Optimisation of prosthetic vision for the detection of domestic objects using a silicon retina

Chantal van Duin[1]

s1004516

Supervisors:

Dr. Bodo. J. Rückauer[1]
Prof. Dr. Marchel A.J. van Gerven [1]

Affiliations:

[1] *Donders Institute for Brain, Cognition and Behaviour, Radboud University, Nijmegen, The Netherlands*

March 2023

Neuroprosthetic visual implants are a promising solution for some forms of blindness by providing some rudimentary form of vision using phosphenes, percepts of points of light. As this restored prosthetic vision is limited compared to standard retinal vision, one must be selective in what visual information and features to convey. In this thesis, efficient and effective means to perform visual scene simplification for prosthetic vision was investigated. In particular, the feature extraction at the sensor level using a silicon retina known as the DVS was evaluated. The DVS is an event sensor which mimics the spiking behaviour of a human retina by asynchronously measuring the per-pixel brightness changes of a visual scene resulting in a high dynamic range and a microsecond temporal resolution. These characteristics in combination with the proposal that bio-inspired retinal models may improve phosphenes, make event sensors of interest for visual prostheses. As the event data of the DVS has a different space-time output compared to standard cameras, its output needs to be pre-processed in order to function optimally in traditional vision algorithms. This thesis evaluated optimal pre-processing methods for the DVS event stream as well as compared the resulting event-based prosthetic vision against Canny Edge Detection, a state-of-the-art scene simplification approach for prosthetic vision. This evaluation was done with the computational model EfficientDet-D0 performance of object detection of common everyday objects. Findings provided a guideline for pre-processing DVS event streams and showed event-based prosthetic vision significantly exceeding traditional Canny Edge prosthetic vision for both a lower and high-end phosphene resolution.

Keywords : phosphenes, event camera, object detection, prosthetic vision, simulated phosphene vision, DVS, silicon retina

1 Introduction

Worldwide, of the 8 billion human population in 2022, 295 million people are estimated to suffer from moderate and severe vision impairment and around 43 million people are estimated to be completely blind (Bourne et al., 2021; Fernandez, 2018). It has been predicted that the number of people with moderate and severe vision impairment will increase to 474 million in 2050 whereas the number of blind people will rise to a total of 61 million (Bourne et al., 2021). For those whose visual impairment is the result of severe degeneration or damage to the retina, the optic nerve, or the brain, no effective form of treatment yet exists (Fernandez, 2018).

For these visual impairment cases, visual neural prostheses might potentially be the solution by restoring some basic vision in the form of phosphenes, percepts of points of light. Phosphenes are generated by converting the visual scene captured by a camera to electrical stimulation of cells in the visual system: either in the retina, in the optic nerve, the lateral geniculate nucleus (LGN), or in the visual cortex itself.

In the case of retinal prostheses, this electrical stimulation is applied by an electrode chip placed within the retinal layers (Fernandez, 2018; Shim et al., 2020). This can be on the photoreceptor layer of the retina near the retinal ganglion cells (subretinal), in the ganglion cell layer (epiretinal), or between the sclera and choroid (suprachoroidal) (Shim et al., 2020; Zrenner, 2002). While retinal implants have been successfully implemented in humans and have the advantage of being an extra-cranial operation and being at the earliest stage of visual perception, they still require intact middle and inner retinal layers resulting in them only serving a small portion of the blind population (Niketeghad & Pouratian, 2019; Pezaris & Eskandar, 2009; Shim et al., 2020; Zrenner, 2002). The same advantages and disadvantage pertain to visual prostheses which apply electrical stimulation directly on the optic nerve (Niketeghad & Pouratian, 2019; Pezaris & Eskandar, 2009; Veraart et al., 1998). Thalamic visual prostheses stimulate the LGN using electrodes placed with deep brain stimulation (DBS) techniques however so far they have only been tested within animal models as they have the disadvantage of being a deep brain structure prosthetic (Fernandez, 2018; Killian et al., 2016; Pezaris & Eskandar, 2009).

Cortical visual prostheses stimulate the early areas of the visual cortex (V1, V2, and V3), either with electrodes arrays placed on the cortex surface (subdural surface electrodes) or by penetrating the cortex (intracortical electrodes) (Fernandez, 2018; Niketeghad & Pouratian, 2019; Pezaris & Eskandar, 2009). In particular, cortical visual prostheses residing on the primary visual cortex (V1) have shown to be the most promising as V1 has a large surface area, allowing for stimulation using a higher amount of electrodes leading to an increased phosphene generation (Fernandez, 2018; Lewis et al., 2015; Niketeghad & Pouratian, 2019). By selective stimulating the electrodes and making use of V1's retinotopic organisation, a controlled phosphene distribution can be generated to create a meaningful representation of the visual environment (Niketeghad & Pouratian, 2019; van Steveninck, Güçlü, et al., 2022). Furthermore, cortical visual prostheses may also help a larger portion of the blind population as it bypasses any possible damage in the visual pathway prior to the visual cortex itself.

Current cortical visual prostheses can generate 100 up to 1000 phosphenes with the highest phosphene resolution of 32 x 32 being obtained by a 1024-channel intracranial prosthesis, which has been successfully implanted in macaque monkeys (Chen et al., 2020). Overall, it can be noted that a lower phosphene resolution is obtained using surface electrodes, which require electrical currents in the milli-ampere range to reach the stimulation threshold, limiting the number of possible electrodes being placed. In addition, surface electrodes generate larger phosphenes which also might get intervened by simultaneous stimulation on nearby electrodes (Chen et al., 2020; Niketeghad & Pouratian, 2019). However, surface electrodes are less invasive compared to intracranial electrodes and might have greater long-term reliability as there likely will be a lesser degree of foreign body response and unwanted tissue responses (Christie et al., 2016; Niketeghad & Pouratian, 2019). Intracranial electrodes, on the other hand, require a lower magnitude of electrical current and can active neurons within hundred micrometers of the visual cortex resulting in smaller and more precise phosphenes hence a higher phosphene resolution (Chen et al., 2020; Niketeghad & Pouratian, 2019). Efforts to further increase the possible phosphene resolution are being made, for example by the NeuraViPer project [1] which aims to develop a cortical prosthetic with up to 4096 stimulation electrodes.

However, the spatial resolution of prosthetic vision, even when considering the more optimistic higher phosphene resolution, is still of a lower degree compared to healthy retinal vision. Therefore in order to produce an effective representation of the visual environment for the prosthetic, one must be selective in what information of the visual field is conveyed. This can be done by developing image pre-processing techniques which select the most in-

[1] https://www.neuraviper.eu

formative visual features into phosphene vision representations depending on the exact visual environment and needs of the prosthetic user. These pre-processing techniques range from more basic image processing methods such as Canny Edge detection to more complex ones such as complete neural networks or deep learning approaches (Küçükoğlu et al., 2022a; van Steveninck, Güçlü, et al., 2022; van Steveninck, van Gestel, et al., 2022). In order to evaluate how well the pre-processing technique can represent the visual information, Simulated Phosphene Vision (SPV) can be used.

Simulated Phosphene Vision is a representation of what people with a visual prosthetic might see in response to electrical stimulation of the implanted electrodes. Conducting research with SPV bypasses the need to use blind participants with implemented visual prostheses, making it less expensive and less time intensive to optimise and test these processing methods. The resulting SPV of the processing method can then be evaluated either with sighted individuals, for example with the use of Virtual Reality, or with computational models and Machine Learning techniques. Research evaluating these processing methods using SPV typically focuses on one main domain of application which might be useful in daily life for blind individuals. Common applications are mobility navigation tasks (Küçükoğlu et al., 2022a; Lieby et al., 2011; van Steveninck, van Gestel, et al., 2022; Vergnieux et al., 2014), object recognition (Li et al., 2018; Zhao et al., 2010) and object localisation (Macé et al., 2015; Sanchez-Garcia et al., 2018), movement recognition(Zhao et al., 2020), letter recognition (Dagnelie et al., 2006; Zhao et al., 2011), face and emotion recognition (Bollen et al., 2019; Irons et al., 2017; Thompson et al., 2003).

This thesis contributes to SPV research by examining the potential of a more novel pre-processing method for prosthetic vision: an event-driven silicon retina known as the DVS. This event camera mimics the spiking behaviour of human retina ganglion cells with events by asynchronously signalling pixel-wise changes in light intensity within the visual scene. The retina-derived filtering of the DVS could potentially benefit functional prosthetic vision. In order to be used optimally in downstream algorithms such as traditional vision algorithms, the event output needs to be pre-processed and transformed into a more traditional space-time output such as image frames. The DVS is evaluated on the computational model performance of object detection of common everyday objects. Using object detection evaluation metrics, the following research questions are addressed:

- Can phosphenes from a silicon retina be used to perform everyday tasks like object detection?
- How do different event representations and processing of event-based vision affect prosthetic vision?
- How well does event-based prosthetic vision perform in comparison to traditional prosthetic vision pre-processing methods such as Canny Edge detection?
- How does the phosphene resolution affect the performance of event-based and Canny Edge prosthetic vision?

1.1 Background Information

1.1.1 Simulated Prosthetic Vision Framework

In general, Simulated Prosthetic Vision frameworks tend to consist of three components, see Figure 1:

- a pre-processing method or encoder, which determines the optimal stimulation protocol based on the visual input, the task and its chosen pre-processing method.
- a phosphene simulator, which converts the stimulation protocol of the encoder into simulated phosphene vision representation.
- evaluation or decoder, which assesses of how well the generated phosphene vision performs according to certain criteria or metrics for a task.

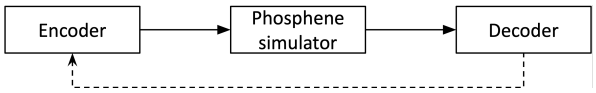

Figure 1. General Simulated Prosthetic Vision Framework.

The complexity and exact details of these components vary heavily among prosthetic vision research as it requires finding a general strategy that can be adapted and tailored to specific tasks and needs of the prosthetic user while working in real-time and under biological constraints. Therefore, SPV research requires determining which visual features to select for a specific task and how these features should be transformed into a suitable stimulation protocol for the generation of phosphenes, leading to a diverse range of developed pre-processing methods.

A well-established pre-processing method for prosthetic vision is Canny Edge Detection (Canny, 1986; Ogawa et al., 2010; Xu et al., 2017), an edge-extracting algorithm which has been shown to work well in real-time for a multitude of tasks (Bollen et al., 2019; Küçükoğlu et al., 2022a; Sanchez-Garcia et al., 2020; van Steveninck, van Gestel, et al., 2022; Vergnieux et al., 2017). As illustrated by Figure 2,

Canny Edge Detection extracts edges of grey-scaled images by :

1. smoothing the image and removing noise using a Gaussian filter.
2. computing the gradient intensity representation to determine the gradient magnitude and gradient direction for each pixel.
3. removing undesirable edges by applying non-maximum suppression (each pixel is checked to see if it is a local maximum in its neighbourhood in the direction of gradient).
4. applying hysteresis thresholds to remove weak edges. Hysteresis thresholding requires two thresholds: a low threshold which removes all pixels which fall below the threshold and a high threshold which marks strong edges consisting of pixels with values above the high threshold. Pixels that fall in between the two thresholds are considered weak edges and are only kept when they are connected to strong edges.

While Canny Edge detection as a pre-processing method shows high promise for prosthetic vision, its resulting performance can be highly dependent on the task and complexity of the visual environment (Rueckauer & van Gerven, 2022). Hence more sophisticated encoders have been developed to overcome these issues, ranging from landmark detection algorithms (Bollen et al., 2019), image segmentation using Fully Convolution Networks (Sanchez-Garcia et al., 2020) to deep-learning inspired approaches such as convolutional deep neural network architectures (Küçükoğlu et al., 2022a; Lozano et al., 2020; van Steveninck, Güçlü, et al., 2022). One of the advantages of using these deep-learning architectures as encoders is that they enable the problem of prosthetic vision (the optimisation of phosphene generation) to be considered as an end-to-end optimisation problem by using the evaluation metrics of the SPV to inform and improve the deep-learning encoder.

Another promising approach for the development of an improved visual prosthetic is to use the human retina as a starting point (Nirenberg & Pandarinath, 2012; Pelayo et al., 2003). This resulted in retinal image encoders which model the biological inner workings of the retina using computational or neural networks (Lozano et al., 2020; Pelayo et al., 2003). Other efforts have included the use of a silicon retina, either to create a more biological-plausible phosphene simulator (Fehervari et al., 2010) or as a pre-processing method (Rueckauer & van Gerven, 2022) which will be examined further in this research.

