

Using Generative Adversarial Networks to Derive Phosphene Patterns in Simulated Prosthetic Vision

Student: Jamin Wu

Student ID: 27025861

Supervisor: Dr Yan Tat Wong

Co-Supervisor: Dr Nicholas Price

Department of Physiology

Department of Electrical & Computer Systems Engineering

Faculty of Medicine, Nursing and Health Sciences, Monash University

Minor thesis submitted in partial fulfillment of the requirements for the
degree **Bachelor of Medical Science (Honours)** at Monash University.

Word Count: 14,934 words

Abstract

Background

Cortical visual prostheses are emerging devices which are implanted on the brain to restore visual sensations to blind people. These prostheses are currently only expected to produce a limited number of small spots of light in the visual fields known as phosphenes, often likened to "stars in a night's sky". Phosphenes, however, tend to be irregularly distributed and display different visual properties between people. It is therefore not clear how phosphenes can be assembled and tailored for people's different visual percepts to convey useful information. Other image processing domains have experienced success with image generation tasks using generative adversarial networks, a machine learning technique often used to assemble pixels into novel imagery. However, generative adversarial networks have not previously been applied to prosthetic vision to assemble phosphenes for arbitrary visual properties.

Aims

We aimed to determine whether generative adversarial networks could be used to generate phosphene patterns for different simulations of prosthetic vision, and whether these generated phosphene patterns were useful to people when applied to a digit representation task.

Methods

We developed a prototype software implementation of a generative adversarial network training architecture and a dynamic phosphene simulator using the Python programming language. We used the phosphene simulator to generate phosphene grids with different visual properties, and ran these grids through the generative adversarial network training architecture to produce phosphene patterns representing single digits, which were qualitatively analysed. We then recruited 11 participants for a psychophysics experiment comparing their ability to learn and recognise digits

using the generated phosphene patterns versus patterns produced by a simple mask-based control.

Results

The prototype implementation we developed was successfully able to produce phosphene patterns for phosphene grids with varying resolution, spatial distribution and sizes of phosphenes. Qualitatively, the generated phosphene patterns were recognisable as digits at higher phosphene grid resolutions, but not at lower resolutions. Despite this, the experimental results demonstrated that people achieved better digit recognition accuracy overall using the generated phosphene patterns versus the mask-based control. While participants did not initially achieve better digit recognition accuracy, participants' accuracy increased faster when learning the generated phosphene patterns versus control.

Conclusion

We demonstrated that a generative adversarial network training architecture can be applied to generate phosphene patterns tailored for different phosphene grids. However, our early implementation was not robust for low-resolution grids. Experimentally, it appears that although low-resolution generated phosphene patterns were not immediately recognisable, they may still be useful to increase the discriminability between digits when learned. As the first attempt at using generative adversarial networks to address the perceptual limitations of phosphenes, our work highlights potential targets for further improving how phosphenes are assembled for different phosphene grid simulations. Further work should aim at improving robustness of the prototype architecture for low-resolution phosphene grids and quantifying the intrinsic ability of low-resolution grids to represent visual information.

Acknowledgements

My experience this year has been humbling, and I am extremely grateful to the people who have made this year into the incredible learning experience it has become.

Firstly, I would like to thank my supervisor Dr Yan Wong for his guidance, encouragement and compassion throughout the year. I have been very fortunate to have been able to discuss and learn from his expertise in both physiology and engineering, and I am immensely grateful for his kindness and understanding whenever things went pear-shaped.

I would also like to thank my co-supervisor Dr Nicholas Price for his advice, support and geniality. I have greatly valued my discussions with him and his friendliness and enthusiasm during our conversations.

I would like to thank Julian Szlawski, who helped organise the very beginning of this project and has helped me feel welcome in the Neurobionics Lab ever since. I would like to thank Dr Zongyuan Ge, who very generously gave his time to offer his advice on useful machine learning approaches for this project. I would like to thank Mojtaba Kermani, Timothy Allison-Walker, Dr Elizabeth Zavitz and Dr Maureen Hagan who supported me in my department oral presentations and Dr Adam Morris who tried to be there too, and with all of whom I have shared interesting conversations and received valuable feedback. I would like to thank Dr Anand Mohan, who patiently inducted me into the lab and made me feel welcome.

I would like to thank all the participants of the psychophysics experiment in this project who generously volunteered their time to participate.

I would like to thank all the members of Journal Club, from whom I learned a great deal about the critical appraisal of journal articles and whose members were brilliant at explaining concepts I previously did not understand. I would like to thank Lucas, Mike and Andrew for having been with me in the Neurobionics Lab, and Victor for having been with me in the Physiology Honours Room. We shared a common fate as students and I am grateful we could journey through our research together.

Declaration

I declare that the contents of this thesis are my own work unless explicit reference indicating otherwise has been made in the text. I declare that this work has not been submitted for the award of any other degrees, diplomas or publication. The background of this thesis is predominantly comprised of a literature review previously submitted for assessment as part of this degree, which has been updated to reflect examiner feedback and recent evidence.

The original premise for this project was devised by Dr Yan Wong. I, under the guidance of Dr Yan Wong and Dr Nicholas Price, developed the aims, hypothesis and methods for the work described in this thesis and wrote all the code required during its conduct and analysis. I performed the data collection and analysis and the interpretation of results of this work were discussed and refined with Dr Yan Wong and Dr Nicholas Price. I wrote and edited this thesis, and its contents were discussed and reviewed with Dr Yan Wong.

Contents

I Background	12
1 Introduction	13
2 Cortical Visual Prostheses	15
3 Qualities of Prosthetic Vision	20
4 Making Prosthetic Vision Useful	30
5 Aims and Hypothesis	41
II Methods	43
6 Generative Adversarial Network Architecture	44
7 Psychophysics Experiment	62
III Results	67
8 Training Results	68
9 Experimental Results	73
IV Discussion	80
10 Aims	81

<i>CONTENTS</i>	7
11 Implications	84
12 Limitations	87
13 Future Directions	90
14 Conclusion	92
V Appendices	93
A Selected Package Dependencies	94
B Selected Code	96
C Supplementary Qualitative Results	102
VI References	104

List of Abbreviations

CVP cortical visual prosthesis

CNN convolutional neural networks

GAN generative adversarial network

GPU graphic processing unit

HPC high performance computing

MASSIVE multi-modal Australian ScienceS Imaging and Visualization Environment

MNIST dataset modified National Institute of Standards and Technology dataset

ReLU rectified linear unit

List of Tables

2.1	Current progress of cortical visual prostheses	17
3.1	Participant demographics of studies looking at cortical phosphenes evoked <i>in vivo</i> in humans	22
9.1	Characteristics of trials for each participant. Control indicates the direct masking processor, and Test indicates the trained phosphene patterns using the GAN architecture.	74
9.2	Predictors of a correct digit classification modelled as a logistic regression. P-values below 0.05, 0.01 and 0.001 are marked with (*), (**) and (***) respectively.	77
A.1	Selected package dependencies for the project.	95

List of Figures

2.1	Modelled render of the Gennaris bionic vision headgear	16
2.2	Basic flowchart of the process of prosthetic vision compared to normal vision	19
3.1	Spatial distribution of phosphenes mapped to the visual fields	24
4.1	Examples of different simulated renders of phosphenes	31
4.2	Example of using an edge-detection algorithm to render a clean simulated phosphene image	33
4.3	Example of using a saliency-based algorithm to highlight a region of interest	34
4.4	Example of using a transformative approach to image processing to re-encode information	35
4.5	Example of conveying information temporally through phosphenes .	36
6.1	Overview of GAN training architecture implementation	45
6.2	A subset of MNIST real handwritten digit samples before and after preprocessing	49
6.3	Schematic of the GAN Generator neural network	51
6.4	Illustration of the phosphene simulation process	53
6.5	Phosphene maps used in the prosthetic vision simulation	54
6.6	Schematic of <i>Modifier</i> architecture.	56
6.7	Examples of passing phosphene images under the <i>Modifier</i> network	58
6.8	Schematic of <i>Discriminator</i> architecture	59

LIST OF FIGURES

11

7.1 Flowchart of psychophysics experiment.	64
8.1 Phosphene patterns generated for the digit 4 for a regular cartesian grid of 576 phosphenes. The digit form approaches a recognisable digit 4 until approximately epoch 20, before the subsequently drifting away.	69
8.2 Phosphene patterns generated for different phosphene resolutions of a regular Cartesian grid for the digit 0. The first column shows the phosphene locations and relative sizes; the second, third and fourth columns show the simulated patterns at the 2nd, 4th and 8th epoch respectively.	70
8.3 Phosphene patterns generated for the digit 6 for 4 different grid arrangements with 576 phosphenes at 10, 20 and 30 epochs with the base phosphene map on the left for comparison.	71
8.4 Phosphene patterns generated for each digit 0-9 for a regular cartesian grid of 576 phosphenes at the 16th epoch.	72
9.1 Mean digit recognition accuracy by condition, pooled participant data	75
9.2 Mean digit recognition accuracy by participant and condition. . . .	76
9.3 Predictors of a correct response in a logistic regression model. Error bars indicate 95% confidence intervals.	78
9.4 Response time per cue by participant. Error bars show standard deviation.	78
9.5 Rolling average of accuracy over pooled trial blocks by condition and participant.	79
C.1 Supplemental qualitative results.	103

Part I

Background

Chapter 1

Introduction

Around 36 million people worldwide were blind in 2015. [1, 2] Many patients are left permanently blind when there is irreversible damage to the visual system such as from glaucoma or trauma. [3, 4] However these patients often still have portions of the visual system intact, which could be leveraged to restore vision artificially.

Cortical visual prostheses are devices which aim to achieve exactly that - devices implanted on the brain which stimulate neurons to directly inject visual sensations into awareness. [5–7]

Studies in the late 20th century showed that electrically stimulating the brain could indeed produce spot-like sensations of light, known as **phosphenes**. [8–11] But these studies also revealed that we can exert only very limited control over what phosphenes look like, where they are located, and how they interact. [9, 12, 13] Of these studies, those conducted in small groups also demonstrated wide variation in the appearance of phosphenes between people. [9, 10] Therefore, whilst we now know we can produce these spots of light by stimulating the brain, it is not clear how to assemble these unyielding percepts into useful vision for all implantees. [14, 15]

To address this, new proposals of cortical visual prostheses are exploring the use of sophisticated computer processing to assemble phosphenes *strategically* so implantees can interpret them in useful ways. [7, 16] These methods generally focus on addressing two significant difficulties of prosthetic vision: its low resolution, and its lack of color and brightness levels, both of which are easy to simulate on graphical displays and test. [17–19] However it is unclear how well these

image processing algorithms could cope with the much wider range of phosphene experiences reported by perceptual studies in humans. Thus, the primary issue facing these algorithms is how they can be made flexible to accommodate the large, uncontrollable variation in what phosphenes look like.

Recently, there have been remarkable advances in the ability of computers to derive means of processing data rather than having algorithms being programmed by hand. In this paradigm, computers iteratively "learn" and integrate patterns between data inputs and outputs through a process known as **machine learning**. [20] By using machine learning methods in training architectures called **generative adversarial networks** (GANs), computers have been successful at generating new image samples from a distribution [?], replicating image styles. [21] Machine learning principles could be similarly applied to prosthetic vision, using inferred patterns to derive image processing algorithms from performance rather than building algorithms around conceptions of what phosphenes look like. This could improve the flexibility of cortical implants for different perceptual experiences and improve the usefulness of cortical visual prostheses for future implantees.

Chapter 2

Cortical Visual Prostheses

Overview of Cortical Visual Prostheses

A **prosthesis** is an artificial, implanted device which aims to restore a lost function to the human body. [22] A **visual prosthesis** is such a device which aims to restore vision to people with visual impairment. [23, 24] A **cortical visual prosthesis** specifically refers to a vision restoration device implanted on the cortical surface of the brain, as opposed to other portions of the visual system such as the eye, optic nerve or thalamus. [6] As a cortical prosthesis is implanted directly on the brain, implantees only require a functional cortical portion of the visual system; the device bypasses any damage to the eyes or nerves leading up to the brain. While visual prostheses at other locations in the visual pathway are also capable of producing visual sensations [25–28], their use-case, implementation and evoked vision differ from those produced by cortical prostheses and are outside the scope of this review.

Research into stimulating the brain to produce vision was first pioneered by Brindley & Lewin [8] and later Dobelle & Mladejovsky [9] in the late 20th century. Using rudimentary hardware, these early experiments showed that a temporary implant composed of a array of electrodes could stimulate the brain in an awake patient and make them see artificial sensations of light. [8, 9] These artificial sensations of light are known as **phosphenes**. Phosphenes are highly variable, but most often appear as dots of light likened to "a star in the sky". [9] These early successes in evoking phosphenes were instrumental in demonstrating the feasibility

of cortical prostheses [13], which have adopted phosphenes as the fundamental building blocks of prosthetic vision.

In the past 50 years since, a small number of research groups have proposed modern cortical prostheses based on the principles of these early results. These include the Gennaris bionic vision system [29, 30], the Intracortical Visual Prosthesis (ICVP) Project [31], CORTIVIS [32] and the Orion Visual Cortical Prosthesis. [33] To illustrate what modern conceptions of a visual cortical prosthesis may look like, Figure 2.1 shows a simulated render of the headgear for the Gennaris bionic vision system.



Figure 2.1: A modelled render of the Gennaris bionic vision headgear. Figure courtesy of Monash Vision Group.

These devices, while still early in development, may eventually be an option for restoring a crude form of vision to patients who would otherwise be left permanently blind. The expectation is that these devices could provide gross light perception which might allow the recognition of basic forms and movement. [29, 30] Current technology cannot reproduce anything close to the trichromatic, approximately 15 million pixel resolution of the human eye [34], and as such, these devices are not yet a full replacement for vision. The development of specific cortical prostheses

has been reviewed previously [35]; we briefly summarise the current progress of these devices in Table 2.1.

Table 2.1: Current progress of cortical visual prostheses. The number of electrodes places a hard upper bound on the resolution (and visual acuity) these devices can provide, so are noted here.

Device	Electrodes	Progress	References
Orion (Second Sight)	60 subdural surface electrodes	FDA-approved clinical trial ongoing 2018-2023, six patients implanted (public scientific results yet to be released).	[33, 35, 36]
Gennaris (Monash Vision Group)	Up to 473 penetrating micro-electrodes	Ethics approved to begin clinical trials, recruiting.	[29, 30, 37]
ICVP Project (Illinois Institute of Technology)	16 penetrating microelectrodes	Preclinical phase	[31]
CORTIVIS (Universidad Miguel Hernández)	100 penetrating microelectrodes	Preclinical phase.	[32]

Mechanism of Cortical Visual Prostheses

While the specific hardware of each device differs, the fundamental mechanism of these devices is similar. Patients must first undergo an operation to surgically implant an electrode array in the primary visual cortex at the back of the brain. [6] Historically, these were subdural surface electrodes sitting atop (but not penetrating) the brain. [8, 9] However, modern prostheses tend to opt for penetrating micro-electrodes [30–32] which are finer and can operate successfully at lower electrical current. [10, 13]

After implantation, the physical components of the system are in place. The system will need to be calibrated and tested before use to determine electrical stimulation thresholds and the spatial correspondence between electrodes and phosphenes in the visual fields. [29, 32]

When in use, an external camera (e.g. on glasses worn by the user) first

captures an image. [6, 30] This image is transmitted to a portable processor, and is converted into electrical parameters for each electrode in the implanted array. Each electrode in the implanted array then delivers pulses of electrical charge into the brain based on its parameters, which electrically stimulates nearby neurons in the cortical tissue. Stimulating neurons in the visual cortex produces patterns of phosphenes which the patient can then perceive and interpret. [8–10] While these patterns may be difficult to decipher at first, it is expected that patients will eventually learn to match phosphene patterns to useful information such as letterforms. [32]

This process from camera image to neural stimulation loops continuously to produce a stream of images like frames of a video. Essentially, this system provides an artificial real-time link between environmental light and visual information; a link ordinarily present in natural vision, but not present in blindness. Figure 2.2 depicts the basic process of prosthetic vision in comparison to normal vision.

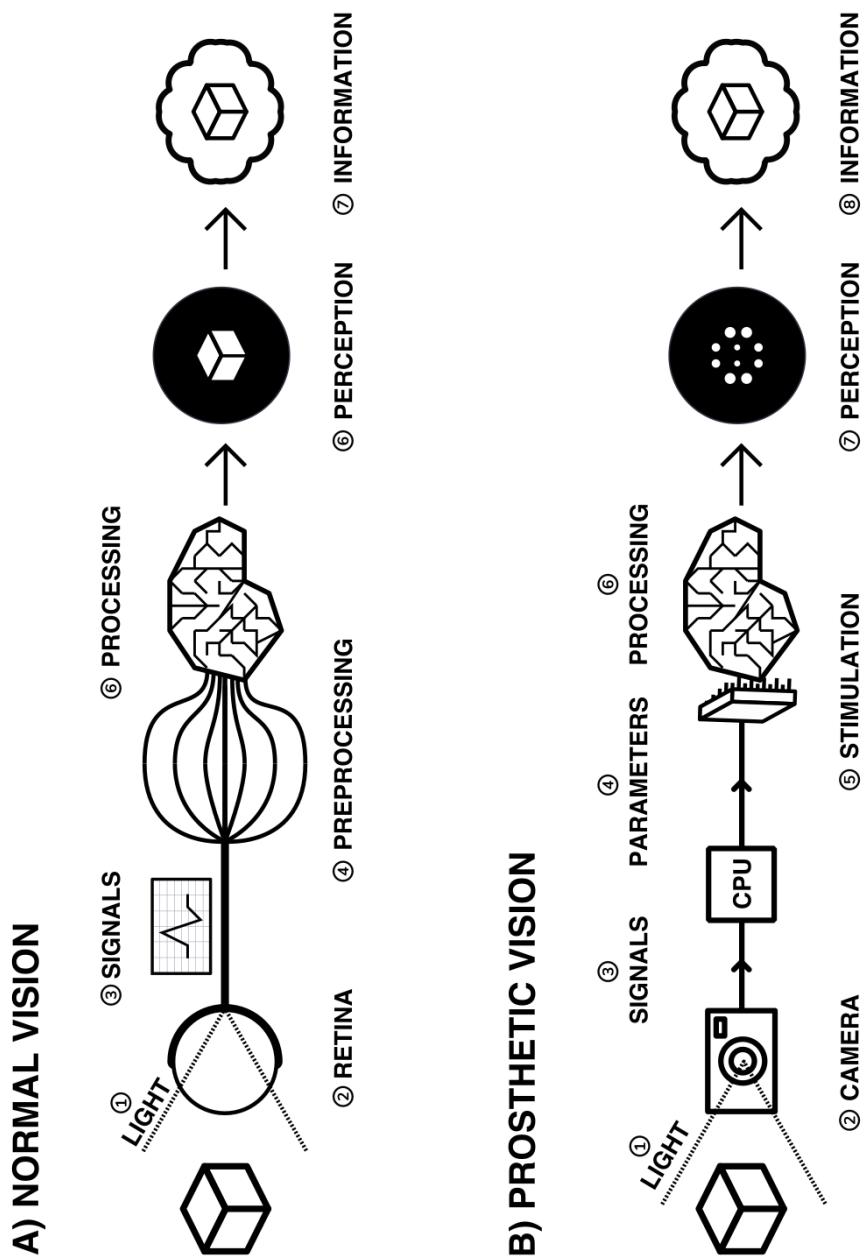


Figure 2.2: A basic flowchart of the process of prosthetic vision compared to normal vision.

Chapter 3

Qualities of Prosthetic Vision

Because cortical visual prostheses use phosphenes as the fundamental building block of prosthetic vision, it is imperative that we be able to compose multiple phosphenes into meaningful imagery. Whether phosphenes can be composed meaningfully depends on the visual and perceptual properties of phosphenes such as their size, color and interaction with other phosphenes.

Unfortunately, due to the technical and ethical issues surrounding stimulating peoples' brains, the number of studies characterising these properties of phosphenes in humans is understandably small. Studies which characterise phosphenes evoked *in vivo* typically fall into two distinct groups:

1. Historical experiments with rudimentary hardware on noble volunteers [8, 10, 38], or
2. Modern but conservative experiments in epilepsy patients who already have electrodes implanted for clinical monitoring. [11, 39–42]

As the pool of phosphene studies in humans is small and the demographics of these studies are skewed towards specific populations, we summarise the pertinent methodological features of each study in Table 3.1. These studies constitute the major perceptual evidence that a cortical visual prosthesis can produce vision, and point towards what type of vision might be possible.

There are several studies which also attempt to characterise phosphenes in non-human primates from trained behavioural responses. [43–46] While the qual-

tative perceptual information offered by these studies is limited, they provide some additional information about the spatial properties of phosphenes inferred from saccades (rapid eye movements).

Table 3.1: The participant demographics of studies which have looked at cortical phosphene evoked *in vivo* in humans.

