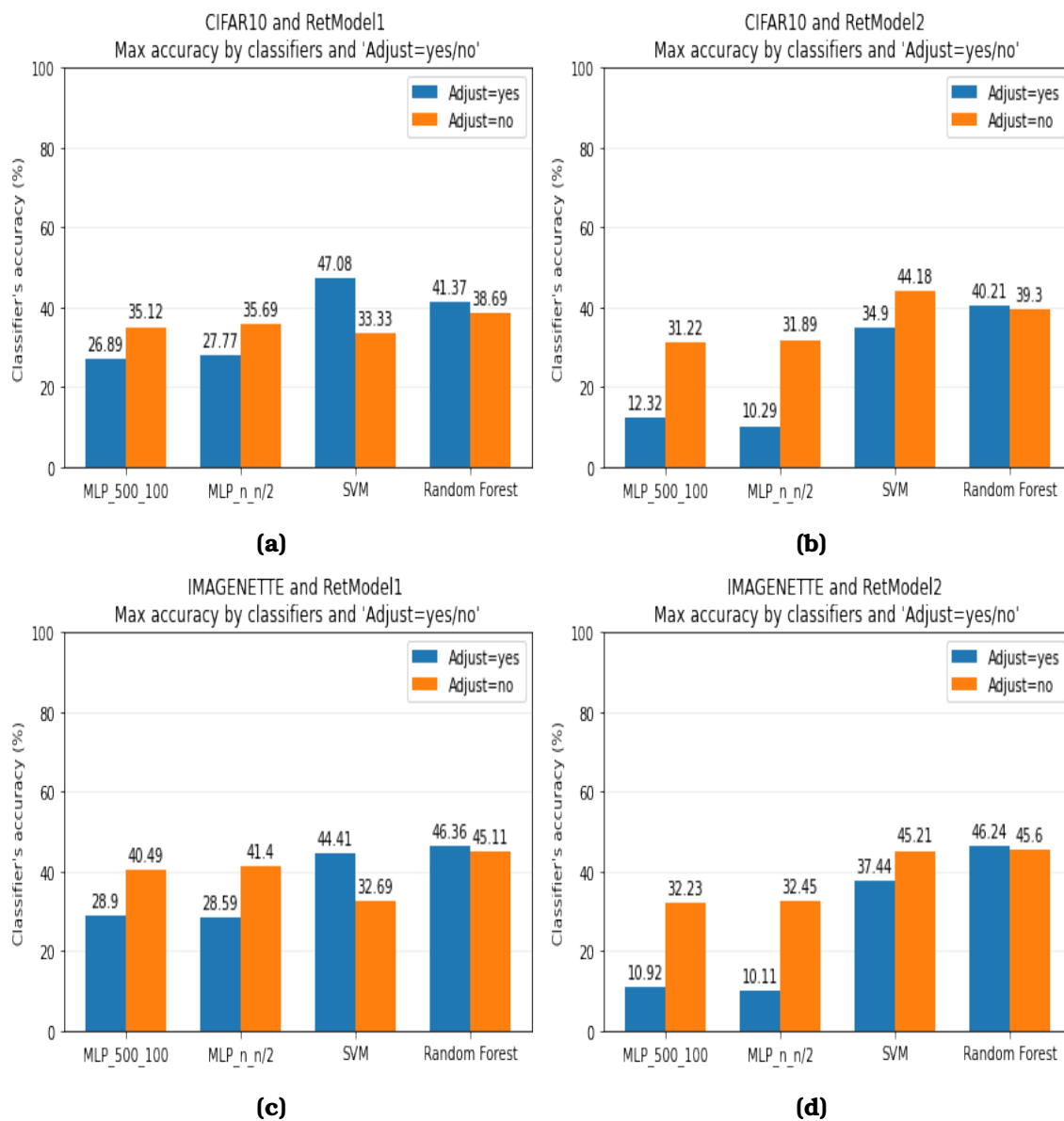


Figure 7.4. Comparison of the maximum performance of classifiers between simulations with "Adjust=yes" and those with "Adjust=no" (Table 6.4) for different datasets (Table 6.1) for both retinal models (Section 6.4.3). (a) Comparison for CIFAR10 using RetModel1. (b) Comparison for CIFAR10 using RetModel2. (c) Comparison for Imagenette using RetModel1. (d) Comparison for Imagenette using RetModel2.

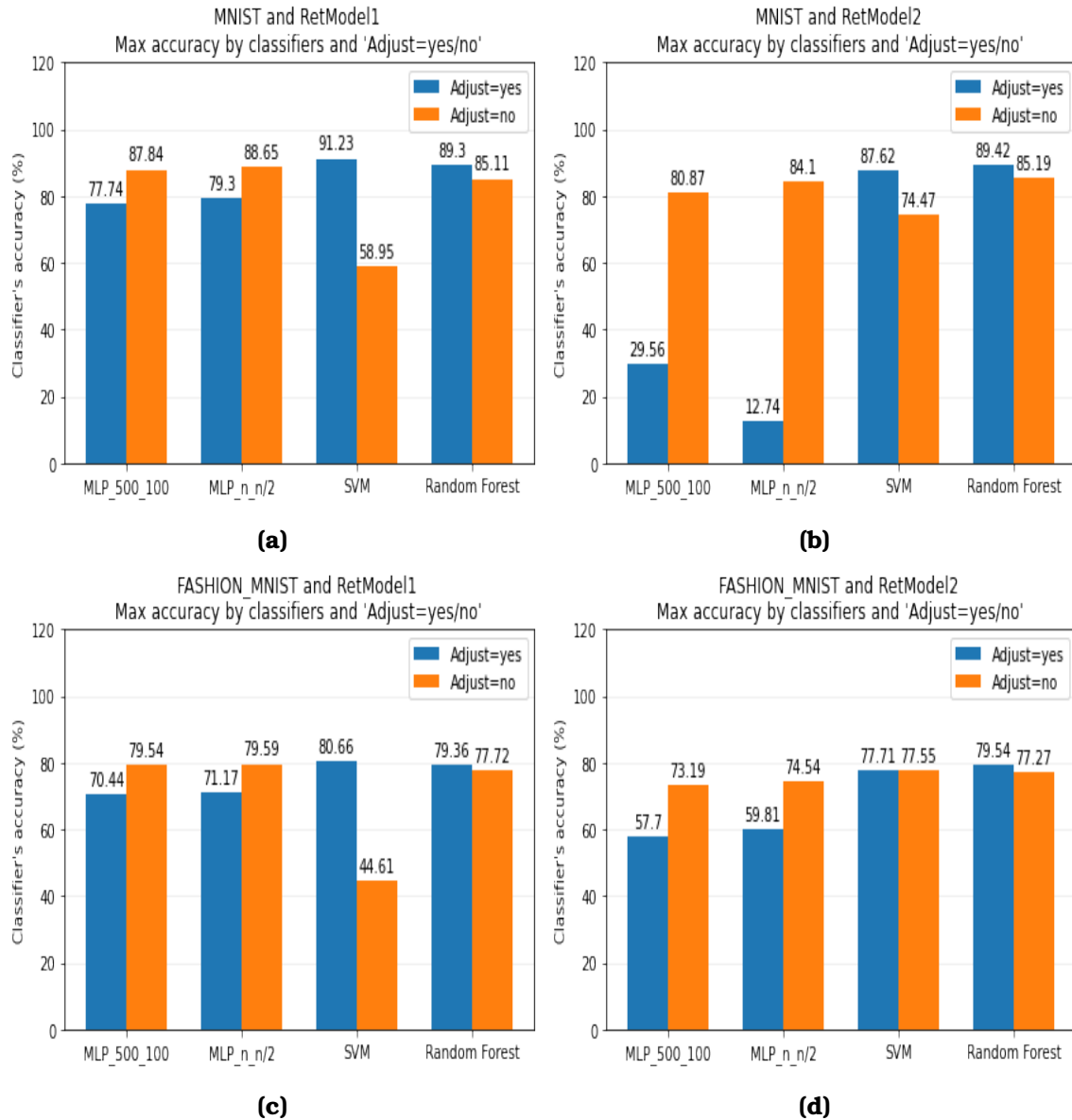


Figure 7.5. Comparison of the maximum performance of classifiers between simulations with "Adjust=yes" and those with "Adjust=no" (Table 6.4) for different datasets (Table 6.1) for both retinal models (Section 6.4.3). (a) Comparison for MNIST using RetModel1. (b) Comparison for MNIST using RetModel2. (c) Comparison for Fashion MNIST using RetModel1. (d) Comparison for Fashion MNIST using RetModel2.

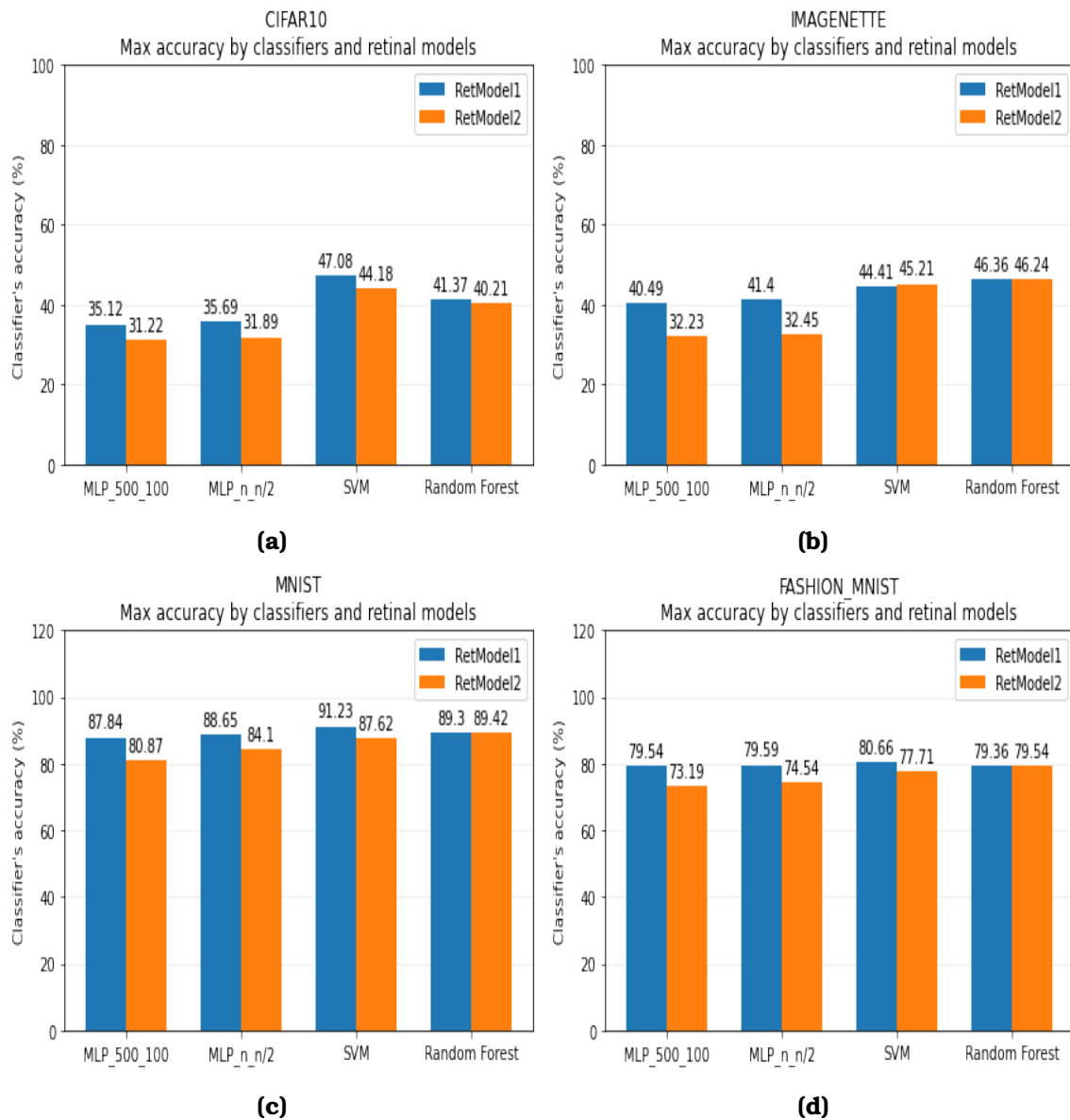


Figure 7.6. Comparison of the maximum performance of classifiers between RetModel1 and RetModel2 (Section 6.4.3) for all datasets. (a) Comparison in CIFAR10. (b) Comparison in Imagenette. (c) Comparison in MNIST. (d) Comparison in Fashion MNIST.

Chapter 8

Discussion

The goal of this thesis was the design of a functional assessment pipeline in order to compare the performance of different retinal models in image recognition tasks and especially, we focused on object and digit recognition tasks. First simulations showed that the more neurons used as features for the classifiers, the better was the performance of the model. On this basis, we tried to artificially increase the restricted number of available neurons (60 in our case) provided by retinal models, so as to increase the performance of several classifiers. To do so, we implemented the idea of splitting images in parts and then feeding each part separately to the retinal model. During experimentation, we investigated a series of questions regarding preprocessing steps, the use of different datasets and classifiers and the performance of retinal models. In this chapter, we discuss results, as presented in Chapter 7 and we describe several limitations that we encountered during this thesis.

8.1 Comparative Assessment of Results

In this section, we present results regarding preprocessing steps, datasets, classifiers and retinal models, as presented in Chapter 7.

8.1.1 Data Preprocessing Steps

As shown in Table 6.3, several parametric decisions should be taken during data preprocessing. More specifically, it should be decided (a) whether or not to apply data augmentation, (b) whether or not to split images and if yes, in how many parts, (c) whether to implement adjustment before feeding the retinal model, (d) whether to choose only valid neurons or not and finally, (e) whether a min or max transformation should be used in order to combine firing rates of the split parts of an image. We fine-tuned these decisions, so as to optimize the model's performance. In particular, simulations highlighted the importance of data aug-

mentation in the performance of the classifiers, a result that is in agreement with best practices of Machine Learning [77]. They also showed that the more neurons used as features to the classifiers, the better they perform. However, using more than 540 ($9 * 60$) neurons does not further increase the accuracy of the classifiers. We also assumed that, by overly splitting an image in parts, the objects in the image are no longer perceptible and thus, the problem of object recognition is degenerated. This means that subimages will be so small, that they will not contain information for the object depicted in the image. Therefore, we concluded that splitting the images in 4 or 9 parts is sufficient for optimal performance. In addition, combining firing rates of split parts of an image with *max* transformation consistently performed better than using *min* transformation. This means that we only keep the neurons that responded better in visual stimuli, which is in accordance with the functionality of biological neurons. Finally, results indicated that image adjustment can be avoided, given that the retinal model performs equally well without it and given that the training of the model is less demanding in resources and time. This is opposed to the way that biological retinal ganglion cells were recorded in [18] and [19]. However, we could claim, based on experiments in Appendix A, that feeding a retinal model with around 10 consecutive frames of an image is enough for most of retinal ganglion cells to respond.

8.1.2 Datasets

We also investigated how the performance of a model differs depending on the task, thus depending on the dataset used. Initially, we used CIFAR10 and MNIST as baseline datasets. Further experimentation took place with more advanced datasets, such as Imagenette and Fashion MNIST. We chose Imagenette as a dataset containing images of higher resolution than CIFAR10, aiming in this way to have higher resolution images, even if we split them. We also chose Fashion MNIST as a more difficult and complex dataset than MNIST. According to results, we assume that image resolution is not a crucial factor in the model's performance. CIFAR10 consists of images of size (32, 32) and given that every image was split from 4 up to 16 parts and then each part reshaped to (50, 50), we see that CIFAR10 provides image data of very low resolution. On the other side, Imagenette was intentionally used after CIFAR10 to increase image resolution. Every image in Imagenette is much bigger than (50, 50), which results in split image parts of much higher resolution. However, in spite of significantly higher quality, Imagenette did not perform significantly better than CIFAR10 and there were even simulations that performed worse. If we also compare the accuracy of CIFAR10 and Imagenette with the accuracy of MNIST and Fashion MNIST, we observe that the latter is twice as high as the former (more than 80%), even if

both MNIST and Fashion MNIST consist of images of size (28, 28). Based on the significant difference in performance between CIFAR10/Imagenette and MNIST / Fashion MNIST, we assume that the structural differences between them could be a differentiating factor. Images in both MNIST and Fashion MNIST have the object for classification situated in the middle and the rest of the image consists of dark pixels. CIFAR10 and Imagenette on the other hand, consist of actual images, which are composed of various colors and shapes and they are situated across the whole image. This could play a key role, if we additionally consider the eye's property to react only in spatiotemporal changes among different image frames. This property favors both MNIST and Fashion MNIST, whose only part that changes is the object that we are interested to classify.

8.1.3 Classifiers

Furthermore, we examined which classifiers perform better in such type of tasks. Initial experimentation with different types of classifiers, led us to choose MLP_500_100 (size of input layer = 500, size of hidden layer = 100), MLP_n_n/2 (size of input layer = n, size of hidden layer = n/2, where n = number of features), SVM (kernel='rbf') and Random Forest (max_features=12). Among them, Random Forest was proved to be the best performing and the most consistent classifier. Even if SVM sometimes approached or even exceeded Random Forest's accuracy, it did not perform that well across the whole range of simulations.

8.1.4 Retinal Models

In the very end of this work, we applied functional assessment in order to compare the performance of two different retinal models. Results showed that the retinal model with the lowest mean squared error (MSE) performed better across the whole range of functional simulations. Therefore, it seems that functional assessment is in accordance with standard assessment technique. Still, by functionally evaluating a model we get a direct estimate of how well an implantee may perform in a task of interest.

8.2 Limitations of Functional Assessment

During the development of this work, several limitations were encountered. The first limitation has to do with the application of retinal prostheses in relationship with functional tasks. Arrays of electrodes in implants provide a limited number of available neurons and thus, retinal models are trained to predict responses for only a few neurons. However, both functional tasks in real life usually require colored images of high resolution and also machine learning models

require data with sufficient number of features in order to perform well in functional tasks. This poses a problem not only when trying to use retinal model to solve object recognition tasks, but also when developing retinal prostheses that could ideally meet the functional needs of real life. Another limitation has to do with the selected modelling approach of the visual system. From a biological perspective, the visual path does not include only retinal processing, but mainly it includes neural processing in the brain's visual centers, which are responsible for higher order functionalities, like object recognition. In this work, we choose to process retinal firing rates with traditional machine learning classifiers. An alternative could be the use of deep learning architectures in order to model brain centers like the Ventral Visual System (VVS), which is the system associated with object recognition in the brain. This approach has been used in high-impact research on biological vision, though with a different rationale: as a means to model higher (cortical) visual functions to elucidate the mechanisms involved in the retina [78].

Chapter 9

Conclusions

9.1 Summary

In this Diploma Thesis, we investigated the problem of functional assessment of retinal models with the use of machine learning techniques. We showed how to functionally compare the performance of different retinal models in image recognition tasks and we built a pipeline for this purpose. In particular, we used CIFAR10, MNIST, Fashion MNIST and Imagenette as datasets and we trained common classifiers, such as MLP, SVM and Random Forest. Considering the limited number of available neurons provided by the retinal model, we used the idea of splitting images in parts and then feeding each part separately to the retinal model, so as to artificially increase the number of neural responses. Based on this concept, we built and optimized an image preprocessing pipeline. We found that Random Forest classifier achieves the highest performance and that models perform better with datasets such as MNIST and Fashion MNIST, where we achieved up to 90% accuracy. In addition, we found that using more neurons improves the overall performance of the model, thus highlighting the need to design retinal prostheses with a larger number of available electrodes. Finally, we applied functional assessment in order to compare the performance of two different retinal models. The results showed that there is an agreement between standard and functional assessment, as the two models had similar performance in both techniques.

9.2 Future Work

On the way to develop a functional assessment framework for retinal models, we came across interesting future directions, that could increase functional assessment reliability. These are briefly described below:

- Apply functional assessment to more retinal models with different architectures.

- Develop functional assessment pipelines for more functional tasks, such as face recognition or motion detection.
- Use deep learning models mimicking brain's visual centers, instead of traditional classifiers. In this way, object detection and other higher functionalities may perform better.
- Use *semantic image segmentation* in order to increase the performance of retinal models in object recognition tasks. In this way, it would be possible to separate regions of interest in an image and fill the rest of it with dark pixels. Therefore, retinal models could more efficiently recognize objects.
- Explore if there is convergence (or divergence) in functionally evaluating a retinal model and using perceptual images. Perceptual images represent a method for visualizing a model's performance by decoding the neural activity (of retina or V1) to a perceived ("mental") image [79].

Appendices

Appendix **A**

Image repetitions in temporal dataset

During data preprocessing, we need to decide the number of image repetitions in temporal dataset. Given that each row of the temporal dataset has "*temporal_interval*" frames, we need to decide how many of them would be actual images and how many would be dark frames (Figure A.1).



Figure A.1. An example of a row in temporal dataset.

To do so, we conducted an experiment in order to compare the firing rates of each neuron as a function of the number of actual images used in the temporal dataset. We plotted firing rates using 1, 5 and 500 different rows of temporal data. In the case of 500 rows, we plotted errorbars with the mean firing rate and the standard deviation for every different number of images used. In our case, the maximum length of each row (i.e. "*temporal_interval*") equals to 30. In Figure A.2, an example with all three types of plots is shown, while in Table A.1 we show errorbars from 8 different neurons.

In order to detect spikes, we can check either in the beginning of the plots (first 1 – 5 frames), where retina first sees the image, or in the very last frames (last 20 – 30 frames), where the retina stops to see the image. This is based on retina's property to react only in spatiotemporal changes between consecutive frames. As we see in Table A.1, in these regions of the plot (in the very beginning or in the very end) errorbars are very small, indicating that neurons have a similar behavior independently of the input image. However, if we want to better classify objects, we care about the regions of the plot with the highest deviation. This is because we expect that neurons would have different behaviors for different image inputs and we will be able to classify them according to neuron's reaction to them.

Based on the figures below, we observe that the regions with the highest deviation range from 10 to 15 images. Therefore, we conclude that if we want the retinal model to better perform in image recognition tasks, rows in temporal datasets should consist of a number between 10 and 15 images.

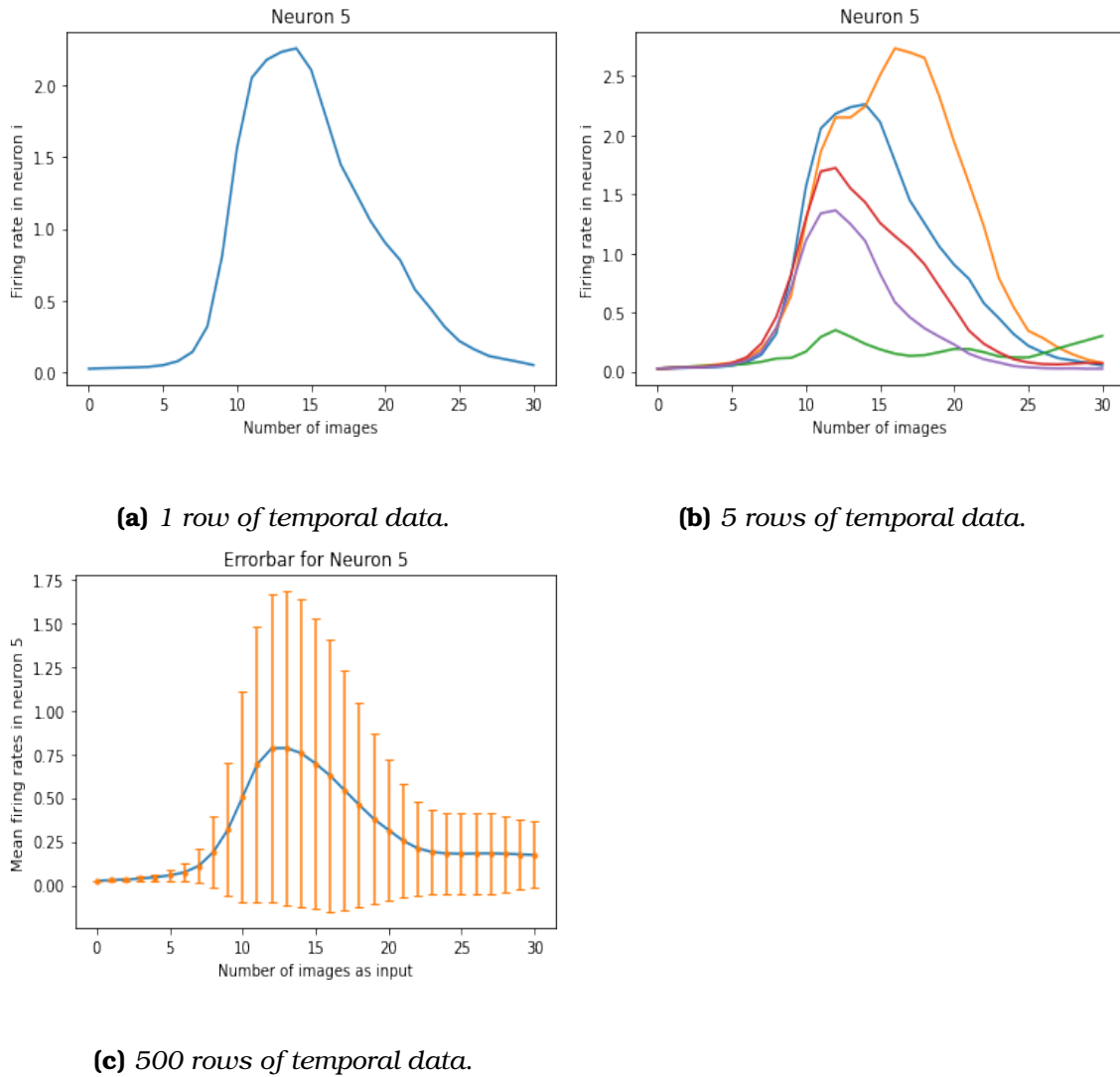
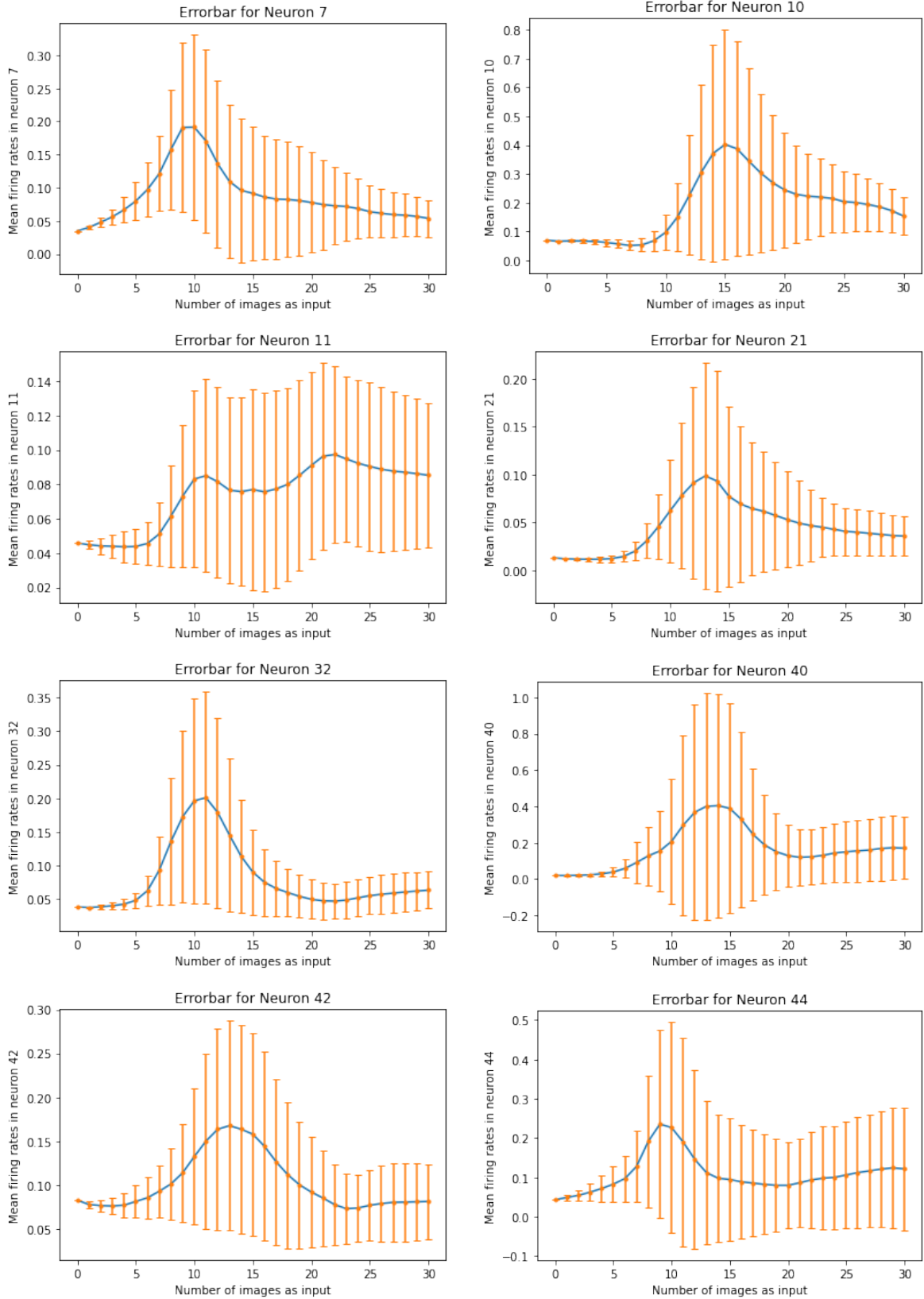


Figure A.2. Graphs representing the firing rates of neuron_5 as a function of different numbers of images in each temporal row. For each diagram, we give different number of images as input to the model and we receive different number of outputs (i.e. firing rates). In Figure A2c, we plot the errorbar, as we have an input of five hundred images.

Table A.1. Errorbars for eight different neurons.



Bibliography

- [1] J. E. Bixler, "FIGURE 3, [Cross section of the eye...].," May 2019. Publisher: American Diabetes Association.
- [2] Wikipedia, "Macula of retina — Wikipedia, the free encyclopedia." <http://en.wikipedia.org/w/index.php?title=Macula%20of%20retina&oldid=996160948>, 2021. [Online; accessed 09-January-2021].
- [3] "Simple Anatomy of the Retina by Helga Kolb – Webvision."
- [4] J. Martins and L. Sousa, *Bioelectronic Vision: Retina Models, Evaluation Metrics, and System Design*. Series on bioengineering and biomedical engineering, World Scientific, 2009.
- [5] D. Heeger, "Perception lecture notes: Retinal ganglion cells." <http://www.cns.nyu.edu/~david/courses/perception/lecturenotes/ganglion/ganglion.html>.
- [6] M. Fee, "Introduction to neural computation, lecture 08." https://ocw.mit.edu/courses/brain-and-cognitive-sciences/9-40-introduction-to-neural-computation-spring-2018/lecture-notes/MIT9_40S18_Lec08.pdf.
- [7] P. Dayan and L. F. Abbott, *Theoretical Neuroscience: Computational and Mathematical Modeling of Neural Systems*. The MIT Press, 2005.
- [8] S. A. Baccus and M. Meister, "Fast and Slow Contrast Adaptation in Retinal Circuitry," *Neuron*, vol. 36, pp. 909–919, Dec. 2002.
- [9] N. Maheswaranathan *et al.*, "The dynamic neural code of the retina for natural scenes," *bioRxiv*, 2019. Publisher: Cold Spring Harbor Laboratory _eprint: <https://www.biorxiv.org/content/early/2019/12/17/340943.full.pdf>.
- [10] J. D. Weiland and M. S. Humayun, "Retinal Prosthesis," *IEEE Transactions on Biomedical Engineering*, vol. 61, pp. 1412–1424, May 2014.
- [11] L. Yue, J. D. Weiland, B. Roska, and M. S. Humayun, "Retinal stimulation strategies to restore vision: Fundamentals and systems," *Progress in Retinal and Eye Research*, vol. 53, pp. 21–47, July 2016.