In SPV studies, phosphenes are usually generated as white or grey-scaled Gaussian dots arranged along a rectangular grid however, the exact size of the grid, the size of the phosphenes, the prevalence of the phosphenes, and the possible luminance levels of the phosphenes may deviate among studies (Chen et al., 2020; Rueckauer & van Gerven, 2022; Sanchez-Garcia et al., 2020; van Steveninck, Güçlü, et al., 2022; Zhao et al., 2010). Though more biological-plausible phosphene simulators, which take into consideration temporal dynamics and limitations of cortical stimulation, are currently being developed(Fehervari et al., 2010; van der Grinten et al., 2022), most SPV studies have used the more simplified but still feasible versions of phosphene simulators for the sake of functionality. The phosphene generation and its parameter optimisation are then either evaluated via task performance metrics by sighted participants or by Machine Learning algorithms, forming the decoder of the SPV framework.

1.1.2 Event-based Vision

Event-based vision is generated by event cameras, which are biological-inspired, asynchronous sensors that measure the per-pixel brightness changes of a visual scene (Delbrück et al., 2010; Gallego et al., 2020). This is in comparison to traditional frame-based cameras which use captured images at a fixed sampling rate instead. Event-based vision is driven by the dynamic information of light within the visual scene, rather than sampling the visual information of a scene based on external timing signals unrelated to the visual sources (Cho & Lee, 2015; Gallego et al., 2020; Posch, 2012).

Event-driven silicon retinas mimic the spiking

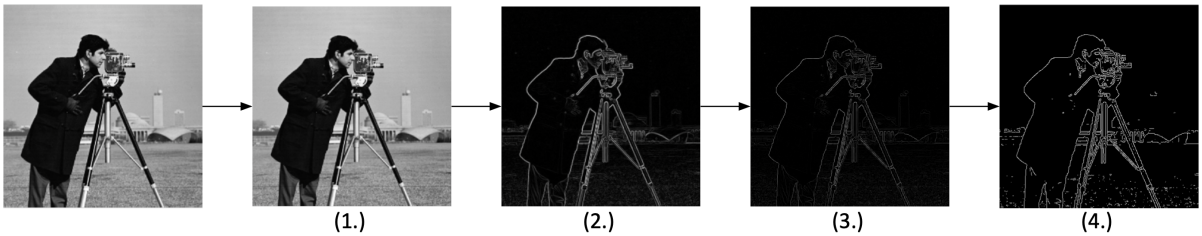

Figure 2. Canny Edge Detection Process

output of retinal ganglion cells to the visual cortex in the form of a sequence of 'events' or 'spikes', where an event is represented as a change of log light intensity of a predefined magnitude at a pixel at a particular time (Gallego et al., 2020; Posch, 2012).

The human retina consists of photoreceptors, bipolar cells, and ganglion cells with the photoreceptors being connected to bipolar cells, which are then connected to ganglion cells. Light is detected, processed and then converted into electrical pulses by the photo-receptors. These electrical pulses are then encoded in spiking patterns by the ganglion cells which are transmitted to the visual cortex via the optic nerve. The translation of light into visual information can be understood from two parallel visual system paths which have their basis in two types of ganglion cells: the Y-cells or Magnocells which are distributed along the Magnocellular pathway and the X-cells or Parvo-cells which are distributed along the Parvocellular pathway. The Magnocellular pathway is also known as the 'where' or transient visual pathway as it seems to be involved in the detection of changes and the detection of objects whereas the Parvocellular pathway also known as the 'what' or sustained visual pathway is more concerned with detailed visual information such as colour information, spatial details, and patterns (Cho & Lee, 2015; Gallego et al., 2020; Posch, 2012).

In this research, a particular silicon retina will be of focus known as the Dynamic Vision Sensor (DVS). The DVS is a type of visual sensor which was modelled after the retinal ganglion Y-cells in particular (Gallego et al., 2020; Lichtsteiner et al., 2008; Posch, 2012). A DVS pixel autonomously responds to relative changes in light intensity at a microsecond temporal resolution by tracking the photocurrent through the logarithmic photoreceptor. Then using a differencing circuit and two comparators, each DVS pixel asynchronously sends out an ON-event when the gradient of the photocurrent of the pixel is positive and an OFF-event when the gradient is negative (Cho & Lee, 2015; Lichtsteiner et al., 2008; Posch, 2012). See Figure 3 below for an illustration of this DVS pixel operation (Posch, 2012). The DVS camera outputs a stream of events in which the following is encoded per event: the event timestamp t, the pixel address (x,y), and the event polarity p.

The DVS measurement of brightness changes is not only very fast resulting in high temporal resolution (µs), but it also removes a lot of redundancy resulting in low power and low memory consumption. As DVS pixels measure on a logarithmic scale and independently from each other without waiting on an external timed shutter, they have a high dynamic range (120 dB). This makes them perform well under various and rapidly changing lighting conditions, this is unlike high-quality, frame-based cameras which have a much lower dynamic range (60 dB). Furthermore, this also results in low latency (µs) and the absence of motion blur (Cho & Lee, 2015; Gallego et al., 2020; Lichtsteiner et al., 2008). These technical characteristics make event sensors highly suitable for robotics and neuromorphic systems. Consequently, for the same advantages, event sensors are also attractive to use for visual neural prostheses (Rueckauer & van Gerven, 2022). Moreover, silicon retinas are conceivably interesting from an encoding perspective as they model key properties of biological vision. So they might potentially make it easier for the cortical visual areas to process as they resemble familiar en-

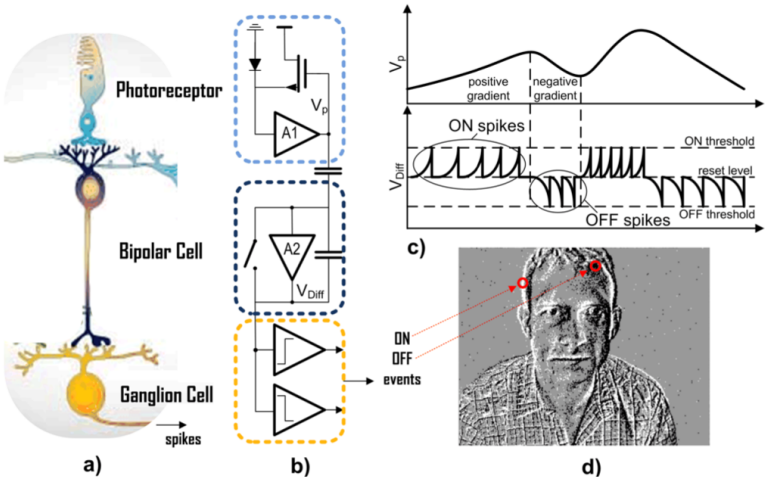

Figure 3. **a)** A simplified three-layer retina model **b)** Corresponding abstract DVS pixel circuitry **c)** Principle of DVS operation **d)** Example of a possible array of DVS pixels to natural scene, events collected over ± 10 ms (Posch, 2012).

coding.

However, using silicon retinas also come with a few limitations. Since intensity information is absent and only binary intensity changes are encoded, in the case of a visual scene with only static motion, the output of a DVS will mainly consist of background noise. This limitation also demonstrates the biggest limitations of a silicon retina, a static visual scene or single traditional image frame will result in no events being generated and thus no useful images. Furthermore, due to the asynchrony of DVS pixels and its output in a stream of spike-like events instead of the traditional image frames, traditional vision algorithms and image processing methods cannot be used. Instead, novel approaches must be developed to extract meaningful information from the events to solve a given vision-processing task. These novel approaches usually consist of the processing of events and transforming them into alternative representations that are better suited for downstream algorithms. Current developed approaches for representations range from simple transformation methods such as the accumulation of events into an image frame either by a constant time interval or by a constant number of events per frame, to more complex methods such as event accumulation into voxel grids, using individual events as input for Spiking Neural Networks or groups of events such as event frames as input for deep-learning Neural Networks (Bardow et al., 2016; Gallego et al., 2020; Pan et al., 2019; Zhu et al., 2018).

1.1.3 Object Detection

Object detection is a computer vision technique which involves both object classification (the allocation of a category to objects) and object localisation by placing rectangular labelled bounding boxes around objects within an image (Jabir et al., 2021; Zhao et al., 2019; Zou et al., 2019). More concisely, it tries to address the question: 'What objects are where?' (Zou et al., 2019). This is a challenging task for the visually impaired, making it of interest for the development of a visual prosthetic (Srikanth et al., 2021).

Historically object detection has been approached in two ways: with traditional detection models and deep learning based detection models. Traditional detection models were based on handcrafted features and their pipeline globally consisted of three stages: informative region selection, feature extraction and classification (Zhao et al., 2019; Zou et al., 2019). However, object detection performance using these traditional handcrafted feature models stagnated after 2010 and shifted to deep learning based models after the emergence of deep learning techniques such as Deep Neural Networks (DNN) and Convolutional Neural Networks (CNN) (Zhao et al., 2019; Zou et al., 2019). Their deeper architecture allows for the learning of more robust, complex features without the need to manually specify the features. Deep learning object detection frameworks can mainly be separated into two categories: 'two-stage detectors' and 'one-stage detectors'. Similar to traditional approaches, two-stage detector frameworks split object detection up into two stages, first by generating region proposals by specifying regions of interest (RoI) with the use of anchor boxes which are then classified in the second stage. One-stage detectors discard the need of region proposal by framing object detection as a unified framework which directly classifies and regresses the candidate anchor boxes (Tan et al., 2020; Zhao et al., 2019; Zou et al., 2019). While two-stage detectors are more flexible and achieve higher accuracy levels, one-stage detectors have higher detection speeds, and are overall more efficient and simple (Huang et al., 2017; Tan et al., 2020; Zou et al., 2019). For those reasons, recent successful efforts have been made to develop one-stage detectors with higher accuracy so that they perform competitively with two-stage detectors (Chen et al., 2019; Law & Deng, 2018; Zhao et al., 2019; Zou et al., 2019).

In order to evaluate an object detection model, different evaluation metrics exist with the most frequently used evaluation metrics being the PASCAL-VOC and the COCO mean Average Precision (mAP). The mAP measures the average detection precision under different recalls over all object categories. It does so by using a metric named Intersection over Union (IoU), which measures the object localisation accuracy by evaluating the overlap between a ground truth bounding box and a predicted bounding box. The IoU is given by the intersection area of the predicted bounding box B_p and the ground truth bounding box B_{gt} divided by the area over union between them:

$$IoU = \frac{B_p \cap B_{gt}}{B_p \cup B_{gt}} \tag{1}$$

The IoU is used to determine whether an object is successfully detected by checking whether the IoU of an object is above a predefined threshold. If so i.e. ($IoU \geq threshold$), then the object detection is seen as valid and labelled as a True Positive(TP). If not i.e. ($IoU \leq threshold$), then the detection is seen as invalid and labelled as a False Positive (FP). False Negative (FN) is applied when a ground truth is not detected at all.

Precision is the model's ability to identify only the relevant items i.e. the percentage of correct positive predictions:

$$Precision = \frac{TP}{TP + FP} \tag{2}$$

Recall is the model's ability to identify all relevant items i.e. the percentage of the true positive detected among all relevant ground truths:

$$Recall = \frac{TP}{TP + FN} \tag{3}$$

The most recent version of PASCAL-VOC AP measures the Average Precision by calculating the area under the Precision x Recall curve using interpolation over all data points:

$$AP = \sum_{n=0} (r_{n+1} - r_n) \rho_{interp}(r_{n+1}) \tag{4}$$

where

$$\rho_{interp}(r_{n+1}) = \max_{r' \geq r} \rho(r') \tag{5}$$

and $\rho(r')$ is the measured precision at recall r' for all data points n. The mean Average Precision (mAP) is then calculated as the mean of AP across all K object categories:

$$mAP = \frac{\sum_{i=1}^{K} AP_i}{K} \tag{6}$$

The PASCAL-VOC defines the mAP metric using a single IoU threshold of 0.5 whereas the COCO metrics define several mAP metrics using different IoU thresholds:

- primary challenge metric: $mAP_{coco} = \frac{1}{10} \sum_{thr \in [0.5:0.05:0.95]} mAP^{IoU=thr}$
- PASCAL-VOC metric: $mAP^{IoU=0.5}$
- strict metric: $mAP^{IoU=0.75}$

In addition to the different IoU thresholds, the COCO metrics include the calculation of the mAP_{coco} across different object scales.

2 Methods

In order to address the research aim of examining the potential of event-based vision as a front-end sensor for prosthetic vision, a pipeline of the SPV framework, see Figure 1, was implemented in Python using Pycharm. Within this pipeline, two encoders were implemented: event-based vision and Canny-Edge detection, followed by a phosphene simulator and a decoder consisting of object detection model EfficientDet-D0, see Figure 7 and Figure 9.

2.1 Data set

The data set that was used for all the experiments conducted in this research was CORe50 (Lomonaco & Maltoni, 2017; Lomonaco et al., 2020). This video data set was designed specifically for Continuous Object Recognition. It consists of 50 typical everyday objects belonging to 10 categories: plug adapters, mobile phones, scissors, light bulbs, cans, glasses, balls, markers, cups and remote controls, making object detection possible at both object level and category level, see Figure 4. The latter will be of focus for the majority of the experiments due to the low resolution of prosthetic vision.