Date	Reference	Setting	Electrodes	Parameters	Patient Demographic
1968 [8]	Acute	80 surface electrodes (array) on occipital cortex	Monophasic trains with 0.2ms pulses of unknown current (power 90mW) at 100Hz	1 patient blind from glaucoma and retinal detachment approx 1 year prior (female, 52 years)	
1974 [9]	Acute	Variable number of surface electrodes on occipital cortex	Monophasic or biphasic trains with 0.25-2ms/phase pulses of up to 1.5mA at 30-200Hz for 1000ms	15 patients with cerebral tumours and partial visual field defects or normal sight (11 male, 4 female, 20-71 years)	
1974 [38]	Acute	64 subdural surface electrodes (array) on occipital cortex	Biphasic trains with 0.5ms/phase pulses of up to 8mA at 50Hz for unknown duration	1 patient blind from congenital cataract in one eye and glaucoma and retinal detachment in the other for 28 years (male, 45 years); and 1 patient blind from trauma for 7 years (male, 28 years)	
1978 [12]	Chronic {5-2 years post implant}	Unknown	Unknown-phase trains with up to 2ms pulses of ?mA at 2-1000Hz for 1-8 pulses	Unknown	
1976, 1979 [47, 48]	Chronic {Unknown years post implant}	64 subdural surface electrodes (array) on occipital cortex	Biphasic trains with 0.25ms/phase pulses of 0.5-4.0mA at 50Hz for 500-1000ms	1 patient blind from trauma 10 years prior to implantation (male, 33 and 35 years)	
1990 [10]	Acute	1-3 intracortical penetrating microelectrodes on occipital cortex	Biphasic trains with 0.2ms/phase pulses of up to 200 μ A at 100Hz for 100-1000ms	3 sighted patients with epilepsy (unknown demographic?)	
1996 [13]	Acute	38 intracortical penetrating microelectrodes on occipital cortex	Biphasic trains with 0.2-0.8ms pulses of up to 80 μ A at 75-200Hz for 125-250ms	1 patient blind from glaucoma 22 years prior (female, 42 years)	
1994, 1999 [49, 50]	Acute	Unknown number of surface electrodes on extrastriate visual cortex	Biphasic trains with 0.2ms pulses of 2-10mA at 50Hz for 5000ms	Unknown	
2000 [51]	Chronic {2-1 years post implant}	64 (subdural?) surface electrodes (array) on occipital cortex	Biphasic trains with 0.5ms/phase of (10-20V) at 30Hz for 1-50 pulses	1 patient blind from trauma 5 years prior to implantation (male, 62 years)	
2000 [39]	Acute	Total 271 subdural surface electrodes on occipital cortex and adjacent areas across all subjects	Biphasic trains with 0.3ms pulses of 1-15mA at 50Hz for 5000ms	23 sighted patients with epilepsy (12 male, 11 female, 16-41 years)	
2009 [41]	Acute	Total 50 subdural surface electrodes on 11 different visual areas across all subjects	Biphasic trains with 0.2ms pulses of 0.49-7mA at 200Hz for 300ms	10 sighted patients with epilepsy (6 male, 4 female, 19-67 years)	
2016 [40]	Acute	1 or 2 subdural surface electrodes on V1 studied per subject	Biphasic trains with 0.2-1ms pulses of 0.2-5mA at 5-100Hz for 200-1000ms	4 sighted patients with epilepsy (3 male, 1 female, 24-40 years)	
2017 [52]	Acute	Up to 16 subdural surface electrodes (array) on early occipital cortex per subject	Biphasic trains with 0.1ms/phase pulses of 0.3-4.0mA at 200Hz for 200-300ms	15 sighted patients with epilepsy (5 male, 10 female, 22-61 years)	
2018 [53]	Acute	Up to 16 subdural surface electrodes (array) on early occipital cortex per subject	Biphasic trains with 0.1ms/phase pulses of 0.3-4.0mA at 200Hz for 200-300ms	8 sighted patients with epilepsy	
2018 [54]	Acute	16 or 24 subdural surface electrodes (array) on early occipital cortex per subject	Biphasic trains with 0.1ms/phase pulses of 0.3-4.0mA at 200Hz for 50-300ms	4 sighted patients with epilepsy (all male, 20-54 years) and 1 patient blind 8 years prior, unspecified reason (female, 35 years)	
2019 [42]	Acute	16 subdural surface electrodes (array) on occipital cortex	Biphasic trains with 1ms pulses of up to 11mA at 60Hz	1 patient with epilepsy and a partial visual field defect for 30 years from AVM haemorrhage (male, 45 years)	

Phosphenes Produced by Stimulating a Single Electrode

The most common result of stimulating a single electrode is a single phosphene characterised as a small, localisable dot of light likened to a star. [8, 9, 13, 39] Every study has, however, demonstrated large variation on this basic percept.

The Quality of a Phosphene

Being able to perceive qualities of light such as brightness and colour gives us richer, more specific information about the world. [55, 56] Composing images with phosphenes of different brightness and colour would enable us to mimic the richness of natural visual information.

It therefore seems promising that one of the most consistently reported features of phosphenes is that different levels of brightness *are* perceivable and even modifiable. The brightness of a phosphene reproducibly increases with stimulation amplitude, pulse duration and pulse frequency. [9, 12, 13, 38, 40, 51] An early study estimated up to 12 distinguishable levels of brightness by varying the stimulation amplitude of a surface electrode. [12]

The colour of phosphenes, however, is not as promising. Phosphene colours range from colourless to vividly coloured with large inter-individual variation. Some patients only report seeing white or colourless phosphenes. [8–10, 51] Others have reported a spectrum across almost every reportable colour and beyond to 'other-wordly' colours. [9, 10, 12, 13, 39, 41, 50] Sighted patients looking at a white background have also reported seeing black phosphenes, though this finding is not well reported elsewhere. [39] While it was previously speculated that blind patients saw colourless phosphenes due to long-term sight deprivation [9], this is not consistently the case, and coloured percepts have also been reported by a patient blind for 22 years. [13] Least promising is that colour is not consistently modifiable using different parameters of electrical stimulation, meaning phosphenes are most often randomly coloured. [12]

Spatial Properties of Phosphenes

Of great concern to cortical visual prostheses is how phosphenes are arranged in visual space, which may affect the shapes of patterns that can be formed by prosthetic devices. The visual cortex, as a sensory surface, is mapped retinotopically i.e. such that regions in the visual field which are next to each other are also next to each other on the cortex (though they may be distorted). [57, 58] Electrodes placed over visual cortex appear to follow this mapping, and relationships between adjacent electrodes are roughly conserved. [8, 48, 54] Figure 3.1 illustrates the mapping of a 64-electrode array to phoshene locations measured by perceptual testing.

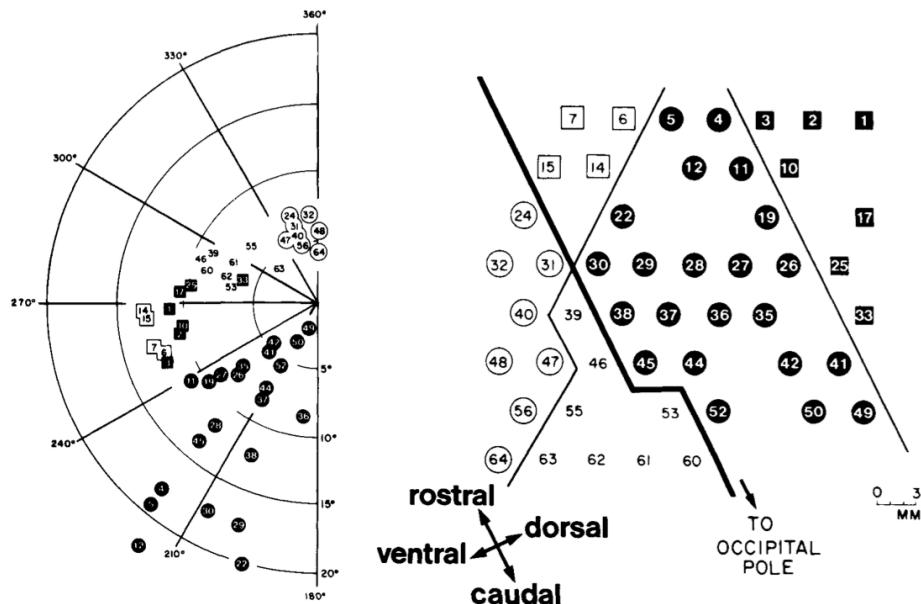


Figure 3.1: Spatial distribution of phosphenes mapped to the visual fields (left) in a patient implanted with an early 64-electrode array (right). Figure from Dobelle et al. 1979 [48]

However, while we can very grossly estimate positions of phosphenes in the visual field (especially in relation to the calcarine sulcus, below which phosphenes correspond to superior fields), the distortion of retinotopy on the visual cortex means precise mapping is not possible until post-implantation. In sighted patients, phoshene locations can be mapped with receptive fields in response to visual stimuli, to which they closely correspond. [52–54] This is clearly not possible in blind patients, so phosphenes are often mapped by indicating directions or relative

positions of pairwise phosphenes. [8, 13, 48, 54] The implication is that while we can roughly determine the quadrant of a phosphene in the visual field at implantation, we cannot know precisely where it is located until stimulation is trialled. In addition, sometimes stimulating one electrode produces more than one phosphene, which may be either adjacent or inverted about the horizontal meridian of the visual field. [8, 9, 13] This is most likely attributable to off-target stimulation of tissue across a sulcus, supported by observations that this phenomena occurs less severely with penetrating microelectrodes (which discharge less than surface electrodes). [13, 48]

The space a phosphene occupies in the visual fields varies with eccentricity and stimulation current. Early evaluations of phosphene size using various objects at arm's length [8, 9, 13, 38, 51] have generally been agreeable with more formal estimates using degrees of visual field. [10, 40, 52] Most phosphenes are 1-2° of visual field in diameter and range from 0.1-10°(a "grain of sago" to a coin at arm's length) . [8, 10, 52] The size of phosphenes depends on where they are located in the visual fields; more peripheral phosphenes are larger and reportedly have less distinct borders. [12, 40, 52] The variation of phosphenes with size is consistent with behavioural studies in non-human primary undergoing cortical stimulation [44] and the phenomenon of cortical magnification, where the central visual field is overproportionately represented on the surface of the brain. [59] Phosphenes also appear to increase in size with stimulation amplitude [12, 40, 52], though one early report of microelectrode stimulation also described instances where phosphene size decreased which have not been subsequently reproduced. [13]

While circular phosphenes are ubiquitous [8–10, 13, 39], other phosphene shapes have been reported. The most consistently reported shape other than circles are elongated elliptical or linear phosphenes oriented in horizontal, oblique or vertical orientations. [8–10, 12, 54] A few reports identify shapes ranging from triangles and stars, to checkerboards, to face or eye-like hallucinatory sensations. [39, 41] Often, more abstract phosphenes appear on stimulation of later visual areas of the brain, which may not be relevant for prostheses targeting only primary visual cortex. [41] While phosphene shapes appear loosely related to the putative

role of different brain regions [39], no studies have been able to deliberately control the shape of phosphenes.

Finally, phosphenes have been repeatedly shown to move with eye movements and have been likened to the movement of retinal afterimages. [8, 13, 38] It appears the whole map of phosphenes moves as multiple phosphenes maintain their relative positions after movement. [13, 38]

Temporal Properties of a Phosphene

Phosphenes generally appear synchronous with stimulation. [13, 54] It is difficult to measure the latency of percepts without also including motor reaction time, but studies comparing phosphene onset reaction times to auditory stimuli suggest that additional latency is minimal. [12] In multiple studies, phosphenes have been sporadically reported to persist for up to 20 minutes after stimulation ceased, particularly after a high-discharge stimulation prior. [8, 12, 13, 38] Perhaps paradoxically, phosphenes purposefully sustained by continuous stimulation demonstrate significant fading in as little as 15 seconds. [9, 13] The fading effect of phosphenes is also reflected over separate trials, where phosphenes progressively dim in each subsequent trial (though they "reset" the next day). [13]

On a shorter time scale, phosphenes elicited by surface stimulation may also flicker. The phosphene flicker produced by surface electrodes is fixed, fast, and asynchronous with hardware or physiological pulses. [8, 9, 12, 38, 51] This differs from the "flicker" produced by two separate successive stimulations, which disappears at stimulation frequencies of approximately 33Hz (though an overlying intrinsic flicker remains). [12] It is unknown whether flicker also occurs in stimulation with microelectrodes; of the few studies of stimulation with penetrating microelectrodes, flicker was not reported. [10, 13]

In summary, we can exert only very limited control over what individual phosphenes look like. Phosphenes are also highly variable, both between-individuals and between-electrodes. While there are points of agreement between studies, such as the effect of stimulation current on brightness, other phenomena, such as colour and flickering, remain contentious. It remains unclear whether these disagreements are

due to differences in stimulation parameters, hardware, participants or pathology.

Phosphenes Produced by Stimulating Multiple Electrodes

The appearance of images containing multiple phosphenes is fundamental to modern cortical prostheses as very little information can be transmitted through only a single electrode at once. [6, 35] The intention for cortical prostheses is to produce perceivable *patterns* which can be interpreted. The eventual hope is to approximate natural images with phosphenes used like pixels of a graphical display. Early chronic implants operated on this principle, albeit with very low resolution. [51]

However, the empirical evidence on *what* is perceived when multiple electrodes are stimulated is surprisingly scarce.

At the most simple level, two electrodes which produce individual phosphenes appear to also produce two separate perceivable phosphenes when stimulated simultaneously. [8, 9] Sometimes, the size of each phoshene decreases compared to individual stimulation, and the distance between phosphenes may increase. [53] When close together, these phosphenes may fuse together into a single percept. [8, 9] However, this is not always the case; in fact, dimmer phosphenes may not be perceived at all [9, 38, 53], though there is some evidence that increasing the stimulation amplitude may reintroduce the dimmer percept. [13] As a result, increasing the number of electrodes may not linearly increase the number of perceived phosphenes.

Several studies have characterised greater numbers of simultaneous phosphenes. Early evidence suggested that four-phosphene patterns (e.g. a square) could be recognised, but not reliably as spurious phosphenes appeared and some expected phosphenes were not perceived. [38] Another patient was able to perceive a six-phosphene vertical line. [13] Modern studies, however, have provided conflicting results. In one study in an epilepsy patient, five electrodes stimulated at once were only able to produce two perceivable phosphenes that was not simply the aggregate of each of the five phosphenes. [54] In non-human-primates, stimulation of visual cortex simultaneously at two spatially distant points did not sum to a joint signal,

further suggesting a separation of processing of simultaneous stimulation. [60]

The reasoning behind this difficulty is thought to be because cortical visual prostheses unselectively stimulate local regions of the brain. In normal primary visual cortex, neurons are typically selectively stimulated by specific image features such as the orientation of lines in the visual fields. [61] When electrodes instead unselectively stimulate neurons, the pattern of neural stimulation is unnatural and later visual areas may not immediately be able to decode the unrecognisable stimulus. [54]

Despite the difficulties of these temporary experiments, chronic studies suggest that patients are able to use this information usefully after a learning period. There are brief reports of a patient with a chronic implant being able to read phosphene patterns on a 64-electrode implant at 30 letters per minute, similar to Braille [47]. Reports on a different patient from the same group described the ability to recognise symbols and letters at an estimated visual acuity of 20/1200 (seeing at 20 metres what could normally be seen at 1200 metres). [51] However, due to the absence of any further chronic studies of implants in blind patients, the upper limit to which people can learn to recognise phosphene patterns is unknown.

Given these limited studies of combinations of phosphenes, there is a tremendous gap between the current knowledge of phosphene patterns and the proposed mechanism of cortical prostheses. It is entirely unclear whether people can perceive phosphene patterns on the order of tens or hundreds, whether people can learn to perceive these patterns in useful ways, or to what degree these patterns may change.

Summary of the Perceptual Limitations of Phosphenes

The major issues surrounding the current literature on phosphenes are therefore:

- 1. There are no modern studies of phosphenes evoked in blind but otherwise-healthy patients, the primary demographic of cortical visual prostheses.**

There are also scant chronic studies, none of which have been conducted with penetrating microelectrodes. [12, 47, 51] Emerging clinical trials will help

resolve this issue, but until such studies bear fruit, our knowledge on what cortical prosthetic vision looks like may not be readily applicable to new devices.

2. **Phosphenes are highly variable.** Almost all features of phosphenes display uncontrollable variability, and the only two properties of phosphenes we have been shown to reliably control are phosphene brightness and size. [12] This variability permeates between electrodes, between patients and between studies. The heterogeneity, low sample size and skewed populations of the literature have made it difficult to distinguish the root cause of such variation.
3. **The interpretability of patterns formed by multiple phosphenes is unclear.** There is conflict amongst studies on whether multiple phosphenes at once can be integrated simultaneously, or whether people can learn to compensate for initial difficulties with interpreting phosphenes.

As a result, there is considerable uncertainty on exactly what visual sensations modern devices can give to implantees on a case-by-case basis.

Chapter 4

Making Prosthetic Vision Useful

Our ability to control the appearance of individual phosphenes and their patterns is clearly limited. In this section, we briefly review current literature on how images can be represented strategically in phosphene space to overcome these limitations.

The Role of Simulated Prosthetic Vision

Because of the difficulties of implanting electrode arrays, little research has been conducted on what methods of representing information in phosphene space are most useful *in vivo*. The only cortical implant which has been connected to a camera in humans was the Dobelle Implant in 2000. [51]. The Dobelle Implant used direct image processing techniques fitting of the software capabilities of the time, which essentially downsampled the camera image and directly mapped the brightness to implant electrodes. [51]. The group briefly entertained the idea of using edge-detection for more selective stimulation, but no subsequent studies reported the outcomes of this idea.

To allow the testing of new image processing algorithms in the absence of access to real implantees, research in image processing algorithms has largely moved to simulations of prosthetic vision. [62, 63] Simulated prosthetic vision is the primary vehicle through which most new image processing algorithms are tested. The features of simulated prosthetic vision have been reviewed previously. [62] Briefly, camera information is processed and rendered onto a head-mounted or

other display as simulated phosphenes. Typically, these simulated phosphenes are rendered as the most commonly reported percept - white dots with a Gaussian blur filter applied. [62] In this way, phosphene "images" are displayed for the user with the aim to approximate the prosthetic vision of an implantee. Examples of these simulated renders are shown in Figure 4.1.

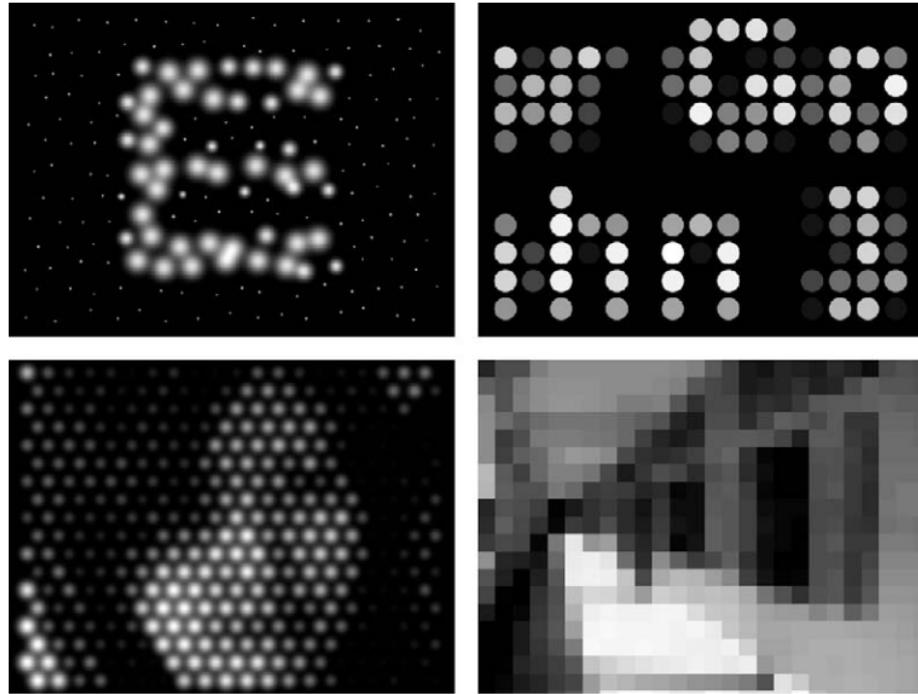


Figure 4.1: Examples of different simulated renders of phosphenes. Figure from Chen et al. 2009. [62]

A Brief Outline of Current Image Processing Approaches

Direct Mask-Based Methods

The prevailing paradigm of image processing for early cortical prostheses was to directly map camera images to a grid of electrodes as though they were superimposed. [13, 51] This produces a phosphene image like a mask full of holes placed on top of the original image. In this way, prosthetic vision began by attempting to emulate natural vision as closely as possible.

Such an approach may work with a large number of electrodes if all phosphenes

could be interpreted correctly as "pixels". One study estimated that approximately 625 phosphenes would be sufficient to reach a visual acuity of 20/30, suitable for most general tasks [64].

However, there are several issues of direct methods when compared with the perceptual limitations of phosphenes: Direct mask-based methods, by virtue of keeping faithful to the original image, tend to produce simulated phosphene renders with large numbers of "on" phosphenes, particularly in well-lit environments. As the ability to interpret multiple phosphenes simultaneously is not well established, stimulating many electrodes at once may not produce the expected visual percept. Because of the high variability of phosphenes, it is also unlikely that the quality and spatial distribution of pixels of a transformed image could be reproduced as faithfully as intended.

Moreover, the quality of these methods very rapidly degrades once resolution drops. [65] No new implants are capable of producing 625 distinct phosphenes. [6] Without the resolution to support the interpretation of low-level features of directly processed images, images can be uninterpretable.

Edge-Based Methods

Edge-detection refers to image processing algorithms methods which highlight the edges of objects only. [66] Edges require less phosphenes at once and may reduce the amount of redundant information in an image. This is important when we can consider that the number of perceivable phosphenes may not increase linearly with the number of stimulated electrodes. [53]

Edge detectors such as the Canny [66] edge detector are widely used. These edge detectors are able to detect fast pixel gradients in images, which typically occur at boundaries. If additional inputs to the processing algorithm are possible, then more sophisticated techniques can be used. For example, the use of a range camera or other depth sensing devices can be used to more intelligently find non-background edges. [67] State-of-the-art convolutional neural networks (CNNs) have also been applied to edge detection for prosthetic vision by semantic pixel labelling of images of rooms and determining edges by boundaries between walls. [68] An

illustration of an edge-detection algorithm combined with a object-filling algorithm is shown in Figure 4.2.