- [12] S. Theodoridis and K. Koutroumbas, *Pattern Recognition, Fourth Edition*. USA: Academic Press, Inc., 4th ed., 2008.
- [13] H.-R. Zhang, F. Min, and X. He, "Aggregated recommendation through random forests," *TheScientificWorldJournal*, vol. 2014, p. 649596, 08 2014.
- [14] F.-F. Li, A. Karpathy, and J. Johnson, "Cs231n: Convolutional neural networks for visual recognition 2016,"
- [15] S. Haykin, *Neural Networks: A Comprehensive Foundation (3rd Edition)*. USA: Prentice-Hall, Inc., 2007.
- [16] M. Peng, C. Wang, T. Chen, G. Liu, and X. Fu, "Dual temporal scale convolutional neural network for micro-expression recognition," *Frontiers in Psychology*, vol. 8, 10 2017.
- [17] N. Maheswaranathan *et al.*, "Deep learning models reveal internal structure and diverse computations in the retina under natural scenes," *bioRxiv*, 2018. Publisher: Cold Spring Harbor Laboratory _eprint: <https://www.biorxiv.org/content/early/2018/06/08/340943.full.pdf>.
- [18] A. Lozano, C. Soto-Sánchez, J. Garrigós, J. J. Martínez, J. M. Ferrández, and E. Fernández, "A 3D Convolutional Neural Network to Model Retinal Ganglion Cell's Responses to Light Patterns in Mice," *International Journal of Neural Systems*, vol. 28, p. 1850043, Dec. 2018.
- [19] L. T. McIntosh, N. Maheswaranathan, A. Nayebi, S. Ganguli, and S. A. Baccus, "Deep learning models of the retinal response to natural scenes," 2017.
- [20] E. Fernandez, "Development of visual neuroprostheses: trends and challenges," *Bioelectronic Medicine*, vol. 4, Dec 2018.
- [21] N. Melanitis and K. S. Nikita, "Biologically-inspired image processing in computational retina models," *Computers in biology and medicine*, vol. 113, p. 103399, 2019.
- [22] A. Alevizaki, N. Melanitis, and K. Nikita, "Predicting eye fixations using computer vision techniques," in *2019 IEEE 19th International Conference on Bioinformatics and Bioengineering (BIBE)*, pp. 309–315, IEEE, 2019.
- [23] B. P. Lepri, "Is acuity enough? other considerations in clinical investigations of visual prostheses," *Journal of Neural Engineering*, vol. 6, p. 035003, Jun 2009.

- [24] A. M. Demchinsky *et al.*, “The first deaf-blind patient in russia with argus II retinal prosthesis system: what he sees and why,” *Journal of Neural Engineering*, vol. 16, p. 025002, feb 2019.
- [25] A. N. Angelopoulos, H. Ameri, D. Mitra, and M. Humayun, “Enhanced depth navigation through augmented reality depth mapping in patients with low vision,” *Scientific Reports*, vol. 9, p. 11230, Aug 2019.
- [26] A. Krizhevsky, V. Nair, and G. Hinton, “Cifar-10 (canadian institute for advanced research),”
- [27] Y. LeCun and C. Cortes, “MNIST handwritten digit database,” 2010.
- [28] H. Xiao, K. Rasul, and R. Vollgraf, “Fashion-mnist: a novel image dataset for benchmarking machine learning algorithms,” 2017.
- [29] J. Howard, “Imagenette.” <https://github.com/fastai/imagenette/>.
- [30] M. Morow *Encyclopedia of the Neurological Sciences || Retina.*, 26-28, 2014.
- [31] G. Field and E. Chichilnisky, “Information processing in the primate retina: Circuitry and coding,” *Annual Review of Neuroscience*, vol. 30, p. 1-30, Jul 2007.
- [32] R. Masland, “The neuronal organization of the retina,” *Neuron*, vol. 76, p. 266-280, Oct 2012.
- [33] R. H. Masland, “Cell populations of the retina: The proctor lecture,” *Investigative Ophthalmology Visual Science*, vol. 52, p. 4581, Jun 2011.
- [34] J. R. Sanes and R. H. Masland, “The Types of Retinal Ganglion Cells: Current Status and Implications for Neuronal Classification,” *Annual Review of Neuroscience*, vol. 38, pp. 221-246, July 2015.
- [35] S. W. Kuffler, “Discharge patterns and functional organization of mammalian retina,” *Journal of Neurophysiology*, vol. 16, no. 1, pp. 37-68, 1953. PMID: 13035466.
- [36] P. H. Schiller and E. J. Tehovnik, “Visual prosthesis,” *Perception*, vol. 37, no. 10, pp. 1529-1559, 2008. PMID: 19065857.
- [37] Wikipedia, “Action potential — Wikipedia, the free encyclopedia.” <http://en.wikipedia.org/w/index.php?title=Action%20potential&oldid=998242198>, 2021. [Online; accessed 09-January-2021].

- [38] D. Heeger, "Poisson model of spike generation," 2000.
- [39] E. Chichilnisky, "A simple white noise analysis of neuronal light responses," *Network (Bristol, England)*, vol. 12, pp. 199–213, 06 2001.
- [40] J. W. Pillow *et al.*, "Spatio-temporal correlations and visual signalling in a complete neuronal population," *Nature*, vol. 454, pp. 995–999, Aug. 2008.
- [41] A. N. Burkitt, "A Review of the Integrate-and-fire Neuron Model: I. Homogeneous Synaptic Input," *Biological Cybernetics*, vol. 95, pp. 1–19, July 2006.
- [42] H. Tanaka *et al.*, "From deep learning to mechanistic understanding in neuroscience: the structure of retinal prediction," *arXiv:1912.06207 [physics, q-bio]*, Dec. 2019. arXiv: 1912.06207.
- [43] "Artificial retina-restoring eyesight due to a photoreceptor replacement implant." <https://www.inserm.fr/en/health-information/health-and-research-from-z/artificial-retina>.
- [44] E. Bloch, Y. Luo, and L. da Cruz, "Advances in retinal prosthesis systems," *Therapeutic Advances in Ophthalmology*, vol. 11, p. 251584141881750, Jan. 2019.
- [45] "Retinal prostheses by ethan d. cohen – webvision." <https://webvision.med.utah.edu/book/part-xv-prosthetics/retinal-prostheses-by-ethan-d-cohen/>.
- [46] K. P. Murphy, *Machine Learning: A Probabilistic Perspective*. The MIT Press, 2012.
- [47] T. M. Mitchell, *Machine Learning*. USA: McGraw-Hill, Inc., 1 ed., 1997.
- [48] A. Zhang, Z. C. Lipton, M. Li, and A. J. Smola, *Dive into Deep Learning*. 2020. <https://d2l.ai>.
- [49] I. Goodfellow, Y. Bengio, and A. Courville, *Deep Learning*. The MIT Press, 2016.
- [50] L. Breiman, J. H. Friedman, R. A. Olshen, and C. J. Stone, *Classification and Regression Trees*. Monterey, CA: Wadsworth and Brooks, 1984.
- [51] N. Patel and S. Upadhyay, "Study of various decision tree pruning methods with their empirical comparison in weka," *International Journal of Computer Applications*, vol. 60, pp. 20–25, 2012.
- [52] Y. Song and Y. Lu, "Decision tree methods: applications for classification and prediction," *Shanghai Archives of Psychiatry*, vol. 27, pp. 130 – 135, 2015.

- [53] L. Breiman, "Bagging predictors," *Mach. Learn.*, vol. 24, p. 123–140, Aug. 1996.
- [54] L. Breiman, "Random forests," *Mach. Learn.*, vol. 45, p. 5–32, Oct. 2001.
- [55] F. Rosenblatt, "The perceptron: A probabilistic model for information storage and organization in the brain.," *Psychological Review*, vol. 65, no. 6, pp. 386–408, 1958.
- [56] C. Sammut and G. I. Webb, eds., *Mean Squared Error*, pp. 653–653. Boston, MA: Springer US, 2010.
- [57] C. Sammut and G. I. Webb, eds., *Mean Absolute Error*, pp. 652–652. Boston, MA: Springer US, 2010.
- [58] N. Srivastava, G. Hinton, A. Krizhevsky, I. Sutskever, and R. Salakhutdinov, "Dropout: A simple way to prevent neural networks from overfitting," *Journal of Machine Learning Research*, vol. 15, no. 56, pp. 1929–1958, 2014.
- [59] L. Bottou, "Large-scale machine learning with stochastic gradient descent," in *Proceedings of COMPSTAT'2010* (Y. Lechevallier and G. Saporta, eds.), (Heidelberg), pp. 177–186, Physica-Verlag HD, 2010.
- [60] J. Duchi, E. Hazan, and Y. Singer, "Adaptive subgradient methods for online learning and stochastic optimization," *Journal of Machine Learning Research*, vol. 12, no. 61, pp. 2121–2159, 2011.
- [61] D. P. Kingma and J. Ba, "Adam: A Method for Stochastic Optimization," *arXiv:1412.6980 [cs]*, Jan. 2017. arXiv: 1412.6980.
- [62] D. E. Rumelhart, G. E. Hinton, and R. J. Williams, "Learning internal representations by error propagation," in *Parallel Distributed Processing: Explorations in the Microstructure of Cognition, Volume 1: Foundations* (D. E. Rumelhart and J. L. McClelland, eds.), pp. 318–362, Cambridge, MA: MIT Press, 1986.
- [63] 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, 1998.
- [64] Y. LeCun, Y. Bengio, and G. Hinton, "Deep learning," *Nature*, vol. 521, pp. 436–444, May 2015.
- [65] "Deep learning models of the retinal response to natural scenes - github repository." <https://github.com/baccuslab/deep-retina>.

- [66] M. Abadi *et al.*, “TensorFlow: Large-Scale Machine Learning on Heterogeneous Distributed Systems,” *arXiv:1603.04467 [cs]*, Mar. 2016. arXiv: 1603.04467.
- [67] F. Chollet *et al.*, “Keras.” <https://github.com/fchollet/keras>, 2015.
- [68] F. Pedregosa *et al.*, “Scikit-learn: Machine learning in Python,” *Journal of Machine Learning Research*, vol. 12, pp. 2825–2830, 2011.
- [69] A. B. Jung *et al.*, “imgaug.” <https://github.com/aleju/imgaug>, 2020. Online; accessed 01-Feb-2020.
- [70] “Tensorflow keras datasets, cifar10.” https://www.tensorflow.org/api_docs/python/tf/keras/datasets/cifar10.
- [71] “Tensorflow keras datasets, mnist.” https://www.tensorflow.org/api_docs/python/tf/keras/datasets/mnist.
- [72] “Tensorflow keras datasets, fashion_mnist.” https://www.tensorflow.org/api_docs/python/tf/keras/datasets/fashion_mnist.
- [73] “Tensorflow datasets, imagenette.” <https://www.tensorflow.org/datasets/catalog/imagenette>.
- [74] S. Ioffe and C. Szegedy, “Batch normalization: Accelerating deep network training by reducing internal covariate shift,” *arXiv:1502.03167 [cs]*, Mar 2015. arXiv: 1502.03167.
- [75] J. Benesty, J. Chen, Y. Huang, and I. Cohen, *Pearson Correlation Coefficient*, pp. 1–4. Berlin, Heidelberg: Springer Berlin Heidelberg, 2009.
- [76] G. Nakopoulos, “Investigation of retinal response at salient image regions (in preparation),” 2021. Diploma Thesis, BioSim Lab, National Technical University of Athens.
- [77] C. Shorten and T. Khoshgoftaar, “A survey on image data augmentation for deep learning,” *Journal of Big Data*, vol. 6, pp. 1–48, 2019.
- [78] J. Lindsey, S. A. Ocko, S. Ganguli, and S. Deny, “A unified theory of early visual representations from retina to cortex through anatomically constrained deep cnns,” *bioRxiv*, 2019.
- [79] N. Parthasarathy *et al.*, “Neural networks for efficient bayesian decoding of natural images from retinal neurons,” *bioRxiv*, 2018.



ΕΘΝΙΚΟ ΜΕΤΣΟΒΙΟ ΠΟΛΥΤΕΧΝΕΙΟ

ΣΧΟΛΗ ΗΛΕΚΤΡΟΛΟΓΩΝ ΜΗΧΑΝΙΚΩΝ ΚΑΙ ΜΗΧΑΝΙΚΩΝ ΥΠΟΛΟΓΙΣΤΩΝ

ΤΟΜΕΑΣ ΣΥΣΤΗΜΑΤΩΝ ΜΕΤΑΔΟΣΗΣ ΠΛΗΡΟΦΟΡΙΑΣ ΚΑΙ ΤΕΧΝΟΛΟΓΙΑΣ ΥΛΙΚΩΝ

Λειτουργική αξιολόγηση μοντέλων αμφιβληστροειδούς με χρήση τεχνικών μηχανικής μάθησης

ΔΙΠΛΩΜΑΤΙΚΗ ΕΡΓΑΣΙΑ

ΤΟΥ

ΝΙΚΟΛΑ ΠΑΠΑΔΟΠΟΥΛΟΥ

Επιβλέπουσα: Κωνσταντίνα Νικήτα
Καθηγήτρια, Ε.Μ.Π.

Αθήνα, Ιανουάριος 2021



ΕΘΝΙΚΟ ΜΕΤΣΟΒΙΟ ΠΟΛΥΤΕΧΝΕΙΟ

ΣΧΟΛΗ ΗΛΕΚΤΡΟΛΟΓΩΝ ΜΗΧΑΝΙΚΩΝ ΚΑΙ ΜΗΧΑΝΙΚΩΝ ΥΠΟΛΟΓΙΣΤΩΝ

ΤΟΜΕΑΣ ΣΥΣΤΗΜΑΤΩΝ ΜΕΤΑΔΟΣΗΣ ΠΛΗΡΟΦΟΡΙΑΣ ΚΑΙ ΤΕΧΝΟΛΟΓΙΑΣ ΥΛΙΚΩΝ

Λειτουργική αξιολόγηση μοντέλων αμφιβληστροειδούς με χρήση τεχνικών μηχανικής μάθησης

ΔΙΠΛΩΜΑΤΙΚΗ ΕΡΓΑΣΙΑ

ΤΟΥ

ΝΙΚΟΛΑ ΠΑΠΑΔΟΠΟΥΛΟΥ

Επιβλέπουσα: Κωνσταντίνα Νικήτα
Καθηγήτρια, Ε.Μ.Π.

Εγκρίθηκε από την τριμελή εξεταστική επιτροπή την 18η Ιανουαρίου 2021.

(Υπογραφή)

(Υπογραφή)

(Υπογραφή)

.....
Κωνσταντίνα Νικήτα
Καθηγήτρια, Ε.Μ.Π.

.....
Ανδρέας-Γεώργιος Σταφυλοπάτης
Καθηγητής, Ε.Μ.Π.

.....
Γεώργιος Στάμου
Αν. Καθηγητής, Ε.Μ.Π.

Αθήνα, Ιανουάριος 2021



ΕΘΝΙΚΟ ΜΕΤΣΟΒΙΟ ΠΟΛΥΤΕΧΝΕΙΟ

ΣΧΟΛΗ ΗΛΕΚΤΡΟΛΟΓΩΝ ΜΗΧΑΝΙΚΩΝ ΚΑΙ ΜΗΧΑΝΙΚΩΝ ΥΠΟΛΟΓΙΣΤΩΝ

ΤΟΜΕΑΣ ΣΥΣΤΗΜΑΤΩΝ ΜΕΤΑΔΟΣΗΣ ΠΛΗΡΟΦΟΡΙΑΣ ΚΑΙ ΤΕΧΝΟΛΟΓΙΑΣ ΥΛΙΚΩΝ

(Υπογραφή)

.....
**Νικόλας
Παπαδόπουλος**

Διπλωματούχος Ηλεκτρολόγος Μηχανικός και Μηχανικός Υπολογιστών Ε.Μ.Π.

Copyright © Νικόλας Παπαδόπουλος, 2021.

Με την επιφύλαξη παντός δικαιώματος. All rights reserved.

Απαγορεύεται η αντιγραφή, αποθήκευση και διανομή της παρούσας εργασίας, εξ ολοκλήρου ή τμήματος αυτής, για εμπορικό σκοπό. Επιτρέπεται η ανατύπωση, αποθήκευση και διανομή για σκοπό μη κερδοσκοπικό, εκπαιδευτικής ή ερευνητικής φύσης, υπό την προϋπόθεση να αναφέρεται η πηγή προέλευσης και να διατηρείται το παρόν μήνυμα. Ερωτήματα που αφορούν τη χρήση της εργασίας για κερδοσκοπικό σκοπό πρέπει να απευθύνονται προς τον συγγραφέα.

Οι απόψεις και τα συμπεράσματα που περιέχονται σε αυτό το έγγραφο εκφράζουν τον συγγραφέα και δεν πρέπει να ερμηνευθεί ότι αντιπροσωπεύουν τις επίσημες θέσεις του Εθνικού Μετσόβιου Πολυτεχνείου.

Περίληψη

Η παρούσα διπλωματική εργασία προτείνει τη λειτουργική αξιολόγηση (functional assessment) μοντέλων αμφιβληστροειδούς (retinal models), ως εναλλακτική στην καθιερωμένη αξιολόγηση των μοντέλων, που βασίζεται στην πιστή αναπαραγωγή της νευρικής απόκρισης. Η λειτουργική αξιολόγηση περιγράφει την ιδέα της αξιολόγησης της απόδοσης ενός μοντέλου μέσα από δοκιμασίες κατανόησης της οπτικής πληροφορίας. Στην εργασία αυτή, αναπτύξαμε ένα σύστημα λειτουργικής αξιολόγησης μοντέλων αμφιβληστροειδούς με χρήση τεχνικών μηχανικής μάθησης, όπου εισάγουμε εικόνες σε ένα μοντέλο αμφιβληστροειδούς, λαμβάνουμε τις νευρικές αποκρίσεις του μοντέλου, με τις οποίες μετά εκπαιδεύουμε ταξινομητές πάνω σε πειράματα αναγνώρισης αντικειμένου και αναγνώρισης ψηφίων. Συγκεκριμένα, χρησιμοποιήσαμε για αναγνώριση αντικειμένων τα σύνολα δεδομένων CIFAR10, Fashion MNIST και Imagenette και για αναγνώριση ψηφίων το MNIST, ενώ εκπαιδεύσαμε διάφορους ταξινομητές μηχανικής μάθησης, όπως Multilayer Perceptron (MLP), Support Vector Machine (SVM) και Random Forest. Εξετάσαμε μια σειρά σχετικών ερωτημάτων: ποιοι ταξινομητές αποδίδουν καλύτερα σε τέτοιου είδους πειράματα, πώς διαφέρει η επίδοση ενός μοντέλου ανάλογα με την δοκιμασία (dataset) και πώς επηρεάζει ο αριθμός των παραγόμενων νευρικών αποκρίσεων την επίδοση των μοντέλων. Δεδομένου επίσης ότι το Convolutional Neural Network (CNN) μοντέλο αμφιβληστροειδούς που χρησιμοποιήσαμε παρέχει έναν περιορισμένο αριθμό νευρικών αποκρίσεων, μελετήσαμε τρόπους να αυξήσουμε τεχνητά αυτόν τον αριθμό. Για τον σκοπό αυτό, διαχωρίσαμε τις εικόνες σε διάφορα τμήματα/μέρη, τροφοδοτήσαμε το μοντέλο αμφιβληστροειδούς με κάθε τμήμα ξεχωριστά και διερευνήσαμε το πώς μπορούν να συνδυαστούν βέλτιστα οι νευρικές αποκρίσεις που δίνει το μοντέλο για κάθε κομμάτι, έτσι ώστε να βελτιώσουμε την επίδοση στα πειράματα αναγνώρισης εικόνας. Βρήκαμε ότι ο ταξινομητής Random Forest επιτυγχάνει την καλύτερη επίδοση και ότι τα μοντέλα αποδίδουν καλύτερα σε δοκιμασίες με σύνολα δεδομένων όπως το MNIST και το Fashion MNIST, όπου πετύχαμε ακρίβεια μέχρι και 90%. Επιπλέον, τα πειράματα έδειξαν ότι η χρήση περισσότερων νευρώνων βελτιώνει τη συνολική απόδοση των μοντέλων, αναδεικνύοντας έτσι την ανάγκη για κατασκευή εμφυτευμάτων αμφιβληστροειδούς που να διαθέτουν μεγαλύτερο αριθμό ηλεκτροδίων. Τέλος, εφαρμόσαμε την τεχνική της λειτουργικής αξιολόγησης σε δύο διαφορετικά μοντέλα αμφιβληστροειδούς. Τα αποτελέσματα έδειξαν ότι υπάρχει συμφωνία μεταξύ της καθιερωμένης και της λειτουργικής αξιολόγησης, καθώς τα δύο μοντέλα είχαν παρόμοιες επιδόσεις στις δύο τεχνικές. Καταλήγουμε επομένως ότι η λειτουρ-

γική αξιολόγηση μπορεί να δώσει αξιόπιστα αποτελέσματα και να χρησιμοποιηθεί ως εναλλακτική στην καθιερωμένη τεχνική αξιολόγησης.

Λέξεις Κλειδιά

μοντέλα αμφιβληστροειδούς, λειτουργική αξιολόγηση, προσθετική αμφιβληστροειδούς, μηχανική μάθηση, νευρωνικά δίκτυα, ταξινόμηση, αναγνώριση ψηφίων, αναγνώριση αντικειμένων

Abstract

This diploma thesis proposes the functional assessment of retinal models, as an alternative to the currently common practice of comparing the similarity of model-generated and ground truth neural responses, as recorded by retinal implants. Functional assessment describes the concept of evaluating the performance of retinal models on image understanding tasks. In this work, we developed a pipeline for functional assessment using machine learning techniques, where we feed retinal models with images and we receive the neural responses of the model, with which we train classifiers on object and digit recognition tasks. In particular, we used CIFAR10, Fashion MNIST and Imagenette datasets for object recognition tasks and MNIST dataset for digit recognition tasks. We also trained common classifiers, such as Multilayer Perceptron (MLP), Support Vector Machine (SVM) and Random Forest. We examined a number of relevant questions: which classifiers perform better in such type of tasks, how the performance of a model differs depending on the task and how the number of available neurons affects the performance of the model. Given also that the Convolutional Neural Network (CNN) retinal model provides a limited number of available neurons (60 in our case), we investigated ways to artificially increase this number. For that, we split the images into different parts, we fed separately each part to the retinal model and we investigated how we can optimally combine the neural responses produced by the model for each part, in order to achieve high performance in image recognition tasks. We found that Random Forest classifier achieves the highest performance and that models perform better with datasets such as MNIST and Fashion MNIST, where we achieved up to 90% accuracy. In addition, experiments indicated that using more neurons improves the overall performance of the model, thus highlighting the need to design retinal prostheses with a larger number of available neurons. Finally, we applied functional assessment in order to compare the performance of two different retinal models. The results showed that there is an agreement between standard and functional assessment, as the two models had similar performance in both techniques. Thus, we conclude that functional assessment produces reliable results and it can be used as an alternative to the standard evaluation technique.

Keywords

retinal models, functional assessment, retinal prosthesis, machine learning, neural networks, classification, digit recognition, object recognition

*“Γιατί τυφλωθήκαμε, Δεν ξέρω, ίσως μια μέρα να καταφέρουμε να μάθουμε το λόγο,
Θέλεις να σου πω αυτό που νομίζω, Λέγε, Νομίζω ότι δεν τυφλωθήκαμε, νομίζω ότι
είμαστε τυφλοί, Τυφλοί που βλέπουν, Τυφλοί που δεν βλέπουν, κι ας βλέπουν.”*

–Περί τυφλότητας, Jose Saramago

Ευχαριστίες

Θα ήθελα να ευχαριστήσω την καθηγήτρια Κωνσταντίνα Νικήτα. Μου έδωσε την ευκαιρία να ασχοληθώ με ένα τόσο ενδιαφέρον θέμα, ήταν παρούσα σε κάθε ζήτημα που προέκυπτε και επέμεινε στις λεπτομέρειες.

Θα ήθελα να ευχαριστήσω τον υπ.Διδάκτορα Νίκο Μελανίτη-Παρασκευά. Οι συμβο(υ)λές του και η ανταπόκρισή του ήταν τέτοιες, που δε μπορούσα να φανταστώ καλύτερες.

Θα ήθελα να ευχαριστήσω τον συμφοιτητή μου Γιώργο Νακόπουλο. Μοιραστήκαμε παρόμοια θέματα και βοήθησε σημαντικά στην εργασία μου.

Θα ήθελα επίσης να ευχαριστήσω τον καθηγητή Βασίλειο Μακίό που ενορχηστώνει τόσο μοναδικά ένα κουαρτέτο, στο οποίο έχω τη χαρά να είμαι μέλος.

Τέλος, θα ήθελα να ευχαριστήσω την οικογένειά μου που πάντα ισορροπεί τα πράγματα, αλλά και τους φίλους μου που πάντα τα εξελίσσουν.

Αθήνα, Ιανουάριος 2021

Νικόλας Παπαδόπουλος

Περιεχόμενα

Περίληψη	1
Abstract	3
Ευχαριστίες	7
Περιεχόμενα	9
Κατάλογος Σχημάτων	12
Κατάλογος Πινάκων	16
1 Εισαγωγή	19
1.1 Κίνητρο	19
1.2 Συνεισφορά της Διπλωματικής Εργασίας	20
1.3 Διάρθρωση της Διπλωματικής Εργασίας	21
2 Φυσιολογία και Προσθετική Αμφιβληστροειδούς	23
2.1 Φυσιολογία του Ανθρώπινου Οφθαλμού	23
2.2 Φυσιολογία του Αμφιβληστροειδούς	23
2.3 Υποδεκτικό Πεδίο των Γαγγλιακών Κυττάρων	24
2.4 Νευρικές Αποκρίσεις του Αμφιβληστροειδούς	26
2.5 Μοντέλα Αμφιβληστροειδούς	26
2.6 Προσθετική Αμφιβληστροειδούς	27
3 Μηχανική Μάθηση	29
3.1 Εισαγωγή	29
3.1.1 Επιβλεπόμενη Μάθηση	29
3.1.2 Μη Επιβλεπόμενη Μάθηση	30
3.1.3 Ενισχυτική Μάθηση	30
3.2 Ταξινομητές Μηχανικής Μάθησης	30
3.2.1 Τυχαία Δάση - Random Forest	30
3.2.2 Μηχανές Διανυσμάτων Υποστήριξης - SVM	31
3.2.3 Πολυεπίπεδα Νευρωνικά Δίκτυα - MLP	31

4	Ανάπτυξη Μεθοδολογίας Λειτουργικής Αξιολόγησης	33
4.1	Πλαίσιο Λειτουργικής Αξιολόγησης	33
4.2	Περιγραφή Εργαλείων Λογισμικού	34
4.3	Περιγραφή Συνόλων Δεδομένων	34
4.4	Περιγραφή του Μοντέλου Αμφιβληστροειδούς των Προσομοιώσεων	34
4.5	Ανάλυση Βημάτων Προεπεξεργασίας Δεδομένων	38
4.6	Οργάνωση των Προσομοιώσεων	43
5	Αποτελέσματα	47
5.1	Απόδοση Ταξινομητών στο Σύνολο Δεδομένων CIFAR10	47
5.2	Απόδοση Ταξινομητών στο Σύνολο Δεδομένων MNIST	48
5.3	Απόδοση Ταξινομητών στο Σύνολο Δεδομένων Fashion MNIST	49
5.4	Απόδοση Ταξινομητών στο Σύνολο Δεδομένων Imagenette	50
5.5	Συγκριτική Ανάλυση Αποτελεσμάτων	54
5.5.1	Αξιολόγηση της απόδοσης των ταξινομητών σαν συνάρτηση του αριθμού των νευρώνων	54
5.5.2	Σύγκριση της απόδοσης των ταξινομητών μεταξύ των συνόλων δεδομένων.	54
5.5.3	Σύγκριση της απόδοσης των ταξινομητών μεταξύ των προσομοι- ώσεων με "Adjust=yes" και "Adjust=no"	55
5.5.4	Σύγκριση της απόδοσης των ταξινομητών μεταξύ των μοντέλων αμφιβληστροειδούς.	55
6	Συζήτηση	63
6.1	Συμπεράσματα	63
6.1.1	Προεπεξεργασία Δεδομένων	63
6.1.2	Σύνολα Δεδομένων	64
6.1.3	Ταξινομητές	65
6.1.4	Μοντέλα Αμφιβληστροειδούς	65
6.2	Περιορισμοί της Λειτουργικής Αξιολόγησης	66
7	Επίλογος	67
7.1	Σύνοψη	67
7.2	Μελλοντικές Προεκτάσεις	68

Παραρτήματα	69
Α' Επαναλήψεις εικόνων στο χρονικό σύνολο δεδομένων	71
Βιβλιογραφία	78