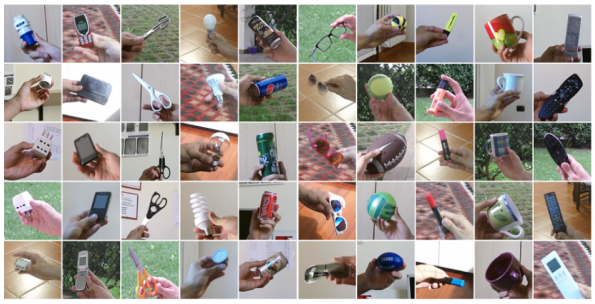

Figure 4. Example images of the 50 CORe50 objects. Each column denotes one of the 10 object categories (Lomonaco & Maltoni, 2017).

For each object, the data set collected 11 video sessions (8 indoors and 3 outdoors) characterised by varied lighting, background, holding hand (left or right), and object occlusions due to the hand position. In each video, the object of interest is handheld using an extended arm and smoothly rotated and moved in front of the camera to showcase the object from various angles. The videos were shot using a Kinect 2.0 sensor at 20 fps over 15 seconds, resulting in 300 350x350 RGB-D frames per video, see Figure 5.

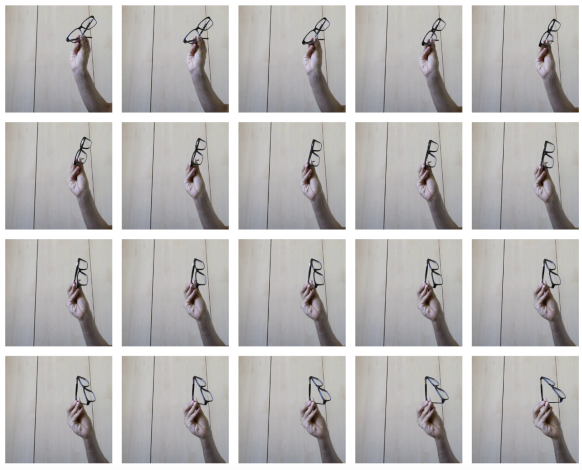

Figure 5. Example of 1-second recording of object 26 (glasses) in session 1.

The data set was chosen as it fulfilled the following requirements:

- The data consist of video input since silicon

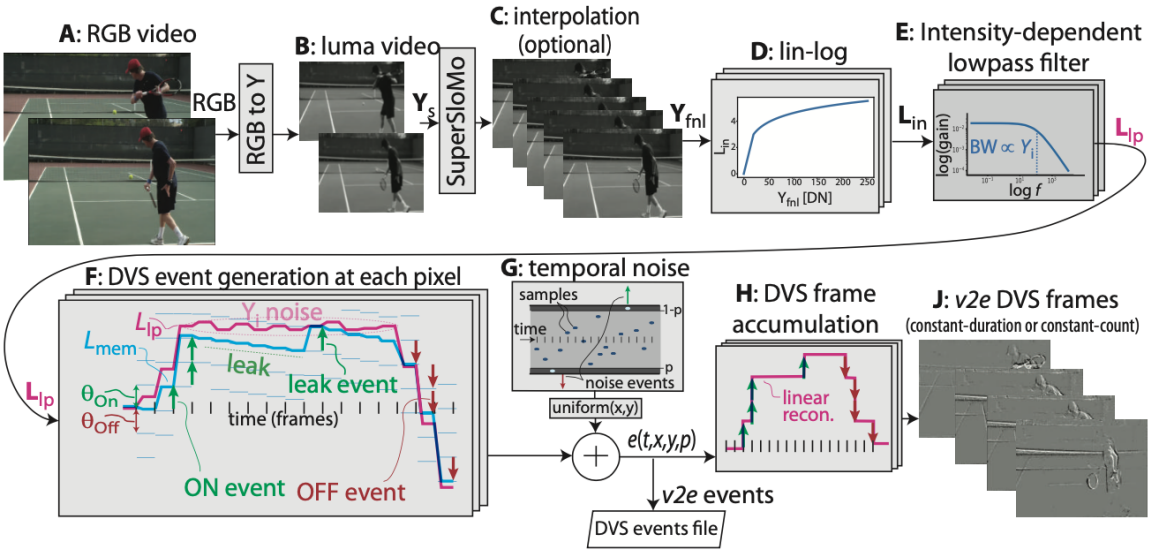

Figure 6. v2e DVS event generation steps. (Hu et al., 2021).

retinas such as the DVS require dynamic visual scenes.

- The data depicts visual scenes that make sense for the task of object detection and for prosthetic vision users.

- The data is clearly and correctly labelled with the bounding boxes being either provided or easily constructed.

2.2 Encoder: Event-based vision

The SPV pipeline using the event-based vision encoder follows previous approaches (Küçükoğlu et al., 2022b; van Steveninck, Güçlü, et al., 2022) of an encoder, followed by a phosphene simulator and ending in a decoder, in this case an object detection decoder, see Figure 7. The event-based vision encoder uses video CORe50 input and transforms it into a stream of events using a python software toolbox known as v2e (Hu et al., 2021). The v2e toolbox generates realistic DVS events from any real conventional frame-based video by modelling an accurate DVS pixel model, see Figure 6.

The DVS events were generated with v2e using the same frame rate as CORe50 (20 fps) and using a preset parameter configuration for v2e. The preset parameter configuration assumes good lighting conditions and low noise levels, which is justified here due to the idealised recording conditions for the CORe50 data.

In order to generate prosthetic vision with the phosphene simulator, the stream of events needs to be transformed into a representation of a traditional image frame. This transformation from an event stream to an event representation is implemented using a python neuromorphic software toolbox known as Tonic (Lenz et al., 2021), which facilitates the manipulation and loading of event-based and spike-based data, see Figure 7.

The pipeline to generate the event representation can be described by the following consecutive transformations, see Figure 8:

1. Merge Polarities: merges the polarities of the

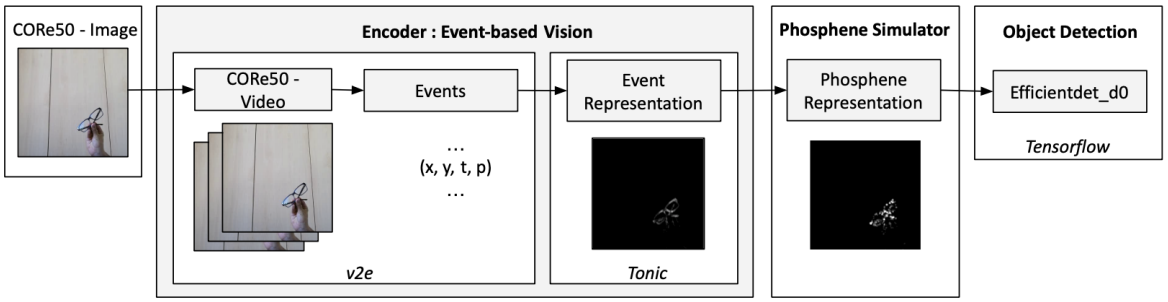

Figure 7. SPV Pipeline using the Event-based vision Encoder.

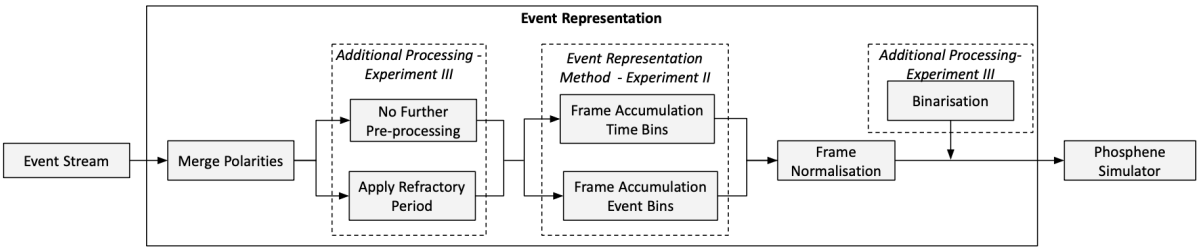

Figure 8. Transformations within Event Representation.

events so that ON and OFF events are merged into singular polarity events, only ON events which are registered by a value of 1. This is to ensure that the OFF events, which are registered as 0 values, do not get factored out in the resulting frame event representation when applying the normalisation step.

2. Event Representation method: creates image frames from events by binning them in two possible ways:
    - Event Frame Accumulation by a constant time interval: accumulate events to frames by a constant number of frames, sliced along the time axis. This has the advantage of resulting in a constant supply of frames meaning that downstream algorithms have a reliable input of frames at regular time intervals. There is no delay in waiting for events so even with a low event rate, there still will be frames transmitted. However as a disadvantage, in the case of a dynamic scene or a scene with a lot of motion, a dense frame representation will be generated. Possibly including motion blur, due to the high event rate.
    - Event Frame Accumulation by a constant number of events per frame: accumulate events to frames by a constant number of frames, sliced along the number of events in the whole recording. This results in event frames whose event count remains constant from frame to frame but whose time interval between frames may vary. This has the advantage of being adaptive to the dynamics present in the scene by having frames with the same amount of event information content. However, this can mean that in the case of a low event rate, no frames will be generated for some time which can be disadvantageous for downstream algorithms.
3. Frame Normalisation: the resulting event representation frames, F, are normalised to pixel values between 0 and 1, according to:

$$F_{norm} = \frac{F - \min F}{\max F - \min F} \tag{7}$$

4. Additional processing method: apply further processing operations besides the event frame accumulation.
    - None: no further pre-processing is applied i.e. no events are dropped or transformed.
    - Refractory Period: drops all events during the specified refractory period to potentially even out the resulting temporal event density. In the case of a pixel getting a big quantity of events in a time interval, it could result in an overly large accumulated value for that pixel, pushing other pixels to very low values during the normalisation step. This would result in some pixels being shown less clearly due their lower normalised values and the over-representation of the event-dense pixel. The application of the refractory period should smooth this effect out to a degree. This transformation is applied before events are accumulated into frames.
    - Binarisation: sets all normalised non-zero event values accumulated within an event frame to a value of 1 to binarise the generated phosphenes. This would result in less intense phosphenes being shown more clearly which could potentially aid in depicting the visual scene. This transformation takes place after the event representation is generated and normalised.

2.3 Encoder: Canny Edge detection

The SPV pipeline using the Canny Edge detection encoder follows the same pipeline as the event-based vision encoder pipeline, see Figure 6, but then uses Canny Edge detection as an encoder and takes grey-scaled CORe50 images as input instead

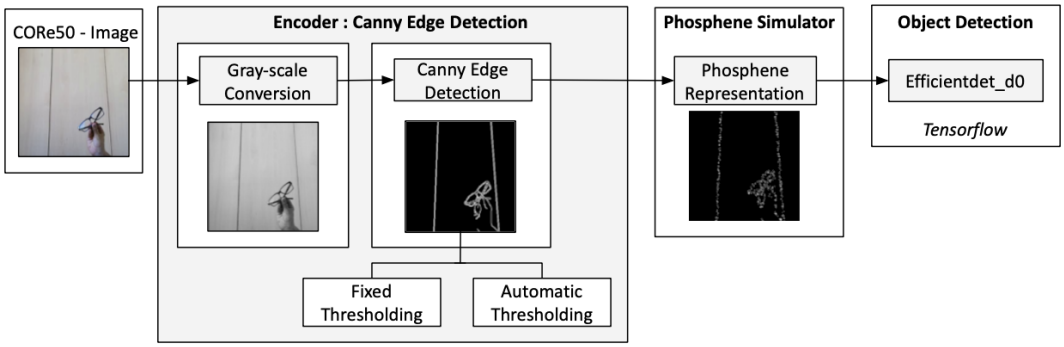

Figure 9. SPV Pipeline using Canny Edge Detection Encoder.

of CORe50 videos, see Figure 9. The Canny Edge detection is implemented using OpenCV (Bradski, 2000), requiring only the specification of the low and high hysteresis thresholds. These thresholds can either be fixed or kept adaptive, both are tested in this thesis. The adaptive thresholds are generated by median value auto-thresholding, as specified by:

$$thr_{low} = \max(0, (1 - \sigma) * mdn(Img)) \quad (8)$$

$$thr_{high} = \min(255, (1 + \sigma) * mdn(Img)) \quad (9)$$

with $\sigma = \frac{1}{3}$.

2.4 Phosphene Simulator

The phosphene simulator was adapted from previous SPV research (van Steveninck, van Gestel, et al., 2022), and simulates phosphenes as white Gaussian blobs with a standard deviation of 1.2 pixel on a jittered, rectangular grid of size either 32 x 32 or 64 x 64, forming the phosphene resolution. The grid sizes correspond to 1024 and 4096 stimulation electrodes respectively. The phosphene simulator generates the phosphenes from the simulation pattern (the frame accumulated event representation frame or the canny edge filtered image) as follows:

1. Resize stimulation pattern to input image size (350x350).
2. Dilate stimulation pattern with a circular kernel of radius 7 pixels.
3. Superimpose resized, dilated stimulation pattern with the phosphene grid to determine the activation of individual electrode sites.
4. Convolve the activation pattern with a Gaussian kernel to generate the phosphenes.
5. Simulate RGB frames by duplicating the resulting phosphene frame for each RGB component.