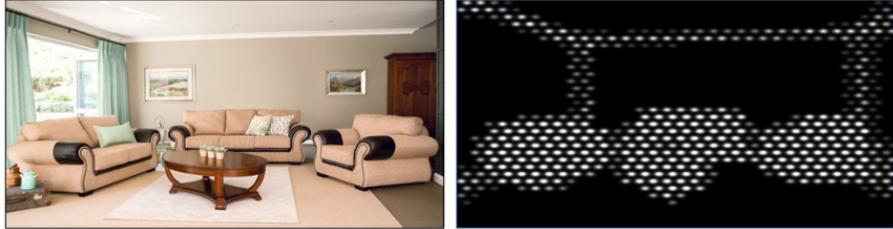


Figure 4.2: An example of an edge-detection algorithm with a CNN-driven object-filling algorithm to render a clean simulated phosphene image. Figure from [68]

The difficulty with edge-based methods is that edges easily degrade when resolution drops, similar to direct methods. [17] One method which aimed to resolve the fragility of edges combined edge-based methods with saliency-based methods to give greater form to objects. [69] Such hybrid methods may be more robust than the use of pure edges when faced with significant downsampling.

Saliency-Based Methods

As opposed to naively translating brightness values of camera images to electrode stimulation, saliency-based measures more intelligently identify the semantics of objects in a scene. Using this semantic structure, the image can be divided or *segmented* into regions of interest which carry a common semantic meaning (e.g. "background" or "foreground"). [70]

With this approach, more deliberate differences between foreground and background can be made as depicted in Figure 4.3. [71] When applied to an image classification task, saliency-based methods improved the recognition accuracy of common objects. [65, 69]

Object detection neural networks have also been applied to highlight particular salient features of an image. [72] In these methods, powerful image classification algorithms are able to detect a specified object and solely highlight that object on the simulated phosphene render. While these methods were constrained to only specific objects, they demonstrate leverage of modern progressions in image

processing to intelligently identify objects.

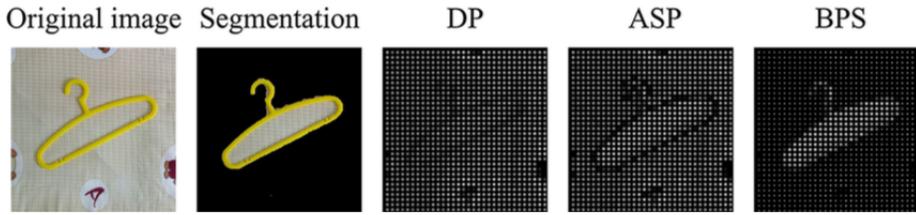


Figure 4.3: An example of a saliency-based algorithm to highlight a region of interest of the image; different panels show progressive stages in the process. Figure from [65]

The methods help highlight what is most likely to be relevant in an image and suppress background, which may otherwise interfere. However the usefulness of this masking approach is importantly constrained by the implantee's ability to subsequently recognise what is being shown. While simulations have demonstrated the utility of this approach, the fidelity of the mask form when phosphenes are irregularly shaped and sized is not clear. While these methods are advantageous compared to direct methods in that irrelevant information may be reduced, it faces the same limitations that low-level forms may be obscured by phosphenes distortions.

Transformative Methods

Recent advances in machine learning have meant that computers are now reaching human-level abilities for tasks such as image classification using deep learning methods. [20, 73] Because the processor in a cortical visual prosthesis has access to the full camera image (as opposed to the user, who can only see the phosphenes version), the processor has more information available to interpret. Instead of expecting the user to interpret high-level information from degraded phosphenes images, some interpretation could be relegated to the computer which can then intelligently re-encode the information in a deliberate manner.

Numerous patents have been filed for such a system. [74, 75] In these systems, important visual cues such as stairs, faces and bank notes are recognised by the computer, which can then remove unnecessary low-level detail and produce compact, abstract images that represent the *concept* of what is seen, not what is actually seen. For example, faces can be recognised and re-encoded as emoticons

which cleanly fit in low-resolution space. [67] An example of this approach is shown in 4.4



Figure 4.4: An example of a transformative approach to image processing to re-encode information. Figure from [67]

The chief benefit of such methods is that useful information can be communicated with less phosphenes. Since many low-level details (e.g. "is this bank note folded at the corner?") are not always relevant, the information burden to the user can be reduced to only what is necessary.

The perceptual issues facing these methods are that these typically rely on producing phosphene images that "mimic" real life (e.g. emoticons, which attempt to mimic faces). [67] However, the perceptual distortions and variability of phosphenes make it unclear whether these mimics could be replicated and thus the ability to specifically evoke these "mimics" *in vivo* with phosphenes is not well established. What may appear cleanly represented in simulated phosphene space may be heavily distorted and even unrecognisable in real implantee settings. A potential rebuttal is that as these methods re-encode information at the bequest of the algorithm implementer, they could be optimised on a case-by-case basis for the particular phosphenes an implantee sees. Individual-level implementations of image processing algorithms are yet to be explored.

Temporal Methods

Given the challenges already faced by patients when trying to interpret multiple simultaneous phosphenes, some groups have begun to explore non-simultaneous methods of conveying patterned information. *Dynamic current steering*, where phosphene patterns are "traced" in quick succession rather than presented all at

once have shown significant benefits for letter recognition tasks. [54, 76] Figure 4.5 illustrates the principle behind this approach. Patients, without prior training, were able to trace the path of phosphenes and interpret simple letterforms accurately. Unlike most of these other methods, dynamic current steering *has* been tested *in vivo* in patients with epilepsy with subdural surface electrodes and in fact arose out of perceptual difficulties noted by the investigators. [54] This highlights one of the issues with simulated prosthetic vision tests; fundamental perceptual differences of electrical neural stimulation may not be discovered and accounted for until tested in real patients.

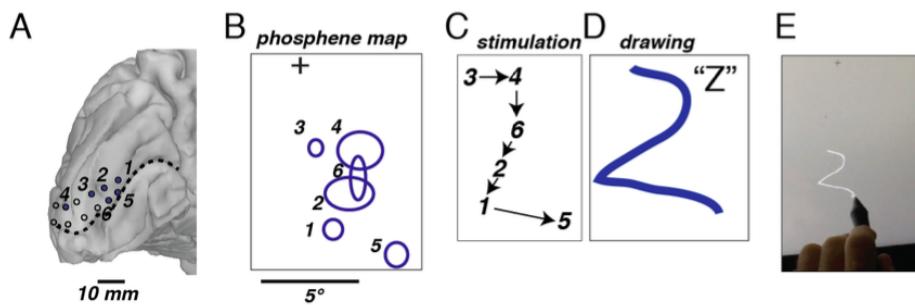


Figure 4.5: An example of a temporal approach to conveying information through phosphenes, where electrodes are stimulated in succession instead of simultaneously. A) implant locations, B) phosphene map, C) stimulation order, D) phosphene form, E) participant interpretation. Figure taken from [54]

These methods attempt to resolve the issues surrounding interpreting multiple simultaneous phosphenes by tapping into the brain's natural ability to interpret gross motion. [77] While temporal methods may be slower at conveying information per unit time, they have the advantage of not requiring simultaneous presentations of phosphenes, and requiring less current as only a limited number of phosphenes need to be conveyed for a single frame. It is clear, however, that the difference between temporal methods and form-based methods cannot be assessed in simulated prosthetic vision as the differences between these two methods is based on differences in *in vivo* perception. More clinical trials are needed to establish whether temporal methods of information transfer as opposed to spatial methods are better suited for cortical prosthetic devices.

The Limitations of Simulated Prosthetic Vision

The crux of many of these methods rests on tests of simulated prosthetic vision, faces significant limitations for generalisability to *in vivo* implants:

1. **The possible difficulties *in vivo* of interpreting phosphenes are not accounted for.** One of the key unknowns in phosphene space are how well the brain can learn to decipher the unselective unnatural stimulation of visual cortex by multiple electrodes simultaneously. [15] Because our uncertainty in this area relates to the unnatural neural stimulation of the visual cortex, this cannot be assessed in a simulated setting. When phosphenes are simply shown on a display, a sighted subject naturally is able to process the displayed patterns making full use of the retinal and neural circuitry distal to the brain. This is most certainly not the case for an implantee. While normal-sighted subjects have often been shown to be able to recognise complex patterns with many simultaneous phosphenes in a simulated setting [63], it is still unclear whether this can be replicated in real implantees.
2. **Phosphene simulation often does not account for all known properties of phosphenes.** The properties we have described above are rarely all accounted for. For example, temporal effects such as fading and accommodation are not implemented in most simulations. Additionally, most simulations render phosphene images as low-resolution greyscale images with uniform circular [65, 68, 78, 79] or hexagonal pixels. [80], though some studies have also incorporated biologically-based retinotopic distortion. [81, 82] Indeed, many simulated prosthetic vision algorithms approach the problem as chiefly one of low resolution, loss of colour and distortion. This does not accurately reflect the rich (but uncontrollable) perceptual experiences previously reported by *in vivo* experiments in shape, size, colour or flicker.
3. **Many simulations use regular and higher-resolution phosphene grids than have previously been achieved.** Sometimes, psychophysical experiments render on the order of a thousand phosphenes [65, 68, 71] This far outstrips the number of phosphenes which have been tested simultaneously

in vivo so far (less than a hundred) [51], and also surpasses the estimates of capabilities of modern prostheses. [6] Ultimately, the ability to reliably evoke many cortical phosphenes regularly and of the calibre of many simulations has not been established.

4. **Most simulations do not specifically target cortical phosphenes.** Most advances in image processing methods for simulated prosthetic vision are targeted at retinal prostheses, for which there are already commercially available devices. [26, 83, 84] The lack of distinction between different biological methods of evoking phosphenes in some experiments of simulated prosthetic vision mean results may not be directly translatable.

The implication of these issues is that studies of image processing algorithms in simulated prosthetic vision are not flexible for different phosphene percepts. The methods we have described are dependent on being able to replicate the simulated renders in *in vivo* implants, but little research has been conducted on this area. It is unclear how the image processing algorithms produced by studies of simulated prosthetic vision could be made flexible for the variability in phosphenes previously described.

Future Directions from Advances in Machine Learning

The remaining goal for better addressing the perceptual limitations for phosphenes is to find flexible ways to reconcile image processing algorithms in simulated phosphene space with the wide variability in what phosphenes look like. Image processing outside of prosthetic vision has experienced a wealth of improvements from advances in machine learning, where computers learn patterns from data without prior knowledge of those patterns. [20] Machine learning for image processing is most often applied to the training of **convolutional neural networks** (CNNs), which are layered architectures of image filters modelled after the function of physiological neurons. [73] CNNs have already discovered widespread use in image recognition tasks [85], video recognition tasks [86] and style transfer tasks, where images are modified to mimic the style of other images. [21]

The benefit of CNNs is that they produce image processing algorithms that are trainable from data without requiring explicit programming. This could be applied to the current gap between simulated prosthetic vision and perceptual limitations of phosphenes. By training CNNs on task performance data rather than programming algorithms to produce explicit patterns of phosphenes, CNNs could learn to process images in phosphene-agnostic ways. CNNs could also be tailored for individuals' perceptions of phosphenes, as the dependency of the algorithm shifts towards task performance tests rather than perceptual tests. This essentially positions the problem of deriving image processing algorithms from bottom-up (from phosphenes to algorithm) to top-down (from performance to algorithm).

Generative adversarial networks are a special type of training architecture for developing CNNs. [87] GANs train two networks - a *Generator* and a *Discriminator* - to perform opposing tasks - one to generate novel imagery, and one to determine whether the generated novel imagery is fake. In this way, CNNs can be trained to produce new images when given samples of images, such as digits. [88–90] Importantly, these novel images are not copies of the provided image samples but represent entirely new samples which exhibit general features similar to the training data. The technical implementation of GANs is beyond the scope of this review, but we describe our implementation in contrast with traditional GAN architectures in Section 6

While GANs have been applied to image-based tasks in other domains, it is not clear how they should be applied to prosthetic vision. An important difference between GANs in other domains versus prosthetic vision is that typically GANs directly manipulate every pixel in images they generate. [87] This gives GANs complete control over what its generated images look like. However, this is not desirable in simulated prosthetic vision where we want to simulate visual experiences which we *cannot* fully control. GANs must instead be used to generate *instructions* to simulated electrodes, which produce a simulated prosthetic vision render independently from the GAN. A useful GAN implementation for simulated prosthetic vision should therefore *not* be based on direct pixel manipulation (as is typically the case).

To our knowledge, GANs have not previously been applied to derive image processing algorithms for prosthetic vision, and thus research on this topic is scant. While neural network architectures have been applied to prosthetic vision to construct new algorithms [68, 72, 91], these have typically been object detection networks involved in segmentation-based processing. There has also been growing interest in using machine learning recognition algorithms for transformative techniques in prosthetic vision. [74] With continuous improvements in the capacity of hardware to support advanced processing [92], it is possible that emergent research in using machine learning to produce flexible, trainable algorithms could improve the utility of a cortical visual prosthesis.

Chapter 5

Aims and Hypothesis

Research Questions

1. Can GANs be used to train a neural network to produce phosphene patterns representing digits, given different simulated properties of prosthetic vision and, if so, how could this be implemented?
2. When applied to a digit representation task, do people find it easier to recognise digits from phosphene patterns derived from training via GANs, compared to a basic comparison (directly masking digits with a phosphene grid)?

Given that the usefulness of an image processing method is highly subjective and task-dependent, we formulated the second research question to confine our assessment of usefulness to a simple single-digit recognition task. Generating images of single digits is well-described in literature of traditional GANs, and the a digit recognition task is both focused and concrete.

Aims and Hypothesis

Aim 1 Develop a preliminary software implementation of a GAN training architecture for generating phosphene patterns, which could be applied to different phosphene simulations with minimal modification. This was the primary aim.

Aim 2 Experimentally test whether people find it easier to recognise digits from phosphene patterns derived from our prototype training implementation, compared to a basic mask-based comparison. This was the secondary aim.

Aim 2 is *not* intended to provide conclusive or compelling evidence on the usefulness of GANs in general for simulated prosthetic vision; it is purely intended as a short-term checkpoint on the performance output of the implementation in Aim 1. Given the novelty of the project, it is not expected that this project will produce a necessarily useful software implementation in Aim 1. The preliminary validation results may instead be used to guide how to better refine the software implementation for future use.

Aim 2 relies on the implementation produced by Aim 1, and can be hypothesis-tested against the following hypothesis:

1. Sighted participants, under simulated conditions, have a higher overall digit recognition accuracy when viewing phosphene patterns derived from our prototype GAN training architecture, compared to when viewing phosphene patterns produced by basic image processing using masking.

Part II

Methods

Chapter 6

Generative Adversarial Network Architecture

Development Context

The code for this project was written by the author using the Python programming language and accompanying package dependencies listed in Appendix A. All code written during the project is made available on a public GitHub repository listed in Appendix B; several small code snippets have also been included. All code was developed on a Windows 10 personal computer, and executed both on the development computer and remotely on the M3 multi-modal Australian ScienceS Imaging and Visualization Environment (MASSIVE) high-performance-computing (HPC) platform. [93]

Architecture Overview

A high-level overview of the implemented GAN training architecture is provided in Figure 6.1.

The overarching goal of this architecture is to train the *Generator* neural network (which is initialised to produce random noise before training) to generate simulation *encodings* for any specified *Simulator* which result in phosphene patterns that look like digits.

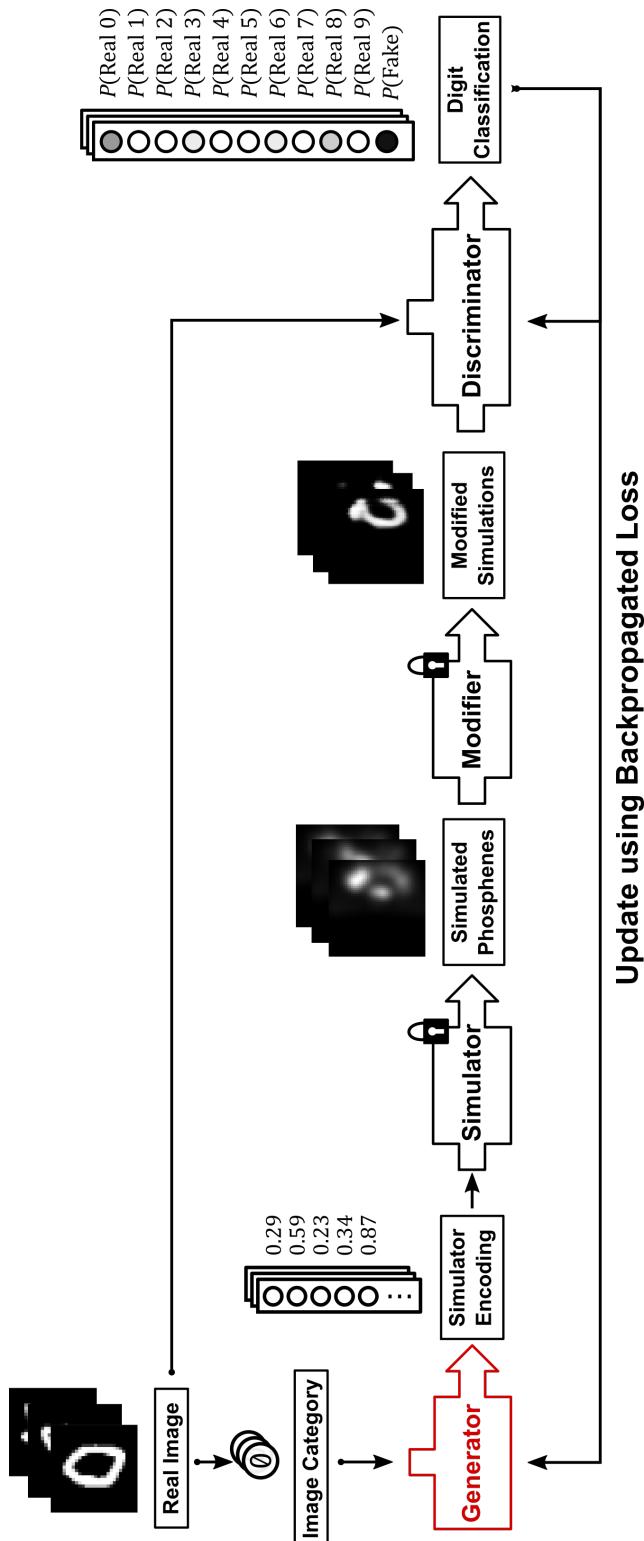


Figure 6.1: Overview of our GAN training architecture implementation demonstrating the flow of data through the architecture. Real images of handwritten digits and "fake" images produced by a *Generator* (red) are supplied to a *Discriminator* and subsequently trained on its output. In this architecture, the *Generator* and *Discriminator* are dynamically updated with each image sample, whereas the *Simulator* and *Modifier* are static (indicated by locks).

The flow of data through the training process proceeds as:

1. A *Simulator* is first chosen for this training run. This can be likened to preparing a phosphene generator for a new implantee with unique phosphene perceptions. Once a *Simulator* is chosen, it is used for the duration of training.
2. A real image sample of a digit is taken from a pool of real digits.
3. The image category is fed into the *Generator*, which produces a candidate encoding for the specified *Simulator* for this training run.
4. The *Simulator* takes the encoding and simulates it as phosphenes, producing a simulated prosthetic vision image.
5. The *Modifier* takes the simulated prosthetic vision image and modifies it to more closely align it with the domain of the real images and reduce systematic differences between the simulated image and the real images.
6. The *Discriminator* takes the real images and the modified simulated prosthetic images and produces predictions whether the digit is real (and what digit it is), or whether it is fake.
7. The predictions of the *Discriminator* are used to adjust the properties of the *Generator* and the *Discriminator* to perform their tasks. The *Generator* wants the *Discriminator* to classify its generated images as real digits; it therefore compares the *Discriminator*'s actual predictions to a hypothetical case in which the *Discriminator* predicted all the generated images were real, and uses the difference to rebalance its internal weights. The *Discriminator* wants to correctly classify the real images as real, and the generated images as fake; it therefore compares its output with a hypothetical case in which its predictions are all correct, and likewise uses the difference to rebalance its internal weights.

As with classical GAN architectures, the *Generator* and *Discriminator* participate in a zero-sum game where each network has opposing goals; the *Generator* aiming to fool the *Discriminator* into thinking the *Simulator*'s modified output is real,

and the *Discriminator* aiming to identify the *Generator*'s fakes amongst the real samples. This architecture closely follows classical GAN architectures, except with the novel additions of the *Simulator* and *Modifier* steps. These two steps differentiate this GAN architecture from traditional GANs, which usually have full pixel-access to the generated images; in this architecture, the GAN instead produces image-agnostic encodings that are supplied to the simulator. The decoupling of the GAN from the generated image pixels allows the GAN to be applied to this simulated prosthetic vision task for different types of simulators, which can be readily substituted.

This basic process is repeated many times over many digit image samples to slowly optimise the *Generator* to produce useful simulator encodings. Each step of this process is discussed in further detail in the sections below.

Real Images and Image Categories

The *Generator* begins training as a naive neural network with no conception of what digits are or what they look like. In order for the architecture to train the *Generator* to produce digit-like encodings, the *Discriminator* must be provided samples of what real digits actually look like. Importantly, these image samples are *not* simply copied or masked by the *Generator*; instead, general features of the image samples are extracted (such as lines and circles) and used to guide generation.

This project used handwritten digit samples from the public modified National Institute of Standards and Technology (MNIST) dataset digit dataset. [94] The MNIST digit dataset consists of 60,000 grayscale images stored as 2D pixel arrays with dimensions 27x27 pixels.

The MNIST digits are advantageous because:

1. It is a publicly-available, comprehensive, labelled, clean, and well-validated dataset (often the *de facto* dataset for benchmarking machine-learning tasks).
2. Handwritten digits ensure there is sufficient variation within the dataset so that the *Generator* and *Discriminator* are incentivised to learn general features of digits (as opposed to rote-memorising whole digits, as might occur with using

digits in a standard font).