Κατάλογος Σχημάτων

2.1	Κάθετη και οριζόντια τομή του ανθρώπινου οφθαλμού [1], [2].	24
2.2	Σχηματική αναπαράσταση του υποδεκτικού πεδίου ενός κυττάρου τύπου ON (αριστερά) κι ενός κυττάρου τύπου OFF (δεξιά) [3].	25
2.3	Υποδεκτικό πεδίο ενός γαγγλιακού κυττάρου τύπου ON, που αναπαρίστανται ως διαφορά δύο γκαουσιανών συναρτήσεων (difference of Gaussians) [4].	25
2.4	Διαφορετικές τεχνικές για τον υπολογισμό νευρικών αποκρίσεων [5].	26
2.5	Το μοντέλο Linear - Nonlinear (LN) [6].	27
2.6	Το μοντέλο Generalized Linear Model (GLM) [7].	27
2.7	Επισκόπηση ενός συστήματος προσθετικής αμφιβληστροειδούς. Μια εικόνα εισέρχεται ως είσοδος μέσω μιας κάμερας και μετατρέπεται σε ηλεκτρικό σήμα από μια μονάδα επεξεργασίας. Το ηλεκτρικό σήμα μεταδίδεται μέσω τηλεμετρίας ραδιοσυχνότητας (RF) ή οπτικής ζεύξης και υφίσταται επεξεργασία από εξωτερική ή εμφυτευμένη διάταξη. Το κύκλωμα παράγει ηλεκτρικό ερέθισμα, που υποκαθιστά το φωτεινό ερέθισμα και εφαρμόζεται στον αμφιβληστροειδή μέσω μιας συστοιχίας μικροηλεκτροδίων τοποθετημένης κοντά στον αμφιβληστροειδή [8].	28
3.1	Αναπαράσταση του αλγορίθμου Random Forest [9].	31
3.2	Αναπαράσταση του αλγορίθμου SVM για διαχωρισμό δύο κλάσεων (Wikipedia).	32
3.3	Δομή ενός ταξινομητή MLP με δύο κρυφά επίπεδα [10].	32
4.1	Μέθοδος λειτουργικής αξιολόγησης.	33
4.2	Αρχιτεκτονική του CNN μοντέλου αμφιβληστροειδούς που χρησιμοποιήθηκε σε αυτή την εργασία [11].	34
4.3	Δείγματα εικόνων από το σύνολο δεδομένων με το οποίο εκπαιδεύσαμε το μοντέλο αμφιβληστροειδούς [12].	37
4.4	Μέρος του temporal dataset με <i>temporal_interval</i> = 30. Η αριστερότερη στήλη κάθε σειράς περιέχει την εικόνα που προβάλλεται εκείνη τη στιγμή στον αμφιβληστροειδή. Οι υπόλοιπες στήλες περιέχουν το ιστορικό των εικόνων που προηγήθηκαν. Φαίνονται επίσης τα αρχικά μαύρα πλαίσια.	37

4.5	Παράδειγμα της τεχνικής επαύξησης δεδομένων.	39
4.6	Παράδειγμα χωρισμού εικόνας σε εννιά τμήματα/μέρη ($p = 3$).	40
4.7	Ένα μέρος του temporal dataset. Κάθε γραμμή στο temporal dataset αναπαριστά ένα δείγμα εισόδου στο μοντέλο αμφιβληστροειδούς και έχει τόσες εικόνες, όσο και η τιμή της μεταβλητής temporal_interval (εδώ, temporal_interval=30). Η εικόνα στην αριστερότερη στήλη αναπαριστά την εικόνα που βλέπει ο αμφιβληστροειδής εκείνη τη στιγμή και οι υπόλοιπες εικόνες αναπαριστούν το ιστορικό των προηγούμενων εικόνων που έβλεπε ο αμφιβληστροειδής. Τα μαύρα πλαίσια αναπαριστούν το σκοτάδι στο οποίο βρίσκεται ο αμφιβληστροειδής πριν δει κάθε εικόνα.	40
5.1	Διαγράμματα ανάλυσης ευαισθησίας με χρήση του RetModel1 (Ενότητα 4.4) για διαφορετικά σύνολα δεδομένων (Πίνακας 4.1), διαφορετικές προσομοιώσεις (Πίνακας 4.5) και διαφορετικούς συνδυασμούς παραμέτρων (Πίνακας 4.4). (α) Σύνολο δεδομένων CIFAR10, προσομοιώσεις A2 & D1 ("Adjust=yes"). (β) Σύνολο δεδομένων CIFAR10, προσομοιώσεις B1 & C1 ("Adjust=no"). (γ) Σύνολο δεδομένων Imagenette, προσομοιώσεις A2 & D1 ("Adjust=yes"). (δ) Σύνολο δεδομένων Imagenette, προσομοιώσεις B1 & C1 ("Adjust=no").	56
5.2	Διαγράμματα ανάλυσης ευαισθησίας με χρήση του RetModel1 (Ενότητα 4.4) για διαφορετικά σύνολα δεδομένων (Πίνακας 4.1), διαφορετικές προσομοιώσεις (Πίνακας 4.5) και διαφορετικούς συνδυασμούς παραμέτρων (Πίνακας 4.4). (α) Σύνολο δεδομένων MNIST, προσομοιώσεις A2 & D1 ("Adjust=yes"). (β) Σύνολο δεδομένων MNIST, προσομοιώσεις B1 & C1 ("Adjust=no"). (γ) Σύνολο δεδομένων Fashion MNIST, προσομοιώσεις A2 & D1 ("Adjust=yes"). (δ) Σύνολο δεδομένων Fashion MNIST, προσομοιώσεις B1 & C1 ("Adjust=no").	57
5.3	Σύγκριση της μέγιστης απόδοσης των ταξινομητών μεταξύ διαφορετικών συνόλων δεδομένων (Πίνακας 4.1) για τα δύο μοντέλα αμφιβληστροειδούς (Ενότητα 4.4). (α) Σύγκριση μεταξύ CIFAR10 και Imagenette για το RetModel1. (β) Σύγκριση μεταξύ CIFAR10 και Imagenette για το RetModel2. (γ) Σύγκριση μεταξύ MNIST και Fashion MNIST για το RetModel1. (δ) Σύγκριση μεταξύ MNIST και Fashion MNIST για το RetModel2.	58

5.4	Σύγκριση της μέγιστης απόδοσης των ταξινομητών μεταξύ των προσομοιώσεων με "Adjust=yes" και αυτών με "Adjust=no" (Πίνακα 4.4) για διαφορετικά σύνολα δεδομένων (Πίνακας 4.1) και για τα δύο μοντέλα αμφιβληστροειδούς (Ενότητα 4.4). (α) Σύγκριση για το CIFAR10 με χρήση του RetModel1. (β) Σύγκριση για το CIFAR10 με χρήση του RetModel2. (γ) Σύγκριση για το Imagenette με χρήση του RetModel1. (δ) Σύγκριση για το Imagenette με χρήση του RetModel2.	59
5.5	Σύγκριση της μέγιστης απόδοσης των ταξινομητών μεταξύ των προσομοιώσεων με "Adjust=yes" και αυτών με "Adjust=no" (Πίνακα 4.4) για διαφορετικά σύνολα δεδομένων (Πίνακας 4.1) και για τα δύο μοντέλα αμφιβληστροειδούς (Ενότητα 4.4). (α) Σύγκριση για το MNIST με χρήση του RetModel1. (β) Σύγκριση για το MNIST με χρήση του RetModel2. (γ) Σύγκριση για το Fashion MNIST με χρήση του RetModel1. (δ) Σύγκριση για το Fashion MNIST με χρήση του RetModel2.	60
5.6	Σύγκριση της μέγιστης απόδοσης των ταξινομητών ανάμεσα στα μοντέλα αμφιβληστροειδούς RetModel1 και RetModel2 (Ενότητα 4.4) για όλα τα σύνολα δεδομένων (Πίνακας 4.1). (α) Σύγκριση στο CIFAR10. (β) Σύγκριση στο Imagenette. (γ) Σύγκριση στο MNIST. (δ) Σύγκριση στο Fashion MNIST.	61
A'.1	Παράδειγμα μιας γραμμής στο temporal dataset.	71
A'.2	Διαγράμματα που αναπαριστούν τις νευρικές αποκρίσεις του νευρώνα_5 συναρτήσει του διαφορετικού αριθμού εικόνων σε κάθε σειρά. Σε κάθε διάγραμμα δίνουμε διαφορετικό αριθμό εικόνων ως είσοδο στο μοντέλο και αντίστοιχα παίρνουμε διαφορετικό αριθμό εξόδων (δηλαδή νευρικών αποκρίσεων). Στο Σχήμα A2γ, σχεδιάζουμε το διάγραμμα μέσης τιμής/τυπικής απόκλισης, καθώς δίνουμε πεντακόσιες εικόνες ως είσοδο.	72

Κατάλογος Πινάκων

4.1	Περιγραφή των συνόλων δεδομένων που χρησιμοποιήθηκαν στις προσομοιώσεις.	35
4.2	Σύνοψη όλων των επιπέδων του CNN μοντέλου αμφίβληστροειδούς που χρησιμοποιήθηκε σε αυτή την εργασία.	36
4.3	Τα βήματα της προεπεξεργασίας δεδομένων.	42
4.4	Το εύρος τιμών για τις παραμέτρους των προσομοιώσεων. Τα σύνολα δεδομένων παρουσιάζονται στον Πίνακα 4.1. Οι υπόλοιπες παράμετροι αναλύονται σε αυτή την ενότητα.	43
4.5	Πέντε τύποι προσομοιώσεων που αφορούν διαφορετικές διαδικασίες προεπεξεργασίας.	44
4.6	Σύνοψη των προσομοιώσεων που έγιναν ανά σύνολο δεδομένων. Όλες οι διαφορετικές τιμές που μπορούν να πάρουν οι παράμετροι των προσομοιώσεων φαίνονται στον Πίνακα 4.4. Τα σύνολα δεδομένων παρουσιάζονται στον Πίνακα 4.1	45
5.1	Μέση τιμή και τυπική απόκλιση της απόδοσης των ταξινομητών στο σύνολο δεδομένων CIFAR10 (Πίνακας 4.1) με χρήση του RetModel1 (Ενότητα 4.4) για τις προσομοιώσεις A2, D1, B1, C1 (Πίνακες 4.4, 4.5). Οι σειρές που φαίνονται με κίτρινο χρώμα αντιστοιχούν σε προσομοιώσεις με Valid=yes, ενώ αυτές με γαλάζιο αντιστοιχούν σε προσομοιώσεις με Valid=no.	48
5.2	Μέση τιμή και τυπική απόκλιση της απόδοσης των ταξινομητών στο σύνολο δεδομένων CIFAR10 (Πίνακας 4.1) με χρήση του RetModel2 (Ενότητα 4.4) για τις προσομοιώσεις A2, C1 (Πίνακες 4.4, 4.5).	49
5.3	Μέση τιμή και τυπική απόκλιση της απόδοσης των ταξινομητών στο σύνολο δεδομένων MNIST (Πίνακας 4.1) με χρήση του RetModel1 (Ενότητα 4.4) για τις προσομοιώσεις A2, D1, B1, C1 (Πίνακες 4.4, 4.5). Οι σειρές που φαίνονται με κίτρινο χρώμα αντιστοιχούν σε προσομοιώσεις με Valid=yes, ενώ αυτές με γαλάζιο αντιστοιχούν σε προσομοιώσεις με Valid=no.	50

5.4	Μέση τιμή και τυπική απόκλιση της απόδοσης των ταξινομητών στο σύνολο δεδομένων MNIST (Πίνακας 4.1) με χρήση του RetModel2 (Ενότητα 4.4) για τις προσομοιώσεις A2, C1 (Πίνακες 4.4, 4.5).	51
5.5	Μέση τιμή και τυπική απόκλιση της απόδοσης των ταξινομητών στο σύνολο δεδομένων Fashion MNIST (Πίνακας 4.1) με χρήση του RetModel1 (Ενότητα 4.4) για τις προσομοιώσεις A2, C1 (Πίνακες 4.4, 4.5).	51
5.6	Μέση τιμή και τυπική απόκλιση της απόδοσης των ταξινομητών στο σύνολο δεδομένων Fashion MNIST (Πίνακας 4.1) με χρήση του RetModel2 (Ενότητα 4.4) για τις προσομοιώσεις A2, C1 (Πίνακες 4.4, 4.5).	52
5.7	Μέση τιμή και τυπική απόκλιση της απόδοσης των ταξινομητών στο σύνολο δεδομένων Imagenette (Πίνακας 4.1) με χρήση του RetModel1 (Ενότητα 4.4) για τις προσομοιώσεις A2, C1 (Πίνακες 4.4, 4.5).	52
5.8	Μέση τιμή και τυπική απόκλιση της απόδοσης των ταξινομητών στο σύνολο δεδομένων Imagenette (Πίνακας 4.1) με χρήση του RetModel2 (Ενότητα 4.4) για τις προσομοιώσεις A2, C1 (Πίνακες 4.4, 4.5).	53
A.1	Διαγράμματα μέσης τιμής και τυπικής απόκλισης για οχτώ διαφορετικούς νευρώνες.	73

Κεφάλαιο **1**

Εισαγωγή

1.1 Κίνητρο

Στη παρούσα διπλωματική εργασία μελετήσαμε τη λειτουργική αξιολόγηση μοντέλων αμφιβληστροειδούς με τη χρήση τεχνικών μηχανικής μάθησης. Στόχος μας είναι να προτείνουμε μια εναλλακτική στην καθιερωμένη πρακτική αξιολόγησης μοντέλων αμφιβληστροειδούς, που βασίζεται στην πιστή αναπαραγωγή της νευρικής απόκρισης.

Η τύφλωση και γενικότερα τα προβλήματα όρασης, είναι μεταξύ αυτών των προβλημάτων που επηρεάζουν μεγάλο αριθμό ανθρώπων παγκοσμίως (πάνω από 40 εκατομμύρια [13]) και παράλληλα, επηρεάζουν αισθητά την καθημερινότητα του ασθενούς. Τα εμφυτεύματα αμφιβληστροειδούς (ή προσθετική αμφιβληστροειδούς) στοχεύουν στην αντιμετώπιση ορισμένων από τις πιο δύσκολες ασθένειες του οπτικού συστήματος και για να το καταφέρουν βασίζονται σε προηγμένα μοντέλα αμφιβληστροειδούς, τα οποία προσομοιώνουν τη λειτουργία των νευρικών κυττάρων του αμφιβληστροειδούς [14]. Η ανάπτυξη μοντέλων αμφιβληστροειδούς είναι επίσης ένα εξαιρετικό εργαλείο για την μελέτη των μηχανισμών της όρασης στο ανθρώπινο μάτι [15]. Και στις δύο περιπτώσεις, είναι επιτακτική ανάγκη να αναπτυχθούν αξιόπιστες μέθοδοι για την αξιολόγηση της απόδοσης διαφορετικών μοντέλων αμφιβληστροειδούς.

Σε αυτήν την εργασία, προτείνουμε μια μέθοδο για τη λειτουργική αξιολόγηση μοντέλων αμφιβληστροειδούς. Η λειτουργική αξιολόγηση μοντέλων αμφιβληστροειδούς περιγράφει την ιδέα της αξιολόγησης της απόδοσης ενός μοντέλου μέσα από προβλήματα (tasks) αναγνώρισης εικόνας. Η τεχνική αυτή αποκλίνει από τις συνηθείς πρακτικές αξιολόγησης μοντέλων αμφιβληστροειδούς, οι οποίες αξιολογούν το μοντέλο συγκρίνοντας τις νευρικές αποκρίσεις που αυτό παράγει με αυτές που παράγουν τα βιολογικά πειράματα πάνω σε ζωντανούς οργανισμούς (όπως ποντίκια, σαλαμάνδρες, κλπ.). Το κίνητρο για τη λειτουργική αξιολόγηση μοντέλων αμφιβλη-

στροειδούς πηγάζει από την παρατήρηση ότι τα εμφυτεύματα αμφιβληστροειδούς στοχεύουν στην αποκατάσταση της ικανότητας των ατόμων να κατανοούν το οπτικό τους περιβάλλον. Επομένως, θα πρέπει να αξιολογήσουμε τα μοντέλα μας απευθείας σε τέτοια προβλήματα. Η ανάγκη για λειτουργική αξιολόγηση της προσθετικής όρασης (δηλαδή της όρασης με εμφυτεύματα αμφιβληστροειδούς ή/και άλλα εμφυτεύματα που διεγείρουν σημεία της οπτικής οδού) εμφανίζεται στη βιβλιογραφία [16] και εφαρμογές της υπάρχουν τόσο για την αξιολόγηση της όρασης σε εμφυτεύματα [17], όσο και σε παρεμβάσεις επαυξημένης πραγματικότητας σε άτομα με σοβαρά προβλήματα όρασης [18]. Οι οπτικές λειτουργίες που εμφανίζονται ως πιο σημαντικές για τα άτομα με προβλήματα όρασης είναι η αναγνώριση κίνησης, η αναγνώριση προσώπου και η ανάγνωση [13].

Ένας προτεινόμενος τρόπος υλοποίησης της λειτουργικής αξιολόγησης είναι να τροφοδοτήσουμε την έξοδο του μοντέλου αμφιβληστροειδούς σε ένα νευρωνικό δίκτυο με σκοπό την επίλυση προβλημάτων κατανόησης εικόνας. Στην εργασία αυτή, μελετήσαμε προβλήματα αναγνώρισης αντικειμένων και αναγνώρισης ψηφίων. Συγκεκριμένα, χρησιμοποιήσαμε τα σύνολα δεδομένων CIFAR10 [19], Fashion_MNIST [20] και Imagenette [21] για προβλήματα αναγνώρισης αντικειμένων και το σύνολο δεδομένων MNIST [22] για προβλήματα αναγνώρισης ψηφίων. Για να γίνει αυτό, τροφοδοτήσαμε με εικόνες το μοντέλο αμφιβληστροειδούς και στη συνέχεια τροφοδοτήσαμε τις νευρικές αποκρίσεις του μοντέλου αμφιβληστροειδούς σε ταξινομητές μηχανικής μάθησης, όπως οι MLP, SVM και Random Forest. Λαμβάνοντας υπόψη τον περιορισμένο αριθμό διαθέσιμων νευρώνων που παρέχονται από το μοντέλο αμφιβληστροειδούς, χωρίσαμε τις εικόνες σε επιμέρους τμήματα/μέρη και μετά τροφοδοτήσαμε κάθε τμήμα ξεχωριστά στο μοντέλο του αμφιβληστροειδούς, ώστε να αυξηθεί τεχνητά ο αριθμός των νευρικών αποκρίσεων. Σε αυτή τη βάση, σχεδιάσαμε και βελτιστοποιήσαμε τη διαδικασία με την οποία οι νευρικές αποκρίσεις από κάθε κομμάτι της εικόνας συνδυάζονται. Επιπλέον, αξιολογήσαμε τη χρήση διαφορετικών συνόλων δεδομένων και ταξινομητών στην απόδοση του συστήματος λειτουργικής αξιολόγησης. Στο τέλος, συγκρίναμε δύο διαφορετικά μοντέλα αμφιβληστροειδούς χρησιμοποιώντας τη μέθοδο της λειτουργικής αξιολόγησης.

1.2 Συνεισφορά της Διπλωματικής Εργασίας

Παρακάτω, αναφέρουμε επιγραμματικά τη συνεισφορά της διπλωματικής εργασίας:

- Σχεδιάσαμε ένα καινοτόμο σύστημα λειτουργικής αξιολόγησης μοντέλων αμφιβληστροειδούς.
- Εφαρμόσαμε λειτουργική αξιολόγηση για να συγκρίνουμε την απόδοση με-

ταξύ δύο διαφορετικών μοντέλων αμφιβληστροειδούς. Με αυτόν τον τρόπο, παρέχεται μια εναλλακτική στην καθιερωμένη τεχνική αξιολόγησης.

- Προσαρμόσαμε κατάλληλα τις παραμέτρους κατά τη διάρκεια της προεπεξεργασίας δεδομένων, ώστε να βελτιστοποιήσουμε την απόδοση του συστήματος λειτουργικής αξιολόγησης.
- Αξιολογήσαμε την απόδοση των μοντέλων αμφιβληστροειδούς σε προβλήματα αναγνώρισης αντικειμένων (CIFAR10, Fashion MNIST, Imagenette) και αναγνώρισης ψηφίων (MNIST), μέσω προσομοιώσεων με τέσσερα διαφορετικά σύνολα δεδομένων. Διαπιστώσαμε ότι τα μοντέλα αποδίδουν καλύτερα σε προβλήματα όπως το MNIST και το Fashion MNIST, όπου επιτύχαμε ακρίβεια έως και 90%.

1.3 Διάρθρωση της Διπλωματικής Εργασίας

Στο πρώτο μέρος, περιγράφουμε το απαραίτητο θεωρητικό υπόβαθρο για να κατανοήσει ο αναγνώστης τη δομή και τη συνεισφορά αυτής της διπλωματικής εργασίας. Συγκεκριμένα, στο Κεφάλαιο 2, παρουσιάζουμε πληροφορίες σχετικά με τη φυσιολογία του ανθρώπινου ματιού και κυρίως του αμφιβληστροειδούς. Επίσης, περιγράφουμε τις έννοιες των μοντέλων αμφιβληστροειδούς και της προσθετικής αμφιβληστροειδούς και παρέχουμε σχετικά παραδείγματα. Στο Κεφάλαιο 3, παρουσιάζουμε μια επισκόπηση της θεωρίας της μηχανικής μάθησης, εστιάζοντας στα μοντέλα που χρησιμοποιήσαμε σε αυτήν την εργασία.

Στο δεύτερο μέρος, περιγράφουμε αναλυτικά τις προσομοιώσεις (*in silico experiments*) που πραγματοποιήσαμε. Πιο συγκεκριμένα, στο Κεφάλαιο 4, περιγράφουμε τη μεθοδολογία που χρησιμοποιήθηκε και τις προσομοιώσεις που πραγματοποιήθηκαν προκειμένου να αναπτυχθεί ένας μηχανισμός για τη λειτουργική αξιολόγηση των μοντέλων αμφιβληστροειδούς. Στο Κεφάλαιο 5, αναλύουμε τα αποτελέσματα των προσομοιώσεων και παρουσιάζουμε συγκριτικούς πίνακες για να αναδείξουμε τη σημασία των αποτελεσμάτων. Στο Κεφάλαιο 6, συζητάμε περαιτέρω τα ευρήματα των προσομοιώσεων.

Τέλος, στο Κεφάλαιο 7, παρουσιάζουμε μια περίληψη της εργασίας, καθώς και μελλοντικές κατευθύνσεις για τη βελτίωση της λειτουργικής αξιολόγησης των μοντέλων αμφιβληστροειδούς.

Κεφάλαιο **2**

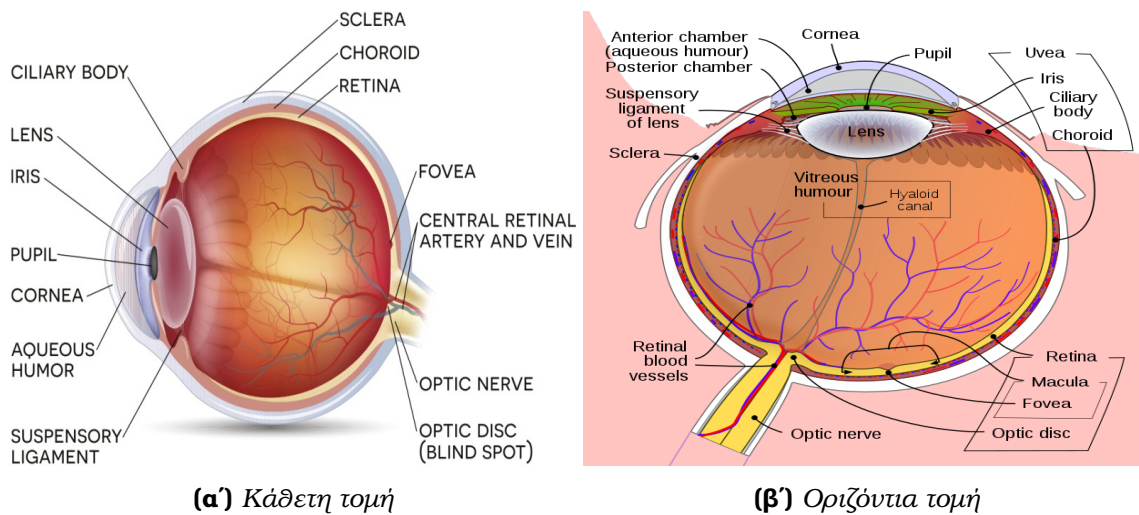
Φυσιολογία και Προσθετική Αμφιβληστροειδούς

2.1 Φυσιολογία του Ανθρώπινου Οφθαλμού

Τα βασικά μέρη του ανθρώπινου οπτικού συστήματος είναι ο αμφιβληστροειδής χιτώνας (retina), ο έξω (πλευρικός) γωνιατός πυρήνας (lateral geniculate nucleus (LGN)) και ο οπτικός φλοιός (visual cortex). Ο αμφιβληστροειδής χιτώνας αποτελεί ένα σημαντικό επίπεδο της οπτικής επεξεργασίας, καθώς μεταφέρει την πληροφορία, η οποία αναπαριστά μία εικόνα του εξωτερικού περιβάλλοντος, στα μέρη εκείνα του εγκεφάλου που είναι υπεύθυνα για την οπτική επεξεργασία. Από την άλλη, ο οπτικός φλοιός είναι η περιοχή εκείνη του εγκεφάλου όπου πραγματοποιείται το μεγαλύτερο μέρος της οπτικής επεξεργασίας, όπως η αναγνώριση αντικειμένων.

2.2 Φυσιολογία του Αμφιβληστροειδούς

Στο Σχήμα 2.1, μπορούμε να δούμε μία σχηματική αναπαράσταση της τομής του ανθρώπινου οφθαλμού. Το μέρος στη πίσω πλευρά του οφθαλμού είναι ο αμφιβληστροειδής χιτώνας. Φωτεινές ακτίνες εισέρχονται στον οφθαλμό μέσω του κερατοειδούς χιτώνα (cornea) και του φακού (lens), οι οποίοι τις εστιάζουν σε ένα σημείο του αμφιβληστροειδούς χιτώνα που ονομάζεται βοθρίο (fovea). Το βοθρίο διαθέτει υψηλότερη ανάλυση σε σχέση με τον υπόλοιπο αμφιβληστροειδή, καθώς υπάρχει μεγάλη συγκέντρωση φωτοϋποδοχέων (photoreceptors) στη συγκεκριμένη περιοχή. Η περιοχή του αμφιβληστροειδούς γύρω από το βοθρίο ονομάζεται ωχρά κηλίδα (macula). Η ωχρά κηλίδα και πιο συγκεκριμένα το βοθρίο είναι υπεύθυνα για την έγχρωμη όραση υψηλής ευκρίνειας υπό καλές συνθήκες φωτισμού και δίνουν στους ανθρώπους τη δυνατότητα να διαβάζουν και να βλέπουν με μεγάλη λεπτομέρεια. Τα υπόλοιπα μέρη του αμφιβληστροειδούς χιτώνα εξασφαλίζουν την περιφερειακή όραση.



Σχήμα 2.1. Κάθετη και οριζόντια τομή του ανθρώπινου οφθαλμού [1], [2].

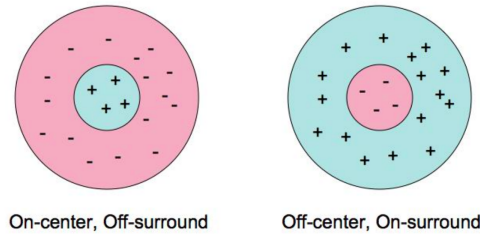
Ο αμφιβληστροειδής χιτώνας αποτελείται από αρκετά επίπεδα νευρικών κυττάρων. Κάθε επίπεδο χαρακτηρίζεται από συγκεκριμένο φάσμα, μορφή και λειτουργία. Τα νευρικά κύτταρα που συνιστούν το τελευταίο επίπεδο του αμφιβληστροειδούς ονομάζονται γαγγλιακά κύτταρα του αμφιβληστροειδούς (Retinal Ganglion Cells - RGCs) και συνιστούν επίσης το οπτικό νεύρο (optic nerve), στο σημείο που οι νευρικές απολήξεις τους ενώνονται σχηματίζοντας συνάψεις. Το σημείο στο οποίο συνδέονται οι νευρικές απολήξεις δεν έχει καθόλου νευρικά κύτταρα του αμφιβληστροειδούς, γι'αυτό και ονομάζεται τυφλό σημείο (blind spot). Τα γαγγλιακά κύτταρα είναι αυτά που αναπτύσσουν τα απαραίτητα δυναμικά ενεργείας για την ενεργοποίηση των νευρώνων. Σε κατάσταση ηρεμίας, αναπτύσσουν δυναμικά ενεργείας σε ένα στοιχειώδη ρυθμό, αλλά όταν διεγείρονται, αναπτύσσουν δυναμικά ενεργείας σε υψηλότερη συχνότητα.

2.3 Υποδεκτικό Πεδίο των Γαγγλιακών Κυττάρων

Η έννοια του υποδεκτικού πεδίου (receptive field) θεωρείται θεμελιώδης για τη νευροεπιστήμη και είναι η τυπική μέθοδος όταν μελετάμε τη σχέση μεταξύ της απόκρισης ενός κυττάρου και του ερεθίσματος που την προκαλεί. Στο οπτικό σύστημα, το υποδεκτικό πεδίο ενός κυττάρου ορίζει την περιοχή του οπτικού πεδίου στο οποίο, εάν ανιχνευτεί ένα ερέθισμα ή μια συγκεκριμένη αλλαγή, το κύτταρο θα διεγερθεί. Στην περίπτωση του αμφιβληστροειδούς, το σχήμα και το μέγεθος του υποδεκτικού πεδίου ποικίλει στις διάφορες περιοχές του αμφιβληστροειδούς. Οι φωτοϋποδοχείς αντιδρούν στα φωτόνια που πέφτουν απευθείας πάνω τους, οπότε έχουν στενά υποδεκτικά πεδία. Τα δίπολα κύτταρα έχουν ένα μεσαίου μεγέθους υποδεκτικό πεδίο καθώς συνδέονται με λίγα κωνία και τέλος, τα οριζόντια κύτταρα έχουν μεγαλύτερο

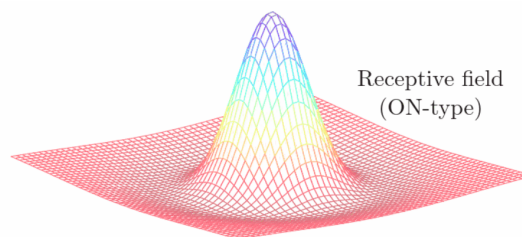
υποδεκτικό πεδίο καθώς ενεργοποιούνται από πολλά κωνία.

Receptive Fields



Σχήμα 2.2. Σχηματική αναπαράσταση του υποδεκτικού πεδίου ενός κυττάρου τύπου ON (αριστερά) κι ενός κυττάρου τύπου OFF (δεξιά) [3].

