Using Generative Adversarial Networks to Derive Phosphene Patterns in Simulated Prosthetic Vision (IN PROGRESS)

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
School of Biomedical Sciences, Monash University

Word Count: <> words

Abstract

Acknowledgements

Declaration

Contents

List of Tables

List of Figures

List of Abbreviations

GAN	Generative adversarial network
CVP	Intracortical Visual Prosthesis

1 Background - CAN SKIP

(Will be a modified version of the literature review, but mostly the same. Here's a quick summary)

Blindness is a significant disability and can be caused by damage at any point along the visual pathway from the eye to the brain. A **cortical visual prosthesis** (CVP) is a vision-restoring device inserted directly into the brain, replacing all parts of the visual pathway except the brain itself. Because a CVP only requires a functioning brain, CVPs are a flexible (and potentially the only) option for many causes of non-cortical blindness.

However, a significant detractor from CVPs is that they are currently only expected to provide very low-quality vision. As CVPs are still in early development with clinical trials only beginning to emerge, we still do not know exactly what people will be able to see. What *is* known is that people tend to see small spots of light called **phosphenes** - like "stars in a night sky" - when small regions of the brain are stimulated. Unfortunately, we cannot yet reliably control the locations of phosphenes, nor their qualities beyond simply being "on" or "off".

To address the perceptual limitations of CVPs, a large body of research has been devoted to investigating how image processing techniques can make phosphenes meaningful. These include using phosphenes to selectively display salient objects, display only prominent edges, or even replacing faces with low-resolution icons. However, most of these studies are premised on static, optimistic simulations of prosthetic vision. Many do not address how to apply traditional image processing strategies when we cannot control how phosphenes are laid out or what qualities they possess. As a result, it is not clear what methods can be used to generalise image processing strategies to different people, who experience phosphenes differently.

Outside of prosthetic vision, there have been advances in using computers to generate novel imagery under specified conditions. **Generative adversarial networks** (GANs) are a machine learning technique to generate new data samples (most often, images) when given many examples of data known prior. GANs are not an image processing algorithm *per se*, but a method of training a neural network to take a seed input and return a generated data sample. GANs work by optimising the features of generated data samples to be as indistinguishable from real data samples as possible. They have been used with varying degrees of success in a number of scene generation, style transfer and image processing tasks.

Because GANs are used to make generated images look more realistic, it is possible they could be used in a prosthetic vision setting to make phosphene patterns look more like what they are supposed to represent. Importantly, GANs are a *training* technique which can be applied to different types of phosphene simulations to produce specifically-tailored results. This means GANs may be able to address the difficulties in finding generalised image processing strategies suitable for different people's experiences of prosthetic vision. To our knowledge, there are no prior studies which attempt to apply GANs to the problems faced by prosthetic vision.

> Should I discuss the choice of a digit recognition task in the background, or in the aims/methods? I'm leaning towards Background, so will probably add it in here.
>
> I will probably also spend some time in the background discussing the actual architecture of a GAN (wasn't in my old literature review).

2 Aims

2.1 Research Questions

As there are no prior studies applying GANs to prosthetic vision, we aimed to address the following 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. 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)?

2.2 Aims and Hypothesis

To address the two research questions, we aimed to:

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.
2. Experimentally test whether people find it easier to recognise digits from phosphene patterns derived from our preliminary training implementation, compared to a basic mask-based comparison.

For Aim 2, we also formulated the following hypothesis:

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

Should this maybe be split into two hypotheses - **initial** digit recognition accuracy, and **overall** (or "final") digit recognition accuracy? The psychophysics experiment itself was tailored more to the latter question (which is why I've kept it as one above), though it probably seems from the Background that I was more trying to lead to the former question. . .

2.3 Rationale

While GANs have been applied to image-based tasks in other domains, it is not yet 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. This gives GANs complete control over what its generated images look like and it can optimise the output of each individual pixel.

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 visual render independent of the GAN based on specified simulation properties. A useful GAN implementation for simulated prosthetic vision should therefore *not* be based on direct pixel manipulation (as is typically the case).

Aim 1 therefore aims to explore and develop a useful GAN implementation based on separating out a simulated rendering step so that it can be applied to simulated prosthetic vision. This is the primary contribution of this project.

GANs are designed to minimise the computer's ability to discriminate between generated images and real image samples. However, computers both find it easy to discriminate features which humans find difficult (e.g. different shades of grey) and find it difficult to discriminate features which humans find easy (e.g. abstract scenes). As a result, while the generated output may be optimised for a computer, its results may not translate to a human. We therefore formulated Aim 2 as a preliminary validation measure to test whether the computer's output was human-benefitting.

We would like to emphasise that 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. We do not expect this project to produce a necessarily useful software implementation in Aim 1 given its novelty, and intend to use the preliminary validation results from Aim 2 to guide how to better refine the software implementation for future use.

3 Methods

4 Results

The results are divided into:

1. Training results, showing qualitative phosphene patterns and training statistics of digit encoders produced by training, and
2. Experimental results, derived from participants' performance during the psychophysics experiment described in Section <>.

4.0.1 Training Results

The cGAN training architecture described in Section <> was tested on a large variety of phosphene grids varying in phosphene location, sizing, regularity and resolution. We have selected a representative sample of grids to highlight several general features of the training results.

1. Generalisability of Training

   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.

Figure <> shows the phosphene patterns for the digits 0-9 after 20 epochs of training for a regular cartesian grid, a regular polar grid, and a randomised polar grid, each with 144 phosphenes.

<FIGURE <»

There were no issues applying the training architecture to grids with arbitrary arrangements and resolutions.

2. Stability of Phosphene Patterns over Epochs

   Phosphene patterns were not stable over epochs; i.e. the phosphene patterns learned by the trained encoder always changed with the next epoch.

   Figure <> shows the trained phosphene representation for the digits 0-9 for a random grid with 144 phosphenes over each of 40 epochs.

   <FIGURE <»

   For comparison, Figure <> shows the equivalent results of Figure <>, trained for a different random grid of only 64 electrodes.

   <FIGURE <»

   From these figures, it is evident that:

   (a) The degree of instability reduces (but does not disappear) as the number of epochs increases.

   (b) The qualitative consequences of instability are greater at low resolution. Small changes in the moderate resolution patterns usually do not disturb the overall form of the digit from one epoch to the next (though the overall form may eventually drift over multiple epochs). However, small changes in low resolution patterns greatly influence the digit form produced by phosphenes, resulting in a relatively higher impact of instability.

4.0.2 Experimental Results

All 11 participants were included in the analysis. 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. All data was considered valid and included in the analysis.

1. Accuracy

5 Discussion

6 Conclusion

7 Appendices

References