In order for the *Discriminator* to produce useful optimisations, there must be no obvious systematic difference between simulated images produced by the *Simulator* and the real images. If such systematic differences were present, the *Discriminator* would have no difficulty discriminating real and generated images and the *Generator*'s output would therefore shift aimlessly as it tries to converge to an impossible goal.

A series of adjustments were therefore performed on the real images to mitigate systematic differences with images achievable by the *Simulator*. In order to more faithfully simulate a single unilateral cortical prosthetic vision device (which only stimulates half the visual fields), we restricted simulations to a single half of the image (the right visual field). We conducted the same adjustment to the MNIST digits by scaling each images' width by half and aligning the result with the right image edge. Each MNIST image's pixel value was also normalised between -1 and 1 (from 0 to 255) and the entire image upscaled to a resolution of 48x48 pixels to equal the pixel resolution of simulated images. Figure 6.2 shows a number of the original samples from the MNIST dataset, and the samples after this preprocessing step was applied.

The MNIST digit dataset includes a digit label for each handwritten data image in the range 0-9 inclusive. These image categories are simply encoded as integers corresponding to their digit label, and fed directly to the *Generator*.

Generator

The *Generator* is a neural network which takes a single digit in the range 0-9, and returns a simulator encoding to be fed into the *Simulator*. The simulator encoding is a vector (list) of numbers in the range 0-1, with one number for each simulated electrode in the *Simulator*; each number specifies the strength with which its corresponding simulated electrode should be stimulated to produce a phosphene pattern. Notably, there is no information encoded about what electrode actually *does*, nor where it is located; the *Generator* network is effectively blind to these

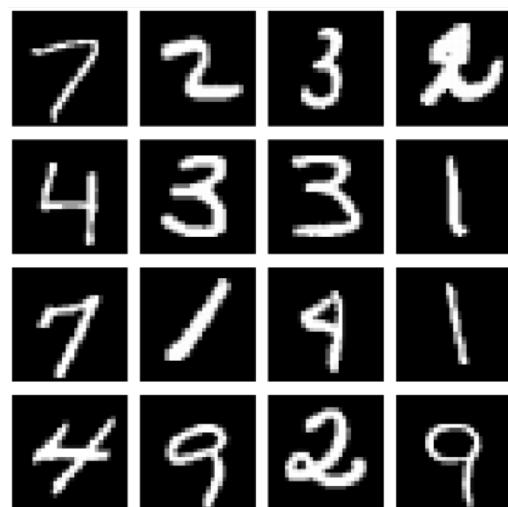
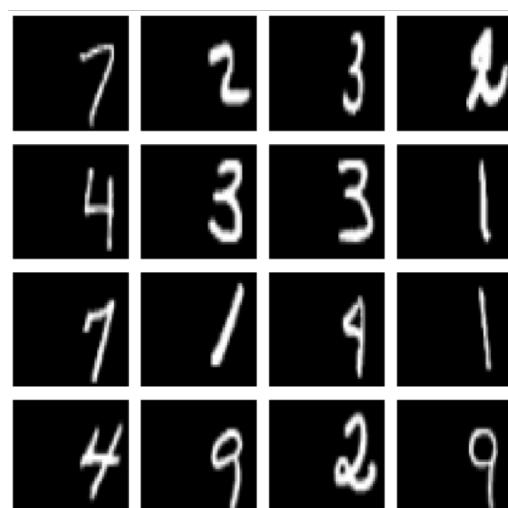
A. Original MNIST**B. Preprocessed MNIST**

Figure 6.2: 16 randomly-selected digits from the MNIST handwritten digit dataset, before (A) and after (B) preprocessing (realignment and normalisation) was applied.

elements of the simulation.

The *Generator* neural network is illustrated in Figure 6.3 and consists of:

1. An input layer, containing a single neuron taking a digit from 0-9 as input.
2. An embedding layer, which maps the single categorical digit class into a 10-element continuous vector. The embedding layer is passed through a batch normalisation step to normalise mini-batches of image samples, and a leaky rectified linear unit (ReLU) activation function to dampen negative values and stabilise the layer.
3. A dense layer, which fully connects each neuron in the previous layer to output a vector of simulated electrode encodings. The dense layer is passed through a sigmoid activation function to constrain all outputs to lie between 0 and 1.

This network was implemented using the machine learning package TensorFlow [95]; the code for this network is shown in Appendix B.

This network is simple with minimal hidden layers, which we justified as being sufficient for a digit-to-encoding task and suitable to reduce training times, given the time limitations of the project. There is an important difference in the *Generator* network described above compared to traditional *Generators* in GANs: the lack of a random numerical seed input. This was an intentional choice. The use of a random input seed into *Generator* networks in traditional GAN architectures permits the production of multi-modal images - i.e. multiple image patterns for a single image category (such as more than one pattern for a digit 0). This is often desirable when we want GANs to produce many different novel images instead of just one; however, for use in simulated prosthetic vision, multi-modal output is a low priority. It is actually desirable to limit the output to one "good" phosphene pattern per digit; in practice, subjects are likely to find it easier to learn one phosphene pattern per digit only, rather than having to learn multiple patterns.

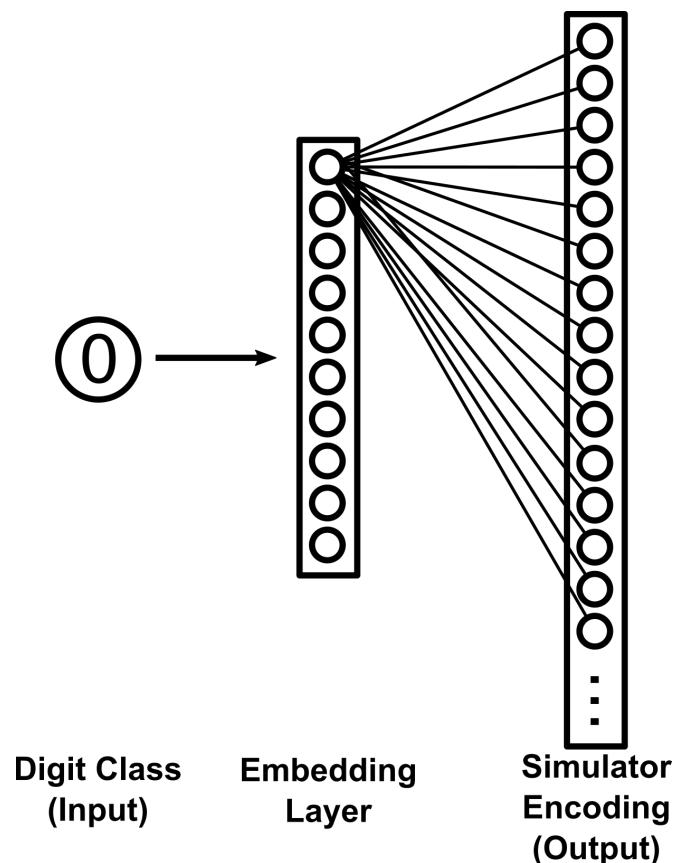


Figure 6.3: Schematic of the GAN generator neural network; a digit class is supplied to an initial embedding layer, and fully connected to the output layer of neurons to produce the simulator encoding. bbb

Simulator

The *Simulator* is a custom simulation of prosthetic vision, which takes instructions for simulated electrodes as input, and outputs the simulated render as a grayscale image. The *Simulator* is a simulated stand-in for real implantees and their perception of vision through a CVP.

The *Simulator* itself is a 3D volume of pre-rendered 2D grayscale image slices (illustrated in Figure 6.4). Each slice corresponds to the visual percept of a single electrode. These pre-rendered grayscale slices allow each electrode to produce a distinct, unique percept; for example, one electrode in the *Simulator* may produce a small phosphene located near the center of vision, while another may produce a much larger phosphene in the peripheries. The *Simulator* takes the simulator encodings from the *Generator* and weights each electrode's slice in the 3D volume according to its corresponding encoded strength. The slices are then aggregated together by summing along the last axis, and clipping values between 0 and 1 to produce a final grayscale render. The final render is then rescaled between -1 and 1 before being supplied to the discriminator.

Because the *Simulator* base is a 3D volume of pre-rendered slices, different subjects and their phosphene maps can be simulated by substituting different pre-rendered volumes without requiring any additional changes. In addition, the use of pre-rendered slices makes simulation efficient, as rendering images consists merely of bulk matrix operations which are easy to parallelise on graphics processing units (GPUs).

In order to generate pre-rendered slices to test, we implemented two different types of phosphene maps:

1. **Cartesian** maps, which place phosphenes in a regular Cartesian plane and assigns the same size and strength to each phosphene. This was chosen as a basic control as this map approximates a normal (albeit low-resolution) pixel image.
2. **Polar** maps*, which place phosphenes in regular Polar grid and assigns each phosphene a size corresponding to its location, as given by a log-polar

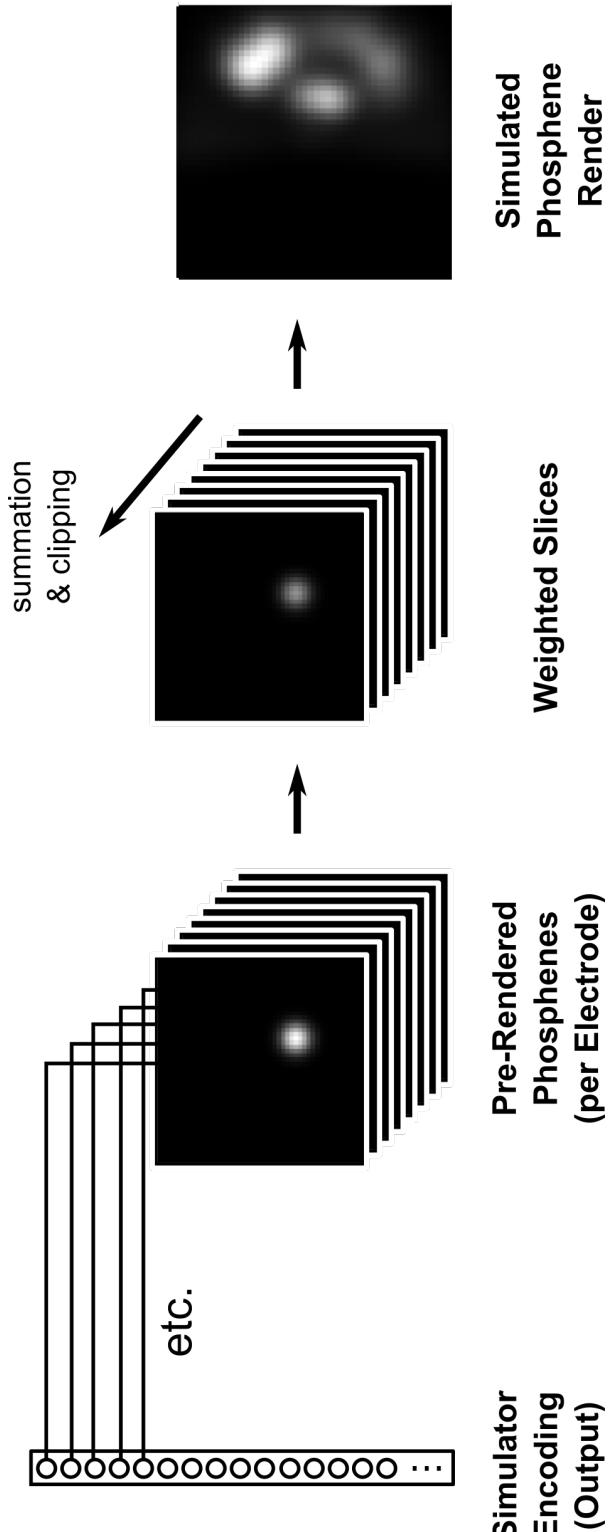


Figure 6.4: Prosthetic vision is simulated by weighting each pre-rendered slice of a 3D volume by provided simulated encodings, and aggregating the weighted slices.

equation $\log(k \times (x^2 + y^2) + a)$ for parametric constants k and a (see Appendix B for the code implementation). This was chosen to more closely reflect the visual qualities of phosphenes discussed in Section 3.

Both options constrained phosphenes in the right visual field. Each of the two options above also had the option of creating randomly-positioned phosphenes to reflect the randomness of *in vivo* phosphene maps.

The *Simulator* can theoretically simulate any number of phosphenes; we focused on phosphene maps containing 64, 144 and 576 phosphenes only to represent low, medium and high resolution phosphene maps. Examples of these phosphene maps are shown in Figure 6.5.

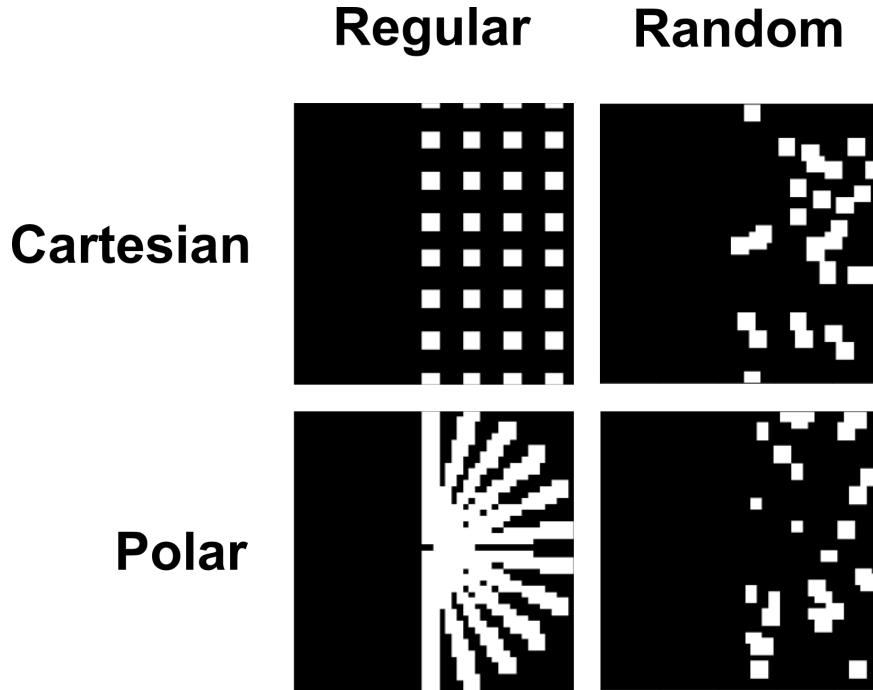


Figure 6.5: Phosphene maps used in the prosthetic vision simulation; regular and random Cartesian and Polar maps were modelled. Shown are four examples of maps with varying phosphene resolutions.

Modifier

In order for GANs to work effectively for the next step (feeding into the *Discriminator*), the generated simulations must have the capability of looking as close to

the real images as possible. Because the generated renders necessarily create simulated phosphenes, there would usually be no chance of the renders actually looking like digits unless the rendering resolution (i.e. number of phosphenes) is unrealistically high. To resolve this problem, a *Modifier* network can be added in order to bring the domain of the generated renders closer to the domain of real handwritten digit images to simulate basic "inference" that humans might make. This amounts to simple "hole-filling" for most purposes - i.e. connecting spaces between phosphenes, as a human might be able to do, and outputting a new "inferred" image corresponding to a holistic outline and form.

For this project, the *Modifier* consisted of a simple style-transfer neural network which mapped between phosphene images and "filled" forms, illustrated in Figure 6.6 and listed in Appendix B. Style-transfer neural networks are well-studied architectures, and the training architecture of the *Modifier* closely followed a traditional style-transfer GAN architecture comprising:

1. An *input* layer, taking the entire 2D grayscale render image (reshaped as a 3D slice with a depth of 1).
2. Multiple *2D convolutional* layers (48, 96, 128 and 128 filters), each followed by a *batch normalization* and a *leaky ReLU* activation layer, which progressively downsample the image into extracted features.
3. Two *2D deconvolutional* layers (48 and 1 filter/s), which progressively upsample the extracted features into a new modified image render. The last layer is passed through a *tanh* activation function to remap the output between -1 and 1, as in the original render.

In order to train the *Modifier*, MNIST digits were passed under 4800 random polar grids of 64 phosphenes with a simple masking operation to generate pairs of phosphene images and corresponding real digit images. These phosphene-real pairs were used to train the weights of the *Modifier* network so it could map between novel phosphene image and real-like digit image pairs. Once the *Modifier* was trained, its weights were saved and frozen for subsequent use as a simple phosphene-to-forms image processor. Examples of the style transfer output of the

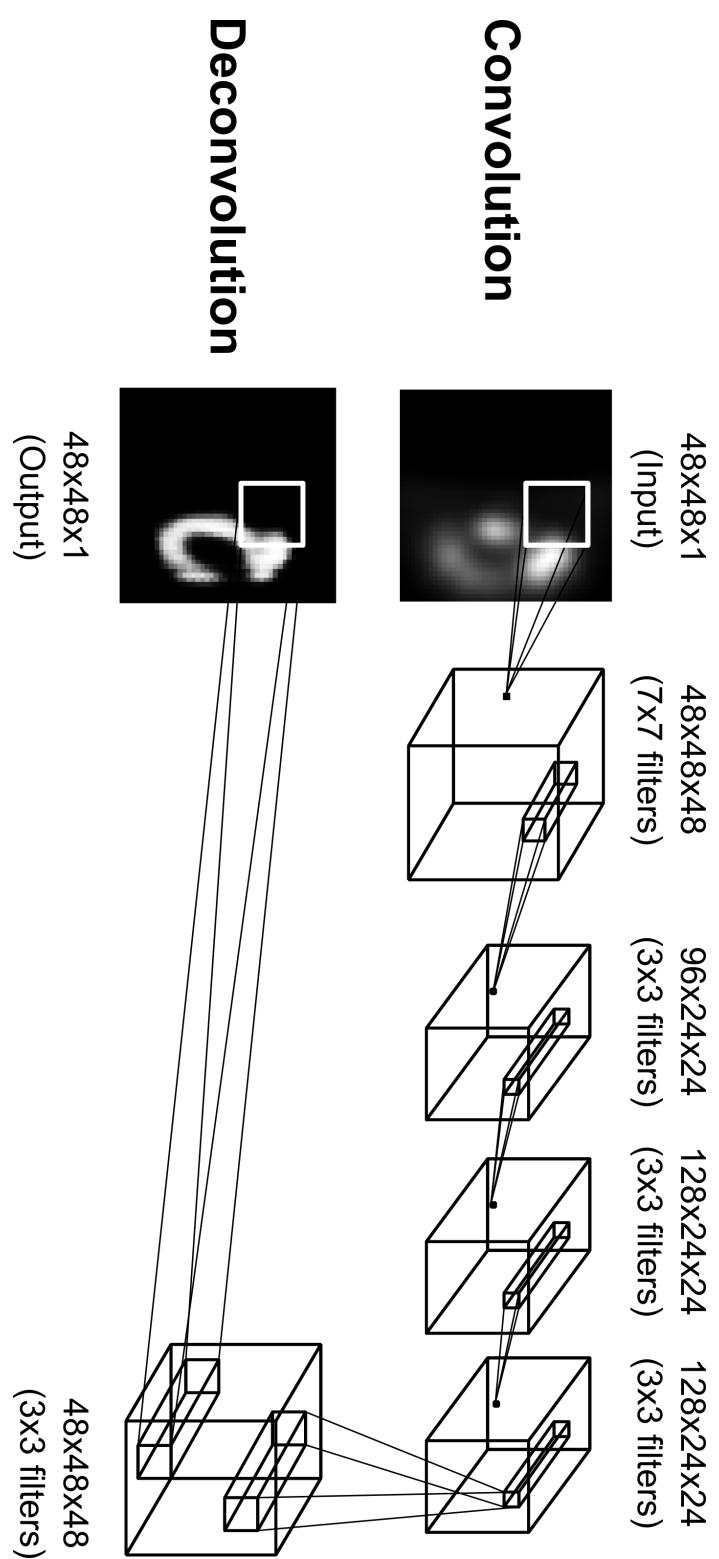


Figure 6.6: Schematic of *Modifier* neural network architecture bring phosphene images closer to the domain of the real handwritten digits with simple inference.

trained *Modifier* is shown in Figure 6.7.

Discriminator

The *Discriminator* is the final major processing step of the GAN training architecture, and the major element involved in actually training the *Generator*. The *Discriminator* takes a modified rendered grayscale image, and outputs an 11-element vector of probabilities. The first ten elements correspond to the probability the *Discriminator* believes the input image is a 0, 1, 2 etc. until the digit 9. The final element corresponds to the probability the *Discriminator* believes that the input image is not a digit at all, and is instead fake. The *Discriminator* is therefore a classifier which classifies images as either belonging to one of 10 digit classifications, or as being fake instead.

The *Discriminator* itself is a convolutional neural network with a relatively small number of layers, illustrated in Figure 6.8 and implemented in Appendix B. It comprises:

1. An *input* layer, taking the 2D grayscale modified render image (reshaped as a 3D slice with a depth of 1).
2. Two *2D convolutional* layers (64 and 128 filters), each followed by a *leaky ReLU* activation function and a 25% *dropout* layer (randomly discarding 25% of prior inputs to improve the robustness of the network).
3. A *flatten* and a *dense* layer, to output an 11-element vector, which is then passed through a *softmax* activation function to normalise all probabilities to sum to 1.

Loss Functions and Training

Once an image, generated or real, has passed through to the *Discriminator* to output probabilities, these probabilities can then be used as the basis for adjusting

A. Simulated Phosphenes



A. Modified Simulations

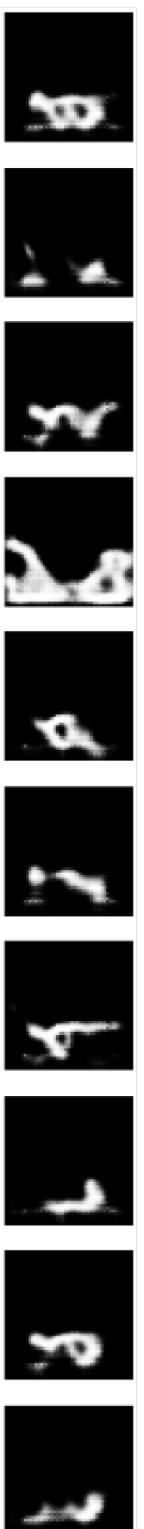


Figure 6.7: Examples of passing phosphene images under the *Modifier* network: simulated phosphenes are shown in A, and their corresponding modified renders in B.