2.5 Decoder: EfficientDet-D0

As object detection decoder, the EfficientDet-D0 model was utilised. EfficientNet-based models are state-of-the-art object detection models, known to provide the best overall performance with the EfficientDet-D7 model 1536x1536 reaching up to 51.2 mAP on the COCO 2017 data set. The EfficientDet-D0 model 512x512 reaches up to 33.6 mAP on the COCO 2017 data set with a speed of 39 ms. The EfficientDet-D0 model is a one-stage

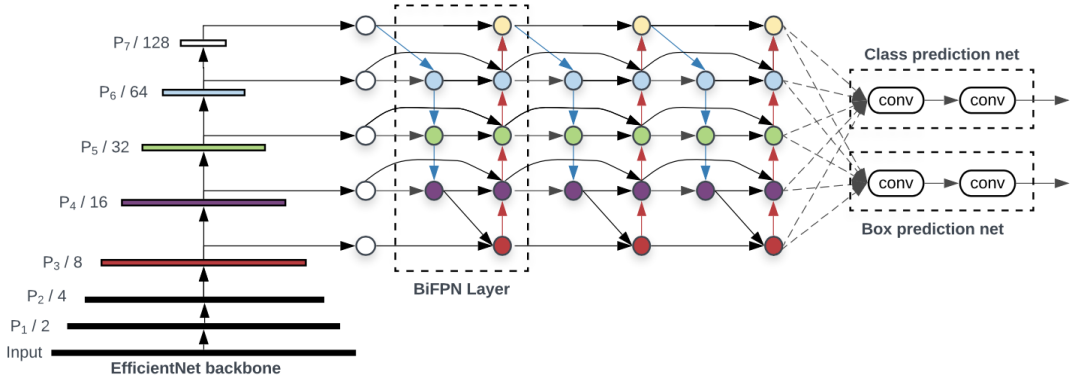

Figure 10. EfficientDet architecture (Tan et al., 2020).

detector comprising 3 parts, see Figure 10 (Tan et al., 2020):

- A pre-trained ImageNet EfficientNet as the backbone network. In the case of EfficientDet-D0, this is EfficientNet-B0. The backbone network serves as the feature extractor for object detection.
- A bidirectional Feature Pyramid Network (BiFPN). The BiFPN network is used to optimise multi-scale feature fusion by serving as the feature network, taking level 3-7 features from the backbone network and repeatedly applying top-down and bottom-up bidirectional feature fusion. For EfficientDet-D0, the BiFPN width is 64 channels, the BiFPN depth is 3 layers and the input resolution size is set at 512.
- A shared class and box prediction network, to which the BiFPN output is given, generating the object class and bounding box prediction. For EfficientDet-D0, the box/class prediction network width is fixed at 3 channels while the depth is fixed at 3 layers.

In this research, EfficientDet-D0 was implemented in TensorFlow 2.0 using the Tensorflow Object Detection API. The weights of EfficientDet-D0 were pre-trained on the COCO 2017 data set and were used to initialise the network with the last layers being fine-tuned for the CORe50 representations in the experiments. For each experiment and experimental condition, the test data set consists of three of the eleven CORe50 sessions, sessions 3,7 and 10, while the remaining 8 sessions were used for the training of the network (Lomonaco & Maltoni, 2017). The training and test sessions were balanced for the expected difficulty with respect to indoor/outdoor scenes, holding hand, and background complexity. Both the training and test data set were shuffled at random for variance reduction to avoid over-fitting and were then batched in batch sizes of 16. For each experimental condition, five experimental runs were performed to account for variations in the runs due to the randomised nature of the training of the model. During each experimental run, the model was trained with an initial cosine decaying learning rate of 0.001 over 30k iterations and L2 regularisation of 4×10^{-5} and was evaluated using the COCO metrics.

3 Experiments and Results

To answer the research questions outlined in the introduction, five experiments are conducted using the above-described SPV pipeline. Each experiment focuses on one particular aspect of event-based prosthetic vision or Canny Edge detected prosthetic vision or their comparison. The EfficientDet-D0 model is trained for each experimental condition of the five experiments. As described in Section 2.5, an mAP optimum is found by searching over multiple iterations and evaluating the performance of each experimental model on the test set, see Figure 11.

As the highest accuracy that can be obtained for all the experimental models is the PASCAL-VOC metric ($mAP^{IoU=0.5}$), it will be used as the main evaluation metric for the performance of each

Table 1: Summary of Results

	mAP @0.5IOU (%)							**Average Training Time (h)**
	run 1	**run 2**	**run 3**	**run 4**	**run 5**	**Mean**	**Std**	
Experiment I - Baseline								
original CORe50 - object	72.10	71.71	64.92	68.00	70.60	69.47	3.0	6:19
original CORe50 - category	82.88	83.25	83.53	81.43	85.91	83.40	1.6	5:34
dvs CORe50 - category	86.98	87.40	87.61	87.91	87.67	**85.21**	5.1	5:33
Experiment II - Event Representation								
10 fps	60.76	59.13	58.40	60.33	60.06	59.74	0.96	5:35
20 fps	60.73	63.29	61.22	52.92	60.91	59.81	3.9	5:34
30 fps	53.34	52.08	52.98	53.29	53.01	52.94	0.51	5:35
40 fps	43.37	42.61	42.78	42.11	43.73	42.92	0.64	5:34
event bins	61.37	61.81	60.47	60.08	61.40	**61.04**	0.73	5:53
Experiment III - Processing Method								
no further processing	60.73	63.29	61.22	52.92	60.91	**59.81**	3.9	5:34
refractory period	62.22	57.77	55.15	60.80	60.17	59.22	2.7	5:34
binarisation	50.86	51.54	50.32	51.09	50.38	50.84	0.51	5:46
Experiment IV - Canny Edge Detection								
20 fps - none	60.73	63.29	61.22	52.92	60.91	**59.81**	3.9	5:34
40 fps - none	43.37	42.61	42.78	42.11	43.73	42.92	0.64	5:34
canny edge - auto	23.31	19.16	18.71	17.60	19.34	19.62	2.1	5:34
canny edge - tight	14.39	15.52	18.48	14.59	16.03	15.80	1.6	5:34
Experiment V - Phosphene Resolution								
20 fps - 64 x 64	60.73	63.29	61.22	52.92	60.91	**59.81**	3.9	5:34
20 fps - 32 x 32	42.52	42.57	42.26	43.22	44.11	42.94	0.75	5:35
canny edge - 64 x 64	23.31	19.16	18.71	17.60	19.34	19.62	2.1	5:34
canny edge - 32 x 32	08.06	12.24	09.02	09.82	10.48	09.92	1.5	5:52

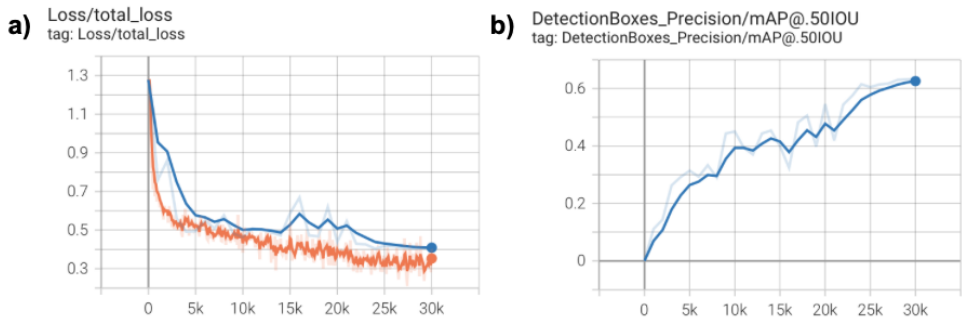

Figure 11. Illustration of an experimental run (the 20 fps event representation for experiment II) during training (orange) and test data (blue) for the curve of the **a)** total loss **b)** $mAP^{IoU=0.5}$.

experimental model. Table 1 shows a summary of the results obtained for each experiment, in particular the obtained mAP (IOU = 0.5) for each experimental runs, the average mAP (IOU = 0.5), its standard deviation, and the average training time of the models. The set-up, motivation and results for each experiment will be discussed below in more detail.

3.1 Experiment I - Standard Sighted Vision Baseline

3.1.1 Set-up and Motivation

The first experiment establishes the EfficientDet-D0 baseline object detection performance fine-tuned for the CORe50 data set for standard sighted vision. Rather than focusing on prosthetic vision, this experiment provides an estimation of how well the chosen object detection model performs when using standard retinal vision and event-based vision in general. The inclusion of event-based vision is used to see whether the DVS is able to provide sufficient information to enable object detection, even for unimpaired vision.

The normally sighted baseline is established by training and evaluating the EfficientDet-D0 model on the non-processed RGB CORe50 images on both the object (e.g. object 26) and category level (e.g. category glasses). Object detection on the object level is only used once to establish the normally sighted baseline. For the rest of the experiments, the category level of object detection is used due to the rudimentary nature of prosthetic vision. We then also expect the object detection on the category level to surpass the object detection on the object level for sighted vision.

EfficientDet-D0 is further also trained on event-based CORe50 images, which can be created with the v2e framework, to establish the normally sighted baseline for DVS frames. The v2e framework allows for video output in the form of a DVS version of the original video input, using a fixed accumulation time of events for the resulting frames. This is used to generate DVS event-based CORe50 images (DVS images) using the same frame rate of the CORe50 data set (20 fps), which models both ON and OFF events as separate polarities in a frame, see Figure 12. Note that this is unlike the frame accumulation along the time axis of the event-based encoder as there the event polarities are merged. We expect that the DVS is able to provide sufficient information to enable object detection but probably not as well as standard RGB frames.

Figure 12. v2e event-generated CORe50 frame of object 26 (glasses) of session 1.

The experimental conditions of experiment I consist of the object detection of:

- Original CORe50 images on the object level, number of classes = 50
- Original CORe50 images on the categorical level, number of classes = 10
- DVS event-based CORe50 images on categorical level, number of classes = 10

3.1.2 Non-prosthetic Baseline Results

As can be seen from Figure 13, the EfficientDet-D0 trained on each individual CORe50 object reached an average accuracy of up to 70% mAP. This is comparable to the accuracy reached when using a COCO pre-trained SSD MobileNetV1 on the CORe50 data set instead of EfficientDet-D0, which achieved an accuracy between 60% and 65% (Lomonaco & Maltoni, 2017). An average mAP of

83% can be obtained by training the EfficientDet-D0 on the original CORe50 object categories and an mAP of 85% when using the DVS event-based CORe50 object categories. As expected, object detection on the category level outperformed the object level. However, surprisingly the DVS frames were able to perform on par with the RGB images with a paired t-test on the mAP(0.5IOU) not revealing a significant preference for either the original or the event-based CORe50 category object detection (p=0.4950).

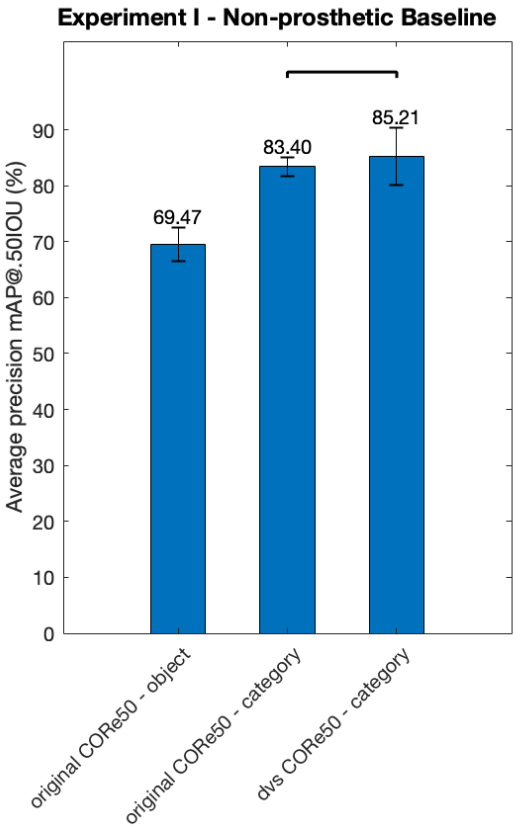

Figure 13. Results of the object detection performance of the three non-prosthetic baselines.