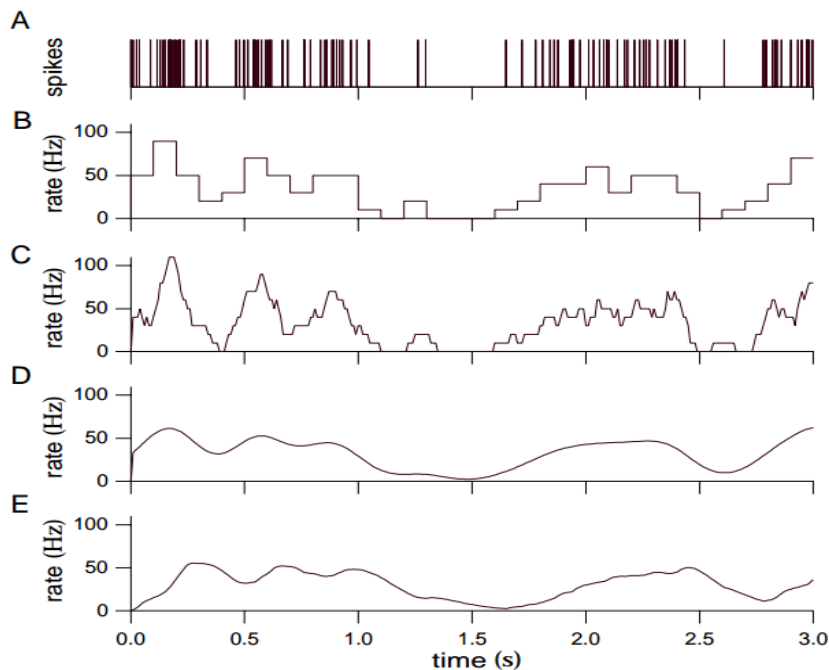
Τα γαγγλιακά κύτταρα του αμφιβληστροειδούς έχουν ένα υποδεκτικό πεδίο που χαρακτηρίζεται από την περιοχή στην οποία το φως μπορεί να επηρεάσει τη νευρική απόκριση και οργανώνεται με βάση την "οργάνωση κέντρου-περιφέρειας" ("*center-surround organization*"), όπως ορίστηκε από τον Kuffler [23]. Πιο συγκεκριμένα, το υποδεκτικό πεδίο των γαγγλιακών κυττάρων αποτελείται από δύο ομόκεντρα ελλειπτικά σχήματα και μπορεί να είναι είτε ON-center είτε OFF-center, και ανταγωνιστικά μεταξύ τους (Σχήμα 2.2). Τα ON-center γαγγλιακά κύτταρα ενεργοποιούνται όταν το φως πέφτει στο κέντρο του υποδεκτικού πεδίου τους και απενεργοποιούνται όταν το φως πέφτει στην περιφέρεια του πεδίου. Τα OFF-center γαγγλιακά κύτταρα δρουν με τον αντίθετο τρόπο, δηλαδή η δραστηριότητά τους αυξάνεται όταν αυξάνεται η φωτεινότητα στην περιφέρεια του υποδεκτικού πεδίου και μειώνεται όταν μειώνεται η φωτεινότητα στο κέντρο του πεδίου. Τα υποδεκτικά πεδία των γαγγλιακών κυττάρων μπορούν να μοντελοποιηθούν σαν διαφορά δύο γκαουσιανών συναρτήσεων (difference of Gaussians) (Σχήμα 2.3). Το υποδεκτικό πεδίο ενός κυττάρου τύπου OFF θα είναι συμμετρικό ως προς αυτό του Σχήματος 2.3. Ο ανταγωνισμός των δύο περιοχών επιτρέπει στους νευρώνες να ανιχνεύουν διαφορές στη φωτεινότητα μεταξύ σημείων, δηλαδή αλλαγές στην αντίθεση. Με άλλα λόγια, οι νευρώνες ενδιαφέρονται μόνο για τις μεταβολές, μια λειτουργία που σχετίζεται με την ανίχνευση ακμών στον τομέα της επεξεργασίας εικόνας [4], [5], [24].



Σχήμα 2.3. Υποδεκτικό πεδίο ενός γαγγλιακού κυττάρου τύπου ON, που αναπαρίσταται ως διαφορά δύο γκαουσιανών συναρτήσεων (difference of Gaussians) [4].

2.4 Νευρικές Αποκρίσεις του Αμφιβληστροειδούς

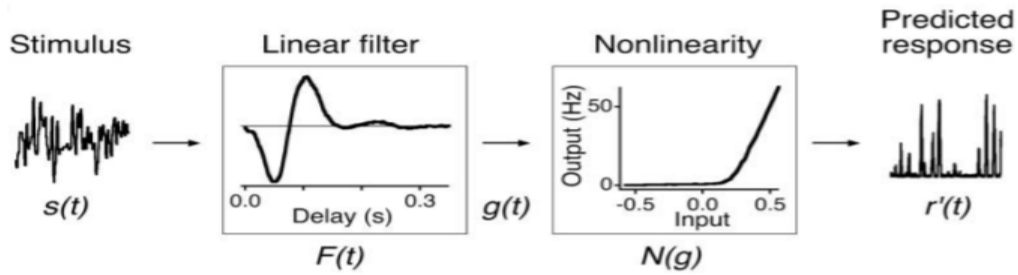
Στο Σχήμα 2.4, φαίνονται συνοπτικά οι τεχνικές/μέθοδοι με τις οποίες υπολογίζεται η νευρική απόκριση ενός νευρώνα. Το Σχήμα 2.4A δείχνει μια νευρική απόκριση ενός νευρώνα. Το Σχήμα 2.4B δείχνει μια προσέγγιση του ρυθμού απόκρισης, όπου χρησιμοποιείται χωρισμός σε παράθυρα μεγέθους 100ms. Στο Σχήμα 2.4C πραγματοποιείται συνέλιξη με τετραγωνικό πυρήνα, ενώ στο Σχήμα 2.4D πραγματοποιείται συνέλιξη με γκαουσιανό πυρήνα.



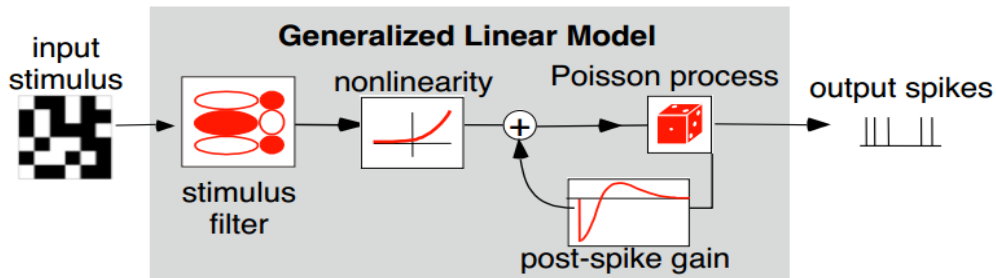
Σχήμα 2.4. Διαφορετικές τεχνικές για τον υπολογισμό νευρικών αποκρίσεων [5].

2.5 Μοντέλα Αμφιβληστροειδούς

Το κίνητρο πίσω από την ανάπτυξη μοντέλων αμφιβληστροειδούς είναι η μελέτη των μηχανισμών που διέπουν τη λειτουργία του αμφιβληστροειδούς. Για το λόγο αυτό, οι επιστήμονες αναπτύσσουν μοντέλα που προσομοιάζουν τη συμπεριφορά των γαγγλιακών κυττάρων του αμφιβληστροειδή. Τα πιο γνωστά εξ αυτών είναι τα μοντέλα Linear Nonlinear (LN) [25] (Σχήμα 2.5), [6], τα Generalized Linear Models (GLMs) μοντέλα [26] (Σχήμα 2.6), τα Integrate-and-Fire μοντέλα [27] και πιο πρόσφατα τα μοντέλα Βαθιάς Μηχανικής Μάθησης (Deep Learning), που κυρίως χρησιμοποιούν Συνελκτικά Νευρωνικά Δίκτυα (CNNs) [28], [12].



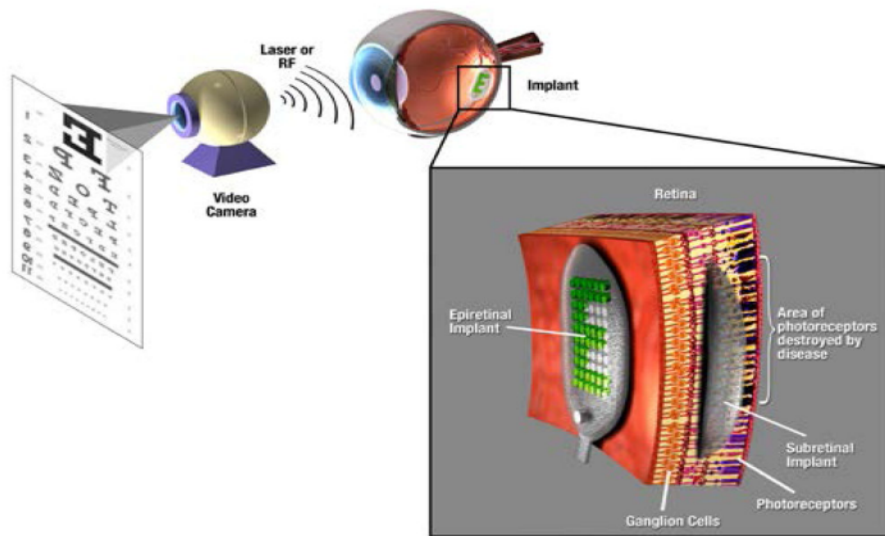
Σχήμα 2.5. Το μοντέλο Linear - Nonlinear (LN) [6].



Σχήμα 2.6. Το μοντέλο Generalized Linear Model (GLM) [7].

2.6 Προσθετική Αμφιβληστροειδούς

Ενώ η χρήση μοντέλων αμφιβληστροειδούς έχει συνδεθεί με την μελέτη των νευρικών λειτουργιών του, η ανάπτυξη παρόμοιων μοντέλων είναι πολύ χρήσιμη στον τομέα της προσθετικής αμφιβληστροειδούς. Τα περισσότερα προσθετικά αμφιβληστροειδούς έχουν ορισμένα κοινά στοιχεία: μια συσκευή λήψης εικόνων, μια μονάδα επεξεργασίας εικόνας, ένα μικροκύκλωμα για τη νευρική διέγερση και μια διάταξη ηλεκτροδίων για την καταγραφή των νευρικών αποκρίσεων (Σχήμα 2.7). Η συσκευή λήψης εικόνων λαμβάνει εικόνες από το οπτικό περιβάλλον, οι οποίες μετατρέπονται σε μοτίβα που διεγείρουν τη συστοιχία πολλαπλών ηλεκτροδίων. Οι οπτικές πληροφορίες επιλέγονται είτε με κάμερα, είτε ανιχνεύονται απευθείας από τις φωτοδιόδους που συνδέονται με ηλεκτρόδια για τη δημιουργία ηλεκτρικών ερεθισμάτων. Άλλα σημαντικά συστατικά που μπορούν να βρεθούν στα περισσότερα εμφυτεύματα είναι η ασύρματη μετάδοση δεδομένων/ισχύος.



Σχήμα 2.7. Επισκόπηση ενός συστήματος προσθετικής αμφιβληστροειδούς. Μια εικόνα εισέρχεται ως είσοδος μέσω μιας κάμερας και μετατρέπεται σε ηλεκτρικό σήμα από μια μονάδα επεξεργασίας. Το ηλεκτρικό σήμα μεταδίδεται μέσω τηλεμετρίας ραδιοσυχνότητας (RF) ή οπτικής ζεύξης και υφίσταται επεξεργασία από εξωτερική ή εμφυτευμένη διάταξη. Το κύκλωμα παράγει ηλεκτρικό ερέθισμα, που υποκαθιστά το φωτεινό ερέθισμα και εφαρμόζεται στον αμφιβληστροειδή μέσω μιας συστοιχίας μικροηλεκτροδίων τοποθετημένης κοντά στον αμφιβληστροειδή [8].

Κεφάλαιο **3**

Μηχανική Μάθηση

3.1 Εισαγωγή

Η Μηχανική Μάθηση (Machine Learning) αποτελεί υποκλάδο της Τεχνητής Νοημοσύνης (Artificial Intelligence), που ασχολείται με την κατασκευή μοντέλων που υλοποιούν συγκεκριμένους αλγόριθμους και χρησιμοποιούν πειραματικά δεδομένα με σκοπό την εξαγωγή χρήσιμων προβλέψεων ή συμπερασμάτων [29]. Ο Mitchell [30] όρισε την μηχανική μάθηση ως εξής:

*“Ένα πρόγραμμα υπολογιστή λέγεται ότι **μαθαίνει** από την εμπειρία E ως προς κάποια κλάση εργασιών T και μέτρο απόδοσης P , αν η απόδοσή του σε εργασίες από το T , όπως μετριέται από το P , βελτιώνεται μέσω της εμπειρίας E .”*

Οι αλγόριθμοι μηχανικής μάθησης χωρίζονται σε τρεις μεγάλες κατηγορίες: Επιβλεπόμενη Μάθηση (Supervised Learning), Μη Επιβλεπόμενη Μάθηση (Unsupervised Learning) και Ενισχυτική Μάθηση (Reinforcement Learning).

3.1.1 Επιβλεπόμενη Μάθηση

Στην επιβλεπόμενη μάθηση χρησιμοποιούνται σύνολα δεδομένων, τα οποία περιέχουν ένα σύνολο εισόδου X και ένα σύνολο εξόδου y . Το X αποτελείται από ένα σύνολο χαρακτηριστικών (features) για κάθε δείγμα, ενώ το y αποτελείται από συγκεκριμένες ετικέτες (labels). Σκοπός των μοντέλων επιβλεπόμενης μάθησης είναι να μάθουν τη συνάρτηση f που αντιστοιχίζει βέλτιστα τα δεδομένα εισόδου X με τις αντίστοιχες ετικέτες y .

$$X \rightarrow y \text{ ή } y = f(X) \quad (3.1)$$

Ο απώτερος στόχος είναι να μπορεί το μοντέλο να αναθέτει σωστές ετικέτες σε νέα, άγνωστα δεδομένα, για τα οποία δεν έχουν οριστεί ακόμα ετικέτες [31]. Κομμάτι της επιβλεπόμενης μάθησης αποτελούν μοντέλα όπως οι Μηχανές Διανυσμάτων

Υποστήριξης (SVM), τα Τυχαία Δάση (Random Forest) και τα Πολυεπίπεδα Νευρωνικά Δίκτυα (Multilayer Perceptrons), με τα οποία θα ασχοληθούμε σε αυτήν την εργασία.

3.1.2 Μη Επιβλεπόμενη Μάθηση

Σε αντίθεση με την προηγούμενη κατηγορία μηχανικής μάθησης, η μη επιβλεπόμενη μάθηση δεν χρησιμοποιεί ζεύγη τιμών εισόδου-εξόδου. Ο στόχος της είναι να αναγνωριστούν μοτίβα στα δεδομένα εισόδου χωρίς την ανατροφοδότηση από τιμές εξόδου. Ένα χαρακτηριστικό παράδειγμα μη επιβλεπόμενης τεχνικής είναι το πρόβλημα της ομαδοποίησης (clustering). Η μέθοδος αυτή στοχεύει στην ομαδοποίηση των δεδομένων εισόδου δημιουργώντας ομάδες των οποίων τα μέλη είναι παρόμοια το ένα με το άλλο, αλλά διαφέρουν σημαντικά από τα μέλη των άλλων ομάδων [31].

3.1.3 Ενισχυτική Μάθηση

Η ενισχυτική μάθηση διαφέρει από τις προηγούμενες κατηγορίες, καθώς εστιάζει στη λήψη βέλτιστων αποφάσεων. Τα μοντέλα ενισχυτικής μάθησης περιέχουν έναν πράκτορα (agent), ο οποίος αλληλεπιδρά με ένα περιβάλλον (environment). Ο πράκτορας λαμβάνει κάποια παρατήρηση (observation) από το περιβάλλον, εκτελεί μια ενέργεια και, στη συνέχεια, λαμβάνει μια ανταμοιβή (reward) από το περιβάλλον. Αυτή η διαδικασία συνεχίζεται επαναληπτικά. Η συμπεριφορά του πράκτορα διέπεται από μια πολιτική (policy), η οποία είναι μια συνάρτηση που αντιστοιχίζει τις παρατηρήσεις του περιβάλλοντος σε ενέργειες. Ο στόχος του αλγορίθμου ενίσχυσης της μάθησης είναι να παράγει μια καλή πολιτική [32].

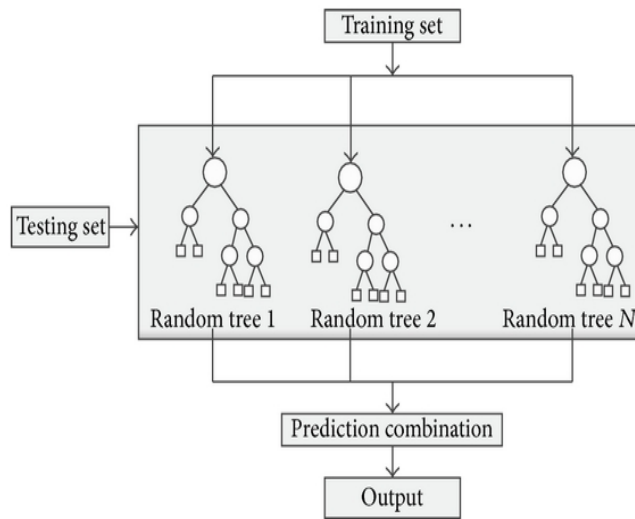
3.2 Ταξινομητές Μηχανικής Μάθησης

Στη συγκεκριμένη εργασία έγινε χρήση των εξής ταξινομητών επιβλεπόμενης μηχανικής μάθησης: Random Forest, SVM και MLP, οι οποίοι παρουσιάζονται παρακάτω.

3.2.1 Τυχαία Δάση - Random Forest

Μία συνηθισμένη τεχνική στην Επιβλεπόμενη Μάθηση είναι η ανεξάρτητη εκπαίδευση πολλαπλών ταξινομητών, συνδυάζοντας τα αποτελέσματά τους στο τέλος για την πρόβλεψη της τελικής τιμής (Ensemble). Ο ταξινομητής Random Forest ανήκει στην προαναφερθείσα τεχνική και συγκεκριμένα αποτελείται από ένα πλήθος Δέντρων Αποφάσεων που ονομάζονται ταξινομητές βάσης, οι οποίοι είναι παράλλη-

λοι και ανεξάρτητοι μεταξύ τους. Στο σχήμα 3.1 φαίνεται η μορφή ενός ταξινομητή Random Forest.



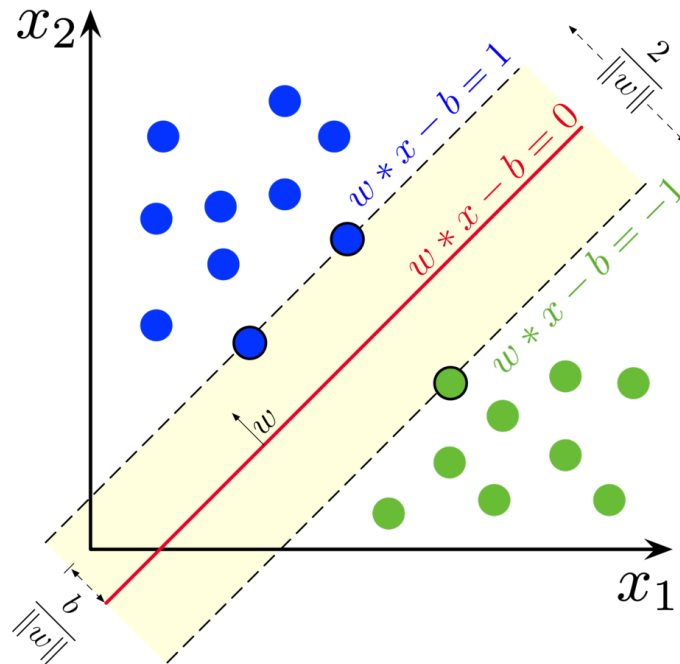
Σχήμα 3.1. Αναπαράσταση του αλγορίθμου Random Forest [9].

3.2.2 Μηχανές Διανυσμάτων Υποστήριξης - SVM

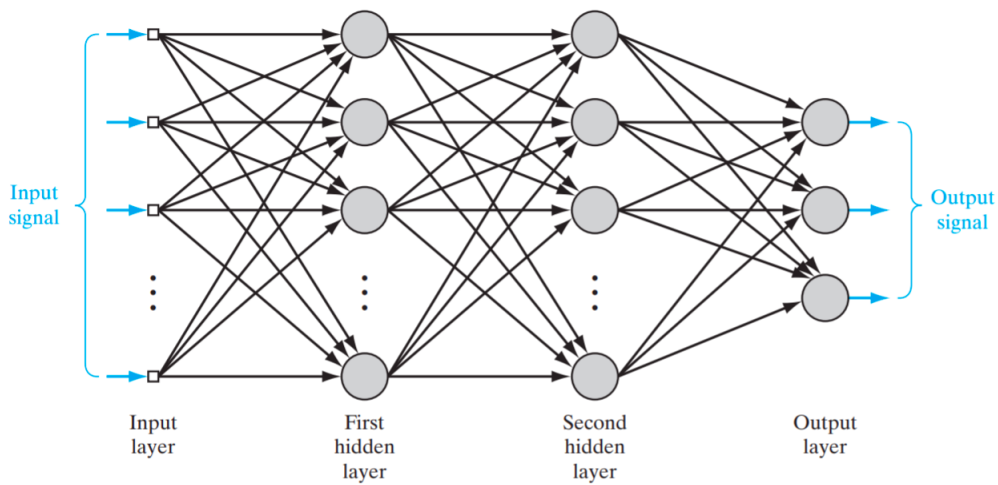
Ο ταξινομητής SVM (Σχήμα 3.2) είναι ένας από τους βασικότερους αλγόριθμους της επιβλεπόμενης μηχανικής μάθησης. Βασίζεται στην χωρική αναπαράσταση των δεδομένων εισόδου και δημιουργώντας ένα υπερεπίπεδο τα διαχωρίζει στις κλάσεις εξόδου. Το υπερεπίπεδο αυτό πρέπει να διαχωρίζει όσο δυνατόν περισσότερο τις κλάσεις εξόδου και ταυτόχρονα να μεγιστοποιεί την απόσταση των δεδομένων εισόδου από αυτό. Η μορφή των δεδομένων μπορεί να τα κατατάσσει στην κατηγορία των γραμμικά διαχωρίσιμων ή μη γραμμικά διαχωρίσιμων δεδομένων. Όταν οι κλάσεις είναι γραμμικά διαχωρίσιμες οι ταξινομητές SVM μπορούν να διαχωρίσουν τα δεδομένα, ενώ όταν οι κλάσεις είναι μη γραμμικά διαχωρίσιμες, απαιτείται η χρήση Μη Γραμμικών SVM, οι οποίοι χρησιμοποιώντας τεχνικές μετασχηματισμού των συντεταγμένων των δεδομένων σε χώρο μεγαλύτερης διαστατικότητας μπορούν να υπολογίσουν υπερεπίπεδα που να διαχωρίζουν τα δεδομένα.

3.2.3 Πολυεπίπεδα Νευρωνικά Δίκτυα - MLP

Ο ταξινομητής MLP είναι από τις πιο απλές μορφές νευρικών δικτύων και περιέχει πολλαπλούς νευρώνες (κόμβους) διατεταγμένους σε επίπεδα. Οι κόμβοι από παρακείμενα επίπεδα έχουν συνδέσεις ή ακμές μεταξύ τους. Όλες αυτές οι συνδέσεις έχουν βάρη που σχετίζονται με αυτές. Γενικά, ένας ταξινομητής MLP περιέχει ένα επίπεδο εισόδου, ένα ή περισσότερα κρυφά επίπεδα και ένα επίπεδο εξόδου, όπως φαίνεται στο Σχήμα 3.3.



Σχήμα 3.2. Αναπαράσταση του αλγορίθμου SVM για διαχωρισμό δύο κλάσεων (Wikipedia).



Σχήμα 3.3. Δομή ενός ταξινομητή MLP με δύο κρυφά επίπεδα [10].

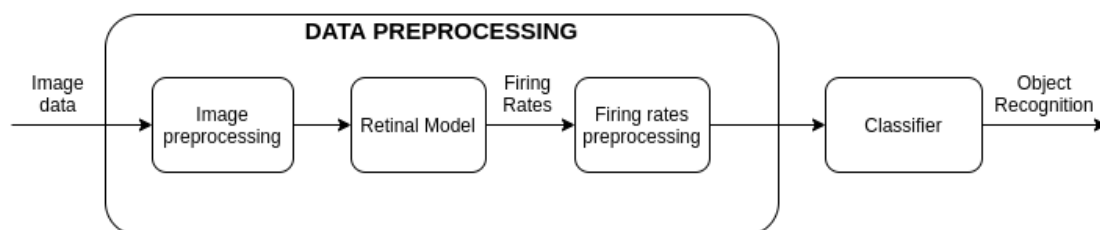
Κεφάλαιο **4**

Ανάπτυξη Μεθοδολογίας Λειτουργικής Αξιολόγησης

4.1 Πλαίσιο Λειτουργικής Αξιολόγησης

Για τις προσομοιώσεις που αναλύονται στην συγκεκριμένη ενότητα, ακολουθήθηκε μια κοινή μεθοδολογία, η οποία φαίνεται στο Σχήμα 4.1 και περιγράφεται από τα παρακάτω βήματα :

1. Εισαγωγή συνόλων δεδομένων
2. Προεπεξεργασία δεδομένων
3. Εισαγωγή δεδομένων στο μοντέλο αμφιβληστροειδούς
4. Εξαγωγή των νευρικών αποκρίσεων
5. Εισαγωγή των νευρικών αποκρίσεων στους ταξινομητές για αναγνώριση αντικειμένου και αναγνώριση ψηφίων
6. Σύγκριση της απόδοσης των ταξινομητών



Σχήμα 4.1. Μέθοδος λειτουργικής αξιολόγησης.

Αρχικά, θα αναφέρουμε τα εργαλεία λογισμικού, αλλά και λεπτομέρειες σχετικά με το μοντέλο αμφιβληστροειδούς που χρησιμοποιήσαμε. Στη συνέχεια, θα περιγράψουμε τη διαδικασία της προεπεξεργασίας των δεδομένων και των προσομοιώσεων που εκτελέστηκαν.

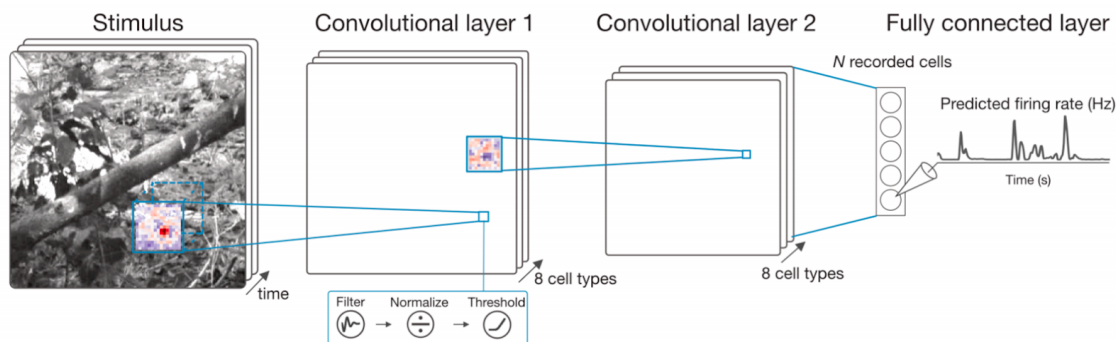
4.2 Περιγραφή Εργαλείων Λογισμικού

Ο κώδικας που χρησιμοποιήσαμε για το μοντέλο αμφιβληστροειδούς μπορεί να βρεθεί στο [33]. Το μοντέλο εκπαιδεύτηκε με τη χρήση των βιβλιοθηκών TensorFlow [34] και Keras [35]. Για τους ταξινομητές MLP, χρησιμοποιήσαμε την υλοποίηση του Keras, ενώ για τους ταξινομητές Random Forest και SVM χρησιμοποιήσαμε υλοποιήσεις Scikit-Learn [36]. Η επαύξηση των δεδομένων έγινε χρησιμοποιώντας τη βιβλιοθήκη `imageaug` [37]. Όλες οι προσομοιώσεις έγιναν με υπολογιστικούς πόρους από το Google Colaboratory.

4.3 Περιγραφή Συνόλων Δεδομένων

Για τις προσομοιώσεις, χρησιμοποιήσαμε τα ακόλουθα σύνολα δεδομένων: CIFAR10 [38], MNIST [22], Fashion MNIST [20] και Imagenette [21]. Μια περιγραφή των συνόλων δεδομένων βρίσκεται στον Πίνακα 4.1. Επιλέξαμε τα συγκεκριμένα σύνολα δεδομένων, επειδή και τα τέσσερα χρησιμοποιούνται σε πειράματα ταξινόμησης με 10 κλάσεις. Στόχος μας ήταν να εξαλείψουμε τους παράγοντες διαφοροποίησης μεταξύ των συνόλων δεδομένων και να συγκρίνουμε ισοδύναμες εργασίες ταξινόμησης. Ξεκινήσαμε με το CIFAR10 και το MNIST ως κλασικά σύνολα δεδομένων για αρχικές δοκιμές. Στη συνέχεια, επιλέξαμε να συγκρίνουμε το CIFAR10 με το Imagenette και το MNIST με το Fashion MNIST. Επιλέξαμε το Imagenette ως σύνολο δεδομένων που περιέχει εικόνες πολύ υψηλότερης ανάλυσης από το CIFAR10. Επιλέξαμε επίσης το Fashion MNIST ως ένα πιο δύσκολο και περίπλοκο σύνολο δεδομένων από το MNIST, όπως σημειώνεται στο [20].

4.4 Περιγραφή του Μοντέλου Αμφιβληστροειδούς των Προσομοιώσεων



Σχήμα 4.2. Αρχιτεκτονική του CNN μοντέλου αμφιβληστροειδούς που χρησιμοποιήθηκε σε αυτή την εργασία [11].

Πίνακας 4.1. Περιγραφή των συνόλων δεδομένων που χρησιμοποιήθηκαν στις προσομοιώσεις.

Dataset	Περιγραφή	Links
CIFAR10	Το CIFAR10 αποτελείται από 60000 32x32 έγχρωμες εικόνες που χωρίζονται σε 10 κλάσεις, με 6000 εικόνες σε κάθε κλάση. Χρησιμοποιείται σε προβλήματα αναγνώρισης εικόνων.	[19], [38]
MNIST	Το MNIST αποτελείται από 70000 28x28 ασπρόμαυρες εικόνες χειρόγραφων ψηφίων από το 0 έως το 9. Τα ψηφία έχουν κανονικοποιηθεί ως προς τις διαστάσεις τους και βρίσκονται στο κέντρο της εικόνας. Χρησιμοποιείται σε προβλήματα αναγνώρισης ψηφίων.	[22], [39]
Fashion MNIST	Το Fashion MNIST αποτελείται από 70000 28x28 ασπρόμαυρες εικόνες από 10 κλάσεις ρούχων. Αυτό το dataset μπορεί να χρησιμοποιηθεί ως εναλλακτική για το MNIST. Χρησιμοποιείται σε προβλήματα αναγνώρισης εικόνων.	[20], [40]
Imagenette	Το Imagenette είναι ένα υποσύνολο του Imagenet και αποτελείται από έγχρωμες εικόνες μεταβλητού μεγέθους που αντιπροσωπεύουν 10 εύκολα ταξινομήσιμες κλάσεις του Imagenet. Χρησιμοποιείται σε προβλήματα αναγνώρισης εικόνων.	[21], [41]

Στην εργασία αυτή, χρησιμοποιήσαμε το CNN μοντέλο αμφιβληστροειδούς, που περιγράφεται στις εργασίες [28], [11] και [33]. Όπως φαίνεται στο Σχήμα 4.2, το μοντέλο αποτελείται από 3 επίπεδα και εκπαιδεύεται για να προβλέπει τις νευρικές αποκρίσεις 60 γαγγλιακών κυττάρων. Επίσης, στον Πίνακα 4.2, φαίνονται λεπτομερώς τα επίπεδα του μοντέλου. Για την εκπαίδευση χρησιμοποιήσαμε ένα σύνολο δεδομένων από αντίστοιχη εργασία [12]. Στη συγκεκριμένη εργασία, προέβαλαν φυσικές εικόνες στον αμφιβληστροειδή του ποντικιού και κατέγραψαν την απόκριση 60 γαγγλίων κυττάρων αμφιβληστροειδούς. Το σύνολο εικόνων που χρησιμοποίησαν αποτελείται από 4890 ασπρόμαυρες εικόνες μεγέθους 50x50 pixels, ένα δείγμα των οποίων φαίνεται στο Σχήμα 4.3.