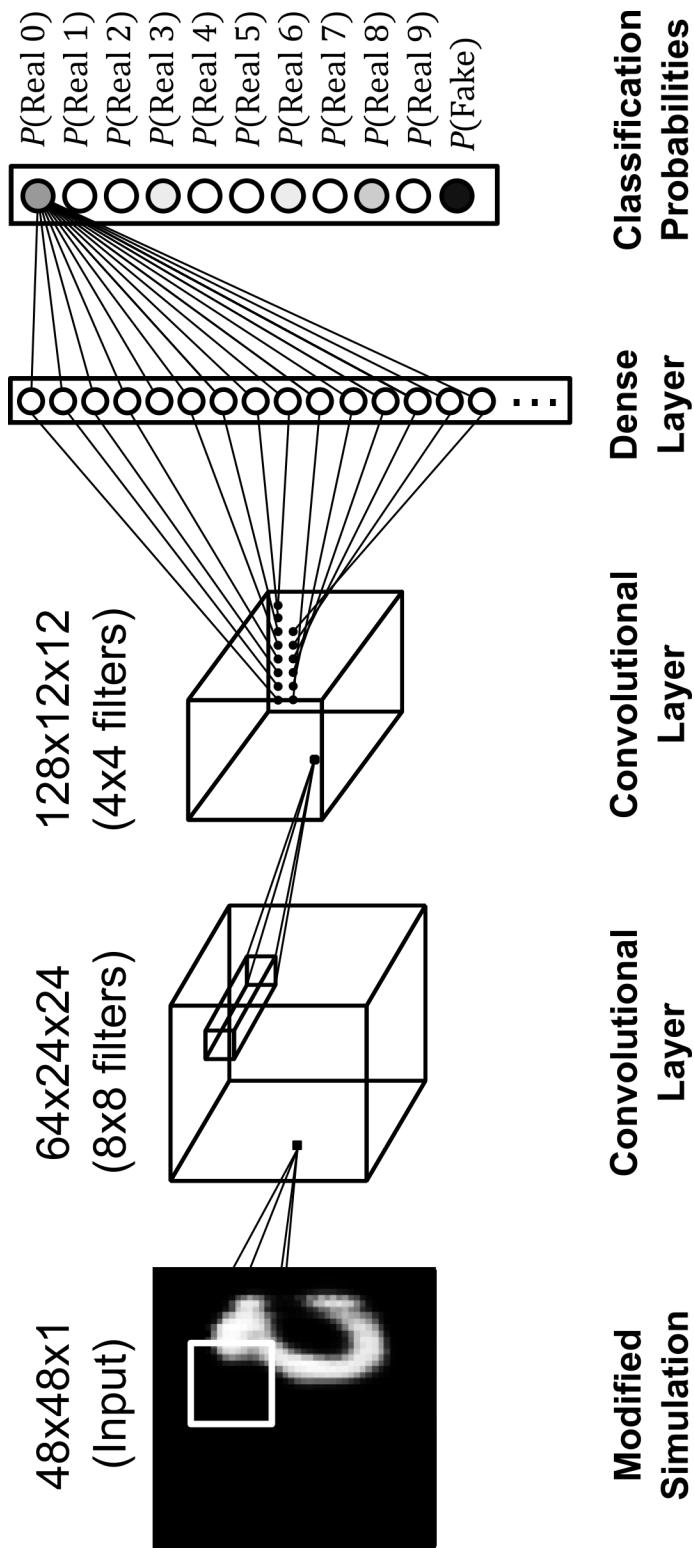


Figure 6.8: Schematic of the *Discriminator* convolutional neural network architecture which classifies images into probabilities of belonging to each digit class, or being fake.

the *Generator* and *Discriminator* to optimise them toward their respective goals. This is the foundational basis of any GAN training architecture.

The *Generator*'s goal is to have the modified renders produced from its encodings look as indistinguishable from real digits as possible. When its modified renders are passed through the *Discriminator*, it should therefore be optimised to produce a probability of 1 corresponding to the original *Generator*'s digit class input and a 0 for every other possibility. The *loss* of the *Generator* is given by the categorical cross-entropy between a one-hot vector corresponding to the original image class input, and the *Discriminator*'s predicted output.

The *Discriminator*'s goal is to correctly classify real handwritten digits into their respective digit, as well as to correctly identify the modified renders produced by the *Generator* as "garbage". The *loss* of the *Discriminator* is therefore given by the sum of categorical cross-entropies between:

1. A one-hot vector corresponding to the real digit class of real images, and the predicted output of real images, and:
2. A one-hot vector corresponding to the fake class, and the predicted output of images from modifier renders produced by the *Generator* and *Simulator*.

These two losses are backpropagated through the network for each image sample and the weights within the *Generator* and *Discriminator* networks are optimised according to their respective gradients over their loss function. For this project, the optimiser for gradient descent used was the Adam optimiser [96] with a learning rate between of 0.001.

Performing Training and Qualitative Assessment

Training was performed in batches of 500 MNIST images, over all 60,000 images for up to 30 to 50 epochs (repetitions of processing all 60,000 images). All training was performed on the M3 MASSIVE high-performance-computing service, and required approximately 12 minutes per epoch. The weights of the *Generator* and intermediate phosphene render results were saved after each epoch to permit early stopping and reuse of any epochs renders. Overall, the training was conducted on

over 100 different grids during development; 11 of which were randomly chosen for primary validation in the psychophysics experiment. Qualitative results of training were evaluated to guide development at multiple stages during the development process.

The trained phosphene patterns were qualitatively assessed for:

1. Variation over training epochs
2. Variation over different phosphene map resolutions
3. Variation over different phosphene map arrangements (regular and random polar and cartesian maps)
4. Variation over different digits

Chapter 7

Psychophysics Experiment

Development

A psychophysics experiment was developed in order to perform preliminary validation of the GAN prototype discussed prior. Ethics approval was obtained prior from the Monash University Human Research Ethics Committee to conduct prosthetic vision simulations in normal-sighted individuals for the development of a bionic eye (MUHREC Project Number 12525). The psychophysics experiment was coded by the author using the Python programming language and the Psychopy psychophysics software package [97], and is included in the repository listed in Appendix B.

Recruitment

11 participants (9 male, 2 female; ages 21-40) with normal or corrected-to-normal vision were recruited from students and staff at Monash University on a volunteer basis with a poster advertisement consistent with the ethics approval for this project. Participants were briefed on the purpose and conduct of the experiment and signed a consent form for participation.

Setting

Psychophysics experiments were held in a single darkened room at Monash University for all participants. The room contained a table, a chair, a computer, headphones, and a chin-rest mounted on the table. Other items, including office stationery and books, were present in the room but were not significant for the experiment and not interacted with by participants. During the experiment, the lights were switched off, the blinds were shut, the door closed and the participant instructed to silence their mobile device, leaving the computer monitor as the only major source of light. The computer monitor's brightness and contrast were adjusted at the beginning of experimentation and were the same for all participants. Participants were allowed to adjust the height of the chair, and vertical height of the chin-rest (but not the horizontal distance from the monitor) as comfortable. All participants used the same pair of headphones, sanitised between each use, and were allowed to adjust the volume of the headphones as comfortable for their hearing level.

Conduct

Prior to the psychophysics experiment commencement, a random grid was allocated to each participant. The experiment consisted of a single 60 to 90 minute session for each participant. All sessions concluded after 90 minutes whether the experiment was complete or not. At the start of the experiment, the participant was asked to sit on a chair in front of a computer screen. The participant was then asked to rest their chin in a chin rest approximately an arms length away from the computer screen. The participant was then given a pair of headphones to wear, and the volume was adjusted for the participant's comfort. The experimental trial blocks, beginning with an example block, were then commenced.

Trials and Trial Blocks

The experiment consisted of a basic digit classification task, divided into **cues** (single phosphene pattern to classify), **trials** (streams of consecutive digits to

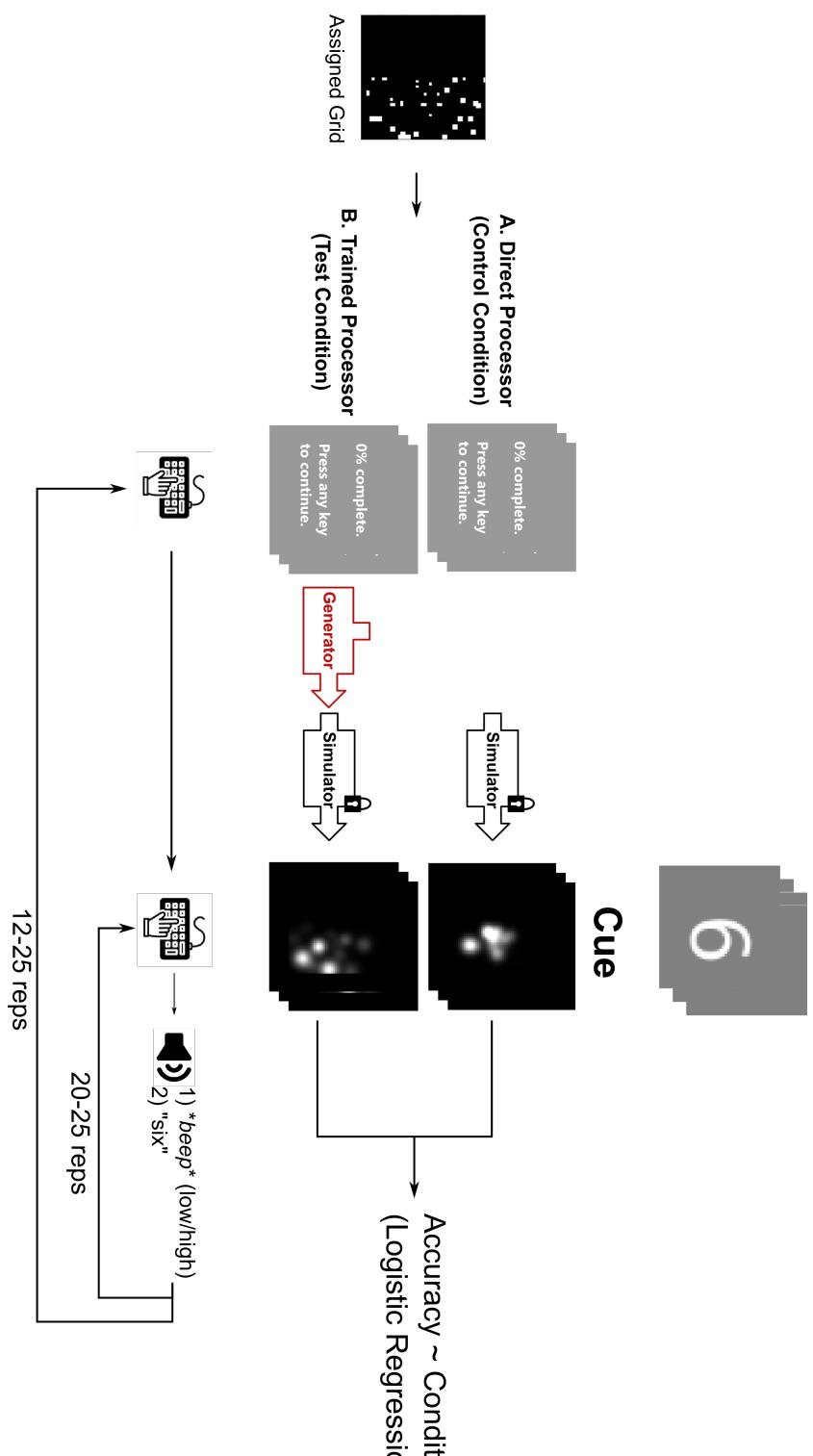


Figure 7.1: Flowchart of the conducted psychophysics experiment. Participants completed both a control block (A) and a test block (B) of digit classification trials with auditory feedback.

classify) and **trial blocks** (consecutive digits to classify with rest in between).

These are defined as:

Trial block an uninterrupted set of classification trials, interspersed with grey screens which participants were instructed they could use to take a short pause.

Trial an uninterrupted set of phosphene-represented digit *cues* which the participant was asked to identify.

Cue a single black screen with a phosphene pattern located in the central quarter of the screen.

A **condition** for this experiment consisted of either the control condition (a brightness-based mask over phosphenes, where the phosphene map is placed directly over the digits and projected through the map) or the test condition (the trained *Generator* for the grid allocated to the participant). The digit images used as the underlying basis for the controls were white Arial sans-serif digits on black backgrounds, aligned to the right border of the image as with the MNIST digits; these are included in the code repository listed in Appendix B.

Each experimental trial block consisted of 12 to 20 trials; each trial consisted of 20 to 25 phosphene representations of digits which a participant was asked to classify, and participants performed either 2 or 4 blocks depending on time constraints; exact numbers of trials are listed in Section 9. At the start of a trial, the participant would be shown a grey progress screen, instructing them to press any key to continue. At a keypress, the participant would be shown the first cue, and the program would await a digit keypress (0-9 on the number pad) from the participant. When the participant pressed a digit, they would immediately hear audio feedback simultaneously playing a tone indicating if they were correct (high) or incorrect (low), and an English voice telling them what the true digit was. The next cue would then immediately be shown, and the program would await the next digit keypress. This would repeat 20-25 times to conclude one trial. At the conclusion of a trial, another grey progress screen would be shown, and the trial process would again repeat. At the conclusion of a trial block, the participant would be given a short break before

the next trial block was begun. The condition participants started with was balanced between both conditions for both scenarios above; participants performing 4 blocks performed the two conditions interleaved.

Prior to the commencement of the trial blocks, the participant was given an example trial block to familiarise themselves with the procedure of the experiment. The example trial block consisted of an identical format to the trial blocks above, but contained only 2 trials of 10 digits each, for 1 single block.

Analysis

Results of the psychophysics experiment were first analysed with regards to the mean overall accuracy of digit recognition for each condition, both pooled across participants and within each participant.

To statistically address the second aim of this project and its corresponding hypothesis and to also incorporate learning effects over time, the classification data was modelled using a logistic regression model, including the effects of:

- Condition, the primary variable of interest.
- Cue, trial and trial block and their corresponding interaction effects, which we expected to reflect participant's ability to learn patterns over repeated trials.
- Digit class, which may reflect intrinsic differences between digits' abilities to be represented in arbitrary grids.
- Participant, which was expected to reflect inter-individual variation.

Effects with a p-value below 0.05 were considered significant.

In addition, we briefly analysed:

- Differences in duration spent classifying digits, a possible confounder, and
- Learning effects by condition, to see if the test versus control condition affected learning.

Part III

Results

Chapter 8

Training Results

We have selected a representative sample of different simulated grid properties to highlight several general features of the training results. The output here is appraised qualitatively for a broad range of grids, with statistical validation done for a small samples of low-resolution grids in Section 9. The qualitative output of all training runs is included in the code repository in Appendix B.

Variation over Epochs

Phosphene patterns initially improved with epochs, but drifted away from recognisable patterns as the number of epochs increased further. This effect permeated all runs of the training architecture for every grid, and the epoch at which digits appeared "best" varied considerably. Figure 8.1 demonstrates the evolution of the generated phosphene pattern for the digit 4 for a regular cartesian grid of 576 phosphenes. Although the phosphene pattern approaches several different depictions of a 4 at epochs 6, epoch 20 and epoch 39, it does not stabilise at a particular point but continues to attempt to reconfigure itself.

Variation over Resolution

In general, phosphene forms more closely resembled digits at higher phosphene resolutions. At the lowest resolution tested (64 phosphenes), phosphenes ceased



Figure 8.1: Phosphene patterns generated for the digit 4 for a regular cartesian grid of 576 phosphenes. The digit form approaches a recognisable digit 4 until approximately epoch 20, before the subsequently drifting away.

to resemble digits even for the most simple grid, the regular Cartesian grid. The phosphene patterns generated for a regular Cartesian grid of 64, 144 and 576 phosphenes at 2, 4 and 8 epochs for the digit 0 is shown in Figure 8.2.

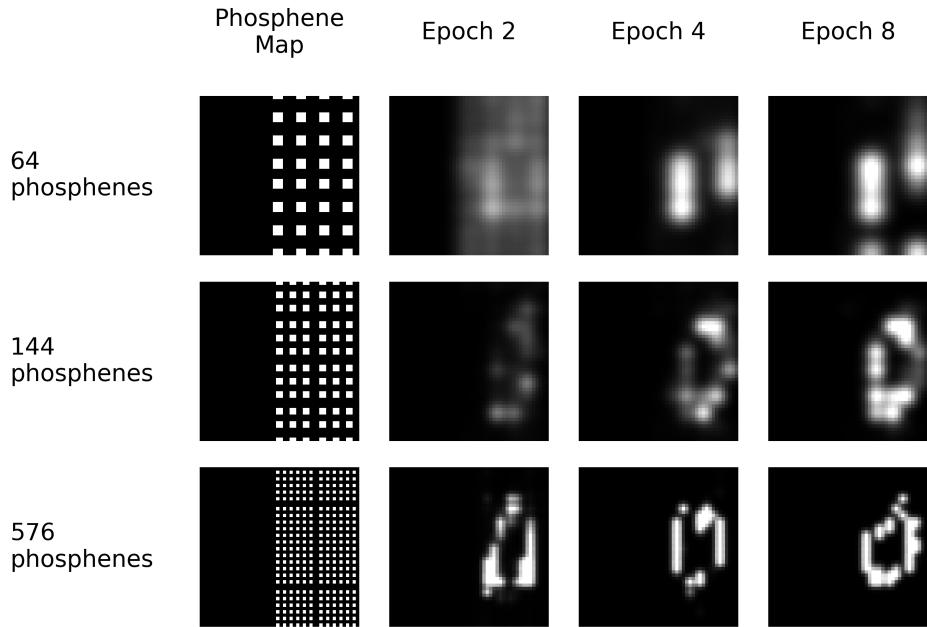


Figure 8.2: Phosphene patterns generated for different phosphene resolutions of a regular Cartesian grid for the digit 0. The first column shows the phosphene locations and relative sizes; the second, third and fourth columns show the simulated patterns at the 2nd, 4th and 8th epoch respectively.

Variation over Grid Arrangements

Phosphene patterns were successfully generated for both cartesian and polar grids with regularly and randomly spaced phosphenes. However, the appearance of these phosphene patterns were often non-optimal at high-resolutions, and non-recognisable at low-resolutions. Figure 8.3 shows the generated phosphene patterns for a grid of 576 phosphenes at 10, 20 ad 30 epochs for the digit 6. Note that the polar grids have considerably larger phosphenes in general (particularly at the peripheries), leading to greater difficulty for the training architecture in producing

any recognisable forms.

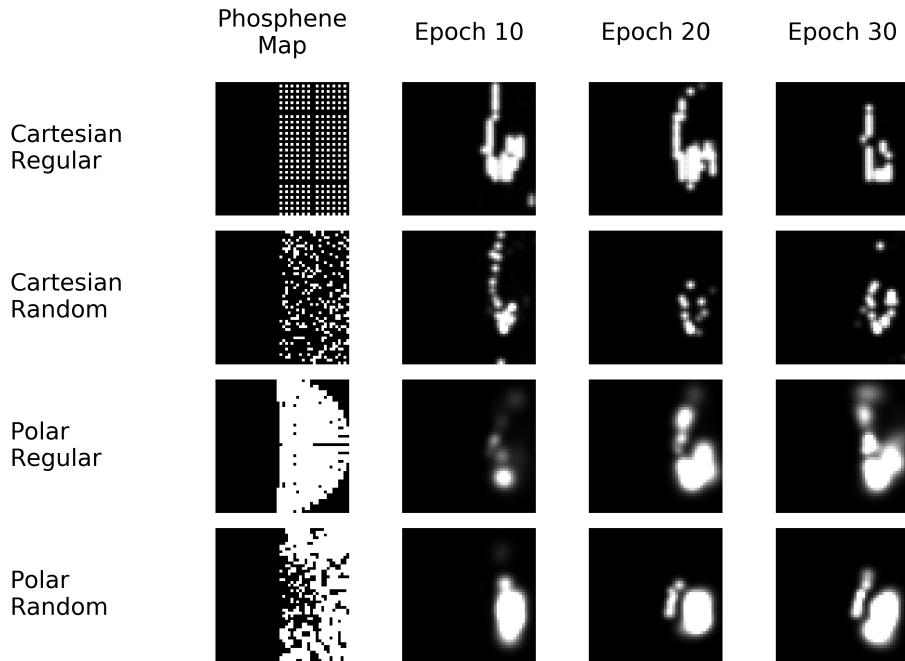


Figure 8.3: Phosphene patterns generated for the digit 6 for 4 different grid arrangements with 576 phosphenes at 10, 20 and 30 epochs with the base phosphene map on the left for comparison.

Variation over Digits

The ability to generate phosphene patterns resembling digits varied dramatically for different resolutions and arrangements of grids. Figure 8.4 shows the generated phosphene patterns at the 16th epoch for a regular cartesian grid of 576 phosphenes, the highest resolution and simplest arrangement tested. Phosphene patterns at this stage were recognisable, though there are notable imperfections such as interrupted lines.