3.2 Experiment II - Event-based Prosthetic Vision: Event Representation

3.2.1 Set-up and Motivation

The second experiment focuses on the optimisation of event-based prosthetic vision object detection by looking at the effect of the selected event representations. As outlined by the event-based encoder of the SPV pipeline, two types of event representations are tested: event frame accumulation by a constant time interval or by a constant number of events. For the time bin frame accumulated representations various possible frame rates commonly used in traditional vision applications are used. We expect to see that the higher the frame rate that is used for time bins, the lower the object detection performance will be for the event representation. This is due to how the higher the frame rate, the lower the number of events will be accumulated per frame meaning that there will be a smaller amount of phosphenes available that can depict the objects. The severity of the expected performance decrease is difficult to predict and is therefore investigated with the experiment.

As for the event count frame accumulation representation, only one is tested: 300 event bins corresponding to the 20 fps of the CORe50 data set. The dynamics of the CORe50 data set frames are relatively small due to it always being a small object rotated at a slow stable pace before a static camera with no rapid unpredictable movements. Therefore we do not expect to see a lot of changes in dynamics so no large changes with regard to the number of events changing from frame to frame hence only one event count frame accumulation representation is tested. We expect to see that the object performance of the event count frame accumulation representation will perform similarly to the time bin accumulation of 20 fps due to the same number of frames meaning that there approximately will be a similar number of events per frame for both representations.

All experimental conditions in this experiment use a phosphene resolution of 64 x 64 and no further additional processing is done for the event representation. The experimental conditions are as follows:

- 10 fps: frame accumulation by 150 time bins, corresponding to a frame rate of 10 fps, the lowest threshold for visual prostheses to benefit from head movements (Li, 2013). As the bounding boxes (bbox) of the CORe50 objects are generated with a frame rate of 20 fps in mind, the bounding box allocation needs to be adjusted when using alternative frame rates. Since the object moves or rotates minimally per ms, the bounding boxes for 20 fps is used instead by taking the bbox for the 20 fps frame closest in time to the adjusted frame rate a_fps (i.e. 10 fps for example) frame, as demonstrated in Figure 14. This bbox allocation for the adjusted frame rate can easily be calculated with $bbox_{a\_fps} = int(f * bbox_{20fps})$ with $f = \frac{20fps}{a\_fps}$.

- 20 fps: frame accumulation by 300 time bins, corresponding to a frame rate of 20 fps. The frame rate of the CORe50 data set and the lowest threshold for real-time SPV frame rates (Chen et al., 2009).

- 30 fps: frame accumulation by number of 450 time bins, corresponding to a frame rate of 30

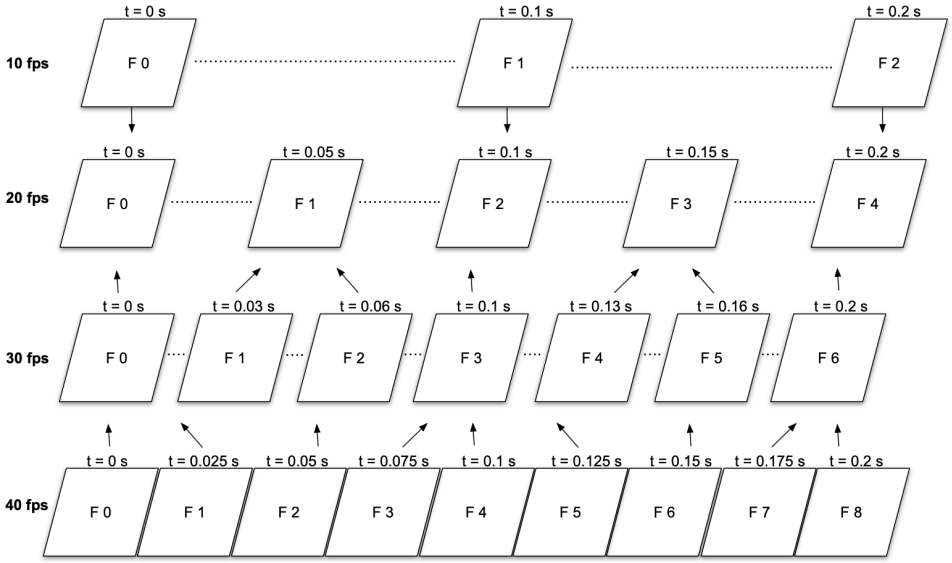

Figure 14. Bounding boxes allocation for the alternative event representation frame rates.

fps, which falls within the range for real-time SPV frame rates (Chen et al., 2009).

- 40 fps: frame accumulation by 600 time bins, corresponding to a frame rate of 40 fps, which falls within the range for real-time SPV frame rates (Chen et al., 2009).
- event bins: frame accumulation by 300 event bins. The bbox allocation for the event bin frame accumulation is generated by taking the bounding boxes for the 20 fps frames of the nearest time point to the lowest time point contained in the event accumulated frame (given by returning the minimum time point in each event frame obtained in the event-based vision encoder).

3.2.2 Event Representation Results

As seen from Figure 15, the highest average mAP that can be acquired with event-based prosthetic vision is 61% with the event bin event frame accumulation event representation. The performance was however not significantly better than the time bin frame accumulation representation of 20 fps, which reached an average mAP of 59.8% (p=0.4696). This falls in line with our expectation of the event count frame accumulation representation performing on a similar level to the time bin frame accumulation representation of 20 fps. However, surprisingly the time bin frame accumulation event representation of 10 fps performed also on par with the time bin frame accumulation representation of 20 fps, reaching an average mAP of 59.7% (p=0.9709). Meaning that there was apparently no extra disadvantage or advantage of having a lower frame rate when using a frame rate between 10 and 20 fps. This is unlike the time bin frame accumulation event representation of 30 fps and 40 fps, which performed significantly worse compared to the 20 fps time accumulation representation, only reaching an average mAP of 53% and 43% respectively ($p < 0.05$ and $p < 0.001$). From 20 fps onward, the results fell in line with our prediction of a higher frame rate resulting in a lower object detection, with an increase of 10 frames per second corresponding to an object detection performance drop of 10% mAP.

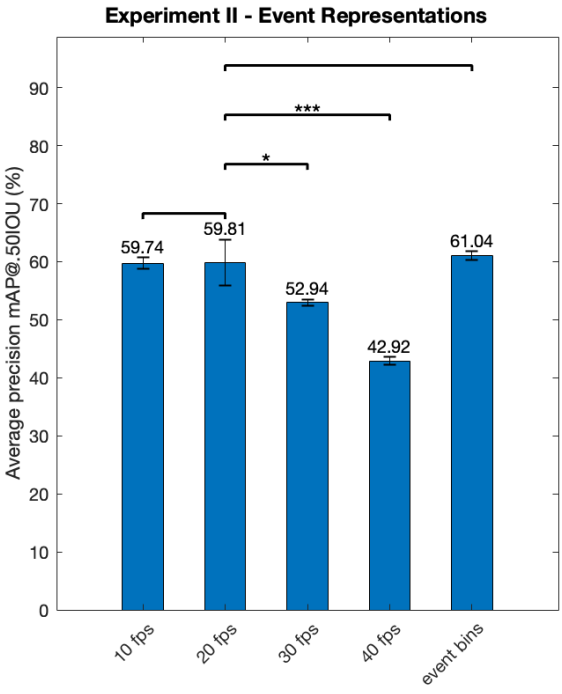

Figure 15. Results of the object detection performance of the tested event representations for event-based prosthetic vision.

3.3 Experiment III - Event-based Prosthetic Vision: Additional Processing

3.3.1 Set-up and Motivation

The third experiment focuses on further optimising event-based prosthetic vision object detection by looking at the effect of additional processing of the event representation. In this experiment, a fixed event frame accumulation representation is further processed using operations outlined within the event representation of the event-based encoder of the SPV pipeline, to see whether the resulting phosphene representation is improved and the object detection performance increased.

All experimental conditions in this experiment use an event representation frame accumulation using 300 frames, sliced along the time axis (20 fps) and a phosphene resolution of 64 x 64. The experimental conditions are as follows, see Figure 16:

- None: no further processing was done. This is the baseline of the 20 fps event representation for event-based prosthetic vision from experiment II.
- Refractory period: a refractory period of 1 ms is applied to even out the resulting temporal event density to a degree. As the data set of CORe50 is not very dynamic, we do not expect to see a big increase in object detection performance but it could potentially lead to a small increase.
- Binarisation: events are binarised in the event representation and no further processing (incl. refractory period) is done to remove the contrast between phosphenes. The removal of contrast could show the outlines of the objects better which we suspect could lead to a better object detection performance.

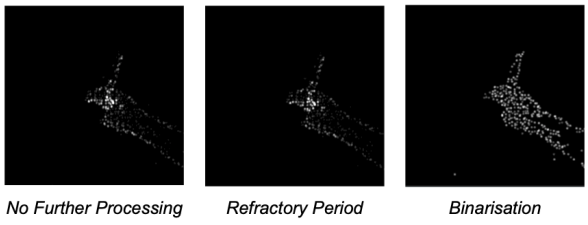

Figure 16. Phosphene vision frame of object 11 (scissors) of session 1 across the various additional processing of the event representation of event-based prosthetic vision.

3.3.2 Additional Processing of Event-based Prosthetic Vision Results

As shown in Figure 17, the processing method of event-based prosthetic vision of the trained EfficientDet-D0 that performed the best was the no further processing method, reaching up to an average mAP of 59.8%. Unlike our expectations, the refractory period processing method did not lead to an increase in performance, reaching an average mAP of 59.2%, nor was this difference significant at all as was revealed with a paired t-test ($p = 0.8281$). Surprisingly, the binarisation processing method did not help the object detection performance. Instead, it resulted in a significant decrease in performance, reaching an average mAP of 50 % ($p < 0.01$).

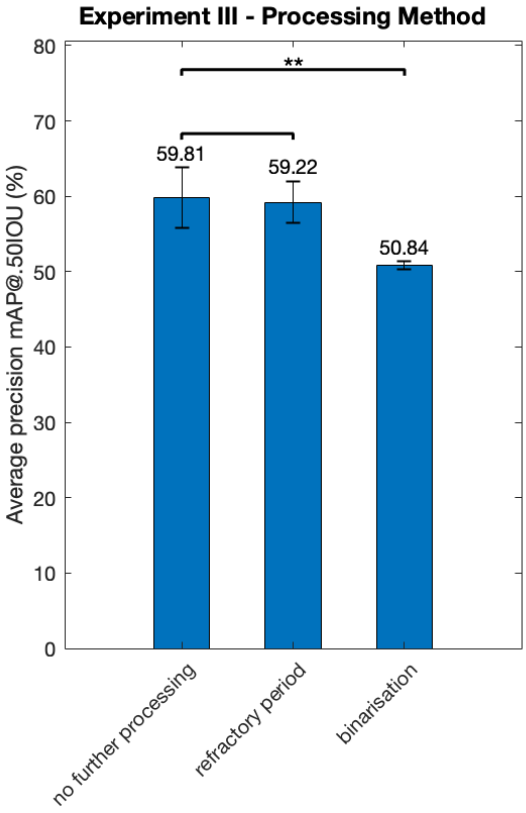

Figure 17. Results of the object detection performance of the tested further processing operations on the event representation for event-based prosthetic vision.

3.4 Experiment IV - Canny Edge Detection Comparison

3.4.1 Set-up and Motivation

The fourth experiment focuses on examining how event-based prosthetic vision compares to Canny Edge prosthetic vision on object detection. In this experiment, it is investigated whether event-based

prosthetic vision indeed outperforms state-of-the-art Canny Edge Detection prosthetic vision as we expected by using a silicon retina as a front-end sensor. How large this expected difference in object detection performance will be is difficult to predict hence the inclusion of both the high and low-end performing event-based prosthetic vision for the comparison to Canny Edge prosthetic vision. Canny Edge prosthetic vision is generated either with optimised adaptive hysteresis thresholds or by using tight hysteresis thresholds. As the adaptive hysteresis thresholds optimise the edges for each frame, we expect to see that Canny Edge prosthetic vision using the auto-generated thresholds will outperform the Canny Edge prosthetic vision using tight thresholds.

All experimental conditions in this experiment use a phosphene resolution of 64 x 64 and are as follows:

- Event - 20 fps: event-based prosthetic vision using frame accumulation by 300 time bins, corresponding to a frame rate of 20 fps. No further processing is done, besides the event frame accumulation. This is the high-end performing event-based prosthetic vision that is used as a comparison.
- Event - 40 fps: event-based prosthetic vision using frame accumulation by 600 time bins, corresponding to a frame rate of 40 fps. No further processing is done, besides the event frame accumulation. This is the low-end performing event-based prosthetic vision that is used as a comparison.
- Canny edge - auto: Canny Edge prosthetic vision using median value auto-generated thresholds to optimise the Canny edges for each frame.
- Canny edge - tight: Canny Edge prosthetic vision using tight fixed thresholds of 225 and 250 for the low and high hysteresis thresholds respectively. The threshold values are chosen as the tight threshold as they reduced the number of edges as much as possible as evaluated by eye.