Η απόκριση των γαγγλιακών κυττάρων του αμφιβληστροειδούς εξαρτάται όχι μόνο από το τρέχον ερέθισμα, αλλά κι από προηγούμενες διεγέρσεις. Προκειμένου να μοντελοποιηθεί αυτή η χρονική εξάρτηση, δημιουργήθηκε ένα χρονικό σύνολο δεδομένων (temporal dataset) πριν από την εισαγωγή εικόνων στο μοντέλο του αμφιβληστροειδούς. Αυτό σημαίνει ότι για κάθε εικόνα που προβάλλεται στον αμφιβληστροειδή, αποθηκεύουμε και το ιστορικό των εικόνων που προβλήθηκαν προηγου-

Πίνακας 4.2. Σύνοψη όλων των επιπέδων του CNN μοντέλου αμφιβληστροειδούς που χρησιμοποιήθηκε σε αυτή την εργασία.

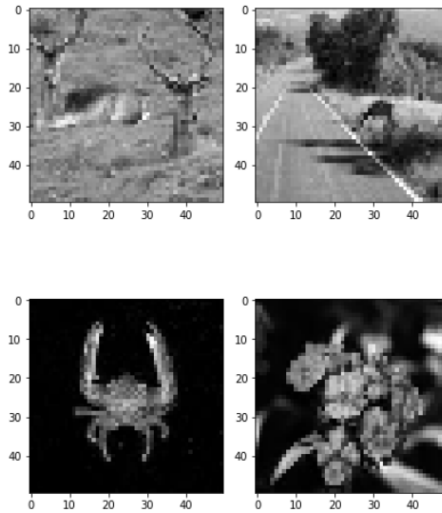
```
Model: "bn_cnn"
```

Layer (type)	Output Shape	Param #
input_1 (InputLayer)	(None, 50, 50, 40)	0
conv2d_1 (Conv2D)	(None, 41, 41, 8)	32008
flatten_1 (Flatten)	(None, 13448)	0
batch_normalization_1 (Batch Normalization)	(None, 13448)	53792
reshape_1 (Reshape)	(None, 41, 41, 8)	0
gaussian_noise_1 (GaussianNoise)	(None, 41, 41, 8)	0
activation_1 (Activation)	(None, 41, 41, 8)	0
conv2d_2 (Conv2D)	(None, 37, 37, 16)	3216
flatten_2 (Flatten)	(None, 21904)	0
batch_normalization_2 (Batch Normalization)	(None, 21904)	87616
reshape_2 (Reshape)	(None, 37, 37, 16)	0
gaussian_noise_2 (GaussianNoise)	(None, 37, 37, 16)	0
activation_2 (Activation)	(None, 37, 37, 16)	0
flatten_3 (Flatten)	(None, 21904)	0
dense_1 (Dense)	(None, 60)	1314240
batch_normalization_3 (Batch Normalization)	(None, 60)	240
activation_3 (Activation)	(None, 60)	0

```

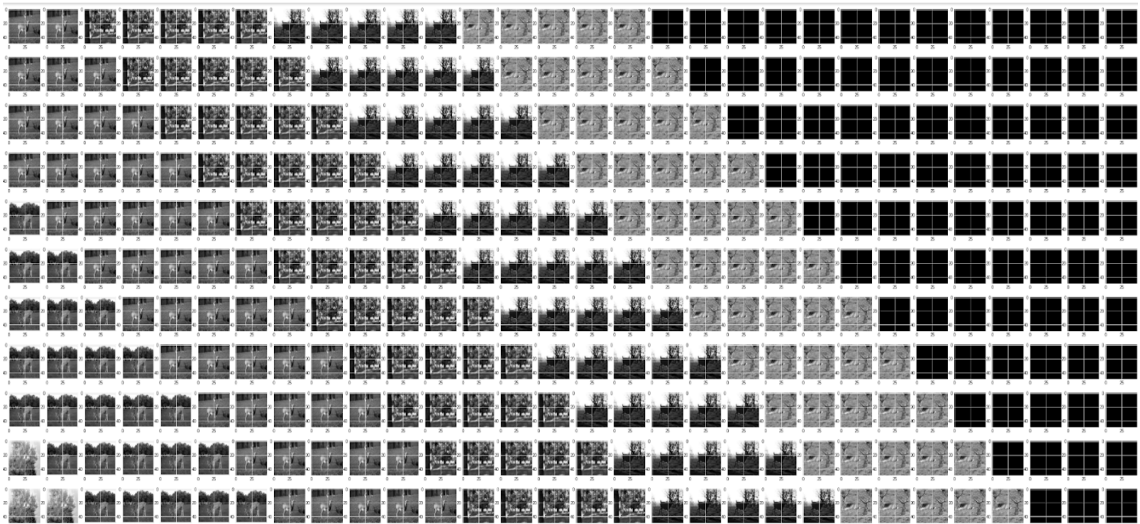
Total params: 1,491,112
Trainable params: 1,420,288
Non-trainable params: 70,824

```



Σχήμα 4.3. Δείγματα εικόνων από το σύνολο δεδομένων με το οποίο εκπαιδεύσαμε το μοντέλο αμφιβληστροειδούς [12].

μένως. Ονομάζουμε `temporal_interval` τον συνολικό αριθμό των εικόνων που χρησιμοποιούνται, ο οποίος αποτελείται από την πραγματική εικόνα που προβάλλεται στον αμφιβληστροειδή μαζί με το ιστορικό των προηγούμενων εικόνων. Επομένως, κάθε δείγμα εισόδου αποτελείται από `temporal_interval` αριθμό εικόνων. Θα πρέπει επίσης να σημειωθεί ότι το ιστορικό της πρώτης εικόνας αποτελείται από σκούρα πλαίσια για να μοντελοποιήσει την αρχική σκοτεινή κατάσταση του αμφιβληστροειδούς. Ένα μέρος του (`temporal dataset`) φαίνεται στο Σχήμα 4.4. Τελικά, η έξοδος του μοντέλου είναι ρυθμός απόκρισης για κάθε ένα από τα 60 γαγγλιακά κύτταρα.



Σχήμα 4.4. Μέρος του `temporal dataset` με `temporal_interval = 30`. Η αριστερότερη στήλη κάθε σειράς περιέχει την εικόνα που προβάλλεται εκείνη τη στιγμή στον αμφιβληστροειδή. Οι υπόλοιπες στήλες περιέχουν το ιστορικό των εικόνων που προηγήθηκαν. Φαίνονται επίσης τα αρχικά μαύρα πλαίσια.

Σε αυτήν την εργασία, χρησιμοποιήσαμε δύο διαφορετικά μοντέλα, ένα εκπαι-

δευμένο με `temporal_interval = 30` (**RetModel1**) και το άλλο εκπαιδευμένο με `temporal_interval = 40` (**RetModel2**). Το RetModel1 είχε καλύτερη απόδοση όσον αφορά τη μέτρηση Pearson Correlation Coefficient (PCC) και το RetModel2 είχε χειρότερες μετρήσεις PCC, αλλά παράγαγε καλύτερα υποδεκτικά πεδία (receptive fields) ¹. Περισσότερες λεπτομέρειες σχετικά με την απόδοση των μοντέλων του αμφιβληστροειδούς μπορούν να βρεθούν στο [42].

4.5 Ανάλυση Βημάτων Προεπεξεργασίας Δεδομένων

Στα σύνολα δεδομένων, εφαρμόστηκε ένας αλγόριθμος προεπεξεργασίας πριν εισαχθούν ως είσοδος στο μοντέλο αμφιβληστροειδούς. Τα βήματα του αλγορίθμου καθώς και οι σχεδιαστικές αποφάσεις που κληθήκαμε να πάρουμε περιγράφονται παρακάτω και φαίνονται στον Πίνακα 4.3.

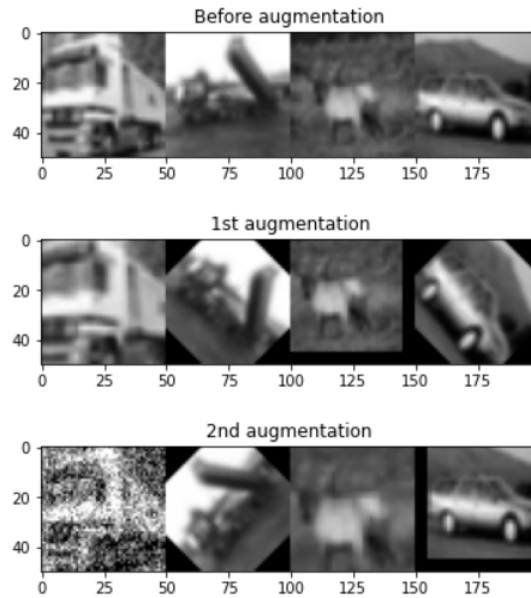
Στο **Βήμα 3**, η τεχνική της επαύξησης δεδομένων έγινε με τη χρήση της βιβλιοθήκης `imgaug` [37]. Σε κάθε εικόνα εφαρμόζονται δύο μετασχηματισμοί, οι οποίοι επιλέγονται τυχαία ανάμεσα από ένα σύνολο πιθανών μετασχηματισμών. Ο πρώτος μετασχηματισμός επιλέγεται τυχαία ανάμεσα από τους ακόλουθους:

1. Περιστροφή (rotate) 45°
2. Γκαουσιανός θόρυβος (Gaussian noise, $\text{scale}=0.1 * 255$)
3. Περικοπή (crop) 5 pixels από κάθε πλευρά
4. Μετατόπιση (translation) στους άξονες x και y από -10% έως $+10\%$ του μεγέθους της εικόνας

Ο δεύτερος μετασχηματισμός επιλέγεται τυχαία ανάμεσα από τους ακόλουθους:

1. Περιστροφή (rotate) -45°
2. Γκαουσιανός θόρυβος (Gaussian noise, $\text{scale}=0.2 * 255$)
3. Περικοπή (crop) 7 pixels από κάθε πλευρά
4. Μετατόπιση (translation) στους άξονες x και y από -10% έως $+10\%$ του μεγέθους της εικόνας

Μετά την επαύξηση, για κάθε εικόνα στο σύνολο δεδομένων υπάρχουν τρεις εικόνες: η αρχική, η πρώτη μετασχηματισμένη και η δεύτερη μετασχηματισμένη. Ένα παράδειγμα επαύξησης δεδομένων φαίνεται στο Σχήμα 4.5, όπου για κάθε μία από τις τέσσερις εικόνες εφαρμόζεται ένας από τους πρώτους κι ένας από τους δεύτερους



Σχήμα 4.5. Παράδειγμα της τεχνικής επαύξησης δεδομένων.

μετασχηματισμούς.

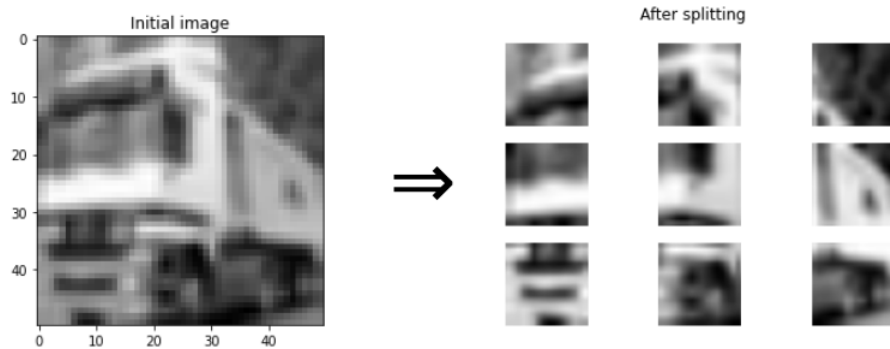
Στο **Βήμα 5**, ο χωρισμός της εικόνας επιτυγχάνεται, χωρίζοντας κάθε μία από τις πλευρές της σε τόσα τμήματα/μέρη, όσα και η μεταβλητή p . Έτσι, δημιουργούνται μικρότερες υποεικόνες μέσα στην αρχική εικόνα. Για $p = 2$, η εικόνα χωρίστηκε σε $p^2 = 4$ τμήματα/μέρη, για $p = 3$ σε $p^2 = 9$ τμήματα/μέρη κ.ο.κ. Σημειώνεται επίσης ότι για $p = 1$, η εικόνα δε χωρίστηκε σε κανένα τμήμα. Στο παρακάτω σχήμα φαίνεται ένα παράδειγμα για $p = 3$, δηλαδή για χωρισμό σε 9 τμήματα/μέρη.

Για το Imagenette και μόνο, επειδή οι εικόνες στο σύνολο δεδομένων έχουν μέγεθος μεγαλύτερο από (50, 50), οι εικόνες που προκύπτουν μετά το split έχουν μέγεθος μεγαλύτερο από (50, 50). Έτσι, σε αντίθεση με τις άλλες περιπτώσεις, ο μετασχηματισμών των υποεικόνων σε (50, 50) είναι σμίκρυνση κι όχι μεγέθυνση. Στο Σχήμα 4.6 φαίνεται ένα παράδειγμα χωρισμού εικόνας σε 9 τμήματα/μέρη ($p = 3$).

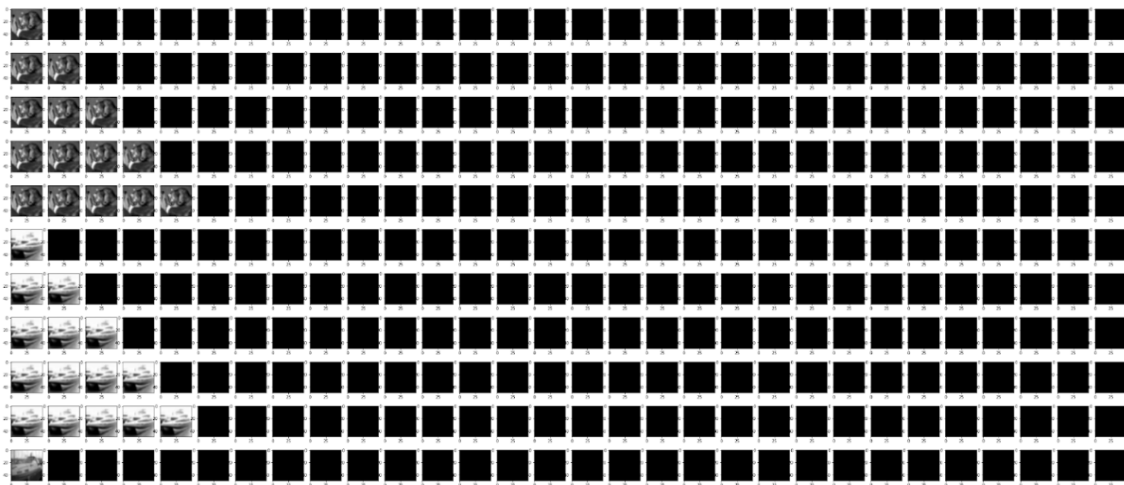
Στο **Βήμα 6**, σκοπός είναι να κατασκευαστεί σε επόμενο βήμα το χρονικό σύνολο δεδομένων (temporal dataset). Για το λόγο αυτό, απαιτείται να επαναλάβουμε κάθε εικόνα πέντε φορές. Για να γίνει ευκολότερα αντιληπτή η έννοια του temporal dataset, παρουσιάζεται το παρακάτω σχήμα :

Το temporal dataset δημιουργείται για να μιμηθεί τη χρονική διάρκεια για την οποία ο αμφιβληστροειδής εκτίθεται σε μία εικόνα. Κάθε μία από τις γραμμές του temporal dataset αποτελεί μία είσοδο στο μοντέλο αμφιβληστροειδούς. Όπως φαίνε-

¹Τα υποδεκτικά πεδία ορίζονται στην Ενότητα 2.3



Σχήμα 4.6. Παράδειγμα χωρισμού εικόνας σε εννιά τμήματα/μέρη ($p = 3$).



Σχήμα 4.7. Ένα μέρος του temporal dataset. Κάθε γραμμή στο temporal dataset αναπαριστά ένα δείγμα εισόδου στο μοντέλο αμφιβληστροειδούς και έχει τόσες εικόνες, όσο και η τιμή της μεταβλητής `temporal_interval` (εδώ, `temporal_interval=30`). Η εικόνα στην αριστερότερη στήλη αναπαριστά την εικόνα που βλέπει ο αμφιβληστροειδής εκείνη τη στιγμή και οι υπόλοιπες εικόνες αναπαριστούν το ιστορικό των προηγούμενων εικόνων που έβλεπε ο αμφιβληστροειδής. Τα μαύρα πλαίσια αναπαριστούν το σκοτάδι στο οποίο βρίσκεται ο αμφιβληστροειδής πριν δει κάθε εικόνα.

ται στο σχήμα, κάθε γραμμή έχει 30 εικόνες (frames), όσες κι η μεταβλητή `temporal_interval`. Το πλαίσιο στην αριστερότερη στήλη υποδηλώνει την εικόνα που βλέπει εκείνη τη στιγμή το μάτι και οι υπόλοιπες 29 είναι το ιστορικό των εικόνων που έβλεπε το μάτι σε προηγούμενες στιγμές. Από το σχήμα, φαίνεται ότι κάθε εικόνα βρίσκεται σε 5 γραμμές και σε κάθε γραμμή η εικόνα επαναλαμβάνεται από 1 έως 5 φορές και τα υπόλοιπα είναι κενά πλαίσια. Ο αριθμός των επαναλήψεων της εικόνας (5 εν προκειμένω) υποδηλώνει τη χρονική διάρκεια για την οποία ο αμφιβληστροειδής εκτίθεται στη συγκεκριμένη εικόνα. Ο αριθμός αυτός αποτελεί αντικείμενο μελέτης της παρούσας εργασίας και οι σχετικές προσομοιώσεις παρουσιάζονται στο Παράρτημα Α'.

Σημειώνεται ότι αν δεν γίνει η προσαρμογή στο Βήμα 6, τότε στο temporal dataset

θα υπάρχει μια γραμμή για κάθε εικόνα και η εικόνα θα επαναλαμβάνεται 10 φορές στη γραμμή αυτή (ο αριθμός 10 προέκυψε σύμφωνα με τις προσομοιώσεις στο Παράρτημα Α). Ενώ, αν γίνει η προσαρμογή, τότε θα υπάρχουν πέντε γραμμές στο temporal dataset όπως περιγράφηκε προηγουμένως.

Στο **Βήμα 7**, το χρονικό σύνολο δεδομένων (temporal dataset) κατασκευάζεται, όπως περιγράφηκε προηγουμένως.

Στο **Βήμα 8**, για κάθε μία από τις γραμμές του (temporal dataset), το μοντέλο αμφιβληστροειδούς προβλέπει τις νευρικές αποκρίσεις για 60 νευρώνες. Έτσι, σε κάθε γραμμή αντιστοιχεί ένα διάνυσμα μεγέθους 60, του οποίου το i στοιχείο περιγράφει την εξομαλυμένη νευρική απόκριση του i νευρώνα, όπου $i = 0, 1, 2, \dots, 59$.

Μια επιπλέον τεχνική προεπεξεργασίας που εφαρμόζεται σε μελέτες πρόβλεψης αποκρίσεων στον αμφιβληστροειδή ή και στον εγκέφαλο, είναι η απόρριψη ορισμένων νευρώνων πριν την εκπαίδευση με βάση κάποιο κριτήριο. Στην περίπτωση μας, μπορούμε να αναγνωρίσουμε τα υποδεκτικά πεδία των νευρώνων με τη χρήση της ανάλυσης STA (Spike Trigger Analysis), όπως περιγράφεται από τον Chichilnisky [25]. Αν η ανάλυση STA καταφέρει να υπολογίσει το κέντρο του υποδεκτικού πεδίου ενός νευρώνα, τότε θεωρούμε έγκυρο αυτόν τον νευρώνα. Για το μοντέλο αμφιβληστροειδούς που χρησιμοποιήσαμε σε αυτήν την εργασία προέκυψαν μόνο 12 έγκυροι νευρώνες στις θέσεις 2, 7, 9, 16, 19, 24, 26, 33, 41, 45, 55 του πίνακα εξόδου με τις αποκρίσεις. Οι παραπάνω έγκυροι νευρώνες υπολογίστηκαν στην εργασία [42], από όπου και πήραμε τα μοντέλα αμφιβληστροειδούς. Οπότε, στην περίπτωση που έγινε η προσαρμογή στο Βήμα 6, αυτό σημαίνει ότι σε κάθε εικόνα αντιστοιχούν 5 διαφορετικά διανύσματα με νευρικές αποκρίσεις για 60 (ή 12 αν έγινε η επιλογή των έγκυρων νευρώνων) νευρώνες. Εφαρμόζοντας στοιχείο προς στοιχείο τον μετασχηματισμό \min ή \max στα 5 διανύσματα της κάθε εικόνας, αυτά συνδυάζονται σε ένα διάνυσμα μεγέθους 60 (ή 12).

Στο **Βήμα 9**, σε περίπτωση που έγινε χωρισμός της εικόνας στο Βήμα 5, τότε για κάθε κομμάτι της εικόνας έχει προκύψει και ένα διάνυσμα νευρικών αποκρίσεων. Αυτά συνενώνονται σε ένα διάνυσμα μεγέθους $a * p^2$, όπου $a = 60$ (ή $a = 12$) και $p = 2, 3, 4$. Για παράδειγμα, αν η εικόνα έχει χωριστεί σε 9 κομμάτια ($p = 3$), τότε το τελικό διάνυσμα έχει μέγεθος $60 * 9 = 540$ για $a = 60$ (ή μέγεθος 108 για $a = 12$). Το σύνολο των τελικών διανυσμάτων μετά το Βήμα 10 είναι το σύνολο δεδομένων με τα οποία θα εκπαιδευτούν τα μοντέλα μηχανικής μάθησης και κατά συνέπεια, το μέγεθος των διανυσμάτων αποτελεί και τον αριθμό των χαρακτηριστικών (features).

Πίνακας 4.3. Τα βήματα της προεπεξεργασίας δεδομένων.

Προεπεξεργασία Δεδομένων

1. **(INPUT)** Πάρε το σύνολο δεδομένων (input) και μόνο για το CIFAR10 και το Imagenette μετασχημάτισε κάθε εικόνα από rgb σε grayscale.
2. **(RESHAPE)** Μετασχημάτισε (reshape) το μέγεθος κάθε εικόνας από το σύνολο δεδομένων σε (50, 50, 1), το οποίο είναι το αποδεκτό μέγεθος εισόδου για το μοντέλο αμφιβληστροειδούς. Στη περίπτωση του Imagenette, επειδή οι αρχικές εικόνες είναι μεγαλύτερες από (50, 50), ο μετασχηματισμός γίνεται μετά τον χωρισμό της εικόνας στο Βήμα 5.
3. **(DATA AUGMENTATION)** Επίλεξε 2 διαφορετικούς μετασχηματισμούς και εφαρμόσε τους στην κάθε εικόνα με την τεχνική της επαύξησης δεδομένων (data augmentation).
 - Στο Βήμα 3, αποφάσισε αν θα εφαρμοστεί ή όχι η επαύξηση.
4. **(NORMALIZATION)** Κανονικοποίησε (normalize) τις εικόνες, έτσι ώστε κάθε pixel της εικόνας να παίρνει τιμές στο (0, 1).
5. **(SPLIT)** Χώρισε (split) την εικόνα σε p^2 κομμάτια, όπου $p = 1, 2, 3, 4$. Μετασχημάτισε το μέγεθος καθενός από τα κομμάτια που θα προκύψουν σε (50, 50).
 - Στο Βήμα 5, αποφάσισε αν θα χωριστεί η εικόνα κι αν ναι, σε πόσα τμήματα/μέρη.
6. **(ADJUSTMENT)** Προσάρμοσε (adjust) το σύνολο δεδομένων, επαναλαμβάνοντας κάθε εικόνα πέντε φορές.
 - Στο Βήμα 6, αποφάσισε αν θα γίνει η προσαρμογή ή όχι.
7. **(TEMPORAL)** Δημιούργησε το χρονικό σύνολο δεδομένων (temporal dataset).
8. **(PREDICT)** Πρόβλεψε (predict) τις νευρικές αποκρίσεις του μοντέλου αμφιβληστροειδούς.
 - Στο Βήμα 8, αποφάσισε αν θα επιλεγθούν μόνο οι έγκυροι νευρώνες ή όχι.
9. **(COMBINE)** Αν έγινε η προσαρμογή στο Βήμα 6, τότε συνδύασε (combine) τις νευρικές αποκρίσεις των 5 γραμμών της εικόνας στο temporal dataset σε μία νευρική απόκριση με τη χρήση μετασχηματισμών *min* ή *max*.
 - Στο Βήμα 9, αποφάσισε αν θα χρησιμοποιηθεί *min* ή *max* μετασχηματισμός.
10. **(CONCATENATE)** Αν έγινε χωρισμός της εικόνας στο Βήμα 5, τότε συνένωσε (concatenate) σε ένα διάνυσμα τις νευρικές αποκρίσεις που προέβλεψε το μοντέλο αμφιβληστροειδούς για όλα τα τμήματα/μέρη της εικόνας. Το διάνυσμα αυτό θα είναι η είσοδος στα μοντέλα μηχανικής μάθησης.

4.6 Οργάνωση των Προσομοιώσεων

Ο σχεδιασμός των προσομοιώσεων βασίστηκε στις εξής παραμέτρους: τις σχεδιαστικές αποφάσεις κατά την προεπεξεργασία δεδομένων, τα σύνολα δεδομένων (datasets) που χρησιμοποιήθηκαν, καθώς και τους ταξινομητές (classifiers) που εκπαιδεύτηκαν. Συγκεκριμένα, κατά την προεπεξεργασία των δεδομένων οι αποφάσεις αφορούσαν την επαύξηση των δεδομένων (**Augment**), τον χωρισμό της εικόνας (**Split**), τον συνδυασμό των νευρικών αποκρίσεων για κάθε εικόνα μετά το temporal dataset (**Combine**), το αν θα γίνει προσαρμογή πριν τη δημιουργία του temporal dataset (**Adjust**) και το αν θα εξαχθούν μόνο οι έγκυροι νευρώνες μετά την πρόβλεψη του μοντέλου αμφιβληστροειδούς (**Valid**). Όσον αφορά τα σύνολα δεδομένων, η επιλογή έγινε ανάμεσα από το CIFAR10, το MNIST, το Fashion_MNIST και το Imagenette. Οι ταξινομητές που επιλέχθηκαν να εκπαιδευτούν είναι οι εξής: MLP, Random Forest και SVM. Σημειώνεται ότι οι δοκιμές έδειξαν ότι η επαύξηση των δεδομένων (data augmentation) αυξάνει την αποδοτικότητα των μοντέλων και μειώνει το overfitting. Ως εκ τούτου, αποφασίστηκε να εφαρμόζεται σε όλα τις προσομοιώσεις η επαύξηση δεδομένων (Augment=yes). Τα προαναφερθέντα στοιχεία συνοψίζονται στον Πίνακα 4.4.

Πίνακας 4.4. Το εύρος τιμών για τις παραμέτρους των προσομοιώσεων. Τα σύνολα δεδομένων παρουσιάζονται στον Πίνακα 4.1. Οι υπόλοιπες παράμετροι αναλύονται σε αυτή την ενότητα.

Augment	Split	Combine	Adjust	Valid	Dataset	Classifier
Yes	None, $p = 1$	min	Yes (5 σειρές / εικόνα)	Yes	CIFAR10	MLP
	4 τμήματα, $p = 2$	max	No (1 σειρά / εικόνα)	No	MNIST	Random Forest
	9 τμήματα, $p = 3$				Fashion MNIST	SVM
	16 τμήματα, $p = 4$				Imagenette	

Με βάση τον παραπάνω πίνακα, σχεδιάστηκαν και εκτελέστηκαν οι προσομοιώσεις (A1, A2, B1, C1, D1), όπως φαίνονται στον Πίνακα 4.5. Για κάθε σύνολο δεδομένων διενεργήθηκαν πέντε είδη προσομοιώσεων, τα οποία αντιστοιχούν στις

διαφορετικές σχεδιαστικές αποφάσεις κατά τη διάρκεια της προεπεξεργασίας των δεδομένων. Για κάθε ένα από τα είδη προσομοιώσεων δοκιμάστηκαν τέσσερις διαφορετικοί χωρισμοί (Split) των εικόνων σε: Κανένα, 4, 9 και 16 τμήματα/μέρη ($p = 1, 2, 3, 4$ αντίστοιχα). Επίσης, για κάθε ένα από τα είδη προσομοιώσεων εκπαιδεύτηκαν τέσσερις διαφορετικοί ταξινομητές μηχανικής μάθησης (Classifier): το MLP_500_100 (μέγεθος *input layer* = 500, μέγεθος *hidden layer* = 100), το MLP_n_n/2 (μέγεθος *input layer* = n , μέγεθος *hidden layer* = $n/2$, όπου n = αριθμός features), το SVM (kernel='rbf') και το Random Forest (*max_features* = 12). Για τις προσομοιώσεις χρησιμοποιήθηκαν 10000 δείγματα από το κάθε σύνολο δεδομένων, τα οποία μετά από την επαύξηση έγιναν 30000. Συνολικά, πραγματοποιήθηκαν περισσότερες από εκατόν είκοσι διαφορετικές δοκιμές. Για κάθε δοκιμή έγιναν δέκα επαναλήψεις της εκπαίδευσης (training) και υπολογίστηκε η μέση τιμή και η τυπική απόκλιση του accuracy².

Πίνακας 4.5. Πέντε τύποι προσομοιώσεων που αφορούν διαφορετικές διαδικασίες προεπεξεργασίας.

Πέντε τύποι προσομοιώσεων

- A1.** Augment=yes, Combine=min, Adjust=yes, Valid= no
- A2.** Augment=yes, Combine=max, Adjust=yes, Valid= no
- B1.** Augment=yes, Combine=max, Adjust=no, Valid=yes
- C1.** Augment=yes, Combine=max, Adjust=no, Valid= no
- D1.** Augment=yes, Combine=max, Adjust=yes, Valid=yes

Όλες οι παραπάνω προσομοιώσεις εκτελέστηκαν από μια φορά, χρησιμοποιώντας το RetModel1. Οι προσομοιώσεις που έδωσαν τα καλύτερα αποτελέσματα, εκτελέστηκαν ακόμα μια φορά χρησιμοποιώντας το RetModel2, με σκοπό να συγκρίνουμε την απόδοση διαφορετικών μοντέλων αμφιβληστροειδούς.

²Οι δέκα επαναλήψεις έγιναν για όλα τα μοντέλα εκτός από το SVM για το οποίο έγινε μία μόνο επανάληψη λόγω του νετερμινιστικού χαρακτήρα του αλγορίθμου.