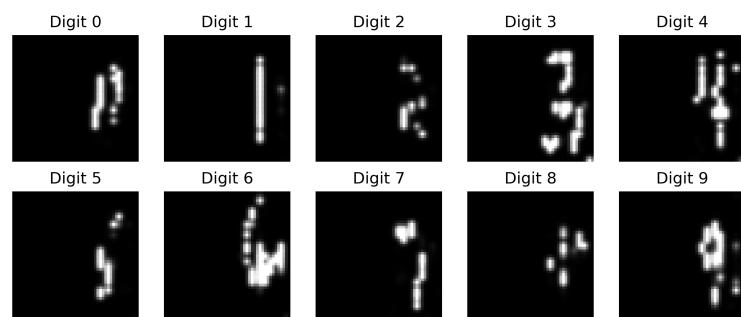


Figure 8.4: Phosphene patterns generated for each digit 0-9 for a regular cartesian grid of 576 phosphenes at the 16th epoch.

Chapter 9

Experimental Results

Grid Selection

While the most recognisable phosphene patterns were produced using regular cartesian grids at higher resolution of 576 phosphenes, these simulations are not the primary area of interest for simulated prosthetic vision where grids are irregular and expected to be much lower resolution. We therefore selected the lowest resolution grids (64 phosphenes) with randomly-arranged polar phosphenes which best reflected the current reality of CVP capabilities. Due to time constraints, we were not able to refine the GAN architecture further and often the digit patterns produced at these low resolutions did not look like digits. However, we proceeded with the experiment given that we still wished to determine whether participants found these generate patterns easier to discern than a simple control.

Participant Trial Characteristics

All participants provided at least one complete block each of digit recognition data for both the control processor and the trained encoder for their grid. The first four participants provided two trial blocks of data with each block containing 20 trials of 25 digit cues. After the first four participants, we shifted the number of digits per block into 20 trials of 20 digit cues and added two additional blocks, as we noticed a significant learning effect during intermediate analysis and wished to take extra

measures to interleave the control and test blocks. The first participant to try this arrangement was unable to complete the four blocks in the allotted time, so the number of trials per block was reduced to 12. One additional participant after this change were unable to complete the two extra blocks in the allotted time, so only the data from the first two blocks was used as with the first five participants.

The characteristics of the trials for each participant are summarised in Table 9.1.

Table 9.1: Characteristics of trials for each participant. Control indicates the direct masking processor, and Test indicates the trained phosphene patterns using the GAN architecture.

Participant	Trials x Cues per Block	Block 1	Block 2	Block 3	Block 4	Total Cues
01	20 trials x 25 cues	Control	Test			500
02	20 trials x 25 cues	Test	Control			500
03	20 trials x 25 cues	Control	Test			500
04	20 trials x 25 cues	Test	Control			500
05	20 trials x 20 cues	Control	Test			400
06	12 trials x 20 cues	Test	Control	Test	Control	480
07	12 trials x 20 cues	Control	Test	Control	Test	480
08	12 trials x 20 cues	Test	Control			240
09	12 trials x 20 cues	Control	Test	Control	Test	480
10	12 trials x 20 cues	Test	Control	Test	Control	480
11	12 trials x 20 cues	Control	Test	Control	Test	480

Overall Accuracy

Pooled

The mean digit recognition accuracy of the test group was higher on average (75.7%, standard error of mean 3.75%) than the mean accuracy of the control group (59.8%,

standard error of mean 4.17%). Figure 9.1 compares the mean digit recognition accuracy for each condition over the pooled participant data.

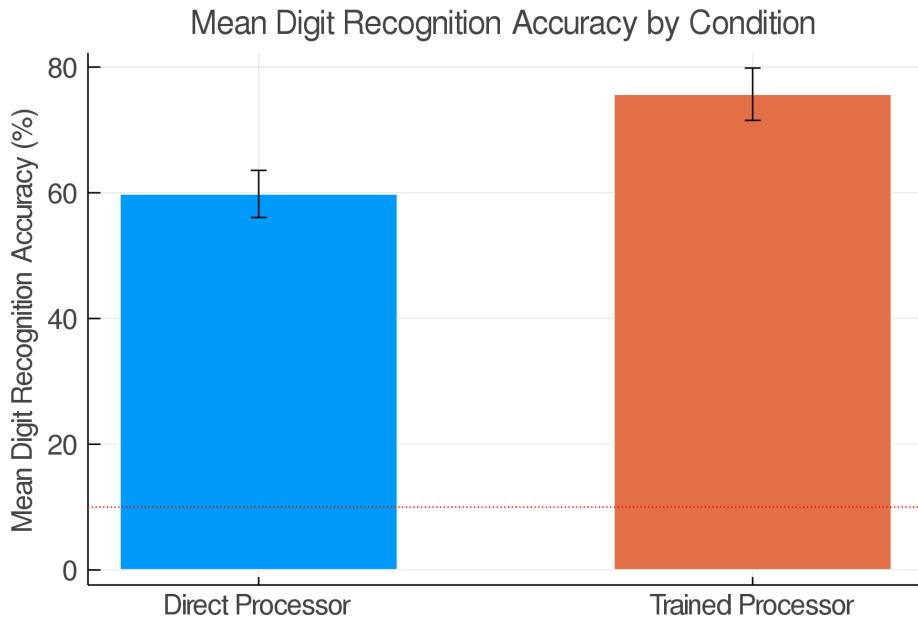


Figure 9.1: Mean digit recognition accuracy by condition, pooled participant data. Error bars show standard error of means.

By Participant

Of 11 participants, 9 participants performed better with the trained processor than the control processor despite variations between participants of baseline performance. Figure 9.2 compares the mean digit recognition accuracy for each condition, stratified by participant.

Statistical Effects

The correctness of participant's digit classification was modelled using logistic regression. These results are summarised in Table 9.2. The test condition significantly improved classification accuracy in this linear regression model ($p < 0.001$); trial ($p < 0.001$) and block ($p < 0.001$) and its interaction effect ($p < 0.05$) also significantly improved performance. There was significant inter-individual variation in baseline

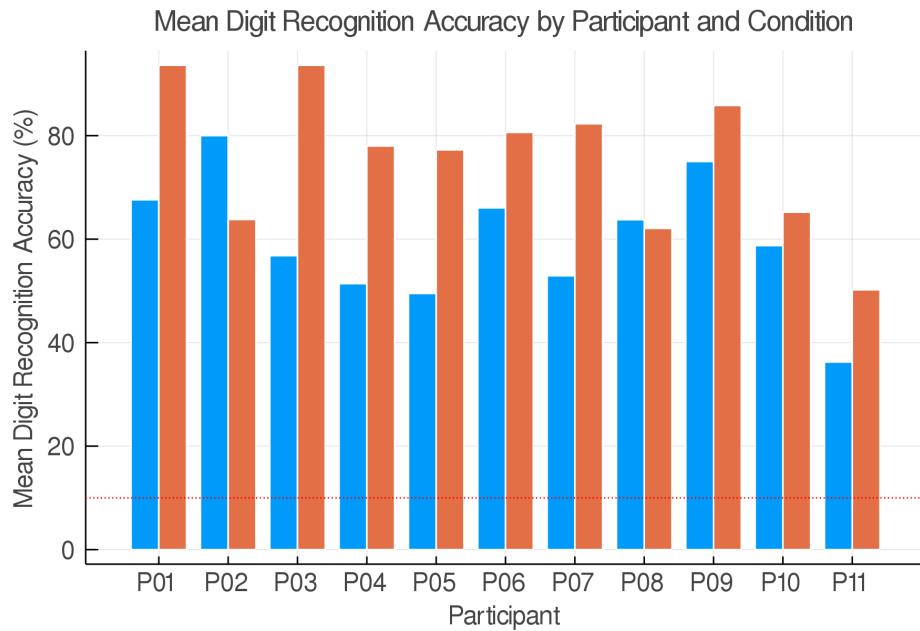


Figure 9.2: Mean digit recognition accuracy by participant and condition.

performance ($p < 0.001$). Several digits were significantly associated with higher classification accuracy including digits 1, 4 and 7 ($p < 0.001$).

Figure 9.3 plots the same data on a scatter plot.

Response Time

The mean response time for a single digit was 2.6 (SEM 0.32 seconds) seconds for the trained processor, versus 3.3 seconds for the trained processor. Figure 9.4 shows the response time per cue for each participant.

Learning Effects

Figure 9.5 shows the a rolling average with a windows size of 50 over pooled trial blocks for each condition and participant. Participants performed with similar accuracies at the start of trial blocks, and both condition displayed a prominent learning effect.

Table 9.2: Predictors of a correct digit classification modelled as a logistic regression.
P-values below 0.05, 0.01 and 0.001 are marked with (*), (**) and (***) respectively.

Predictor	Coefficient	P-Value	Significance
condition	0.959624	<1e-77	***
cue	0.0184719	0.2237	
trial	0.176255	<1e-12	***
block	0.758783	<1e-17	***
cue & trial	0.00174088	0.3815	
cue & block	0.00346854	0.6395	
trial & block	0.0140752	0.2867	
cue & trial & block	-0.00226474	0.0400	*
participant: P02	-0.746689	<1e-9	***
participant: P03	-0.391573	0.0016	**
participant: P04	-1.15042	<1e-21	***
participant: P05	-1.1289	<1e-18	***
participant: P06	-0.682311	<1e-7	***
participant: P07	-0.914266	<1e-12	***
participant: P08	-0.632393	<1e-5	***
participant: P09	-0.0339482	0.7966	
participant: P10	-1.32654	<1e-26	***
participant: P11	-2.32435	<1e-74	***
digit: 1	1.7064	<1e-40	***
digit: 2	0.233331	0.0300	*
digit: 3	0.314268	0.0035	*
digit: 4	0.645442	<1e-8	***
digit: 5	0.105155	0.3319	
digit: 6	-0.0685792	0.5264	
digit: 7	1.02026	<1e-18	***
digit: 8	0.275374	0.0120	*
digit: 9	0.065778	0.5451	
(Intercept)	-2.08177	<1e-20	***

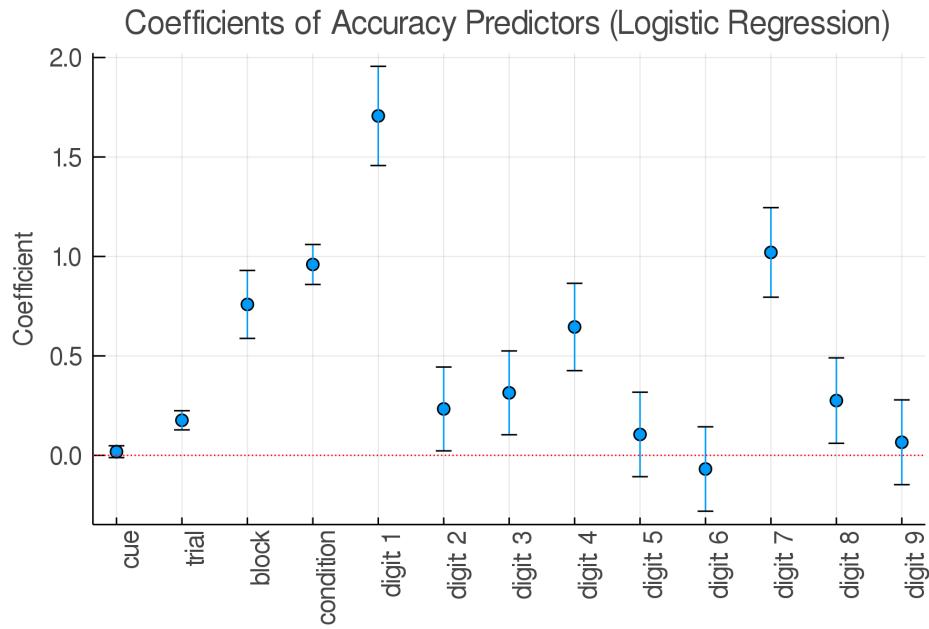


Figure 9.3: Predictors of a correct response in a logistic regression model. Error bars indicate 95% confidence intervals.

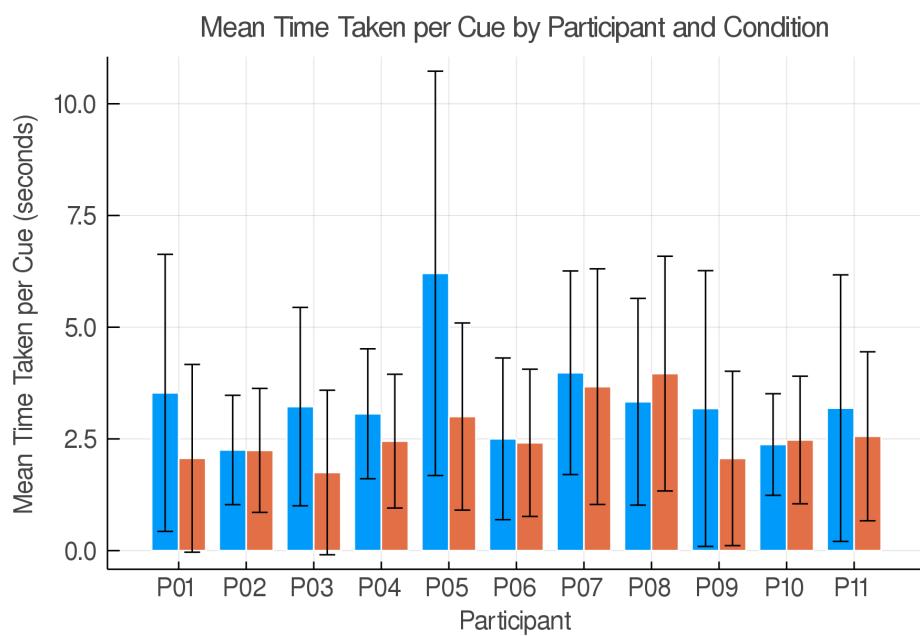


Figure 9.4: Response time per cue by participant. Error bars show standard deviation.

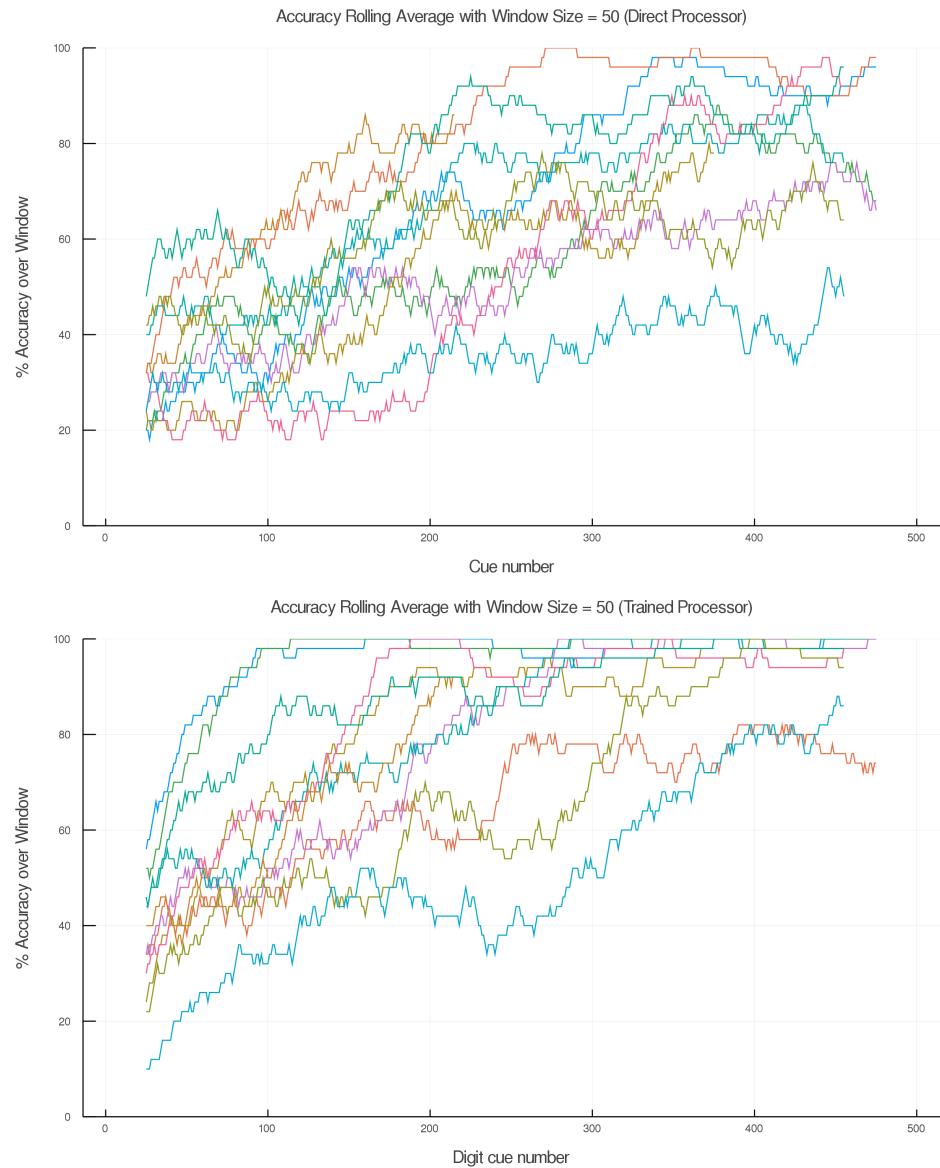


Figure 9.5: Rolling average of accuracy over pooled trial blocks by condition and participant.

Part IV

Discussion

Chapter 10

Aims

The GAN training architecture described in Section 6 was tested on a large variety of phosphene grids varying in phosphene location, sizing, regularity and resolution. We were successful in applying the training process to different types of phosphene grids, which varied primarily in spatial distribution, phosphene size and phosphene resolution.

A prototype GAN training architecture was successfully developed and was capable of producing phosphene encodings for different simulated grid properties. We have approached this by implementing independent Rendering and Modification steps inside the GAN training architecture, and by allowing the Renderer be supplied different simulated phosphenes for each electrode. In this respect, the implementation presented here serves as a proof-of-concept that a generalisable training framework is indeed possible.

Unfortunately, the output of this GAN training architecture does not always resemble their intended forms (digits). Because this difficulty was present even in a regular Cartesian grid (which approximates pixels at higher resolutions), this is likely due to more than just intrinsic difficulties in representing digits in a grid. Clearly, there is still a significant portion of work required before this training architecture can be considered usable for its intended purpose.

The biggest issue confronting the GAN training architecture was stability; phosphene forms did not converge over epochs, as demonstrated by Figure 8.1, and indeed drifted. While we made our best effort to align the domains of generated

images with real images by using a modifier and preprocessing techniques, this was clearly not enough. It is not clear whether this limitation will severely impact on the ability of GANs to be applied to simulated prosthetic vision, as there may be other methods of combatting issue; however we were not able to do so.

At low resolutions, it was also clear that phosphene patterns were highly unrepresentable on some grids, due to the inherent distribution of phosphenes on the grid themselves. This may be telling that GANs are not suitable to the problem of low-resolution phosphene grids, but may still be useful to tailor phosphene patterns for higher-resolution (but distorted) grids.

Figure 8.3 display roughly equivalent forms over the different types of phosphene grids; this is encouraging, though we note that these did not exactly resemble digits. This effect was notable for all qualitative results - while forms looked similar across grids, these appeared to exhibit merely general features of digits (such as the circle of a six with a vertical line) rather than holistically combining into digits themselves. This is a known problem with GANs [87], which may become too focused on minutiae and lose larger forms. Figure 8.4 demonstrated that some digits were additionally much harder to represent (such as an 8), even in a simple Cartesian grid. This is likely due to the fineness detail of the training data after preprocessing, and could possibly be mitigated by further thickening lines in the training data samples.

Due to the time limitations of this project, we proceeded with the psychophysics evaluation despite not achieving particularly "good-looking" digits. Although the phosphene forms did not look like digits, it was still possible that people might find them easier to learn than the control.

Participants achieved a greater overall digit classification accuracy with the trained processor versus the brightness-based control. This effect was statistically significant when analysed using a multiple logistic regression, supporting the hypothesis that the trained processor confers better digit classification accuracy than the brightness-based control.

Notably, participant performance at the beginning of each trial block was similar between the trained and control processor. This suggests that the trained processor

phosphene forms looked no more like digits than the control processor.

It therefore appears that the better participant performance was due to a better ability to learn and distinguish the phosphene forms, rather than the the ability to find the phosphene forms immediately recognisable. We did not statistically assess the learning effect of condition as the qualitative results displayed in Figure 9.5 were sufficient information for guiding its use as a preliminary validation measure. We emphasise that the primary contribution of this project lies in Aim 1 - the development phase - with the statistical results merely hoping to guide further development in the future.

Chapter 11

Implications

To our knowledge, this is the first attempt at tailoring phosphene patterns to specific grids through an automated process. Although previous research has demonstrated how image processing techniques can be applied to simulated prosthetic vision, these techniques are often coupled with the simulations they run in. This makes it hard to generalise how such techniques can be applied when the simulations strongly differ from reality. This project, by using a dynamic simulator capable of representing arbitrary arrangements and properties of phosphenes, is therefore novel in investigating an approach to image processing decoupled from the simulation itself.

The approach we have taken in this project is an early implementation using GANs. We have demonstrated that GANs are capable of producing digit-like forms with "high" resolution cartesian grids, though with imperfections. This closely reflects the domain in which GANs are known to excel; where GANs have close to pixel-level access on generated images, and can optimise at a pixel-wise level.

However, we were not successful in producing reasonable digit-like forms at "low" resolution grids. The GAN training architecture was able to produce a phosphene pattern output, but these did not resemble digits. This is outside the domain for which GANs have previously been applied - when there is greater distance between the output of generated encodings and the simulated render. Because GANs have not been applied in this area before, it was unknown how they would cope in this type of task. This project therefore provides some information on the potential

issues with using GANs for an encoding task with great distance between generator and renderer, as well as a basis for suggestions addressing these issues.

A prominent issue arising from this project is how to evaluate how intrinsically capable different grids are for representing recognisable digits. It is not altogether clear whether the issues with representing digits arose due to limitations of the GAN architecture, or the impossibility of representing digits for some types of grids in the first place. A contrived example is where a grid is composed of phosphenes arranged entirely in a single vertical line; it would be impossible to make any recognisable digit from this grid except for a digit 1. This is especially a problem with low-resolution grids with other distortions (such as size variance), where the contribution and valubility of each phosphene is relatively high. This problem - the lack of a means of evaluating the informative capacity of different phosphene grids - has not previously been addressed.