3.4.2 Comparison of Event-based Prosthetic Vision to Canny Edge Detection Prosthetic Vision Results

The highest performance that could be reached for prosthetic vision using Canny Edge detection was an average mAP of 19.6% with the auto-generated Canny Edge thresholds, see Figure 18. As expected, the performance for the Canny Edge detected prosthetic vision using tight edges was lower but not significantly, reaching up to 15.8% average mAP ($p = 0.0537$). Falling in line with our expectations, the auto-generated Canny Edge detected prosthetic vision performed in comparison to event-based phosphene significantly worse. With the performance of auto-generated Canny Edge detected prosthetic vision only reaching a third of the average mAP of the higher-end performing event-based prosthetic vision ($p < 0.001$) and half of the lowest-end performing event-based prosthetic vision mAP ($p < 0.001$).

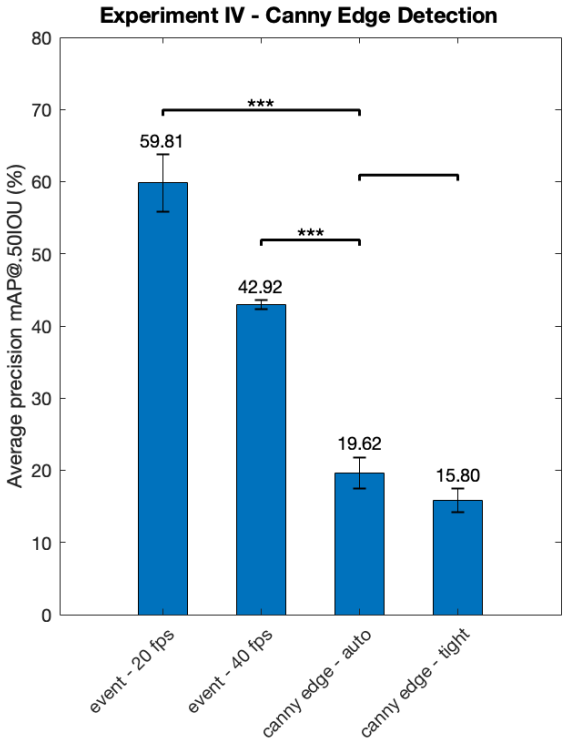

Figure 18. Results of the object detection performance of the comparison of event-based prosthetic vision to Canny Edge Detection prosthetic vision.

3.5 Experiment V - The Effect of Phosphene Resolution

3.5.1 Set-up and Motivation

The fifth experiment focuses on examining the effect of the phosphene resolution for event-based and Canny Edge prosthetic vision on object detection. As an optimistic future aspiration phosphene resolution of 64 x 64 was used throughout previous experiments, this experiment investigates how much the object detection performance drops when using the currently highest phosphene resolution of 32 by 32 for both event-based and Canny Edge prosthetic vision instead. We expect to see that for both event-based and Canny Edge prosthetic vision the object detection performance drops when using the lower phosphene resolution but that event-based prosthetic vision still performs better than

Canny Edge prosthetic vision.

The experimental conditions of this experiment are as follows:

- event - 64 x 64: event-based prosthetic vision using frame accumulation by 300 time bins (20 fps) and a phosphene resolution of 64 x 64. No further processing is done, besides the event frame accumulation.
- event - 32 x 32: event-based prosthetic vision using frame accumulation by 300 time bins (20 fps) and a phosphene resolution of 32 x 32. No further processing is done, besides the event frame accumulation.
- Canny edge - 64 x 64: Canny Edge prosthetic vision using median value auto-generated thresholds and a phosphene resolution of 64 x 64.
- Canny edge - 32 x 32: Canny Edge prosthetic vision using median value auto-generated thresholds and a phosphene resolution of 32 x 32.

3.5.2 Effect of the Phosphene Resolution for Prosthetic Vision Results

As expected, event-based prosthetic vision outperforms Canny Edge prosthetic vision even when using a lower phosphene resolution, see Figure 19. Using a phosphene resolution of 32 by 32, an average mAP of 42.9% could be reached with event-based prosthetic vision, which is significantly better than the auto-generated Canny Edge prosthetic vision performance of 9.9% ($p < 0.001$). In line with our expectations, the performance for both the event-based and the Canny Edge prosthetic vision significantly drops when using the lower phosphene resolution. However, this decrease was larger than we originally expected with the average mAP going from 59.8% to 42.9% ($p < 0.001$) for event-based prosthetic vision and the Canny Edge prosthetic vision performance decreasing from 19.6% to 9.9% ($p < 0.01$).

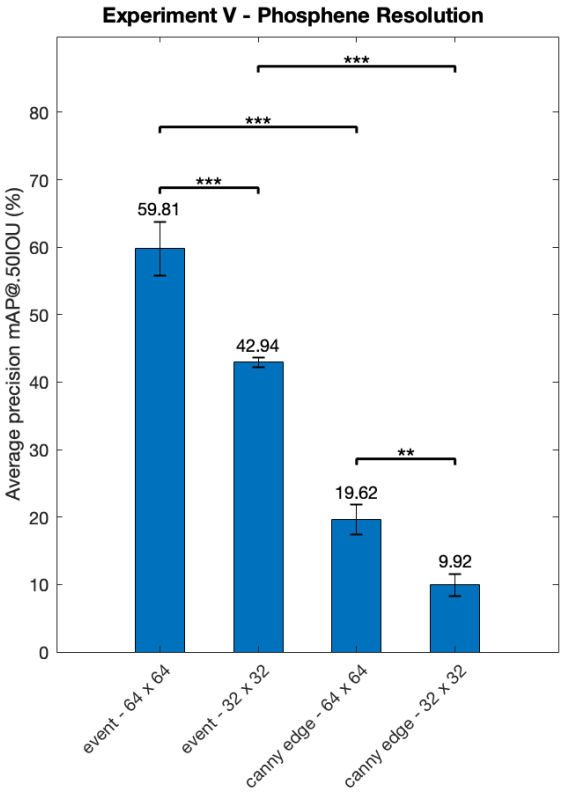

Figure 19. Results of the object detection performance of the effect of the phosphene resolution for event-based and Canny Edge prosthetic vision.

4 Discussion

In this work, the potential of using a DVS silicon retina as a front-end sensor for visual prosthetic vision was evaluated. This was investigated with an object detection task of everyday objects using the Simulated Prophetic Vision framework and the EfficientDet-D0 model. Besides establishing a baseline of how well event-based prosthetic vision performs at object detection, it was assessed how different event processing and representations affect the performance of event-based prosthetic vision. Furthermore, the performance of event-based prosthetic vision was compared to Canny Edge Detection prosthetic vision along with how the phosphene resolution affected these performances. Below, these findings are discussed in further detail and some directions for future research are provided.

4.1 The Baseline Performance of Event-based Prosthetic Vision

As the results presented in the preceding section demonstrate, the event-based prosthetic vision is able to reach up to a relatively high performance, 60% mAP compared to the standard retinal and

event-based vision mAP of ± 85% for the CORe50 data set. It is difficult to relate this performance to other findings of prosthetic vision research, as this is to our current knowledge, the only SPV research that focuses on object detection. Other research mainly consists of either object recognition or object localisation, though some may use object detection or recognition computer algorithms to generate the simulated phosphene vision during the task (Hu et al., 2014; Macé et al., 2015; Rueckauer & van Gerven, 2022). However, the event-based prosthetic object detection performance seems to fall in line with the recognition and localisation SPV research.

For example, using salience segmentation, Li et al found that human participants were able to get up to ± 70% recognition accuracy for the phosphene vision of 40 objects with a phosphene resolution of 32 by 32 (Li et al., 2018). Hu et al used a discriminating task of objects with similar profiles, consisting of objects from 3 image data sets (cups, tomatoes, cars) that contained 10 images of different objects, to test the effects of phosphene array parameters for simulated prosthetic vision using human subjects. They were able to get an accuracy level of 78.5% under the optimum condition (resolution: 32 × 32, grey level: 8, distortion: k = 0, dropout: 0%) (Hu et al., 2014). Mace et al performed a reach-and-touch task requiring object discrimination and localization in which SPV was coupled with object recognition system SPikeNet Vision (SNV). Participants were asked to locate a precise object among seven common objects positioned on a table, with the object of interest being recognised and localised within the camera image using SNV and its position displayed by switching on a unique phosphene at the corresponding location. The average time for correct reaching was 18.2 s and the accuracy was 85.4% (Macé et al., 2015). Finally, Rueckauer et al were able to get up to a DNN accuracy of 91% for phosphene event-based vision and up to 94% for DVS event frames for a DVS gesture recognition task of 11 gestures, using a DNN classifier (Rueckauer & van Gerven, 2022).

4.2 Optimising Event-based Prosthetic Vision

Experiments II and III were conducted to optimise event-based prosthetic vision by assessing the effect of various event representations and further event processing on object detection performance. For the chosen event representation, a time bin accumulated frame of 10 fps and 20 fps along with the event bin frame accumulation seems to perform the best with no significant difference between them. However, the performance seems to significantly drop when using a time bin accumulated frame of 30 fps and even lower when using a time bin accumulated frame of 40 fps. As briefly mentioned, this drop in performance when using a higher frame rate could be due to how the higher the number of frames is, the lesser number of events are accumulated in the resulting frames. This in turn could result in an object being depicted less clearly due to a lesser number of phosphenes being available. This would also explain why the event bin frame accumulation is on par with the time bin accumulated frame of 20 fps, as they both correspond to 300 frames in total with them approximately having a similar number of events across frames.

See Figure 20, for a demonstration of the differences in the accumulation of events in the image frames for the various event representations of experiment II. As demonstrated with this figure, the phosphene distribution with regard to the count, intensity and density of the phosphene of the higher frame rates time bin event representation of 30 fps and 40 fps clearly differs compared to those of 10 fps and 20 fps. Additionally, the figure also demonstrates the frequency of the output of the event representations. The time bin event representations output frames with a consistent frequency whereas the event bin event representation only outputs one image frame within the time interval of 6.5 and 6.6 seconds with the next image frame being outputted at 6.66 seconds, showcasing its varied time interval of frames.

The phenomenon of higher frame rates resulting in fewer phosphenes could be explained in part due to how the events are generated. This SPV research uses the recordings of the CORe50 data set which were recorded with a traditional camera using a frame rate of 20 fps and then synthetically generates events from these recordings as an approximation of an actual physical DVS camera. As a result, the events are generated according to a 20 fps event camera. So in the case of a higher frame rate time bin representation such as 30 fps or 40 fps, the finite event stream of a 20 fps event camera is accumulated into event representations meant for a higher frame rate. Therefore, it is possible that the issue of a lesser depicted phosphene representation due to a higher frame rate will not be an issue when using either a recording or physical DVS camera with the corresponding higher frame rate.

This is not the only limitation that is introduced due to not using a real DVS camera. In a realistic setting, one would not have the full event sequence available before the accumulation of events in frames. Instead, events are accumulated in an event representation according to its chosen accumulation method in real-time. As a result, the advantages, disadvantages and subjective experience of the event representations are not explored fully in this research. For example, human participants

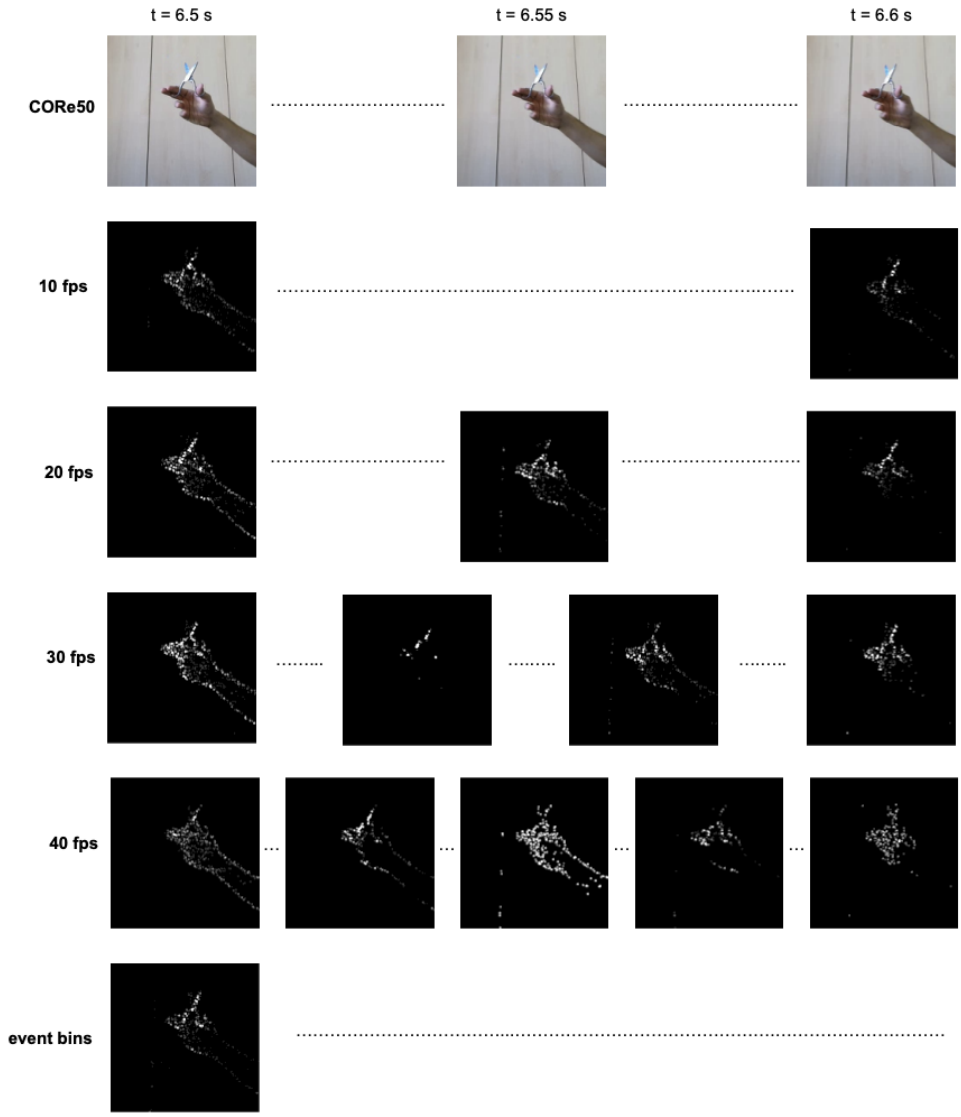

Figure 20. Comparison Event Representation Image Frames for object 11 (scissors) of session 1 corresponding to CORe50 frames 130 (6.5 s), 131 (6.55 s) and 132 (6.6 s).