Πίνακας 4.6. Σύνοψη των προσομοιώσεων που έγιναν ανά σύνολο δεδομένων. Όλες οι διαφορετικές τιμές που μπορούν να πάρουν οι παράμετροι των προσομοιώσεων φαίνονται στον Πίνακα 4.4. Τα σύνολα δεδομένων παρουσιάζονται στον Πίνακα 4.1

RetModel1				RetModel2			
CIFAR10	MNIST	Fashion_MNIST	Imagenette	CIFAR10	MNIST	Fashion_MNIST	Imagenette
A1	A1						
A2	A2	A2	A2	A2	A2	A2	A2
B1	B1						
C1	C1	C1	C1	C1	C1	C1	C1
D1	D1						

Κεφάλαιο 5

Αποτελέσματα

Για κάθε μία από τις προσομοιώσεις που αναφέρθηκαν στον Πίνακα 4.6, παρουσιάζονται παρακάτω δύο πίνακες για κάθε σύνολο δεδομένων με την απόδοση (accuracy) των μοντέλων. Στους πίνακες φαίνονται η μέση τιμή και η τυπική απόκλιση για κάθε δοκιμή. Ο πρώτος πίνακας για κάθε σύνολο δεδομένων έχει τα αποτελέσματα για τις προσομοιώσεις A2 και D1, στα οποία έγινε το Βήμα 6 της προεπεξεργασίας (Adjust=yes) και στο Βήμα 9 επιλέχθηκε να συνδυαστούν οι 5 νευρικές αποκρίσεις σε μία με τον μετασχηματισμό max (Combine=max). Αντίστοιχα, ο δεύτερος πίνακας για κάθε σύνολο δεδομένων έχει τα αποτελέσματα για τις προσομοιώσεις B1 και C1, στα οποία δεν έγινε το Βήμα 6 της προεπεξεργασίας (Adjust=no) και στο Βήμα 9 επιλέχθηκε να συνδυαστούν οι 5 νευρικές αποκρίσεις σε μία με τον μετασχηματισμό max (Combine=max). Σημειώνεται ότι στους πίνακες, οι σειρές που φαίνονται με κίτρινο χρώμα αντιστοιχούν σε προσομοιώσεις με Valid=yes, ενώ αυτές με γαλάζιο αντιστοιχούν σε προσομοιώσεις με Valid=no.

5.1 Απόδοση Ταξινομητών στο Σύνολο Δεδομένων CIFAR10

Στον Πίνακα 5.1 εξετάζεται η απόδοση των ταξινομητών στο πρόβλημα αναγνώρισης αντικειμένου στο σύνολο δεδομένων CIFAR10 με χρήση του RetModel1. Παρουσιάζονται τα αποτελέσματα για τις προσομοιώσεις A2 και D1 με "Adjust=yes" και για τις προσομοιώσεις B1 και C1 με "Adjust=no".

Στον Πίνακα 5.2 εξετάζεται η απόδοση των ταξινομητών στο πρόβλημα αναγνώρισης αντικειμένου στο σύνολο δεδομένων CIFAR10 με χρήση του RetModel2. Παρουσιάζονται τα αποτελέσματα για τις προσομοιώσεις A2 με "Adjust=yes" και C1 με "Adjust=no".

Πίνακας 5.1. Μέση τιμή και τυπική απόκλιση της απόδοσης των ταξινομητών στο σύνολο δεδομένων CIFAR10 (Πίνακας 4.1) με χρήση του RetModel1 (Ενότητα 4.4) για τις προσομοιώσεις A2, D1, B1, C1 (Πίνακες 4.4, 4.5). Οι σειρές που φαίνονται με κίτρινο χρώμα αντιστοιχούν σε προσομοιώσεις με Valid=yes, ενώ αυτές με γαλάζιο αντιστοιχούν σε προσομοιώσεις με Valid=no.

CIFAR10				
Simulations A2 & D1 (Adjust=yes, Combine=max)				
n_features (n)	Classifiers			
	MLP_500_100	MLP_n_n/2	SVM	Random Forest
12 (Valid=yes, p=1)	13.179% (+/-4.173)	15.687% (+/-0.856)	19.567% (+/-0.000)	30.890% (+/-0.241)
48 (Valid=yes, p=2)	11.590% (+/-3.605)	15.378% (+/-4.673)	26.667% (+/-0.000)	39.208% (+/-0.300)
60 (Valid=no, p=1)	21.912% (+/-0.365)	20.537% (+/-0.730)	24.850% (+/-0.000)	41.365% (+/-0.219)
108 (Valid=yes, p=3)	11.147% (+/-2.972)	19.046% (+/-4.575)	31.550% (+/-0.000)	40.968% (+/-0.460)
240 (Valid=no, p=2)	24.567% (+/-0.380)	25.283% (+/-0.872)	33.333% (+/-0.000)	40.740% (+/-0.585)
540 (Valid=no, p=3)	26.288% (+/-1.264)	27.186% (+/-1.108)	41.200% (+/-0.000)	40.945% (+/-0.383)
960 (Valid=no, p=4)	26.895% (+/-0.744)	27.771% (+/-0.896)	47.083% (+/-0.000)	40.990% (+/-0.320)
Simulations B1 & C1 (Adjust=no, Combine=max)				
n_features (n)	Classifiers			
	MLP_500_100	MLP_n_n/2	SVM	Random Forest
12 (Valid=yes, p=1)	18.027% (+/-0.422)	21.258% (+/-0.204)	19.150% (+/-0.000)	28.152% (+/-0.219)
48 (Valid=yes, p=2)	28.501% (+/-0.304)	26.259% (+/-0.318)	30.700% (+/-0.000)	34.548% (+/-0.330)
60 (Valid=no, p=1)	28.138% (+/-0.547)	25.254% (+/-0.171)	32.333% (+/-0.000)	38.685% (+/-0.518)
108 (Valid=yes, p=3)	31.460% (+/-0.387)	30.054% (+/-0.422)	33.333% (+/-0.000)	35.667% (+/-0.420)
240 (Valid=no, p=2)	32.063% (+/-0.526)	31.562% (+/-0.515)	30.167% (+/-0.000)	35.225% (+/-0.369)
540 (Valid=no, p=3)	35.121% (+/-0.277)	35.685% (+/-0.517)	19.417% (+/-0.000)	36.255% (+/-0.458)

5.2 Απόδοση Ταξινομητών στο Σύνολο Δεδομένων MNIST

Στον Πίνακα 5.3 εξετάζεται η απόδοση των ταξινομητών στο πρόβλημα αναγνώρισης αντικειμένου στο σύνολο δεδομένων MNIST με χρήση του RetModel1. Παρουσιάζονται τα αποτελέσματα για τις προσομοιώσεις A2 και D1 με "Adjust=yes" και για

Πίνακας 5.2. Μέση τιμή και τυπική απόκλιση της απόδοσης των ταξινομητών στο σύνολο δεδομένων CIFAR10 (Πίνακας 4.1) με χρήση του RetModel2 (Ενότητα 4.4) για τις προσομοιώσεις A2, C1 (Πίνακες 4.4, 4.5).

CIFAR10					
Model2	Simulation A2 (<i>Adjust=yes, Combine=max</i>)				
	<i>n_features (n)</i>	Classifiers			
		MLP_500_100	MLP_n_n/2	SVM	Random Forest
	540 (Valid=no, p=3)	12.321% (+/-3.423)	10.117% (+/-0.151)	31.650% (+/-0.000)	40.212% (+/-0.389)
	960 (Valid=no, p=4)	11.172% (+/-2.628)	10.288% (+/-0.109)	34.900% (+/-0.000)	38.220% (+/-0.388)
	Simulation C1 (<i>Adjust=no, Combine=max</i>)				
	<i>n_features (n)</i>	Classifiers			
		MLP_500_100	MLP_n_n/2	SVM	Random Forest
540 (Valid=no, p=3)	31.217% (+/-1.081)	31.893% (+/-0.703)	44.183% (+/-0.000)	39.302% (+/-0.455)	
960 (Valid=no, p=4)	30.903% (+/-0.646)	31.271% (+/-0.685)	36.300% (+/-0.000)	37.190% (+/-0.312)	

τις προσομοιώσεις B1 και C1 με "Adjust=no".

Στον Πίνακα 5.4 εξετάζεται η απόδοση των ταξινομητών στο πρόβλημα αναγνώρισης αντικειμένου στο σύνολο δεδομένων MNIST με χρήση του RetModel2. Παρουσιάζονται τα αποτελέσματα για τις προσομοιώσεις A2 με "Adjust=yes" και C1 με "Adjust=no".

5.3 Απόδοση Ταξινομητών στο Σύνολο Δεδομένων Fashion MNIST

Στον Πίνακα 5.5 εξετάζεται η απόδοση των ταξινομητών στο πρόβλημα αναγνώρισης αντικειμένου στο σύνολο δεδομένων Fashion MNIST με χρήση του RetModel1. Παρουσιάζονται τα αποτελέσματα για τις προσομοιώσεις A2 με "Adjust=yes" και C1 με "Adjust=no".

Στον Πίνακα 5.6 εξετάζεται η απόδοση των ταξινομητών στο πρόβλημα αναγνώρισης αντικειμένου στο σύνολο δεδομένων Fashion MNIST με χρήση του RetModel2. Παρουσιάζονται τα αποτελέσματα για τις προσομοιώσεις A2 με "Adjust=yes" και C1 με "Adjust=no".

Πίνακας 5.3. Μέση τιμή και τυπική απόκλιση της απόδοσης των ταξινομητών στο σύνολο δεδομένων MNIST (Πίνακας 4.1) με χρήση του RetModel1 (Ενότητα 4.4) για τις προσομοιώσεις A2, D1, B1, C1 (Πίνακες 4.4, 4.5). Οι σειρές που φαίνονται με κίτρινο χρώμα αντιστοιχούν σε προσομοιώσεις με *Valid=yes*, ενώ αυτές με γαλάζιο αντιστοιχούν σε προσομοιώσεις με *Valid=no*.

MNIST				
Simulations A2 & D1 (<i>Adjust=yes, Combine=max</i>)				
n_features (n)	Classifiers			
	MLP_500_100	MLP_n_n/2	SVM	Random Forest
12 (Valid=yes, p=1)	44.269% (+/-0.955)	27.558% (+/-5.680)	47.167% (+/-0.000)	65.377% (+/-0.208)
48 (Valid=yes, p=2)	56.251% (+/-1.502)	48.671% (+/-1.433)	72.967% (+/-0.000)	82.377% (+/-0.385)
60 (Valid=no, p=1)	68.492% (+/-1.268)	58.146% (+/-1.185)	72.033% (+/-0.000)	82.723% (+/-0.262)
108 (Valid=yes, p=3)	59.221% (+/-2.234)	59.208% (+/-2.279)	83.467% (+/-0.000)	87.838% (+/-0.160)
240 (Valid=no, p=2)	75.112% (+/-1.885)	75.376% (+/-1.019)	87.717% (+/-0.000)	87.975% (+/-0.241)
540 (Valid=no, p=3)	77.737% (+/-1.146)	79.301% (+/-1.964)	91.233% (+/-0.000)	89.300% (+/-0.223)
Simulations B1 & C1 (<i>Adjust=no, Combine=max</i>)				
n_features (n)	Classifiers			
	MLP_500_100	MLP_n_n/2	SVM	Random Forest
12 (Valid=yes, p=1)	48.507% (+/-0.352)	34.019% (+/-0.640)	44.667% (+/-0.000)	55.225% (+/-0.285)
48 (Valid=yes, p=2)	72.159% (+/-0.412)	59.804% (+/-0.253)	58.950% (+/-0.000)	74.032% (+/-0.374)
60 (Valid=no, p=1)	75.445% (+/-0.402)	65.940% (+/-0.486)	57.883% (+/-0.000)	76.327% (+/-0.280)
108 (Valid=yes, p=3)	78.693% (+/-0.501)	74.670% (+/-0.403)	44.500% (+/-0.000)	81.632% (+/-0.200)
240 (Valid=no, p=2)	84.258% (+/-0.390)	83.844% (+/-0.495)	27.500% (+/-0.000)	82.165% (+/-0.369)
540 (Valid=no, p=3)	87.844% (+/-0.716)	88.648% (+/-0.794)	16.317% (+/-0.000)	85.115% (+/-0.221)

5.4 Απόδοση Ταξινομητών στο Σύνολο Δεδομένων Imagenette

Στον Πίνακα 5.7 εξετάζεται η απόδοση των ταξινομητών στο πρόβλημα αναγνώρισης αντικειμένου στο σύνολο δεδομένων Imagenette με χρήση του RetModel1. Παρουσιάζονται τα αποτελέσματα για τις προσομοιώσεις A2 με "Adjust=yes" και C1 με "Adjust=no".

Πίνακας 5.4. Μέση τιμή και τυπική απόκλιση της απόδοσης των ταξινομητών στο σύνολο δεδομένων MNIST (Πίνακας 4.1) με χρήση του RetModel2 (Ενότητα 4.4) για τις προσομοιώσεις A2, C1 (Πίνακες 4.4, 4.5).

MNIST					
Model2	Simulation A2 (Adjust=yes, Combine=max)				
	n_features (n)	Classifiers			
		MLP_500_100	MLP_n_n/2	SVM	Random Forest
	540 (Valid=no, p=3)	24.389% (+/-16.507)	12.740% (+/-4.736)	81.700% (+/-0.000)	88.417% (+/-0.292)
	960 (Valid=no, p=4)	29.564% (+/-16.920)	11.384% (+/-0.000)	87.617% (+/-0.000)	89.442% (+/-0.111)
	Simulation C1 (Adjust=no, Combine=max)				
	n_features (n)	Classifiers			
		MLP_500_100	MLP_n_n/2	SVM	Random Forest
	540 (Valid=no, p=3)	79.795% (+/-1.116)	81.186% (+/-1.013)	74.467% (+/-0.000)	83.668% (+/-0.302)
	960 (Valid=no, p=4)	80.874% (+/-2.188)	84.104% (+/-1.648)	64.117% (+/-0.000)	85.195% (+/-0.214)

Πίνακας 5.5. Μέση τιμή και τυπική απόκλιση της απόδοσης των ταξινομητών στο σύνολο δεδομένων Fashion MNIST (Πίνακας 4.1) με χρήση του RetModel1 (Ενότητα 4.4) για τις προσομοιώσεις A2, C1 (Πίνακες 4.4, 4.5).

FASHION MNIST					
Model1	Simulation A2 (Adjust=yes, Combine=max)				
	n_features (n)	Classifiers			
		MLP_500_100	MLP_n_n/2	SVM	Random Forest
	240 (Valid=no, p=2)	66.742% (+/-1.356)	67.505% (+/-1.392)	76.687% (+/-0.000)	78.255% (+/-0.238)
	540 (Valid=no, p=3)	70.440% (+/-0.941)	70.948% (+/-1.841)	80.657% (+/-0.000)	79.226% (+/-0.100)
	960 (Valid=no, p=4)	69.933% (+/-1.406)	71.167% (+/-1.271)	79.616% (+/-0.000)	79.359% (+/-0.126)
	Simulation C1 (Adjust=no, Combine=max)				
	n_features (n)	Classifiers			
		MLP_500_100	MLP_n_n/2	SVM	Random Forest
	240 (Valid=no, p=2)	76.695% (+/-1.006)	75.911% (+/-0.586)	44.606% (+/-0.000)	75.684% (+/-0.314)
540 (Valid=no, p=3)	78.853% (+/-0.781)	79.254% (+/-0.704)	19.323% (+/-0.000)	77.525% (+/-0.197)	
960 (Valid=no, p=4)	79.544% (+/-0.609)	79.591% (+/-0.694)	13.586% (+/-0.000)	77.718% (+/-0.180)	

Στον Πίνακα 5.8 εξετάζεται η απόδοση των ταξινομητών στο πρόβλημα αναγνώρισης αντικειμένου στο σύνολο δεδομένων Imagenette με χρήση του RetModel2. Παρουσιάζονται τα αποτελέσματα για τις προσομοιώσεις A2 με "Adjust=yes" και C1 με "Adjust=no".

Πίνακας 5.6. Μέση τιμή και τυπική απόκλιση της απόδοσης των ταξινομητών στο σύνολο δεδομένων Fashion MNIST (Πίνακας 4.1) με χρήση του RetModel2 (Ενότητα 4.4) για τις προσομοιώσεις A2, C1 (Πίνακες 4.4, 4.5).

FASHION MNIST					
Model2	Simulation A2 (Adjust=yes, Combine=max)				
	n_features (n)	Classifiers			
		MLP_500_100	MLP_n_n/2	SVM	Random Forest
	540 (Valid=no, p=3)	57.700% (+/-2.267)	59.808% (+/-1.096)	75.444% (+/-0.000)	79.125% (+/-0.185)
	960 (Valid=no, p=4)	56.576% (+/-3.655)	56.978% (+/-2.118)	77.707% (+/-0.000)	79.544% (+/-0.190)
	Simulation C1 (Adjust=no, Combine=max)				
n_features (n)	Classifiers				
	MLP_500_100	MLP_n_n/2	SVM	Random Forest	
540 (Valid=no, p=3)	72.422% (+/-1.582)	74.012% (+/-0.962)	77.545% (+/-0.000)	76.776% (+/-0.136)	
960 (Valid=no, p=4)	73.192% (+/-0.605)	74.542% (+/-0.771)	64.434% (+/-0.000)	77.268% (+/-0.159)	

Πίνακας 5.7. Μέση τιμή και τυπική απόκλιση της απόδοσης των ταξινομητών στο σύνολο δεδομένων Imagenette (Πίνακας 4.1) με χρήση του RetModel1 (Ενότητα 4.4) για τις προσομοιώσεις A2, C1 (Πίνακες 4.4, 4.5).

IMAGENETTE					
Model1	Simulation A2 (Adjust=yes, Combine=max)				
	n_features (n)	Classifiers			
		MLP_500_100	MLP_n_n/2	SVM	Random Forest
	240 (Valid=no, p=2)	25.160% (+/-0.629)	25.648% (+/-1.375)	34.576% (+/-0.000)	44.905% (+/-0.200)
	540 (Valid=no, p=3)	26.944% (+/-0.673)	27.403% (+/-0.722)	43.384% (+/-0.000)	45.964% (+/-0.238)
	960 (Valid=no, p=4)	28.900% (+/-0.592)	28.589% (+/-1.479)	44.414% (+/-0.000)	46.364% (+/-0.195)
Simulation C1 (Adjust=no, Combine=max)					
n_features (n)	Classifiers				
	MLP_500_100	MLP_n_n/2	SVM	Random Forest	
240 (Valid=no, p=2)	36.070% (+/-0.423)	34.220% (+/-0.491)	32.687% (+/-0.000)	43.665% (+/-0.188)	
540 (Valid=no, p=3)	39.416% (+/-0.616)	40.588% (+/-0.765)	29.030% (+/-0.000)	45.108% (+/-0.329)	
960 (Valid=no, p=4)	40.488% (+/-0.464)	41.401% (+/-0.650)	21.929% (+/-0.000)	45.087% (+/-0.444)	

Πίνακας 5.8. Μέση τιμή και τυπική απόκλιση της απόδοσης των ταξινομητών στο σύνολο δεδομένων Imagenette (Πίνακας 4.1) με χρήση του RetModel2 (Ενότητα 4.4) για τις προσομοιώσεις A2, C1 (Πίνακες 4.4, 4.5).

IMAGENETTE					
Model2	Simulation A2 (<i>Adjust=yes, Combine=max</i>)				
	<i>n_features (n)</i>	Classifiers			
		MLP_500_100	MLP_n_n/2	SVM	Random Forest
	540 (Valid=no, p=3)	10.917% (+/-2.451)	10.036% (+/-0.119)	32.667% (+/-0.000)	45.933% (+/-0.160)
	960 (Valid=no, p=4)	10.158% (+/-0.073)	10.109% (+/-0.111)	37.444% (+/-0.000)	46.239% (+/-0.311)
	Simulation C1 (<i>Adjust=no, Combine=max</i>)				
	<i>n_features (n)</i>	Classifiers			
		MLP_500_100	MLP_n_n/2	SVM	Random Forest
	540 (Valid=no, p=3)	30.923% (+/-0.840)	31.276% (+/-0.693)	42.121% (+/-0.000)	45.598% (+/-0.174)
	960 (Valid=no, p=4)	32.227% (+/-0.569)	32.447% (+/-0.906)	45.213% (+/-0.316)	35.152% (+/-0.000)

5.5 Συγκριτική Ανάλυση Αποτελεσμάτων

Σε αυτήν την ενότητα, παρουσιάζουμε και συζητάμε τα συγκριτικά αποτελέσματα των προσομοιώσεων που διεξάγαμε. Συγκρίνουμε τα διαφορετικά μοντέλα αμφιβληστροειδούς, τα διαφορετικά σύνολα δεδομένων και τους διαφορετικούς ταξινομητές που χρησιμοποιήσαμε. Αρχικά, παρουσιάζουμε την ανάλυση ευαισθησίας μέσω διαγραμμάτων (sensitivity plots), τα οποία συσχετίζουν την απόδοση κάθε ταξινομητή με τον αριθμό των νευρώνων που χρησιμοποιήθηκαν ως χαρακτηριστικά (features). Δεύτερον, παρουσιάζουμε ιστογράμματα που συγκρίνουν την απόδοση των ταξινομητών μεταξύ των τεσσάρων συνόλων δεδομένων (CIFAR10, Imagenette, MNIST, Fashion MNIST). Έπειτα, παρουσιάζουμε ιστογράμματα που συγκρίνουν την απόδοση των ταξινομητών ανάμεσα σε προσομοιώσεις που έγιναν με "Adjust=yes" κι εκείνες που έγιναν με "Adjust=no". Τέλος, παρουσιάζουμε ιστογράμματα που συγκρίνουν την απόδοση των ταξινομητών ανάμεσα στο RetModel1 και στο RetModel2 για όλα τα σύνολα δεδομένων.

5.5.1 Αξιολόγηση της απόδοσης των ταξινομητών σαν συνάρτηση του αριθμού των νευρώνων

Με βάση τα διαγράμματα ανάλυσης ευαισθησίας sensitivity plots στο Σχήμα 5.1 και στο Σχήμα 5.2, μπορούμε να κάνουμε μερικές χρήσιμες παρατηρήσεις. Πρώτα απ' όλα, παρατηρούμε ότι όσο περισσότερους νευρώνες (ή $n_features$) χρησιμοποιούμε, τόσο καλύτερη απόδοση έχουμε. Στις περισσότερες περιπτώσεις, επιτυγχάνουμε την υψηλότερη απόδοση για 540 νευρώνες, ενώ δεν παρατηρείται περαιτέρω βελτίωση για παραπάνω από 240 νευρώνες και ειδικά για παραπάνω από 540 νευρώνες, όπως φαίνεται στα διαγράμματα 5.1γ, δ και 5.2γ, δ. Δεύτερον, το Random Forest αποδίδει σταθερά καλύτερα από τους υπόλοιπους ταξινομητές, ακολουθούμενο από το SVM και στο τέλος, υπάρχουν οι MLP ταξινομητές με τη χαμηλότερη απόδοση σε όλα τις προσομοιώσεις. Ωστόσο, στις προσομοιώσεις B1 και C1 (όπου "Adjust = no"), η απόδοση του SVM φαίνεται να καταρρέει μετά από 240 νευρώνες. Τρίτον, παρατηρούμε ότι οι ταξινομητές αποδίδουν σημαντικά καλύτερα στα σύνολα δεδομένων MNIST και Fashion MNIST, επιτυγχάνοντας περίπου 80% ακρίβεια ή περισσότερο. Από την άλλη πλευρά, οι ταξινομητές επιτυγχάνουν περίπου 40% στα CIFAR10 και Imagenette.

5.5.2 Σύγκριση της απόδοσης των ταξινομητών μεταξύ των συνόλων δεδομένων.

Στο Σχήμα 5.3, σχεδιάζουμε ιστογράμματα για να συγκρίνουμε τη μέγιστη απόδοση διαφορετικών ταξινομητών μεταξύ CIFAR10 και Imagenette και μεταξύ MNIST

και Fashion MNIST. Όπως αναφέρθηκε προηγουμένως, επιλέξαμε το Imagenette ως ένα βελτιωμένο σύνολο δεδομένων (εικόνες με υψηλότερη ανάλυση) σε σχέση με το CIFAR10 και το mbox Fashion MNIST ως ένα πιο δύσκολο και πιο περίπλοκο σύνολο δεδομένων σε σχέση με το MNIST και γι αυτό τα συγκρίνουμε σε ζευγάρια. Το MNIST αποδίδει περίπου 10% καλύτερα από το Fashion MNIST σε όλες τις προσομοιώσεις, μια διαφορά που είναι συνεπής με τα αποτελέσματα της βιβλιογραφίας [20]. Όσον αφορά τη σύγκριση μεταξύ CIFAR10 και Imagenette, παρατηρούμε διαφορετικά αποτελέσματα ανάλογα με τον ταξινομητή που χρησιμοποιείται και το μοντέλο αμφιβληστροειδούς. Συγκεκριμένα, τόσο το MLP όσο και το Random Forest αποδίδουν καλύτερα με το Imagenette. Το SVM αποδίδει καλύτερα με το CIFAR10 χρησιμοποιώντας το RetModel1, ενώ αποδίδει καλύτερα με το Imagenette χρησιμοποιώντας το RetModel2.

5.5.3 Σύγκριση της απόδοσης των ταξινομητών μεταξύ των προσομοιώσεων με "Adjust=yes" και "Adjust=no"

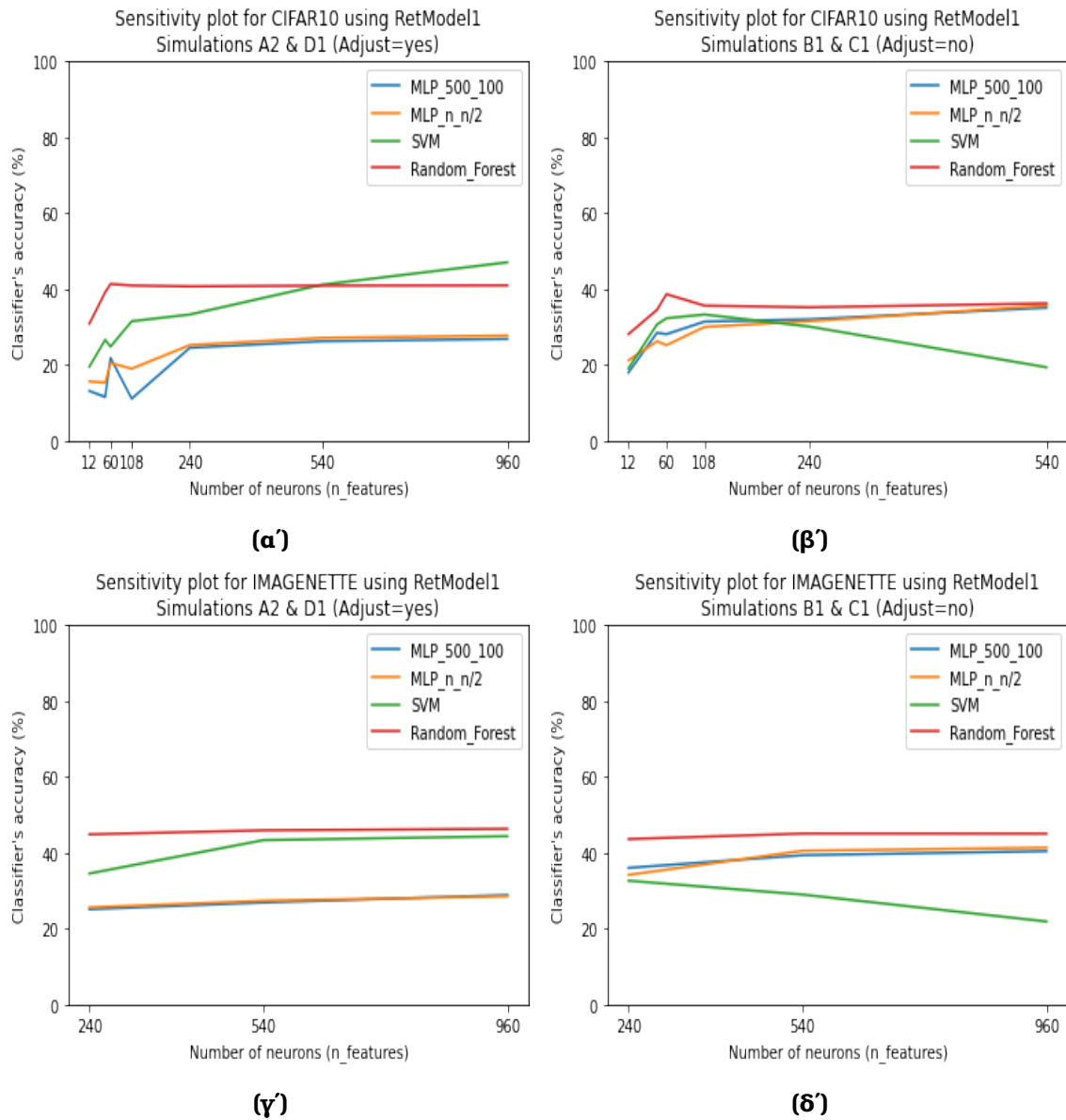
Στο Σχήμα 5.4, συγκρίνουμε τη μέγιστη απόδοση διαφορετικών ταξινομητών μεταξύ των προσομοιώσεων που πραγματοποιήθηκαν με "Adjust = yes" και εκείνων που πραγματοποιήθηκαν με "Adjust = no", για το CIFAR10 και το Imagenette. Και για τα δύο σύνολα δεδομένων παρατηρούμε όμοια απόδοση. Random Forest αποδίδει πάντα σχεδόν το ίδιο τόσο "Adjust = yes" για όσο και για "Adjust = no". Το SVM με "Adjust = yes" αποδίδει καλύτερα για το RetModel1, ενώ το SVM με "Adjust = no" δίνει καλύτερα αποτελέσματα για το RetModel2. Όσον αφορά τα MLPs, δίνουν σταθερά καλύτερα αποτελέσματα με "Adjust = no" και για τα δύο μοντέλα αμφιβληστροειδούς, ενώ αποτυγχάνουν εντελώς με το "Adjust = yes" και το RetModel2.

Στο Σχήμα 5.5, συγκρίνουμε τη μέγιστη απόδοση διαφορετικών ταξινομητών μεταξύ των προσομοιώσεων που πραγματοποιήθηκαν με "Adjust = yes" και εκείνων που πραγματοποιήθηκαν με "Adjust = no", για το MNIST και το Fashion MNIST. Και για τα δύο σύνολα δεδομένων, το SVM και το Random Forest αποδίδει καλύτερα με "Adjust = yes". Τα MLPs αποδίδουν καλύτερα με "Adjust = no" και για τα δύο μοντέλα αμφιβληστροειδούς, ενώ αποτυγχάνουν εντελώς με "Adjust = yes" και το RetModel2 και το MNIST.