Nevertheless, we have demonstrated that computer-generated phosphene forms may still be useful to improve the discriminability of phosphene patterns, even though they do not look like digits. The results of our psychophysics experiment, which we undertook at low-resolution, irregular and distorted grids, demonstrate that although people did not recognise the forms as digits, they were able to learn their meanings better than a direct mask-based control. There is no doubt that manually-crafted phosphene patterns would have easily surpassed the performance of our GAN implementation as it currently stands. However, the usefulness of this GAN implementation is that it is an automated process, and not exclusive to digit forms (provided there is training data of other forms which you wish to represent). There is therefore an advantage to refining automated processes of generating phosphene forms rather than relying on the ability of humans to manually craft phosphene patterns representing digits.

The current results of this project are not compelling for, nor against, the use of GANs in addressing the perceptual limitations of prosthetic vision. We have demonstrated how a GAN can be implemented to perform a phosphene generation task for different simulated grids, and how this performance varies with different simulated properties. The GAN was not robust against all different simulated

properties.

Chapter 12

Limitations

Simulated Prosthetic Vision

One of the criticisms highlighted in the Background section was the difficulty of dealing with differences between simulated prosthetic vision and prosthetic vision *in vivo*. While this project addresses one of these differences - coping with different *spatial* phosphene properties - many differences remain for which the translation of techniques from simulated space to *in vivo* space is unclear. These include:

1. **Temporal qualities of phosphenes, such as flickering and dimming over time.** This is important in understanding how vision can be provided to implantees continuously like natural vision, and how temporal properties could be used to improve implantees' visual understanding. Our training implementation does not currently allow for producing or testing anything more than static patterns. Notably however, static patterns are not necessarily a poor choice since continuously-changing phosphene patterns might provoke confusion in implantees, particularly in low-resolution implants as is currently expected.
2. **Effects of simultaneously eliciting phosphenes.** In this experiment, simultaneously stimulated phosphenes were shown simultaneously for simplicity; however, empirical evidence suggests that simultaneously elicited phosphenes may not combine additively. This aspect of the simulation could

theoretically be incorporated in our GAN training architecture, but was not trialled in this experiment.

3. The movement of phosphenes with eye movements. Phosphenes are known to move with eye movements, meaning that peripheral phosphenes would always be perceived peripherally. Our psychophysics experiment did not incorporate eye movement tracking as it was intended purely as a preliminary validation measure rather than a true test of applicability. Participants were therefore able to scan and study phosphene patterns, which may not be possible *in vivo*. This feature could, however, be modelled in a future experiment with greater resources and refinement of phosphene patterns as a stronger test of applicability.

The results presented here should therefore still be considered within the context of simulated prosthetic vision and all its caveats, as it primarily focused on addressing one particular issue of simulated prosthetic vision rather than all.

Experimental Features

The sample size of the psychophysics experiment used in this project was notably small with 11 participants only. The rationale for this sample size was that it served as an intermediate checkpoint to point-out weaknesses in the current architecture and guide further development. The investment required for recruiting and conducting the experiment on more participants was not justified by the status of the GAN implementation as an early prototype. However, should the prototype be further developed, it is important to validate in a greater number of participants.

Choice of Task

The digit recognition task in this experiment was chosen to optimise data collection so as many data samples could be collected as possible from each participant. The task - recognising a single digit known to belong to the class of 0-9 - is therefore extremely simple and somewhat contrived.

The use of a simple, constrained task was necessary for this project as it investigated a single feature - the discriminability of phosphene patterns. At the current stage of implementation, it is not as informative for development to use "natural" tasks - such as navigation, reading or unconstrained object identification - which often incorporate multiple aspects of visual information as well as other senses. However, the reality is that natural visual tasks are of greatest interest to future implantees. Whilst in early development, the use of simple, constrained tasks is crucial to directly highlight strengths and weaknesses of image processing techniques, these must eventually be tested in more natural environments. This project is not yet at that stage.

Chapter 13

Future Directions

Given that this is a prototypical implementation, it is important that the results of this work can be used to improve research on the use of GANs in simulated prosthetic vision even though the early results are not conclusive.

Training Data

We used MNIST digits as the source of real digits for our training implementation and modified them slightly to suit the visual properties of our simulation. We also attempted to bring the renders closer to the domain of MNIST digits using a style-transfer network to "fill holes" in the phosphene and extract forms.

However it does not appear our measures were enough for the network to remain stable. It was evident from the training loss curves that the *Generator* was never able to fool the *Discriminator* reliably. As a result, the *Generator*'s produced encodings often drifted rather than converging to an optimal solution.

An easy improvement to make (and one which does not require changing the actual training architecture itself) is to use digit forms closer to the image domain of simulated renders as training data. This could be sourced by creating

In addition, training data representing other forms could be used instead, such as outlines of common objects (such as chairs) or letterforms. Using different forms of training data does not require any major changes to the architecture, as the training data is not part of the architecture itself (and therefore can simply be

"mixed and matched"). A potential source of outline data is the Google Quick, Draw! Dataset, which contains millions of simple, line-based drawings of objects and animals. As the ability to substitute different forms of training data is one of the benefits of using GANs, this could potentially provide further information on whether GANs are justified for simulated prosthetic vision.

Grid Evaluation

This project faced major difficulties in creating recognisable phosphene patterns for low-resolution distorted grids. Part of this difficulty is due to intrinsic limitations on *what* can be represented on grids with a small quantity of phosphenes. Certain types of grid, such as a single linear "grid" of phosphenes, clearly constrain the types of shapes that can be represented.

It would be informative and time-saving to be able to evaluate grids beforehand to quantify their usefulness for representing specific types of phosphene patterns. This could be as simple as a measure of dispersion of phosphenes in 2D space for simple grid simulations. There is scant literature on the use of image processing techniques for grid resolutions as low as those tested in this project (64 phosphenes), which may partially be why this has not previously been an issue. However, given that first-generation CVPs are only designed to produce up to a maximum of 473 phosphenes (with most implants far below this number), being able to quantify phosphene grids' representability would be useful for early *in vivo* experiments.

Chapter 14

Conclusion

This project has explored how generative adversarial networks (GANs) may be used to generate phosphene patterns for arbitrary simulated properties of prosthetic vision, and to our knowledge is the first attempt at addressing phosphene pattern generation decoupled from simulation properties. We developed a prototypical software implementation of a GAN training architecture to generate phosphene patterns of digits, and tested whether these helped people recognise digits in a psychophysics experiment compared to a mask-based control. The early training implementation we developed was successfully able to produce phosphene digit patterns for different grid properties including phosphene distribution, sizing and resolution. However, the produced phosphene digit patterns were only recognisable as digits at high grid resolutions and not at low grid resolutions, and the generated patterns proved unstable over epochs. Nevertheless, at low resolution, participants were still able to learn the generated phosphene patterns better than the mask-based control and achieved a higher recognition accuracy overall. The results of this checkpoint suggest that the current GAN implementation may be helpful in increasing the discriminability of phosphene patterns for a simple classification task, but are not yet robust against low-resolution grids and spatial distortions. This project provides useful information on potential targets for improving a GAN implementation; further work should focus on improving the stability of GAN training and quantifying intrinsic capabilities of low-resolution grids to represent form-based information.

Part V

Appendices

Appendix A

Selected Package Dependencies

A selected number of direct dependencies of the code implementation of this project and their version numbers are listed below. A full listing of dependent packages, including indirect dependencies, is available from the file `requirements.txt` in the linked GitHub repository.

Table A.1: Selected package dependencies for the code written in this project.

Package Dependency	Version Number
h5py	2.9.0
imageio	2.5.0
imageio-ffmpeg	0.3.0
Keras-Aplications	1.0.8
Keras-Preprocessing	1.1.0
matplotlib	3.1.0
numpy	1.16.4
numpydoc	0.9.1
opencv-python	4.1.0.25
pandas	0.25.1
Pillow	6.1.0
PsychoPy	3.1.5
pygame	1.9.6
pyglet	1.3.0
Pygments	2.4.2
pypiwin32	223
PyQt5	5.9
PyQt5-sip	4.19.18
pywin32	224
pyWinhook	1.6.1
pywintypes	0.5.5
QtAwesome	0.5.7
qtconsole	4.5.1
QtPy	1.8.0
scikit-image	0.15.0
scikit-learn	0.20.3
scipy	1.3.1
tb-nightly	1.15.0a20190806
tblib	1.4.0
tensorflow	2.0.0rc0
tf-estimator-nightly	1.14.0.dev2019080601
wxPython	4.0.6

Appendix B

Selected Code

The entire codebase is available from <https://github.com/wjmn/bmedsc-exploratory>.

This codebase includes all code written during the course of this project, including:

- Prototypes
- Exploratory data analysis
- The generative adversarial network architecture
- Psychophysics experiments
- Data analysis
- Plotting code

As the codebase is too large to include in hardcopy, only a few small, selected snippets are reproduced here where relevant.

Models

Each of the below shows the code implementation of each of the neural networks involved in the GAN training architecture.

Generator

```
from tensorflow.keras import Sequential, Model  
from tensorflow.keras.layers import *
```

```
# For a phosphene map with 64 phosphenes
NUM_PHOSPHENES = 64

def generator_basic():
    input_labels = Input(shape=(1,))
    layer_labels = Embedding(10, 10)(input_labels)
    layer_labels = Flatten()(layer_labels)
    layer_labels = BatchNormalization()(layer_labels)
    layer_labels = LeakyReLU()(layer_labels)
    layer_labels = Dense(NUM_PHOSPHENES, activation=tf.nn.sigmoid)(layer_labels)
    return Model(input_labels, layer_labels)
```

Modifier

```
from tensorflow.keras import Sequential, Model
from tensorflow.keras.layers import *

def modifier():
    in_image = Input(shape=(renderSize1, renderSize2))

    l_image = Reshape((renderSize1, renderSize2, 1))(in_image)
    l_image = Conv2D(48, (7,7), strides=(1,1), padding='same')(l_image)
    l_image = BatchNormalization()(l_image)
    l_image = LeakyReLU()(l_image)

    l_image = Conv2D(96, (3,3), strides=(2,2), padding='same')(l_image)
    l_image = BatchNormalization()(l_image)
    l_image = LeakyReLU()(l_image)

    l_image = Conv2D(128, (3,3), strides=(1,1), padding='same')(l_image)
    l_image = BatchNormalization()(l_image)
    l_image = LeakyReLU()(l_image)

    l_image = Conv2D(128, (3,3), strides=(1,1), padding='same')(l_image)
    l_image = BatchNormalization()(l_image)
    l_image = LeakyReLU()(l_image)

    l_image = Conv2DTranspose(48, (3,3), strides=(2,2), padding='same')(l_image)
    l_image = BatchNormalization()(l_image)
    l_image = LeakyReLU()(l_image)

    l_image = Conv2DTranspose(1, (3,3), strides=(1,1), padding='same', activation="tanh")(l_image)
    l_image = Reshape((renderSize1, renderSize2))(l_image)

    model = Model([in_image], l_image)

    return model
```

Discriminator

```
from tensorflow.keras import Sequential, Model
```

```

from tensorflow.keras.layers import *

X_RENDER = 48
Y_RENDER = 48

def discriminator_basic():
    input = Input(shape=(Y_RENDER, X_RENDER))
    layer = Reshape((Y_RENDER, X_RENDER, 1))(input)
    layer = Conv2D(64, (8,8), padding='same', strides=(2,2))(layer)
    layer = LeakyReLU()(layer)
    layer = Dropout(0.25)(layer)
    layer = Conv2D(128, (4,4), padding='same', strides=(2,2))(layer)
    layer = LeakyReLU()(layer)
    layer = Dropout(0.25)(layer)
    layer = Flatten()(layer)
    layer = Dense(11, activation=tf.nn.softmax)(layer)
    return Model(inputs=input, outputs=layer)

```

Grids

Each of the below shows the code implementation of the Cartesian and Polar phosphene map generators; these inherit from an Abstract base class implementation included in the GitHub repository as it is too long to reproduce here..

Cartesian Grid

```

class CartesianGrid(AbstractGrid):

    def __init__(
        self,
        x_phosphenes: int,
        y_phosphenes: int,
        x_render: int,
        y_render: int,
        half: bool = False,
        random: bool = False
    ) -> None:
        """
        Create a cartesian grid of phosphenes (regularly sized).

        :param x_phosphenes: the number of phosphenes in the x dimension.
        :param y_phosphenes: the number of phosphenes in the y dimension.
        :param x_render: the number of pixels in the x dimension of the render.
        :param y_render: the number of pixels in the y dimension of the render.
        :param half: whether to only fill the right-half of the render (default False).
        :param random: Whether to randomise the locations of phosphenes (default False).
        """

```

```

        self._set_id()
        self.grid_type = GridType.CARTESIAN
        self.num_phosphenes = x_phosphenes * y_phosphenes
        self.render_shape = (x_render, y_render)
        self.half = half
        self.random = random

        if random:
            self.locations = np.random.rand(x_phosphenes * y_phosphenes, 2)
        else:
            self.locations = np.array([
                (x / (x_phosphenes - 1), y / (y_phosphenes - 1))
                for x in range(x_phosphenes)
                for y in range(y_phosphenes)
            ])

        if half:
            self.locations[:, 0] = self.locations[:, 0] + 1 / 2

        self.sizes = np.array([
            ((x_render // x_phosphenes) // 2,
             (y_render // y_phosphenes) // 2)
            for x in range(x_phosphenes)
            for y in range(y_phosphenes)
        ])

        self.strengths = np.array([
            1
            for x in range(x_phosphenes)
            for y in range(y_phosphenes)
        ])

        self.volume = self.prerender_volume(
            x_render,
            y_render,
        )
    
```

Polar Grid

```

class PolarGrid(AbstractGrid):

    def __init__(
        self,
        n_theta: int,
        n_radius: int,
        x_render: int,
        y_render: int,
        half: bool = False,
        random: bool = False,
    ) -> None:
    
```

```

""" Create a polar grid of phosphenes (size varies with eccentricity).

:param n_theta: number of angles in the grid.
:param n_radius: number of radii in the grid.
:param x_render: the number of pixels in the x dimension of the render.
:param y_render: the number of pixels in the y dimension of the render.
:param half: whether to fill the right-half of the render (default False)
:param random: whether to randomise the locations of phosphenes (default False).
"""

self._set_id()
self.grid_type = GridType.POLAR
self.num_phosphenes = n_theta * n_radius
self.render_shape = (x_render, y_render)
self.half = half
self.random = random

if random:
    self.locations = np.random.rand( n_theta * n_radius, 2 )
    if half:
        self.locations[:, 0] = self.locations[:, 0] + 1 / 2
else:
    self.locations = np.array([
        (0.5 + (i_radius / n_radius * np.cos(self.i_angle(i_theta, n_theta, half))) / 2,
         0.5 + (i_radius / n_radius * np.sin(self.i_angle(i_theta, n_theta, half))) / 2,)
        for i_radius in range(1, n_radius + 1)
        for i_theta in range(n_theta)
    ])

k = x_render / 2 + y_render / 2
a = e * (x_render + y_render) / 32

self.sizes = np.array([
    (np.log(k * ((x - 0.5) ** 2 + (y - 0.5) ** 2) + a),
     np.log(k * ((x - 0.5) ** 2 + (y - 0.5) ** 2) + a))
    for (x, y) in self.locations
])

self.strengths = np.array([
    1
    for _ in range(n_theta)
    for _ in range(n_radius)
])

self.volume = self.prerender_volume(
    x_render,
    y_render,
)

def i_angle(self, i_theta: float, n_theta: float, half: bool = False) -> float:
    """ Calculates the angle for angle of index i in range n.
    """
    if half:

```

```
    angle = (np.pi / (n_theta - 1) * i_theta) - (np.pi / 2)
else:
    angle = 2 * np.pi / n_theta * i_theta
return angle
```

Appendix C

Supplementary Qualitative Results

Below are several randomly selected additional qualitative training results; qualitative results from all training runs can be found in the code repository listed in Appendix B.



Figure C.1: Supplemental qualitative results of phosphene pattern outputs for the trained architecture, randomly sampled; patterns are shown in rows of 10, each corresponding to the output of digits 0 - 9 respectively.

Part VI

References

Bibliography

- [1] Bourne RRA, Flaxman SR, Braithwaite T, Cicinelli MV, Das A, Jonas JB, et al. Magnitude, temporal trends, and projections of the global prevalence of blindness and distance and near vision impairment: a systematic review and meta-analysis. *The Lancet Global Health*. 2017 Sep;5(9):e888–e897. Available from: [https://www.thelancet.com/journals/langlo/article/PIIS2214-109X\(17\)30293-0/abstract](https://www.thelancet.com/journals/langlo/article/PIIS2214-109X(17)30293-0/abstract).
- [2] Flaxman SR, Bourne RRA, Resnikoff S, Ackland P, Braithwaite T, Cicinelli MV, et al. Global causes of blindness and distance vision impairment 1990–2020: a systematic review and meta-analysis. *The Lancet Global Health*. 2017 Dec;5(12):e1221–e1234. Available from: <https://linkinghub.elsevier.com/retrieve/pii/S2214109X17303935>.
- [3] Lee DA, Higginbotham EJ. Glaucoma and its treatment: A review. *American Journal of Health-System Pharmacy*. 2005 Apr;62(7):691–699. Available from: <https://academic.oup.com/ajhp/article/62/7/691/5134357>.
- [4] Zachariades N, Papavassiliou D, Christopoulos P. Blindness after facial trauma. *Oral Surgery, Oral Medicine, Oral Pathology, Oral Radiology, and Endodontology*. 1996 Jan;81(1):34–37. Available from: <http://www.sciencedirect.com/science/article/pii/S1079210496801442>.
- [5] Normann RA, Greger BA, House P, Romero SF, Pelayo F, Fernandez E. Toward the development of a cortically based visual neuroprosthesis. *Journal of Neural Engineering*. 2009 May;6(3):035001. Available from: <https://doi.org/10.1088%2F1741-2560%2F6%2F3%2F035001>.
- [6] Lewis PM, Ackland HM, Lowery AJ, Rosenfeld JV. Restoration of vision in

- blind individuals using bionic devices: A review with a focus on cortical visual prostheses. *Brain Research*. 2015 Jan;1595:51–73. Available from: <http://www.sciencedirect.com/science/article/pii/S0006899314015674>.
- [7] Foroushani AN, Pack CC, Sawan M. Cortical visual prostheses: from microstimulation to functional percept. *Journal of Neural Engineering*. 2018 Feb;15(2):021005. Available from: <https://doi.org/10.1088%2F1741-2552%2Faaa904>.
- [8] Brindley GS, Lewin WS. The sensations produced by electrical stimulation of the visual cortex. *The Journal of Physiology*. 1968 May;196(2):479–493. Available from: <http://doi.wiley.com/10.1113/jphysiol.1968.sp008519>.
- [9] Dobelle WH, Mladejovsky MG. Phosphenes produced by electrical stimulation of human occipital cortex, and their application to the development of a prosthesis for the blind. *The Journal of Physiology*. 1974 Dec;243(2):553–576. Available from: <http://doi.wiley.com/10.1113/jphysiol.1974.sp010766>.
- [10] Bak M, Girvin J, Hambrecht DFT, Kufta CV, Loeb GE, Schmidt EM. Visual sensations produced by intracortical microstimulation of the human occipital cortex. *Medical and Biological Engineering and Computing*. 1990;28:257–259.
- [11] Bosking WH, Beauchamp MS, Yoshor D. Electrical Stimulation of Visual Cortex: Relevance for the Development of Visual Cortical Prosthetics. *Annual Review of Vision Science*. 2017 Sep;3(1):141–166. Available from: <http://www.annualreviews.org/doi/10.1146/annurev-vision-111815-114525>.
- [12] Rushton DN, Brindley GS. Properties of Cortical Electrical Phosphenes. In: Cool SJ, Smith EL, editors. *Frontiers in Visual Science*. Springer Series in Optical Sciences. Springer Berlin Heidelberg; 1978. p. 574–593.
- [13] Schmidt EM, Bak MJ, Hambrecht FT, Kufta CV, O'Rourke DK, Vallabhanath P. Feasibility of a visual prosthesis for the blind based on intracortical micro stimulation of the visual cortex. *Brain*. 1996 Apr;119(2):507–522. Available from: <https://academic.oup.com/brain/article/119/2/507/382434>.