might prefer an event frame accumulation along a pre-determined count of events such as with the event bins frame accumulation to avoid overloading the subject with visual information or being supplied with useless information. On the other hand, it might be that people dislike the inconsistent output characteristic of this method and prefer the reliable output of a time interval event frame accumulation instead. It is difficult to determine the event representations fully without testing with human participants. Additionally, people might prefer and benefit more from a higher fps rate when using prosthetic vision than a downstream algorithm would.

This research however is not aimed to capture the evaluation and optimisation of event representations for event-based prosthetic vision in its entirety. But rather, it aims to examine and establish whether the premise of event-based prosthetic vision has any real scientific merit and provides a starting point for its facilitation of further development and research. With these research aims in mind, this SPV research chose to select a data set carefully instead of creating a new data set with a physical DVS camera aimed at prosthetic vision research, which would have been a waste of resources if event-based prosthetic vision proved to be more unfruitful than previously predicted. Moreover, the way the SPV pipeline is set up allows for the easy implementation of switching the event stream of the event-based encoder to an event camera from a physical DVS camera for the extension of future research with a real DVS camera. Lastly, the matter of the full sequence of events being available before event accumulation in event representation, does not influence the evaluation of the event represen-

tations from the perspective of the object detection model as only the resulting image frames are evaluated.

Still the results of experiment II are promising, as not only was it shown that event bins frame accumulation was able to reach adequate performance levels, the same could be said for the time accumulated event frames of 10 fps. As mentioned previously, 10 fps was shown to be the lowest threshold for visual prostheses that could benefit from head movements (Li, 2013).

The event-based prosthetic vision in this study was able to reach the 60% mAP performance, using the event representation of the time accumulated event frames of 20 fps, without any further additional event processing as experiment III showed. That the flatting of the temporal event density by applying a refractory period did not result in an increase in object detection performance at all, could be seen as surprising. However, a reason for this could be due to the low dynamics present within the CORe50 data set resulting in a relatively low event density level differences in the data set. The 'clean' DVS model parameter setting of the v2e toolbox, which specifies the DVS events generation for the CORe50 data set by setting the parameters of the DVS pixel model, might do enough evening out by de-noising to account for these low dynamics that applying a refractory period is not necessary. This v2e setting automatically applies a denoising transformation by generating clean events, with lesser non-idealities, which is easily replicable when using a physical DVS camera by specifying similar DVS parameters as the clean v2e DVS model.

Remarkably, binarising the events did not seem to aid the distinction between objects for the event-based phosphene vision, it made it worse. An explanation for this could be that the phosphenes are clumped together more when binarising them, resulting in the loss of nuance of visual information due to the loss of edges/boundaries within and between the visual objects, making it more difficult for the object detection model to discriminate between different objects. Naturally, it still needs to be determined whether this is also the case when using human participants to do the discrimination, who might prefer the more joined phosphene representation of the object when binarising events.

4.3 Comparison of Event-based Prosthetic Vision and Canny Edge Prosthetic Vision

As shown by the result of experiment IV, event-based prosthetic vision significantly outperformed Canny Edge prosthetic vision with the best-performing version of Canny Edge prosthetic vision not even reaching the lowest-performing event-based prosthetic vision. This could be because the DVS is better at filtering out the more noisy backgrounds compared to Canny Edge detection. Canny Edge detection is an edge-extracting algorithm and in the case that the background is found to consist of strong enough edges or strong edge-connected weak edges, it will be included in the resulting Canny Edge detected image. This difference in the ability to filter out backgrounds of the two pre-processing techniques is demonstrated in Figure 21 below. In this Figure, CORe50 object 11 (scissors) was depicted for both the auto-generated Canny Edge detection representation and the 20 fps time accumulated event frame representation and their resulting phosphene vision representation using the 64 x 64 phosphene resolution for the test data set sessions.

Since the CORe50 data sets were balanced for the expected object detection processing difficulty, the three different test sessions clearly show how the different pre-processing strategies convert the CORe50 image into an image frame representation and its resulting simulated phosphene vision representation. For example, session 3 consists of a simple indoor scene with soft artificial light resulting in little to no shadows. Besides the CORe50 object, papers are hung on the wall in the background whose edges and writing are partially picked up by the Canny Edge detection and are thus also represented with phosphenes in the Canny Edge detected SPV image. The event-based representation has completely filtered out the hung papers as they are static objects given that the camera remained static as well during the filming of the CORe50 objects, leaving only the object and the hand to be represented by the phosphenes.

Session 7 also consists of an indoor scene but with a floor ledge depicted in the background instead. Additionally, natural lighting comes from a window, casting harsh shadows in both the background and on the object holding hand. Besides the object and the ledge being depicted, Canny Edge detection also picks up on the shadows in the background. These shadows are not picked up by the event-based representation however the DVS did pick up on the shadows on the hand. This is likely due to how when the hand moves the object around, the shadows in the background remain relatively static but the light and shadows on the hand will move as well resulting in them being captured in events.

Session 10 depicts an outdoor session consisting of a bush in the background with natural light, in addition to some backlight. Canny Edge detection represents the bush almost in its entirety, making it hard to identify and detect the scissors in the hand as they blend in with the bush, for both

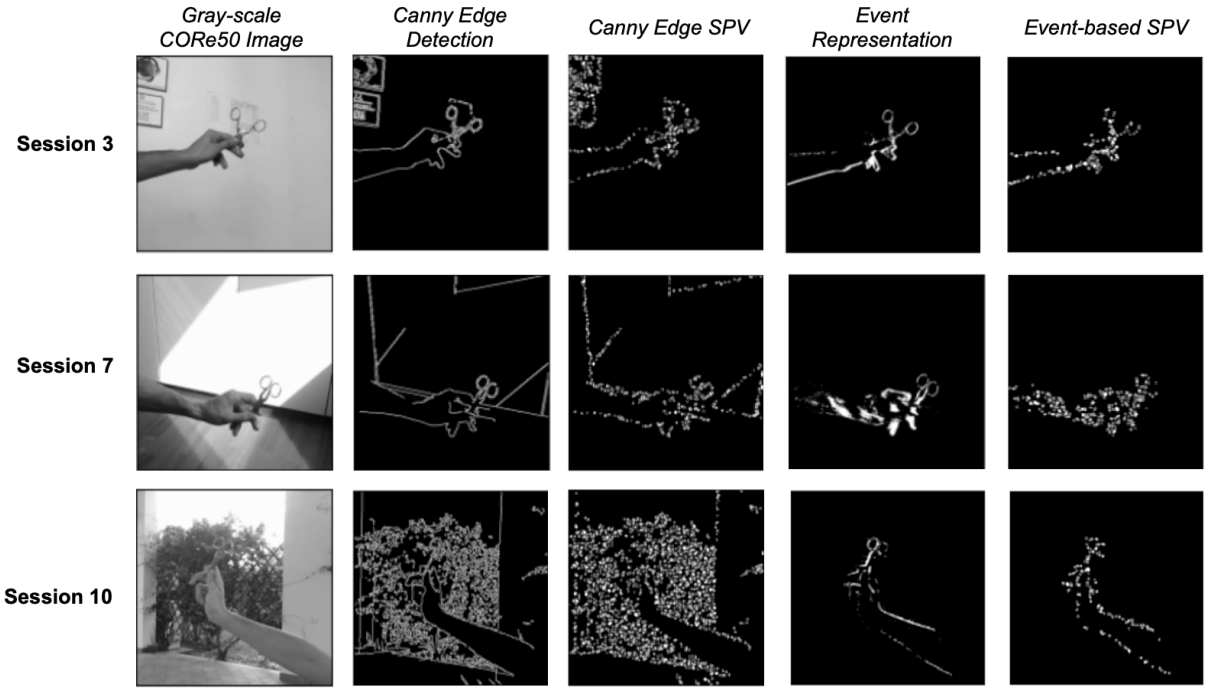

Figure 21. Comparison of Canny Edge detection and event-based phosphene representations.

the Canny Edge detected image and its resulting phosphene representation. On the other hand, the DVS does not represent the bush at all in events, leaving the object to be clearly displayed in the resulting phosphene representation.

These sessions demonstrate how the complexity of the visual scene affects Canny Edge detection and as a result makes it more difficult for the object to be detected and classified accordingly. As this difference is due to the inner workings of the pre-processing methods, event-based prosthetic vision still outperforms Canny Edge prosthetic vision. Because even though there might be more frames where the objects are depicted less clearly when using the 30 and 40 fps time accumulated event frame representation compared to the 20 fps time accumulated event frame representations, they are still able to depict the object better for object detection than Canny Edge detection due to the DVS filtering out the background more. The Canny Edge detection using tight fixed hysteresis thresholds performs even worse than when using auto-generated Canny Edge detection, as the edges, in that case, are no longer optimised for each frame resulting in making it more challenging to depict the object adequately for object detection.

As acknowledged in earlier sections, it has to be seen whether human subjects favour event-based prosthetic vision over Canny Edge prosthetic vision as well. It might be possible that they prefer having more available information about the visual environment as provided by the edges in the background by Canny Edge detection, even if it might make objects more difficult to detect and identify.

4.4 Phosphene Resolution

Throughout the experiments, a phosphene resolution of 64 by 64 was used, corresponding to 4096 stimulation electrodes. As the task of object detection is relatively complex and could require a high number of phosphene to represent the objects accurately, the optimistic case of 4096 electrodes was used. However, the currently highest cortical visual prosthesis has 1024 stimulation electrodes. Furthermore, even when a prospective cortical visual prosthesis is theoretically able to have 4096 stimulation electrodes, it is unlikely that all electrodes will be usable and available at any given moment due to drop-out and severed connections to the neural tissue. Furthermore, unlike in the used simplified phosphene simulator, real phosphenes will not be nicely distributed in a regular grid over the visual field. As not only are cortical maps arranged with a larger number of neurons being allocated for the processing of the visual field centre and a lesser amount for peripheral vision but they are also arranged into clusters, with a cluster being defined as a group of cortical maps with parallel eccentricity representations while within in a cluster, the visual field map angle representations alternate (Wandell et al., 2005; Wandell et al., 2007). This means that phosphenes in a realistic setting will probably only cover some unconnected patches of the visual scene at a small visual angle.