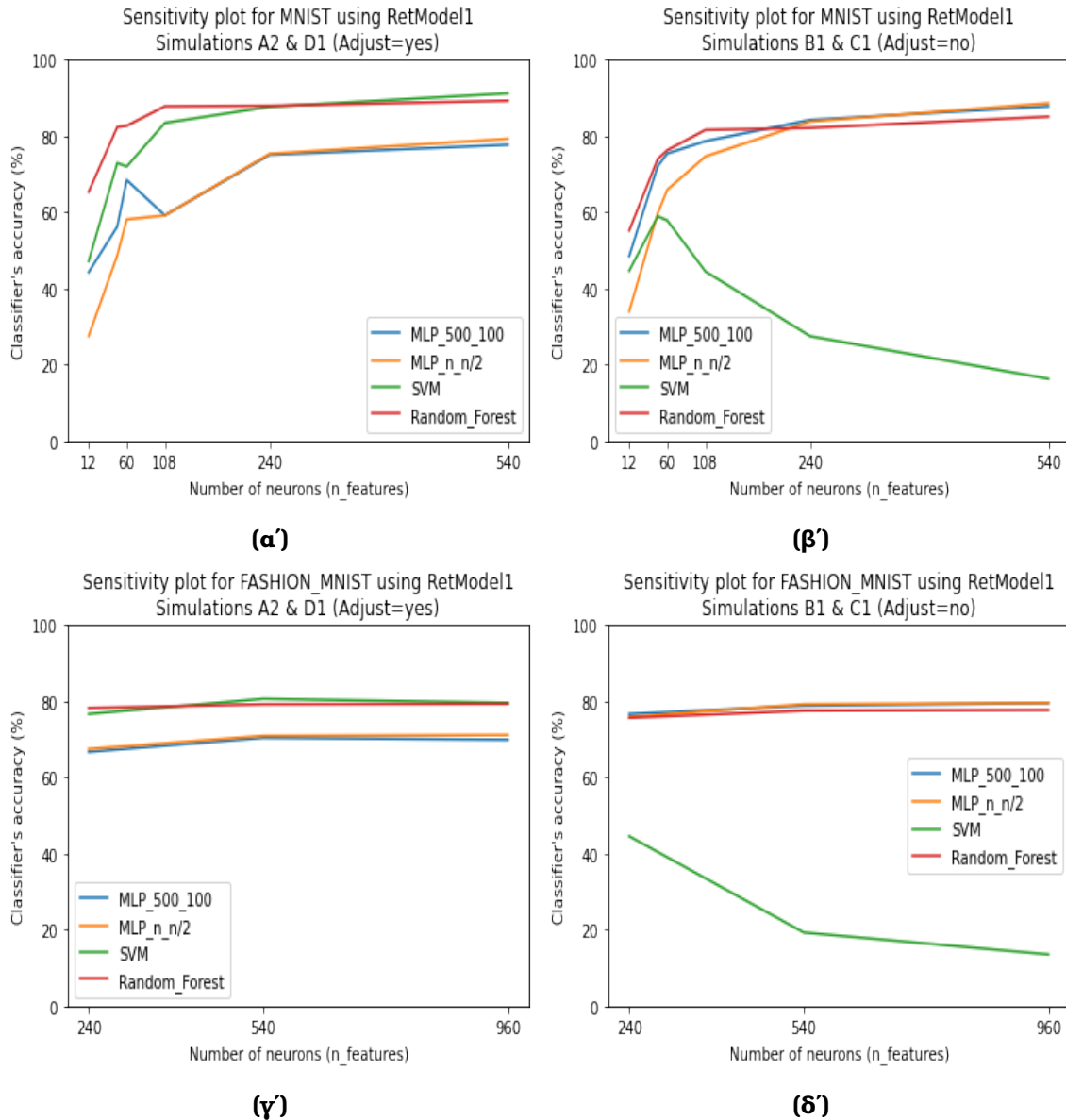
5.5.4 Σύγκριση της απόδοσης των ταξινομητών μεταξύ των μοντέλων αμφιβληστροειδούς.

Τέλος, συγκρίνουμε, για όλα τα σύνολα δεδομένων, την απόδοση διαφορετικών ταξινομητών μεταξύ RetModel1 και RetModel2, όπως φαίνεται στο Σχήμα 5.6. Μια άμεση παρατήρηση είναι ότι το RetModel1 αποδίδει καλύτερα (αν και μόνο για λίγο

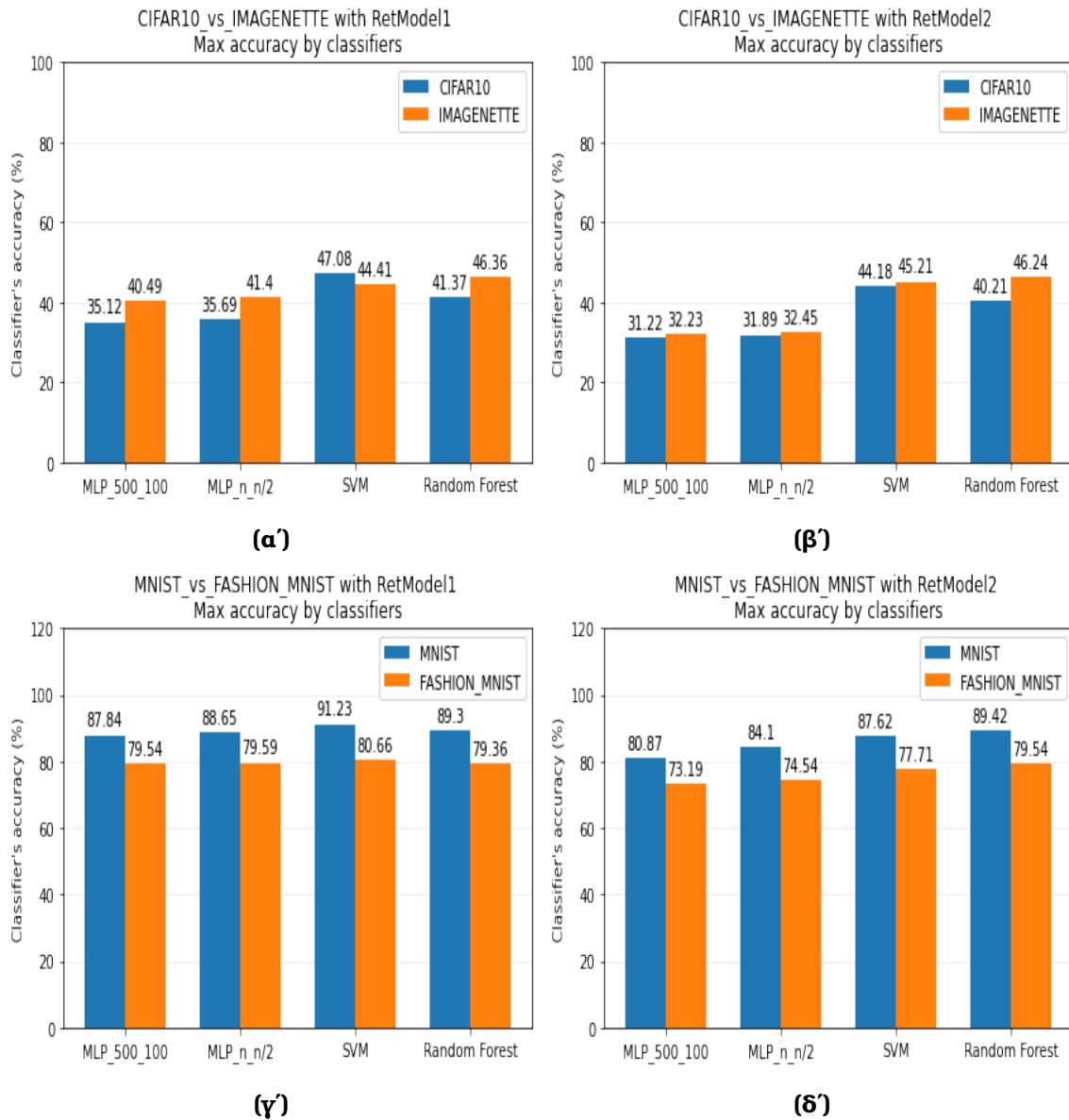
σε ορισμένες περιπτώσεις) από το RetModel2.



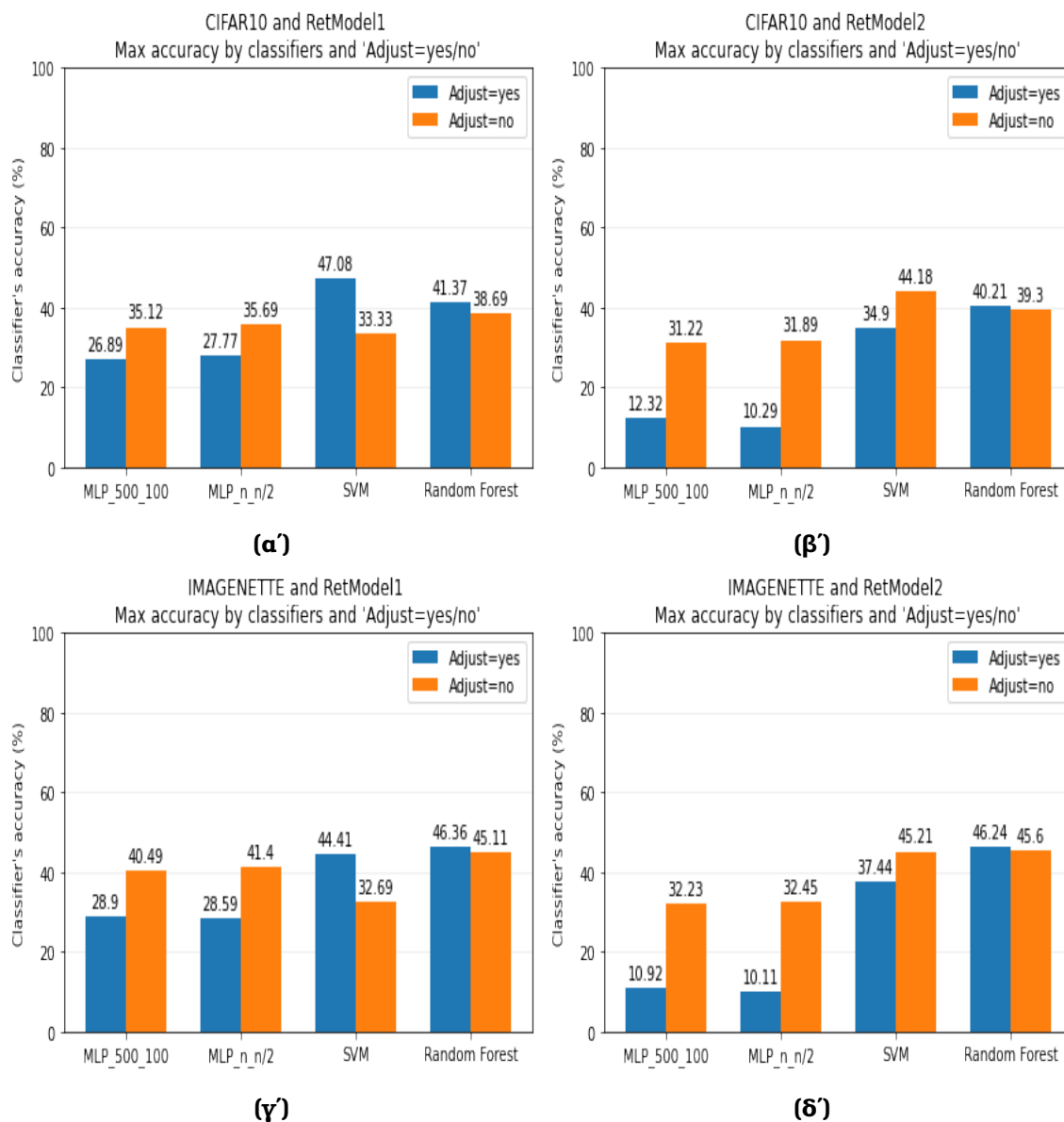
Σχήμα 5.1. Διαγράμματα ανάλυσης ευαισθησίας με χρήση του RetModel1 (Ενότητα 4.4) για διαφορετικά σύνολα δεδομένων (Πίνακας 4.1), διαφορετικές προσομοιώσεις (Πίνακας 4.5) και διαφορετικούς συνδυασμούς παραμέτρων (Πίνακας 4.4). (α) Σύνολο δεδομένων CIFAR10, προσομοιώσεις A2 & D1 ("Adjust=yes"). (β) Σύνολο δεδομένων CIFAR10, προσομοιώσεις B1 & C1 ("Adjust=no"). (γ) Σύνολο δεδομένων Imagenette, προσομοιώσεις A2 & D1 ("Adjust=yes"). (δ) Σύνολο δεδομένων Imagenette, προσομοιώσεις B1 & C1 ("Adjust=no").



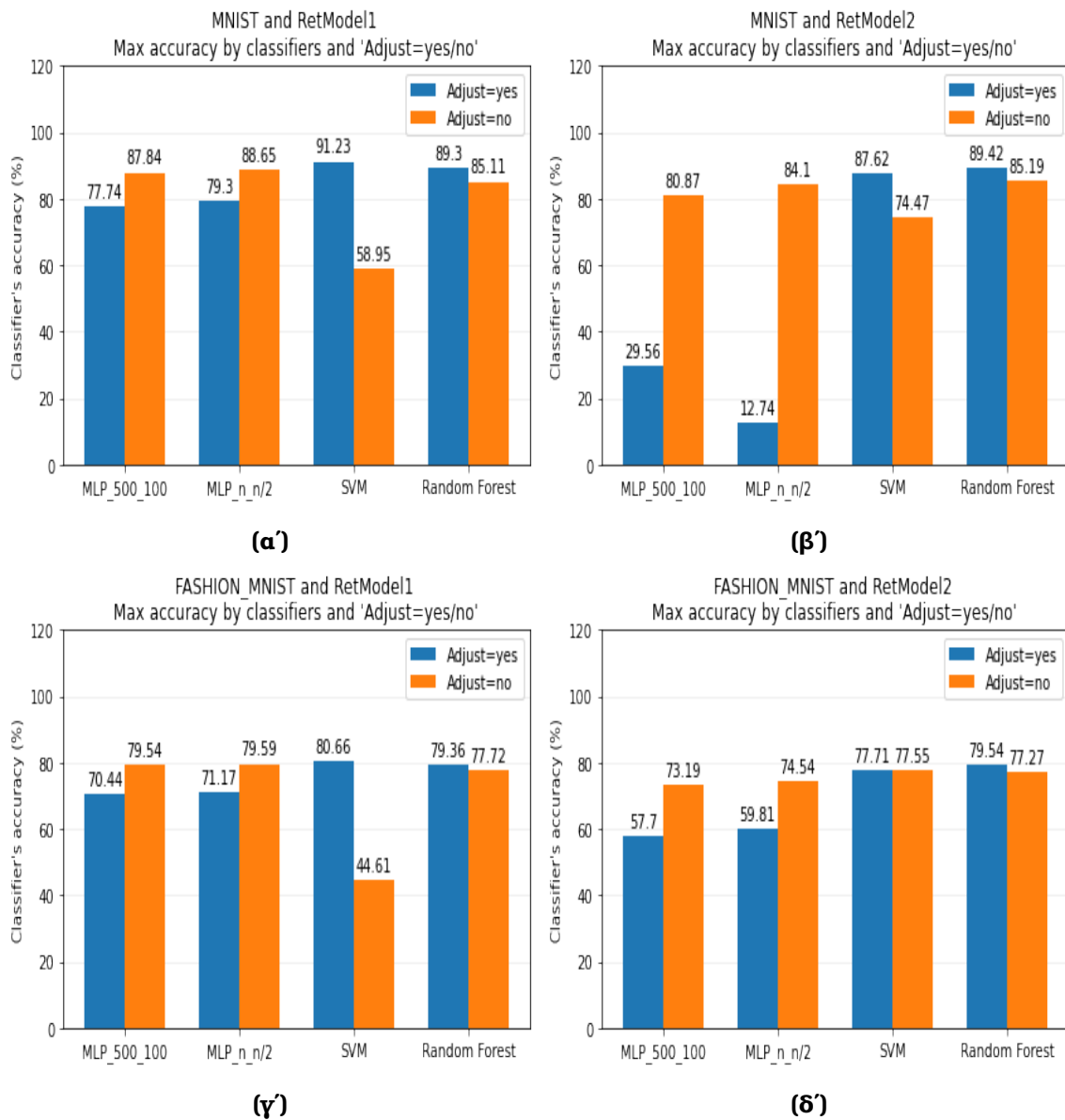
Σχήμα 5.2. Διαγράμματα ανάλυσης ευαισθησίας με χρήση του RetModel1 (Ενότητα 4.4) για διαφορετικά σύνολα δεδομένων (Πίνακας 4.1), διαφορετικές προσομοιώσεις (Πίνακας 4.5) και διαφορετικούς συνδυασμούς παραμέτρων (Πίνακας 4.4). (α) Σύνολο δεδομένων MNIST, προσομοιώσεις A2 & D1 ("Adjust=yes"). (β) Σύνολο δεδομένων MNIST, προσομοιώσεις B1 & C1 ("Adjust=no"). (γ) Σύνολο δεδομένων Fashion MNIST, προσομοιώσεις A2 & D1 ("Adjust=yes"). (δ) Σύνολο δεδομένων Fashion MNIST, προσομοιώσεις B1 & C1 ("Adjust=no").



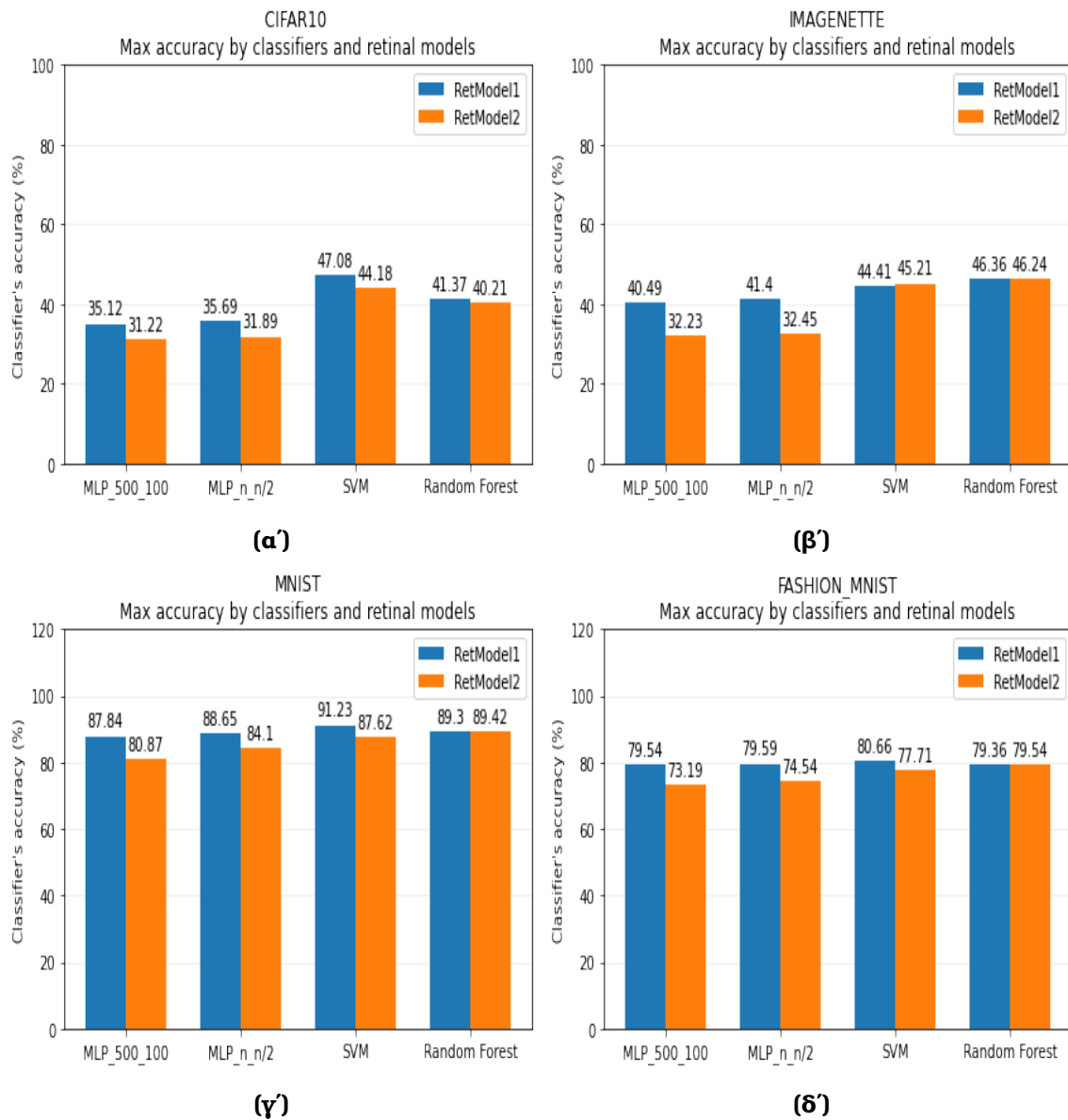
Σχήμα 5.3. Σύγκριση της μέγιστης απόδοσης των ταξινομητών μεταξύ διαφορετικών συνόλων δεδομένων (Πίνακας 4.1) για τα δύο μοντέλα αμφιδληστροειδούς (Ενότητα 4.4). (α) Σύγκριση μεταξύ CIFAR10 και Imagenette για το RetModel1. (β) Σύγκριση μεταξύ CIFAR10 και Imagenette για το RetModel2. (γ) Σύγκριση μεταξύ MNIST και Fashion MNIST για το RetModel1. (δ) Σύγκριση μεταξύ MNIST και Fashion MNIST για το RetModel2.



Σχήμα 5.4. Σύγκριση της μέγιστης απόδοσης των ταξινομητών μεταξύ των προσομοιώσεων με "Adjust=yes" και αυτών με "Adjust=no" (Πίνακα 4.4) για διαφορετικά σύνολα δεδομένων (Πίνακας 4.1) και για τα δύο μοντέλα αμφιβήητροειδούς (Ενότητες 4.4). (α) Σύγκριση για το CIFAR10 με χρήση του RetModel1. (β) Σύγκριση για το CIFAR10 με χρήση του RetModel2. (γ) Σύγκριση για το Imagenette με χρήση του RetModel1. (δ) Σύγκριση για το Imagenette με χρήση του RetModel2.



Σχήμα 5.5. Σύγκριση της μέγιστης απόδοσης των ταξινομητών μεταξύ των προσομοιώσεων με "Adjust=yes" και αυτών με "Adjust=no" (Πίνακα 4.4) για διαφορετικά σύνολα δεδομένων (Πίνακας 4.1) και για τα δύο μοντέλα αμφιβληστροειδούς (Ενότητα 4.4). (α) Σύγκριση για το MNIST με χρήση του RetModel1. (β) Σύγκριση για το MNIST με χρήση του RetModel2. (γ) Σύγκριση για το Fashion MNIST με χρήση του RetModel1. (δ) Σύγκριση για το Fashion MNIST με χρήση του RetModel2.



Σχήμα 5.6. Σύγκριση της μέγιστης απόδοσης των ταξινομητών ανάμεσα στα μοντέλα αμφιβληστροειδούς RetModel1 και RetModel2 (Ενότητα 4.4) για όλα τα σύνολα δεδομένων (Πίνακας 4.1). (α) Σύγκριση στο CIFAR10. (β) Σύγκριση στο Imagenette. (γ) Σύγκριση στο MNIST. (δ) Σύγκριση στο Fashion MNIST.

Κεφάλαιο **6**

Συζήτηση

Στόχος της διπλωματικής εργασίας ήταν ο σχεδιασμός ενός συστήματος λειτουργικής αξιολόγησης προκειμένου να συγκριθεί η απόδοση διαφορετικών μοντέλων αμφιβληστροειδούς σε προβλήματα αναγνώρισης εικόνας και συγκεκριμένα, σε προβλήματα αναγνώρισης αντικειμένων και ψηφίων. Οι πρώτες προσομοιώσεις έδειξαν ότι όσο περισσότεροι νευρώνες χρησιμοποιούνται ως χαρακτηριστικά για τους ταξινομητές τόσο καλύτερη είναι η απόδοση του μοντέλου. Με βάση αυτό, προσπαθήσαμε να αυξήσουμε τεχνητά τον περιορισμένο αριθμό διαθέσιμων νευρώνων (60 στην περίπτωση μας) των μοντέλων αμφιβληστροειδούς, προκειμένου να αυξήσουμε την απόδοση διαφόρων ταξινομητών. Για να το επιτύχουμε αυτό, χωρίσαμε την κάθε εικόνα σε επιμέρους κομμάτια και τροφοδοτήσαμε κάθε ένα από αυτά ξεχωριστά στο μοντέλο αμφιβληστροειδούς. Κατά τη διάρκεια των προσομοιώσεων, αναλύσαμε μια σειρά ερωτημάτων σχετικά με τα βήματα προεπεξεργασίας, τη χρήση διαφορετικών συνόλων δεδομένων και ταξινομητών και την απόδοση των μοντέλων αμφιβληστροειδούς. Τα αποτελέσματα, όπως παρουσιάστηκαν στο Κεφάλαιο 5, σχολιάζονται παρακάτω.

6.1 Συμπεράσματα

6.1.1 Προεπεξεργασία Δεδομένων

Όπως παρουσιάζεται στον Πίνακα 4.3, αρκετές παραμετρικές αποφάσεις χρειάζεται να ληφθούν υπόψιν κατά την προεπεξεργασία των δεδομένων. Συγκεκριμένα, πρέπει να αποφασιστεί (α) αν θα υλοποιηθεί ή όχι η επαύξηση των δεδομένων, (β) αν θα διαχωριστούν ή όχι οι εικόνες και στην περίπτωση που διαχωριστούν σε πόσα τμήματα, (γ) αν θα υλοποιηθεί adjustment πριν τροφοδοτηθούν στο μοντέλο αμφιβληστροειδούς, (δ) αν θα επιλεχθούν ή όχι οι (valid) έγκυροι νευρώνες και τελικά (ε) αν θα πρέπει να χρησιμοποιηθεί min ή max μετασχηματισμός προκειμένου να συνδυαστούν οι ρυθμοί νευρωνικών αποκρίσεων των επιμέρους κομματιών της εικόνας. Μελετήσαμε όλες αυτές τις αποφάσεις, με σκοπό την βελτιστοποίηση της

απόδοσης του μοντέλου. Συγκεκριμένα, οι προσομοιώσεις ανέδειξαν την σημασία της επαύξησης δεδομένων για την απόδοση των ταξινομητών, αποτέλεσμα που έρχεται σε συμφωνία με συνήθεις πρακτικές της Μηχανικής Μάθησης [43]. Επιπλέον, δείξαμε ότι όσο περισσότεροι νευρώνες χρησιμοποιηθούν ως χαρακτηριστικά των ταξινομητών, τόσο καλύτερα αυτοί αποδίδουν. Παρόλα αυτά, η χρήση περισσότερων από 540 ($9 * 60$) νευρώνων δεν αυξάνει περαιτέρω την επίδοση των ταξινομητών. Ακόμα, υποθέτουμε ότι διαμερίζοντας υπερβολικά μια εικόνα σε υποτιμήματα, τα αντικείμενα της εικόνας σταματάνε να είναι αντιληπτά, και συνεπώς το πρόβλημα της αναγνώρισης αντικειμένων εκφυλίζεται. Ως εκ τούτου, συμπεράναμε πως η διαίρεση σε 4 ή 9 τμήματα/μέρη είναι αρκετή για να δώσει υψηλές αποδόσεις στα μοντέλα. Επιπρόσθετα, συνδυάζοντας τους ρυθμούς νευρικών αποκρίσεων των επιμέρους κομματιών μιας εικόνας με το \max μετασχηματισμό απέδιδε με συνέπεια καλύτερα από τον μετασχηματισμό \min . Αυτό σημαίνει ότι κρατάμε μόνο τους νευρώνες που αποκρίνονται καλύτερα στα οπτικά ερεθίσματα το οποίο έρχεται σε συμφωνία με τη λειτουργία των βιολογικών νευρώνων. Τέλος, τα αποτελέσματα υπέδειξαν ότι το adjustment των εικόνων, όπως περιγράφεται στο Κεφάλαιο 4.5, μπορεί να παραληφθεί δεδομένου ότι το μοντέλο αμφιβληστροειδούς αποδίδει εξίσου καλά και χωρίς αυτή τη διαδικασία. Αυτό έρχεται σε αντίθεση με τον τρόπο που καταγράφηκε η απόκριση των γαγγλιακών κυττάρων στις εργασίες [12], [28]. Παρόλα αυτά, βασισμένοι στα πειράματα του Παραρτήματος Α', μπορούμε να ισχυριστούμε ότι αν δώσουμε στο μοντέλο αμφιβληστροειδούς περίπου δέκα επαναλήψεις της ίδιας εικόνας, αυτό είναι αρκετό για να πάρουμε ικανοποιητικές αποκρίσεις από τα κύτταρα.

6.1.2 Σύνολα Δεδομένων

Ερευνήσαμε ακόμα το πώς η απόδοση του μοντέλου διαφοροποιείται σε σχέση με το πρόβλημα αναγνώρισης εικόνας και κατά συνέπεια σε σχέση με το σύνολο δεδομένων που χρησιμοποιείται. Αρχικά, χρησιμοποιήσαμε το CIFAR10 και το MNIST ως baseline σύνολα δεδομένων. Στη συνέχεια, δοκιμάσαμε και πιο σύνθετα σύνολα δεδομένων όπως το Imagenette και το Fashion MNIST. Επιλέξαμε το Imagenette ως σύνολο δεδομένων που περιέχει εικόνες υψηλότερης ανάλυσης από το CIFAR10, στοχεύοντας με αυτόν τον τρόπο να έχουμε υψηλής ανάλυσης εικόνες ακόμα και αφότου τις διαιρέσουμε σε υποτιμήματα. Επιπλέον, επιλέξαμε το Fashion MNIST ως ένα δυσκολότερο και πιο σύνθετο σύνολο δεδομένων συγκριτικά με το MNIST. Σύμφωνα με τα αποτελέσματα, φαίνεται ότι η ανάλυση της εικόνας δεν είναι τόσο σημαντικός παράγοντας για την απόδοση του μοντέλου. Το CIFAR10 αποτελείται από εικόνες μεγέθους (32,32) και δεδομένου ότι κάθε εικόνα διαιρείται σε 4 έως και 16 τμήματα και μετά κάθε τμήμα μετασχηματίζεται σε μέγεθος (50,50), παρατηρούμε ότι το CIFAR10 παρέχει εικόνες πολύ χαμηλής ανάλυσης. Ωστόσο, ενώ το Imagenette περιέχει εικόνες πολύ υψηλότερης ανάλυσης, δεν αποδίδει εξίσου κα-

λύτερα σε σχέση με το CIFAR10, και σε κάποιες προσομοιώσεις μάλιστα αποδίδει χειρότερα. Αν επίσης συγκρίνουμε και την απόδοση στα CIFAR10 και Imagenette με την ακρίβεια στα MNIST και Fashion MNIST, παρατηρούμε ότι η απόδοση στα τελευταία είναι διπλάσια από τα πρώτα (πάνω από 80%) ακόμα και αν τα MNIST και Fashion MNIST απαρτίζονται από εικόνες μεγέθους (28,28). Με βάση αυτή τη σημαντική διαφορά στην απόδοση μεταξύ των CIFAR10/Imagenette και MNIST/Fashion MNIST, υποθέτουμε ότι οι δομικές διαφορές τους είναι ένας κομβικός παράγοντας. Οι εικόνες τόσο στο MNIST όσο και στο Fashion MNIST έχουν τα αντικείμενα προς ταξινόμηση τοποθετημένα στο κέντρο ενώ τα υπόλοιπα τμήματα/μέρη της εικόνας απαρτίζονται από σκοτεινά pixels. Τα CIFAR10 και Imagenette, από την άλλη, απαρτίζονται από πραγματικές εικόνες οι οποίες αποτελούνται από ποικίλα χρώματα και σχήματα και τα αντικείμενα μπορεί να είναι τοποθετημένα σε όλο το πλαίσιο της εικόνας. Οπότε, αυτή η διαφοροποίηση θα μπορούσε να είναι πολύ σημαντική, δεδομένης της ιδιότητας του ματιού να αντιδρά μόνο σε χωροχρονικές μεταβολές μεταξύ διαφορετικών εικόνων. Αυτή η ιδιότητα ευνοεί τόσο το MNIST όσο και το Fashion MNIST, στα οποία το μόνο μέρος της εικόνας που μεταβάλλεται είναι το αντικείμενο προς ταξινόμηση.