- [14] Fernández E, Pelayo F, Romero S, Bongard M, Marin C, Alfaro A, et al. Development of a cortical visual neuroprosthesis for the blind: the relevance of neuroplasticity. *Journal of Neural Engineering*. 2005 Nov;2(4):R1–R12. Available from: <https://doi.org/10.1088%2F1741-2560%2F2%2F4%2Fr01>.
- [15] Beyeler M, Rokem A, Boynton GM, Fine I. Learning to see again: biological constraints on cortical plasticity and the implications for sight restoration technologies. *bioRxiv*. 2017 Mar;p. 115188. Available from: <https://www.biorxiv.org/content/10.1101/115188v1>.
- [16] Barnes N. The role of computer vision in prosthetic vision. *Image and Vision Computing*. 2012 Aug;30(8):478–479. Available from: <http://www.sciencedirect.com/science/article/pii/S0262885612000832>.
- [17] Buffoni LX, Coulombe J, Sawan M. Image Processing Strategies Dedicated to Visual Cortical Stimulators: A Survey. *Artificial Organs*. 2005;29(8):658–664. Available from: <http://onlinelibrary.wiley.com/doi/abs/10.1111/j.1525-1594.2005.29104.x>.
- [18] Chang MH, Kim HS, Shin JH, Park KS. Facial identification in very low-resolution images simulating prosthetic vision. *Journal of Neural Engineering*. 2012 Aug;9(4):046012. Available from: <http://stacks.iop.org/1741-2552/9/i=4/a=046012?key=crossref.82f47f235753f422323435341cb2f4b0>.
- [19] Sharmili N, Swapna N, Ramakrishna G. Comparative analysis of image processing algorithms for visual prosthesis. In: 2017 International Conference on Communication and Signal Processing (ICCSP); 2017. p. 1120–1124.
- [20] Guo Y, Liu Y, Oerlemans A, Lao S, Wu S, Lew MS. Deep learning for visual understanding: A review. *Neurocomputing*. 2016 Apr;187:27–48. Available from: <http://www.sciencedirect.com/science/article/pii/S0925231215017634>.
- [21] Gatys LA, Ecker AS, Bethge M. Image Style Transfer Using Convolutional Neural Networks. In: 2016 IEEE Conference on Computer Vision and Pat-

- tern Recognition (CVPR). Las Vegas, NV, USA: IEEE; 2016. p. 2414–2423. Available from: <http://ieeexplore.ieee.org/document/7780634/>.
- [22] Thurston AJ. Paré and Prosthetics: The Early History of Artificial Limbs. ANZ Journal of Surgery. 2007;77(12):1114–1119. Available from: <https://onlinelibrary.wiley.com/doi/abs/10.1111/j.1445-2197.2007.04330.x>.
- [23] Weiland JD, Humayun MS. Visual Prosthesis. Proceedings of the IEEE. 2008 Jul;96(7):1076–1084.
- [24] Ong JM, da Cruz L. The bionic eye: a review. Clinical & Experimental Ophthalmology. 2012 Jan;40(1):6–17. Available from: <https://onlinelibrary.wiley.com/doi/full/10.1111/j.1442-9071.2011.02590.x>.
- [25] Humayun MS, de Juan E, Dagnelie G, Greenberg RJ, Propst RH, Phillips DH. Visual perception elicited by electrical stimulation of retina in blind humans. Archives of Ophthalmology (Chicago, Ill: 1960). 1996 Jan;114(1):40–46.
- [26] Stingl K, Schippert R, Bartz-Schmidt KU, Besch D, Cottrall CL, Edwards TL, et al. Interim Results of a Multicenter Trial with the New Electronic Subretinal Implant Alpha AMS in 15 Patients Blind from Inherited Retinal Degenerations. Frontiers in Neuroscience. 2017;11. Available from: <https://www.frontiersin.org/articles/10.3389/fnins.2017.00445/full>.
- [27] Veraart C, Raftopoulos C, Mortimer JT, Delbeke J, Pins D, Michaux G, et al. Visual sensations produced by optic nerve stimulation using an implanted self-sizing spiral cuff electrode. Brain Research. 1998 Nov;813(1):181–186. Available from: <http://www.sciencedirect.com/science/article/pii/S0006899398009779>.
- [28] Panetsos F, Sanchez-Jimenez A, Diaz-de Cerio ER, Diaz-Guemes I, Sanchez FM. Consistent Phosphenes Generated by Electrical Microstimulation of the Visual Thalamus. An Experimental Approach for Thalamic Visual Neuroprostheses. Frontiers in Neuroscience. 2011;5. Available from: <https://www.frontiersin.org/articles/10.3389/fnins.2011.00084/full>.

- [29] Lowery AJ, Rosenfeld JV, Lewis PM, Browne D, Mohan A, Brunton E, et al. Restoration of vision using wireless cortical implants: The Monash Vision Group project. In: 2015 37th Annual International Conference of the IEEE Engineering in Medicine and Biology Society (EMBC); 2015. p. 1041–1044.
- [30] Lowery AJ, Rosenfeld JV, Rosa MGP, Brunton E, Rajan R, Mann C, et al. Monash Vision Group's Gennaris Cortical Implant for Vision Restoration. In: Gabel VP, editor. Artificial Vision: A Clinical Guide. Cham: Springer International Publishing; 2017. p. 215–225. Available from: https://doi.org/10.1007/978-3-319-41876-6_17.
- [31] Troyk PR. The Intracortical Visual Prosthesis Project. In: Gabel VP, editor. Artificial Vision: A Clinical Guide. Cham: Springer International Publishing; 2017. p. 203–214. Available from: https://doi.org/10.1007/978-3-319-41876-6_16.
- [32] Fernández E, Normann RA. CORTIVIS Approach for an Intracortical Visual Prostheses. In: Gabel VP, editor. Artificial Vision: A Clinical Guide. Cham: Springer International Publishing; 2017. p. 191–201. Available from: https://doi.org/10.1007/978-3-319-41876-6_15.
- [33] SecondSight. Second Sight Medical Products, Inc. Presents Positive Interim Results at the Fifth Annual BRAIN Initiative® Investigators Meeting;p. 2.
- [34] Deering MF. The Limits of Human Vision. In: 2nd International Immersive Projection Technology Workshop. vol. 2; 1998. p. 6.
- [35] Niketeghad S, Pouratian N. Brain Machine Interfaces for Vision Restoration: The Current State of Cortical Visual Prosthetics. Neurotherapeutics. 2019 Jan;16(1):134–143. Available from: <https://doi.org/10.1007/s13311-018-0660-1>.
- [36] SecondSight. Early Feasibility Study of the Orion Visual Cortical Prosthesis System - Full Text View - ClinicalTrials.gov;. Available from: <https://clinicaltrials.gov/ct2/show/NCT03344848>.

- [37] ANZCTR. The first safety study of a Bionic Vision Prosthesis in adults with complete and untreatable blindness; 2018. Available from: <https://www.anzctr.org.au/Trial/Registration/TrialReview.aspx?id=373899>.
- [38] Dobelle WH, Mladejovsky MG, Girvin JP. Artificial Vision for the Blind: Electrical Stimulation of Visual Cortex Offers Hope for a Functional Prosthesis. *Science*. 1974 Feb;183(4123):440–444. Available from: <http://www.sciencemag.org/cgi/doi/10.1126/science.183.4123.440>.
- [39] Lee HW, Hong SB, Seo DW, Tae WS, Hong SC. Mapping of functional organization in human visual cortex: Electrical cortical stimulation. *Neurology*. 2000 Feb;54(4):849–854. Available from: <http://www.neurology.org/cgi/doi/10.1212/WNL.54.4.849>.
- [40] Winawer J, Parvizi J. Linking Electrical Stimulation of Human Primary Visual Cortex, Size of Affected Cortical Area, Neuronal Responses, and Subjective Experience. *Neuron*. 2016 Dec;92(6):1213–1219. Available from: <https://linkinghub.elsevier.com/retrieve/pii/S0896627316308467>.
- [41] Murphrey DK, Maunsell JHR, Beauchamp MS, Yoshor D. Perceiving electrical stimulation of identified human visual areas. *Proceedings of the National Academy of Sciences*. 2009 Mar;106(13):5389–5393. Available from: <https://www.pnas.org/content/106/13/5389>.
- [42] Collins KL, Sarma D, Hakimian S, Tsai JJ, Ojemann JG. Preserved evoked conscious perception of phosphenes with direct stimulation of deafferented primary visual cortex. *Epilepsy & Behavior Case Reports*. 2019 Jan;11:84–86. Available from: <http://www.sciencedirect.com/science/article/pii/S2213323218301270>.
- [43] Tehovnik EJ, Slocum WM, Carvey CE, Schiller PH. Phosphene induction and the generation of saccadic eye movements by striate cortex. *Journal of Neurophysiology*. 2005 Jan;93(1):1–19.
- [44] Tehovnik EJ, Slocum WM. Phosphene Induction by Microstimulation of

- Macaque V1. *Brain research reviews*. 2007 Feb;53(2):337–343. Available from: <https://www.ncbi.nlm.nih.gov/pmc/articles/PMC1850969/>.
- [45] Tehovnik EJ, Slocum WM. Microstimulation of V1 delays visually guided saccades: a parametric evaluation of delay fields. *Experimental Brain Research*. 2007 Jan;176(3):413–424.
- [46] Tehovnik EJ, Slocum WM, Smirnakis SM, Tolias AS. Microstimulation of visual cortex to restore vision. *Progress in Brain Research*. 2009;175:347–375.
- [47] Dobelle WH, Mladejovsky MG, Evans JR, Roberts TS, Girvin JP. ‘Braille’ reading by a blind volunteer by visual cortex stimulation. *Nature*. 1976 Jan;259(5539):111–112. Available from: <http://www.nature.com/articles/259111a0>.
- [48] Dobelle WH, Turkel J, Henderson DC, Evans JR. Mapping the Representation of the Visual Field by Electrical Stimulation of Human Visual Cortex. *American Journal of Ophthalmology*. 1979 Oct;88(4):727–735. Available from: <https://linkinghub.elsevier.com/retrieve/pii/0002939479906731>.
- [49] Allison T, Ginter H, McCarthy G, Nobre AC, Puce A, Luby M, et al. Face recognition in human extrastriate cortex. *Journal of Neurophysiology*. 1994 Feb;71(2):821–825. Available from: <https://www.physiology.org/doi/abs/10.1152/jn.1994.71.2.821>.
- [50] Puce A, Allison T, McCarthy G. Electrophysiological Studies of Human Face Perception. III: Effects of Top-down Processing on Face-specific Potentials. *Cerebral Cortex*. 1999 Jul;9(5):445–458. Available from: <https://academic.oup.com/cercor/article/9/5/445/310738>.
- [51] Dobelle WH. Artificial Vision for the Blind by Connecting a Television Camera to the Visual Cortex. *ASAIO Journal*. 2000 Feb;46(1):3. Available from: https://journals.lww.com/asaiojournal/Fulltext/2000/01000/Artificial_Vision_for_the_Blind_by_Connecting_a_2.aspx.
- [52] Bosking WH, Sun P, Ozker M, Pei X, Foster BL, Beauchamp MS, et al. Saturation in Phosphene Size with Increasing Current Levels Delivered to Human

- Visual Cortex. *The Journal of Neuroscience: The Official Journal of the Society for Neuroscience*. 2017;37(30):7188–7197.
- [53] Bosking WH, Foster B, Sun P, Beauchamp MS, Yoshor D. Rules Governing Perception of Multiple Phosphenes by Human Observers. *Neuroscience*; 2018. Available from: <http://biorxiv.org/lookup/doi/10.1101/302547>.
- [54] Beauchamp MS, Bosking W, Sun P, Foster B, Niketeghad S, Pouratian N, et al. Dynamic Electrical Stimulation of Sites in Visual Cortex Produces Form Vision in Sighted and Blind Humans. *bioRxiv*. 2018 Nov;p. 462697. Available from: <https://www.biorxiv.org/content/10.1101/462697v1>.
- [55] Solomon SG, Lennie P. The machinery of colour vision. *Nature Reviews Neuroscience*. 2007 Apr;8(4):276–286. Available from: <https://www.nature.com/articles/nrn2094>.
- [56] Vladusich T, Lucassen MP, Cornelissen FW. Brightness and Darkness as Perceptual Dimensions. *PLOS Computational Biology*. 2007 Oct;3(10):e179. Available from: <https://journals.plos.org/ploscompbiol/article?id=10.1371/journal.pcbi.0030179>.
- [57] Fox PT, Miezin FM, Allman JM, Essen DV, Raichle ME. Retinotopic organization of human visual cortex mapped with positron- emission tomography. *Journal of Neuroscience*. 1987 Mar;7(3):913–922. Available from: <http://www.jneurosci.org/content/7/3/913>.
- [58] Engel SA, Glover GH, Wandell BA. Retinotopic organization in human visual cortex and the spatial precision of functional MRI. *Cerebral Cortex*. 1997 Mar;7(2):181–192. Available from: <https://academic.oup.com/cercor/article/7/2/181/318870>.
- [59] Born RT, Trott AR, Hartmann TS. Cortical magnification plus cortical plasticity equals vision? *Vision Research*. 2015 Jun;111:161–169. Available from: <http://www.sciencedirect.com/science/article/pii/S0042698914002351>.
- [60] Ghose K, Maunsell JHR. A Strong Constraint to the Joint Processing of Pairs

- of Cortical Signals. *Journal of Neuroscience*. 2012 Nov;32(45):15922–15933. Available from: <http://www.jneurosci.org/content/32/45/15922>.
- [61] Ben-Yishai R, Bar-Or RL, Sompolinsky H. Theory of orientation tuning in visual cortex. *Proceedings of the National Academy of Sciences*. 1995 Apr;92(9):3844–3848. Available from: <https://www.pnas.org/content/92/9/3844>.
- [62] Chen SC, Suaning GJ, Morley JW, Lovell NH. Simulating prosthetic vision: I. Visual models of phosphenes. *Vision Research*. 2009 Jun;49(12):1493–1506. Available from: <http://www.sciencedirect.com/science/article/pii/S0042698909000467>.
- [63] Chen SC, Suaning GJ, Morley JW, Lovell NH. Simulating prosthetic vision: II. Measuring functional capacity. *Vision Research*. 2009 Sep;49(19):2329–2343. Available from: <https://linkinghub.elsevier.com/retrieve/pii/S0042698909003253>.
- [64] Cha K, Horch K, A Normann R. Simulation of a phosphene-based visual field: Visual acuity in a pixelized vision system. *Annals of biomedical engineering*. 1992 Feb;20:439–49.
- [65] Li H, Su X, Wang J, Kan H, Han T, Zeng Y, et al. Image processing strategies based on saliency segmentation for object recognition under simulated prosthetic vision. *Artificial Intelligence in Medicine*. 2018 Jan;84:64–78. Available from: <http://www.sciencedirect.com/science/article/pii/S0933365716304195>.
- [66] Canny J. Readings in computer vision. In: *A Computational Approach to Edge Detection*. Morgan Kaufmann; 1987. p. 184–203.
- [67] Lui WLD, Browne D, Kleeman L, Drummond T, Wai Ho Li. Transformative Reality: Improving bionic vision with robotic sensing. In: *2012 Annual International Conference of the IEEE Engineering in Medicine and Biology Society*. San Diego, CA: IEEE; 2012. p. 304–307. Available from: <http://ieeexplore.ieee.org/document/6345929/>.

- [68] Sanchez-Garcia M, Martinez-Cantin R, Guerrero JJ. Structural and object detection for phosphene images. arXiv:180909607 [cs]. 2018 Sep;ArXiv: 1809.09607. Available from: <http://arxiv.org/abs/1809.09607>.
- [69] Han T, Li H, Lyu Q, Zeng Y, Chai X. Object recognition based on a foreground extraction method under simulated prosthetic vision. In: 2015 International Symposium on Bioelectronics and Bioinformatics (ISBB); 2015. p. 172–175.
- [70] Pal NR, Pal SK. A review on image segmentation techniques. Pattern Recognition. 1993 Sep;26(9):1277–1294. Available from: <http://www.sciencedirect.com/science/article/pii/003132039390135J>.
- [71] Guo F, Yang Y, Gao Y. Optimization of Visual Information Presentation for Visual Prosthesis. International Journal of Biomedical Imaging. 2018;2018:1–12. Available from: <https://www.hindawi.com/journals/ijbi/2018/3198342/>.
- [72] Macé MJM, Guivarch V, Denis G, Jouffrais C. Simulated Prosthetic Vision: The Benefits of Computer-Based Object Recognition and Localization: Computer Vision for Low Resolution Visual Implants. Artificial Organs. 2015 Jul;39(7):E102–E113. Available from: <http://doi.wiley.com/10.1111/aor.12476>.
- [73] Rawat W, Wang Z. Deep Convolutional Neural Networks for Image Classification: A Comprehensive Review. Neural Computation. 2017 Jun;29(9):2352–2449. Available from: https://doi.org/10.1162/neco_a_00990.
- [74] Chichilnisky EJ, Greschner M, Jepson L. SMART PROSTHESIS FOR FACILITATING ARTIFICIAL VISION USING SCENE ABSTRACTION. 2018;p. 27.
- [75] Li WH, Tang TJJ, Lui WLD. Going beyond vision to improve bionic vision. In: 2013 IEEE International Conference on Image Processing. Melbourne, Australia: IEEE; 2013. p. 1555–1558. Available from: <http://ieeexplore.ieee.org/document/6738320/>.
- [76] Spencer TC, Fallon JB, Shivdasani MN. Creating virtual electrodes with 2D current steering. Journal of Neural Engineering. 2018 Mar;15(3):035002. Available from: <https://doi.org/10.1088%2F1741-2552%2Faab1b8>.

- [77] Grossman ED, Blake R. Brain Areas Active during Visual Perception of Biological Motion. *Neuron*. 2002 Sep;35(6):1167–1175. Available from: <http://www.sciencedirect.com/science/article/pii/S0896627302008978>.
- [78] McCarthy C, Walker JG, Lieby P, Scott A, Barnes N. Mobility and low contrast trip hazard avoidance using augmented depth. *Journal of Neural Engineering*. 2014 Nov;12(1):016003. Available from: <https://doi.org/10.1088%2F1741-2560%2F12%2F1%2F016003>.
- [79] Hu J, Xia P, Gu C, Qi J, Li S, Peng Y. Recognition of Similar Objects Using Simulated Prosthetic Vision. *Artificial Organs*. 2014;38(2):159–167. Available from: <http://www.onlinelibrary.wiley.com/doi/abs/10.1111/aor.12147>.
- [80] Chen SC, Lovell NH, Suaning GJ. Effect on prosthetic vision visual acuity by filtering schemes, filter cut-off frequency and phosphene matrix: a virtual reality simulation. In: The 26th Annual International Conference of the IEEE Engineering in Medicine and Biology Society. vol. 2; 2004. p. 4201–4204.
- [81] Josh H, Yong B, Kleeman L, Road W. A Real-time FPGA-based Vision System for a Bionic Eye. 2011;p. 8.
- [82] Josh H, Mann C, Kleeman L, Lui WLD. Psychophysics testing of bionic vision image processing algorithms using an FPGA Hatpack. In: 2013 IEEE International Conference on Image Processing. Melbourne, Australia: IEEE; 2013. p. 1550–1554. Available from: <http://ieeexplore.ieee.org/document/6738319/>.
- [83] Luo YHL, da Cruz L. The Argus® II Retinal Prosthesis System. *Progress in Retinal and Eye Research*. 2016 Jan;50:89–107. Available from: <http://www.sciencedirect.com/science/article/pii/S1350946215000701>.
- [84] Markowitz M, Rankin M, Mongy M, Patino BE, Manusow J, Devenyi RG, et al. Rehabilitation of lost functional vision with the Argus II retinal prosthesis. *Canadian Journal of Ophthalmology*. 2018 Feb;53(1):14–22. Available from: <https://linkinghub.elsevier.com/retrieve/pii/S0008418217309729>.

- [85] Krizhevsky A, Sutskever I, Hinton GE. ImageNet Classification with Deep Convolutional Neural Networks. In: Pereira F, Burges CJC, Bottou L, Weinberger KQ, editors. Advances in Neural Information Processing Systems 25. Curran Associates, Inc.; 2012. p. 1097–1105. Available from: <http://papers.nips.cc/paper/4824-imagenet-classification-with-deep-convolutional-neural-networks.pdf>.
- [86] Karpathy A, Toderici G, Shetty S, Leung T, Sukthankar R, Fei-Fei L. Large-scale Video Classification with Convolutional Neural Networks; 2014. p. 1725–1732. Available from: https://www.cv-foundation.org/openaccess/content_cvpr_2014/html/Karpathy_Large-scale_Video_Classification_2014_CVPR_paper.html.
- [87] Goodfellow I, Pouget-Abadie J, Mirza M, Xu B, Warde-Farley D, Ozair S, et al. Generative Adversarial Nets. In: Ghahramani Z, Welling M, Cortes C, Lawrence ND, Weinberger KQ, editors. Advances in Neural Information Processing Systems 27. Curran Associates, Inc.; 2014. p. 2672–2680. Available from: <http://papers.nips.cc/paper/5423-generative-adversarial-nets.pdf>.
- [88] Springenberg JT. Unsupervised and Semi-supervised Learning with Categorical Generative Adversarial Networks; 2015.
- [89] Durugkar I, Gemp I, Mahadevan S. Generative Multi-Adversarial Networks; 2016.
- [90] Mirza M, Osindero S. Conditional Generative Adversarial Nets; 2014.
- [91] Ge C, Kasabov N, Liu Z, Yang J. A spiking neural network model for obstacle avoidance in simulated prosthetic vision. *Information Sciences*. 2017 Aug;399:30–42. Available from: <http://www.sciencedirect.com/science/article/pii/S0020025517305819>.
- [92] Moore GE. Cramming More Components Onto Integrated Circuits. *Proceedings of the IEEE*. 1998 Jan;86(1):82–85. Available from: <http://ieeexplore.ieee.org/document/658762/>.

- [93] Goscinski WJ, McIntosh P, Felzmann U, Maksimenko A, Hall CJ, Gureyev T, et al. The multi-modal Australian ScienceS Imaging and Visualization Environment (MASSIVE) high performance computing infrastructure: applications in neuroscience and neuroinformatics research. *Frontiers in Neuroinformatics*. 2014 Mar;8. Available from: <https://doi.org/10.3389/fninf.2014.00030>.
- [94] Lecun Y, Bottou L, Bengio Y, Haffner P. Gradient-based learning applied to document recognition. *Proceedings of the IEEE*. 1998;86(11):2278–2324. Available from: <https://doi.org/10.1109/5.726791>.
- [95] Abadi M, Agarwal A, Barham P, Brevdo E, Chen Z, Citro C, et al.. TensorFlow: Large-Scale Machine Learning on Heterogeneous Systems; 2015. Software available from tensorflow.org. Available from: <https://www.tensorflow.org/>.
- [96] Kingma DP, Ba J. Adam: A Method for Stochastic Optimization; 2014.
- [97] Peirce J, Gray JR, Simpson S, MacAskill M, Höchenberger R, Sogo H, et al. PsychoPy2: Experiments in behavior made easy. *Behavior Research Methods*. 2019 Feb;51(1):195–203. Available from: <https://doi.org/10.3758/s13428-018-01193-y>.