6.1.3 Ταξινομητές

Ακόμα, εξετάσαμε ποιοι ταξινομητές αποδίδουν καλύτερα σε τέτοιου είδους προβλήματα. Μετά από αρχικούς πειραματισμούς, καταλήξαμε στην επιλογή των MLP_500_100 (μέγεθος αρχικού στρώματος = 500, μέγεθος κρυφού στρώματος = 100), MLP_n_n/2 (μέγεθος αρχικού στρώματος = n , μέγεθος κρυφού στρώματος = $n/2$, όπου n = αριθμός των χαρακτηριστικών features), SVM (kernel='rbf') και Random Forest (max_features=12). Μεταξύ αυτών, το Random Forest αποδείχθηκε κατ'επανάληψη να είναι ο πιο αποδοτικός ταξινομητής. Ακόμα και αν το SVM προσέγγισε ή και ξεπέρασε σε κάποιες προσομοιώσεις την ακρίβεια του Random Forest, δεν απέδωσε εξίσου καλά σε όλα τις προσομοιώσεις.

6.1.4 Μοντέλα Αμφιβληστροειδούς

Στο τελικό στάδιο της εργασίας, εφαρμόσαμε τη λειτουργική αξιολόγηση προκειμένου να συγκρίνουμε την απόδοση δύο διαφορετικών μοντέλων αμφιβληστροειδούς. Τα αποτελέσματα έδειξαν ότι τα μοντέλα αμφιβληστροειδούς με το χαμηλότερο μέσο τετραγωνικό σφάλμα απέδωσαν καλύτερο σε όλο το εύρος των λειτουργικών προσομοιώσεων. Συνεπώς φαίνεται ότι η λειτουργική αξιολόγηση είναι σύμφωνη με την καθιερωμένη τεχνική αξιολόγησης. Ωστόσο, αξιολογώντας λειτουργικά το μοντέλο έχουμε το πλεονέκτημα ότι λαμβάνουμε μια άμεση εκτίμηση του πόσο καλά ένα εμφύτευμα θα μπορούσε να αποδώσει στην πράξη στα εκάστοτε λειτουργικά προβλήματα.

6.2 Περιορισμοί της Λειτουργικής Αξιολόγησης

Κατά την εκπόνηση αυτής της εργασίας, ήρθαμε αντιμέτωποι με αρκετούς περιορισμούς. Ο πρώτος περιορισμός έχει να κάνει με την απόδοση των εμφυτευμάτων αμφιβληστροειδούς σε λειτουργικά προβλήματα, όπως η αναγνώριση αντικειμένου. Συγκεκριμένα, τα διαθέσιμα ηλεκτρόδια στα εμφυτεύματα παρέχουν έναν περιορισμένο αριθμό νευρικών αποκρίσεων και συνεπώς τα μοντέλα αμφιβληστροειδούς εκπαιδεύονται στο να προβλέπουν αποκρίσεις μόνο για μερικούς νευρώνες. Παρόλα αυτά, και τα λειτουργικά προβλήματα της πραγματικής ζωής απαιτούν έγχρωμες εικόνες υψηλής ανάλυσης, αλλά και τα μοντέλα μηχανικής μάθησης απαιτούν δεδομένα με έναν επαρκή αριθμό χαρακτηριστικών ώστε να αποδώσουν καλά. Αυτό δημιουργεί πρόβλημα όχι μόνο όταν προσπαθούμε να χρησιμοποιήσουμε μοντέλα αμφιβληστροειδούς στην επίλυση προβλημάτων ταξινόμησης αντικειμένων αλλά και κατά τον σχεδιασμό προσθετικών μελών αμφιβληστροειδούς που θα μπορούσαν ιδανικά να ικανοποιήσουν τις λειτουργικές ανάγκες της καθημερινότητας. Άλλος ένας περιορισμός αφορά την επιλογή της μεθόδου μοντελοποίησης του οπτικού συστήματος. Από βιολογική σκοπιά, το οπτικό μονοπάτι δεν εμπεριέχει μόνο την νευρική επεξεργασία στον αμφιβληστροειδή, αλλά κυρίως εμπεριέχει μια νευρική επεξεργασία στα οπτικά κέντρα του εγκεφάλου, τα οποία είναι υπεύθυνα για υψηλότερης τάξης λειτουργίες, όπως η αναγνώριση αντικειμένων. Σε αυτή την εργασία, επέλεξαμε να επεξεργαστούμε τις νευρικές αποκρίσεις με παραδοσιακούς ταξινομητές Μηχανικής Μάθησης. Μια εναλλακτική προσέγγιση θα μπορούσε να είναι η χρήση αρχιτεκτονικών βαθιάς μάθησης προκειμένου να μοντελοποιηθούν τα οπτικά κέντρα του εγκεφάλου. Αυτή η προσέγγιση έχει χρησιμοποιηθεί σε έρευνες με υψηλό αντίκτυπο στην βιολογική όραση, με σκοπό τη μελέτη των μηχανισμών που λαμβάνουν χώρα στον αμφιβληστροειδή [44].

Κεφάλαιο **7**

Επίλογος

7.1 Σύνοψη

Σε αυτήν τη διπλωματική εργασία, διερευνήσαμε το πρόβλημα της λειτουργικής αξιολόγησης των μοντέλων αμφιβληστροειδούς με τη χρήση τεχνικών μηχανικής μάθησης. Δείξαμε πώς να συγκρίνουμε λειτουργικά την απόδοση διαφορετικών μοντέλων αμφιβληστροειδούς σε προβλήματα αναγνώρισης αντικειμένου και αναγνώρισης ψηφίων και δημιουργήσαμε τις ανάλογες προσομοιώσεις για αυτόν τον σκοπό. Συγκεκριμένα, χρησιμοποιήσαμε τα σύνολα δεδομένων CIFAR10, MNIST, Fashion MNIST και Imagenette και εκπαιδεύσαμε κοινούς ταξινομητές, όπως MLP, SVM και Random Forest. Λαμβάνοντας υπόψη τον περιορισμένο αριθμό διαθέσιμων νευρώνων που παρέχονται από το μοντέλο του αμφιβληστροειδούς, διαχωρίσαμε τις εικόνες σε τμήματα/μέρη και στη συνέχεια τροφοδοτήσαμε κάθε τμήμα ξεχωριστά στο μοντέλο του αμφιβληστροειδούς, ώστε να αυξήσουμε τεχνητά τον αριθμό των νευρικών αποκρίσεων. Με βάση αυτήν την ιδέα, δημιουργήσαμε και βελτιστοποιήσαμε έναν αλγόριθμο προεπεξεργασίας των εικόνων. Διαπιστώσαμε ότι ο ταξινομητής Random Forest επιτυγχάνει την υψηλότερη επίδοση και ότι τα μοντέλα αποδίδουν καλύτερα με σύνολα δεδομένων όπως το MNIST και το Fashion MNIST, όπου επιτύχαμε ακρίβεια έως και 90%. Επιπλέον, διαπιστώσαμε ότι η χρήση περισσότερων νευρώνων βελτιώνει τη συνολική απόδοση του μοντέλου, υπογραμμίζοντας έτσι την ανάγκη σχεδιασμού προσθετικών αμφιβληστροειδών με μεγαλύτερο αριθμό διαθέσιμων ηλεκτροδίων. Τέλος, εφαρμόσαμε τη λειτουργική αξιολόγηση προκειμένου να συγκρίνουμε την απόδοση δύο διαφορετικών μοντέλων αμφιβληστροειδούς. Τα αποτελέσματα έδειξαν ότι υπάρχει συμφωνία μεταξύ της καθιερωμένης και της λειτουργικής αξιολόγησης, καθώς τα δύο μοντέλα είχαν παρόμοια απόδοση και στις δύο τεχνικές.

7.2 Μελλοντικές Προεκτάσεις

Σχετικά με την περαιτέρω ανάπτυξη ενός πλαισίου λειτουργικής αξιολόγησης μοντέλων αμφιβληστροειδούς, προτείνονται ενδιαφέρουσες μελλοντικές κατευθύνσεις, που θα μπορούσαν να αυξήσουν την αξιοπιστία της. Συγκεκριμένα προτείνεται:

- Να εφαρμοστεί η τεχνική της λειτουργικής αξιολόγησης σε περισσότερα μοντέλα αμφιβληστροειδούς με διαφορετική αρχιτεκτονική.
- Να μελετηθεί η λειτουργική αξιολόγηση σε διαφορετικά προβλήματα, όπως η αναγνώριση προσώπου ή ανίχνευση κίνησης.
- Να γίνει χρήση μοντέλων βαθιάς μάθησης που να περιγράφουν τα οπτικά κέντρα του εγκεφάλου, αντί των παραδοσιακών ταξινομητών. Με αυτόν τον τρόπο, η ανίχνευση αντικειμένων και άλλες υψηλότερες οπτικές λειτουργίες ενδέχεται να έχουν καλύτερη απόδοση.
- Να γίνει χρήση μοντέλων βαθιάς μάθησης, τα οποία θα εξάγουν τα σημεία ενδιαφέροντος σε κάθε εικόνα και θα κατευθύνουν προς τα εκεί την οπτική προσοχή του αμφιβληστροειδή.
- Να μελετηθεί εάν υπάρχει σύγκλιση (ή απόκλιση) στη λειτουργική αξιολόγηση ενός μοντέλου αμφιβληστροειδούς και στη χρήση αντιληπτικών εικόνων (perceptual images). Οι αντιληπτικές εικόνες περιγράφουν μια μέθοδο οπτικοποίησης της απόδοσης ενός μοντέλου, μεταφράζοντας (decoding) την νευρική δραστηριότητα σε εικόνες που μπορεί να αντιληφθεί ο άνθρωπος (perceived, "mental" images) [45].

Παραρτήματα

Παράρτημα **A'**

Επαναλήψεις εικόνων στο χρονικό σύνολο δεδομένων

Κατά τη διάρκεια της προεπεξεργασίας δεδομένων χρειάζεται να αποφασίσουμε τον αριθμό των επαναλήψεων στο χρονικό σύνολο δεδομένων (temporal dataset). Δεδομένου ότι κάθε γραμμή του temporal dataset έχει τόσα καρέ, όσες και η μεταβλητή "*temporal_interval*", χρειάζεται να αποφασίσουμε πόσα από αυτά τα καρέ θα αποτελούν πραγματικές εικόνες και πόσα θα είναι μαύρα καρέ (Σχήμα [A.1](#)).

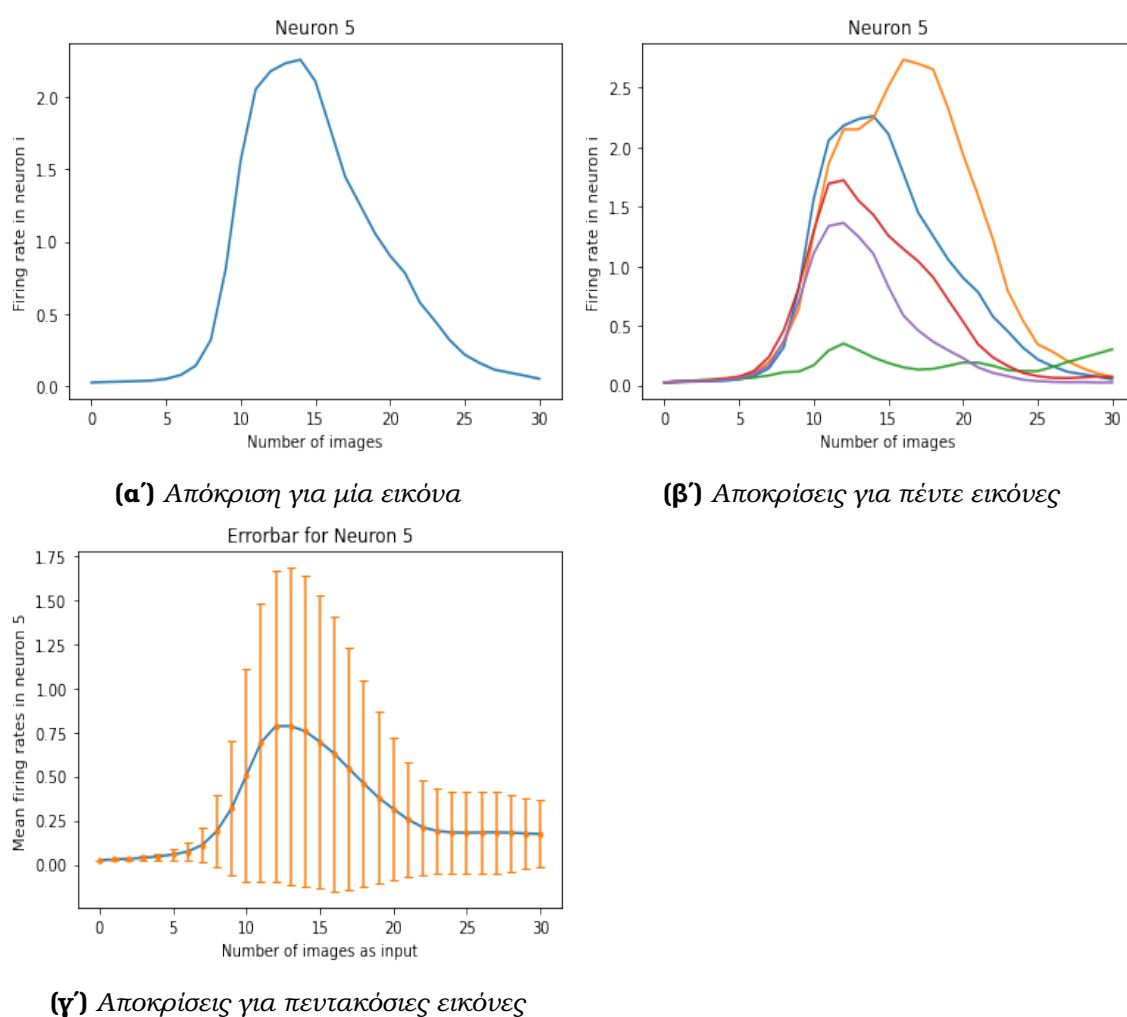


Σχήμα A.1. Παράδειγμα μιας γραμμής στο temporal dataset.

Για να πάρουμε αυτήν την απόφαση, διεξάγαμε ένα πείραμα όπου συγκρίναμε τις νευρικές αποκρίσεις του κάθε νευρώνα συναρτήσει του αριθμού των πραγματικών εικόνων που χρησιμοποιήθηκαν στο temporal dataset. Συγκεκριμένα, σχεδιάσαμε διαγράμματα με τη μέση τιμή και την τυπική απόκλιση των νευρικών αποκρίσεων για κάθε διαφορετικό αριθμό εικόνων που χρησιμοποιήθηκαν στο temporal dataset. Στην περίπτωση μας, το μέγιστο μήκος κάθε σειράς (δηλαδή η μεταβλητή "*temporal_interval*") είναι ίσο με 30. Στο Σχήμα [A.2](#) φαίνεται ένα παράδειγμα με τους τρεις διαφορετικούς τύπους διαγραμμάτων που σχεδιάσαμε, ενώ στον Πίνακα [A.1](#) φαίνονται τα διαγράμματα μέσης τιμής/τυπικής απόκλισης για 8 διαφορετικούς νευρώνες (από τους 60).

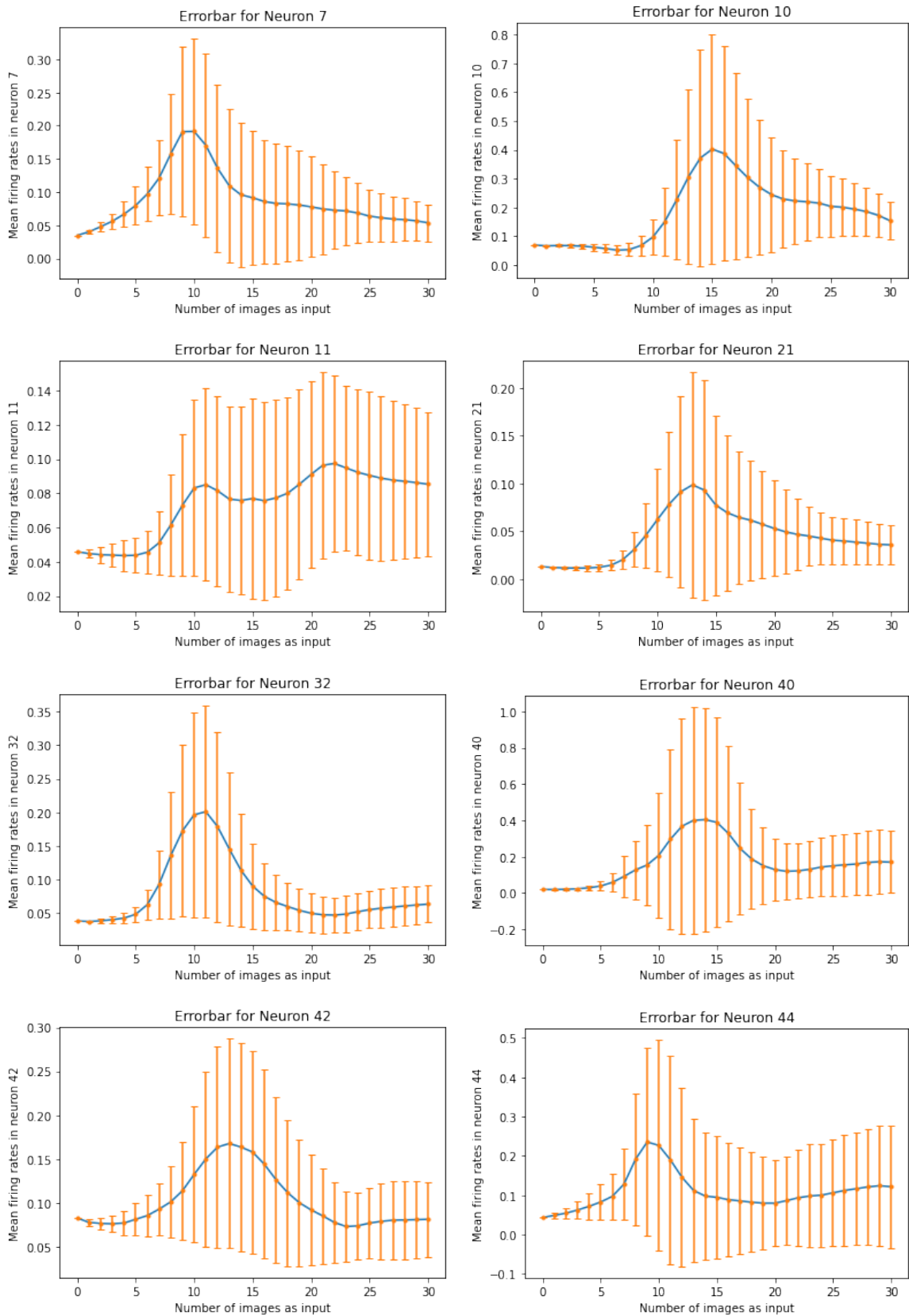
Για να εντοπίσουμε τις διεγέρσεις των νευρώνων, μπορούμε να ελέγξουμε είτε στην αρχή των διαγραμμάτων (1 – 5 πρώτα καρέ), όπου ο αμφιβληστροειδής βλέπει για πρώτη φορά την εικόνα, είτε στα 20 – 30 τελευταία καρέ, όπου ο αμφιβληστροειδής σταματάει να βλέπει την εικόνα. Αυτή η παρατήρηση βασίζεται στην ιδιότητα του αμφιβληστροειδούς να διεγείρεται μόνο από χωροχρονικές μεταβολές μεταξύ συνεχόμενων καρέ εικόνων. Όπως φαίνεται στον Πίνακα [A.1](#), οι περιοχές αυτές (στην πολύ αρχή ή πολύ τέλος) η τυπική απόκλιση είναι πολύ μικρή, υποδεικνύοντας

ότι οι νευρώνες έχουν παρόμοια συμπεριφορά ανεξάρτητα από την εικόνα εισόδου. Ωστόσο, αν θέλουμε να πετύχουμε καλύτερη ταξινόμηση, ενδιαφερόμαστε για περιοχές που έχουν την υψηλότερη απόκλιση. Αυτό γιατί περιμένουμε ότι οι νευρώνες θα έχουν διαφορετική συμπεριφορά για τις διάφορες εικόνες κι άρα, θα μπορούμε να τις ταξινομήσουμε αν ελέγξουμε τη συμπεριφορά των νευρώνων. Βασισμένοι στα παρακάτω διαγράμματα, παρατηρούμε ότι οι περιοχές με τη μεγαλύτερη απόκλιση κυμαίνονται μεταξύ 10 και 15 εικόνων. Επομένως, συμπεραίνουμε ότι αν θέλουμε το μοντέλο αμφιβληστροειδούς να αποδώσει καλύτερα σε προβλήματα αναγνώρισης εικόνων, οι σειρές στο temporal dataset θα πρέπει να περιέχουν έναν αριθμό εικόνων μεταξύ 10 και 15.



Σχήμα Α'.2. Διαγράμματα που αναπαριστούν τις νευρικές αποκρίσεις του νευρώνα_5 συναρτήσει του διαφορετικού αριθμού εικόνων σε κάθε σειρά. Σε κάθε διάγραμμα δίνουμε διαφορετικό αριθμό εικόνων ως είσοδο στο μοντέλο και αντίστοιχα παίρνουμε διαφορετικό αριθμό εξόδων (δηλαδή νευρικών αποκρίσεων). Στο Σχήμα Α2γ, σχεδιάζουμε το διάγραμμα μέσης τιμής/τυπικής απόκλισης, καθώς δίνουμε πεντακόσιες εικόνες ως είσοδο.

Πίνακας Α΄.1. Διαγράμματα μέσης τιμής και τυπικής απόκλισης για οχτώ διαφορετικούς νευρώνες.



Βιβλιογραφία

- [1] J. E. Bixler, “FIGURE 3, [Cross section of the eye...].,” May 2019. Publisher: American Diabetes Association.
- [2] Wikipedia, “Macula of retina — Wikipedia, the free encyclopedia.” <http://en.wikipedia.org/w/index.php?title=Macula%20of%20retina&oldid=996160948>, 2021. [Online; accessed 09-January-2021].
- [3] D. Heeger, “Perception lecture notes: Retinal ganglion cells.” <http://www.cns.nyu.edu/~david/courses/perception/lecturenotes/ganglion/ganglion.html>.
- [4] J. Martins and L. Sousa, *Bioelectronic Vision: Retina Models, Evaluation Metrics, and System Design*. Series on bioengineering and biomedical engineering, World Scientific, 2009.
- [5] P. Dayan and L. F. Abbott, *Theoretical Neuroscience: Computational and Mathematical Modeling of Neural Systems*. The MIT Press, 2005.
- [6] S. A. Baccus and M. Meister, “Fast and Slow Contrast Adaptation in Retinal Circuitry,” *Neuron*, vol. 36, pp. 909–919, Dec. 2002.
- [7] N. Maheswaranathan *et al.*, “The dynamic neural code of the retina for natural scenes,” *bioRxiv*, 2019. Publisher: Cold Spring Harbor Laboratory _eprint: <https://www.biorxiv.org/content/early/2019/12/17/340943.full.pdf>.
- [8] J. D. Weiland and M. S. Humayun, “Retinal Prosthesis,” *IEEE Transactions on Biomedical Engineering*, vol. 61, pp. 1412–1424, May 2014.
- [9] H.-R. Zhang, F. Min, and X. He, “Aggregated recommendation through random forests,” *TheScientificWorldJournal*, vol. 2014, p. 649596, 08 2014.
- [10] S. Haykin, *Neural Networks: A Comprehensive Foundation (3rd Edition)*. USA: Prentice-Hall, Inc., 2007.
- [11] N. Maheswaranathan *et al.*, “Deep learning models reveal internal structure and diverse computations in the retina under natural scenes,” *bioRxiv*, 2018. Publisher: Cold Spring Harbor Laboratory _eprint: <https://www.biorxiv.org/content/early/2018/06/08/340943.full.pdf>.

- [12] A. Lozano, C. Soto-Sánchez, J. Garrigós, J. J. Martínez, J. M. Ferrández, and E. Fernández, “A 3D Convolutional Neural Network to Model Retinal Ganglion Cell’s Responses to Light Patterns in Mice,” *International Journal of Neural Systems*, vol. 28, p. 1850043, Dec. 2018.
- [13] E. Fernandez, “Development of visual neuroprostheses: trends and challenges,” *Bioelectronic Medicine*, vol. 4, Dec 2018.
- [14] N. Melanitis and K. S. Nikita, “Biologically-inspired image processing in computational retina models,” *Computers in biology and medicine*, vol. 113, p. 103399, 2019.
- [15] A. Alevizaki, N. Melanitis, and K. Nikita, “Predicting eye fixations using computer vision techniques,” in *2019 IEEE 19th International Conference on Bioinformatics and Bioengineering (BIBE)*, pp. 309–315, IEEE, 2019.
- [16] B. P. Lepri, “Is acuity enough? other considerations in clinical investigations of visual prostheses,” *Journal of Neural Engineering*, vol. 6, p. 035003, Jun 2009.
- [17] A. M. Demchinsky *et al.*, “The first deaf-blind patient in russia with argus II retinal prosthesis system: what he sees and why,” *Journal of Neural Engineering*, vol. 16, p. 025002, feb 2019.
- [18] A. N. Angelopoulos, H. Ameri, D. Mitra, and M. Humayun, “Enhanced depth navigation through augmented reality depth mapping in patients with low vision,” *Scientific Reports*, vol. 9, p. 11230, Aug 2019.
- [19] A. Krizhevsky, V. Nair, and G. Hinton, “Cifar-10 (canadian institute for advanced research),”
- [20] H. Xiao, K. Rasul, and R. Vollgraf, “Fashion-mnist: a novel image dataset for benchmarking machine learning algorithms,” 2017.
- [21] J. Howard, “Imagenette.” <https://github.com/fastai/imagenette/>.
- [22] Y. LeCun and C. Cortes, “MNIST handwritten digit database,” 2010.
- [23] S. W. Kuffler, “Discharge patterns and functional organization of mammalian retina,” *Journal of Neurophysiology*, vol. 16, no. 1, pp. 37–68, 1953. PMID: 13035466.
- [24] P. H. Schiller and E. J. Tehovnik, “Visual prosthesis,” *Perception*, vol. 37, no. 10, pp. 1529–1559, 2008. PMID: 19065857.

- [25] E. Chichilnisky, "A simple white noise analysis of neuronal light responses," *Network (Bristol, England)*, vol. 12, pp. 199–213, 06 2001.
- [26] J. W. Pillow *et al.*, "Spatio-temporal correlations and visual signalling in a complete neuronal population," *Nature*, vol. 454, pp. 995–999, Aug. 2008.
- [27] A. N. Burkitt, "A Review of the Integrate-and-fire Neuron Model: I. Homogeneous Synaptic Input," *Biological Cybernetics*, vol. 95, pp. 1–19, July 2006.
- [28] L. T. McIntosh, N. Maheswaranathan, A. Nayebi, S. Ganguli, and S. A. Baccus, "Deep learning models of the retinal response to natural scenes," 2017.
- [29] K. P. Murphy, *Machine Learning: A Probabilistic Perspective*. The MIT Press, 2012.
- [30] T. M. Mitchell, *Machine Learning*. USA: McGraw-Hill, Inc., 1 ed., 1997.
- [31] I. Goodfellow, Y. Bengio, and A. Courville, *Deep Learning*. The MIT Press, 2016.
- [32] A. Zhang, Z. C. Lipton, M. Li, and A. J. Smola, *Dive into Deep Learning*. 2020. <https://d2l.ai>.
- [33] "Deep learning models of the retinal response to natural scenes - github repository." <https://github.com/baccuslab/deep-retina>.
- [34] M. Abadi *et al.*, "TensorFlow: Large-Scale Machine Learning on Heterogeneous Distributed Systems," *arXiv:1603.04467 [cs]*, Mar. 2016. arXiv: 1603.04467.
- [35] F. Chollet *et al.*, "Keras." <https://github.com/fchollet/keras>, 2015.
- [36] F. Pedregosa *et al.*, "Scikit-learn: Machine learning in Python," *Journal of Machine Learning Research*, vol. 12, pp. 2825–2830, 2011.
- [37] A. B. Jung *et al.*, "imgaug." <https://github.com/aleju/imgaug>, 2020. Online; accessed 01-Feb-2020.
- [38] "Tensorflow keras datasets, cifar10." https://www.tensorflow.org/api_docs/python/tf/keras/datasets/cifar10.
- [39] "Tensorflow keras datasets, mnist." https://www.tensorflow.org/api_docs/python/tf/keras/datasets/mnist.
- [40] "Tensorflow keras datasets, fashion_mnist." https://www.tensorflow.org/api_docs/python/tf/keras/datasets/fashion_mnist.

- [41] “Tensorflow datasets, imagenette.” <https://www.tensorflow.org/datasets/catalog/imagenette>.
- [42] G. Nakopoulos, “Investigation of retinal response at salient image regions (in preparation),” 2021. Diploma Thesis, BioSim Lab, National Technical University of Athens.
- [43] C. Shorten and T. Khoshgoftaar, “A survey on image data augmentation for deep learning,” *Journal of Big Data*, vol. 6, pp. 1–48, 2019.
- [44] J. Lindsey, S. A. Ocko, S. Ganguli, and S. Deny, “A unified theory of early visual representations from retina to cortex through anatomically constrained deep cnns,” *bioRxiv*, 2019.
- [45] N. Parthasarathy *et al.*, “Neural networks for efficient bayesian decoding of natural images from retinal neurons,” *bioRxiv*, 2018.