For these reasons, the event-based simulated

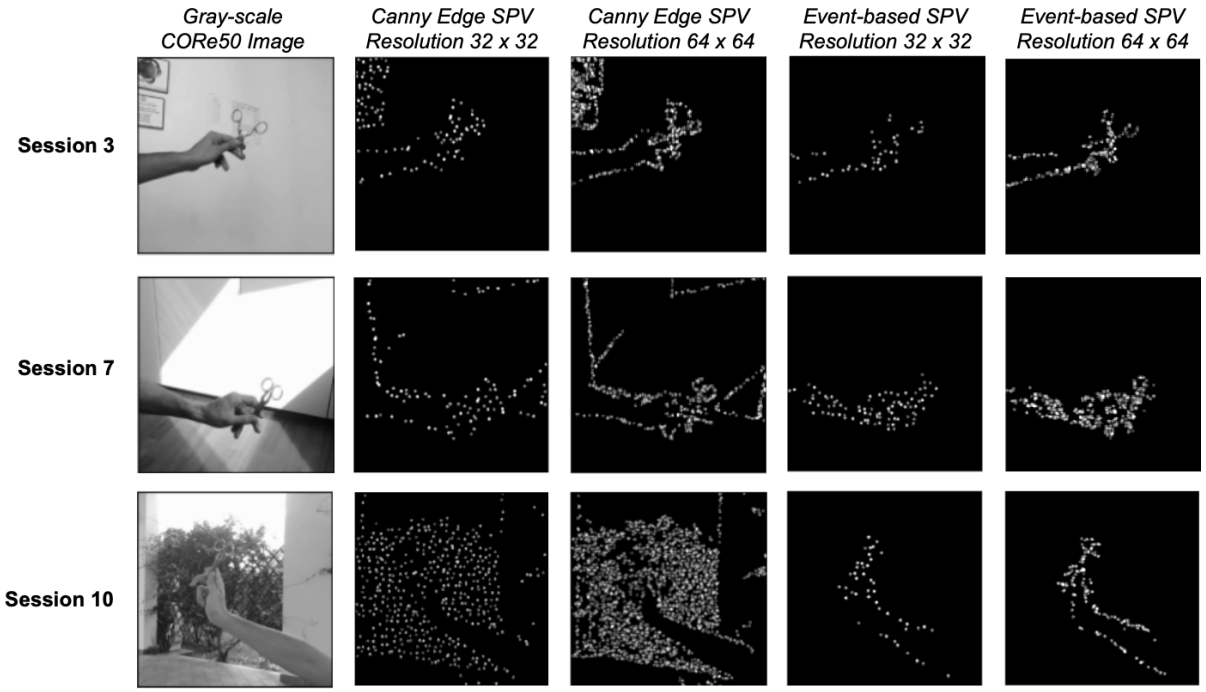

Figure 22. Comparison of Canny Edge detection and event-based SPV for different phosphene resolutions.

prosthetic vision using the 20 fps time bin accumulated frame representation and the auto-generated Canny Edge detection prosthetic vision was computed as well with a phosphene resolution of 32 by 32 instead of 64 by 64. This was done to provide a rough estimate of the expected accuracy drop when using a current theoretical optimal visual prosthetic or when 75% of the electrodes are unable to provide stimulation. As the results showed, the event-based simulated prosthetic vision significantly dropped from 60% to 43% mAP while the Canny Edge simulated prosthetic vision significantly dropped from 20% to 10% mAP. Event-based simulated prosthetic vision still significantly outperforms Canny Edge detection, even when lowering the phosphene resolution.

As Figure 22 demonstrates, a lower phosphene resolution means that there are fewer phosphenes available to depict the object. As a result, it is harder to decipher the object for both preprocessing techniques. For the Canny Edge detection prosthetic vision, much-needed phosphenes are allocated to redundant edges, leaving fewer phosphenes to depict the object in question. For the event-based prosthetic vision, a lot of nuance of the object is lost as a result of the fewer phosphenes. Session 7 especially illustrates this as the lighting on the hand uses up phosphenes that would be more useful for the object instead. Whether the effect of the drop in accuracy is as noticeable when using human subjects, is yet to be seen. For now, the issue of a lower phosphene resolution as outlined in the findings above highlights the importance of the development of more resistant and a higher amount of available stimulation electrodes to create better visual prostheses.

4.5 Limitations and Future Work

As this SPV research is to our knowledge, one of the only SPV studies that focus on object detection and use event-based vision, a few issues and limitations that were discovered in the present study can provide some directions for future research. First, the limited amount of data sets appropriate for event-based vision. There are few data sets that consist of videos, with even lesser ones that are clearly and correctly labelled. A simulated data set to virtually recreate a hallway environment from an earlier behavioural prosthetic research (van Steveninck, van Gestel, et al., 2022) with the use of Virtual Reality was considered briefly for the testing of a navigation task instead. This however revealed the limitation of event-based vision requiring dynamic visual scenes with no large temporal and spatial jumps between movements, which was not possible with that iteration of the visual environment data set. Second, beyond navigation tasks, it was difficult to relate the findings of this study with other SPV studies. To streamline the development and research of prosthetic vision better, benchmarks and shared peer-reviewed databases for prosthetic vision should be discussed and established. Third, in this study object detection was implemented using

TensorFlow API. This resulted in only being able to fine-tune the last layers of the EfficentDet-D0 model for prosthetic vision. However, it might be more favourable for event-based prosthetic vision to be able to train the first layers as well as to take full advantage of the spiking behaviour of the silicon retina from an encoding perspective. Fourth, the task of object detection using object detection algorithms. There is the issue that some objects are shown from unfavourable angles in some frames within the CORe50 data set which would make it difficult to facilitate object recognition in those frames for object detection algorithms. This might be less of a problem in practice with human subjects as objects in real life are not restricted to 2D as they are with images, but are instead situated in a 3D space in which it is relatively straightforward to change one's perspective to be more favourable if unsure. Fifth, the simulated prosthetic vision that was used in the current study is a simplified phosphene simulator. Future research could extend the created pipeline with a more realistic and biologically plausible phosphene simulator (van der Grinten et al., 2022).

Finally, this study only provides a starting point for event-based prosthetic vision, now that it has shown potential as a front-end sensor for a visual prosthesis that might be more advantageous than Canny Edge detection. There are still a lot of research venues that remain unexplored. Some are relatively simple such as the extension of the pipeline with more diverse and complex event representations such as voxel grids or changing the event-based encoder from the v2e computational DVS model to a physical DVS camera. The latter would allow for the testing of the pipeline with human participants, which as discussed in previous studies are necessary to see whether the findings expand to human subjects as well and to what degree. The extended or current pipeline could be used for other data sets and other, more complex task domains such as navigation tasks. This would be beneficial to establish the reach of the potential of event-based prosthetic vision.

A more complex extension would be to incorporate other visual information into the event-based encoder or use the event-based encoder as a component for a more complex encoder. The event-based vision of the DVS is limited by three factors: its absence of intensity information of the visual field, the requirement of motion to generate events and the high amount of generated background events generated with a moving camera. The latter two can partially be mitigated by learning how to control the camera movement to effectively sample the visual scene and by the incorporation of additional noise-filtering algorithms. This was not needed for the present study as the camera was held stable during the filming of the CORe50 objects with the objects being moved in front of the camera lens instead. This is not an unrealistic scenario for a prosthetic user to encounter in their daily life as for example sorting out a drawer, sorting laundry or eating would result in a similar visual scene with regard to dynamics.

The former limitation could be diminished by the incorporation of other visual information, this can be in the form of a normal camera to obtain pixel intensity information or using another silicon retina such as the ATIS or the DAVIS instead of the DVS as the event-based encoder. The ATIS (Asynchronous, Time-based Image Sensor) is a silicon retina which has a DVS subpixel which triggers another subpixel to read out the absolute intensity (grayscale value) as an exposure measurement if a brightness change of a certain magnitude has been detected. The ATIS models a simplified functional Parvo-cellular pathway model in this manner but has the disadvantage that its pixels are at least double the area of DVS pixels (Gallego et al., 2020; Posch, 2012). The DAVIS (Dynamic and Active Pixel Vision Sensor) combines a conventional frame-based active pixel sensor (APS) in the same pixel as the DVS pixel to get intensity readouts alongside event generation (Gallego et al., 2020). Due to a shared photodiode, the APS is much smaller compared to the ATIS but the dynamic range APS readout is on par with conventional cameras. This puts a big limitation on what the DVS made so attractive for visual prostheses in the first place: its high dynamic range useful for various lighting conditions, its high temporal resolution and its low latency. So the incorporation of other visual information should be done in a careful manner to not lose the advantages of event cameras in the process.

It is likely that a final implementable version of a visual prosthesis will consist of a combination of traditional frame sensors and event sensors like the DVS to either switch between the two information sources depending on which one is more suitable for the current visual environment and task. Or a method which uses both information streams such as for example with deep learning encoders, which have been of interest of late. Future work could look into how a deep learning approach compares to event-based vision or how event-based vision could be incorporated within deep learning encoders. The pipeline of this SPV research can provide the facilitation for the above-outlined future research endeavours.

5 Conclusion

In this SPV research, a pipeline for event-based simulated prosthetic vision was presented. The

pipeline was used to examine the potential of the silicon retina DVS as a front-end sensor for prosthetic vision using object detection of everyday objects. It provided a guideline on the effects of various event processing and representations for event-based prosthetic vision. Moreover, the findings showed how event-based prosthetic vision exceeded traditional Canny Edge prosthetic vision while also advocating for the development of higher phosphene resolution for future visual prostheses. This study provides a foundation for further testing of the potential of event-based prosthetic vision for future visual neuroprosthetic research.

Acknowledgement

This thesis was part of the European Union's Horizon 2020 research and innovation program under grant agreement No 899287.
All source code for this research available in Github repository: github.com/ChantalvDuin/event-prosthetic-vision.

References

Bardow, P., Davison, A. J., & Leutenegger, S. (2016). Simultaneous optical flow and intensity estimation from an event camera. *Proceedings of the IEEE conference on computer vision and pattern recognition*, 884–892.

Bollen, C. J., van Wezel, R. J., van Gerven, M. A., & Güçlütürk, Y. (2019). Emotion recognition with simulated phosphene vision. *Proceedings of the 2nd Workshop on Multimedia for Accessible Human Computer Interfaces*, 1–8.

Bourne, R., Steinmetz, J. D., Flaxman, S., Briant, P. S., Taylor, H. R., Resnikoff, S., Casson, R. J., Abdoli, A., Abu-Gharbieh, E., Afshin, A., et al. (2021). Trends in prevalence of blindness and distance and near vision impairment over 30 years: An analysis for the global burden of disease study. *The Lancet global health*, *9*(2), e130–e143.

Bradski, G. (2000). The opencv library. *Dr. Dobb's Journal: Software Tools for the Professional Programmer*, *25*(11), 120–123.

Canny, J. (1986). A computational approach to edge detection. *IEEE Transactions on pattern analysis and machine intelligence*, (6), 679–698.

Chen, K., Li, J., Lin, W., See, J., Wang, J., Duan, L., Chen, Z., He, C., & Zou, J. (2019). Towards accurate one-stage object detection with ap-loss. *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 5119–5127.

Chen, S. C., Suaning, G. J., Morley, J. W., & Lovell, N. H. (2009). Simulating prosthetic vision: I. visual models of phosphenes. *Vision research*, *49*(12), 1493–1506.

Chen, X., Wang, F., Fernandez, E., & Roelfsema, P. R. (2020). Shape perception via a high-channel-count neuroprosthesis in monkey visual cortex. *Science*, *370*(6521), 1191–1196.

Cho, D.-i. D., & Lee, T.-j. (2015). A review of bioinspired vision sensors and their applications. *Sensors and Materials*, *27*(6), 447–463.

Christie, B. P., Ashmont, K. R., House, P. A., & Greger, B. (2016). Approaches to a cortical vision prosthesis: Implications of electrode size and placement. *Journal of neural engineering*, *13*(2), 025003.

Dagnelie, G., Barnett, D., Humayun, M. S., & Thompson, R. W. (2006). Paragraph text reading using a pixelized prosthetic vision simulator: Parameter dependence and task learning in free-viewing conditions. *Investigative ophthalmology & visual science*, *47*(3), 1241–1250.

Delbrück, T., Linares-Barranco, B., Culurciello, E., & Posch, C. (2010). Activity-driven, event-based vision sensors. *Proceedings of 2010 IEEE International Symposium on Circuits and Systems*, 2426–2429.

Fehervari, T., Matsuoka, M., Okuno, H., & Yagi, T. (2010). Real-time simulation of phosphene images evoked by electrical stimulation of the visual cortex. *International Conference on Neural Information Processing*, 171–178.

Fernandez, E. (2018). Development of visual neuroprostheses: Trends and challenges. *Bioelectronic medicine*, *4*(1), 1–8.

Gallego, G., Delbrück, T., Orchard, G., Bartolozzi, C., Taba, B., Censi, A., Leutenegger, S., Davison, A. J., Conradt, J., Daniilidis, K., et al. (2020). Event-based vision: A survey. *IEEE transactions on pattern analysis and machine intelligence*, *44*(1), 154–180.

Hu, J., Xia, P., Gu, C., Qi, J., Li, S., & Peng, Y. (2014). Recognition of similar objects using simulated prosthetic vision. *Artificial organs*, *38*(2), 159–167.

Hu, Y., Liu, S.-C., & Delbruck, T. (2021). V2e: From video frames to realistic dvs events. *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 1312–1321.

Huang, J., Rathod, V., Sun, C., Zhu, M., Korattikara, A., Fathi, A., Fischer, I., Wojna, Z., Song, Y., Guadarrama, S., et al. (2017).

Speed/accuracy trade-offs for modern convolutional object detectors. *Proceedings of the IEEE conference on computer vision and pattern recognition*, 7310–7311.

Irons, J. L., Gradden, T., Zhang, A., He, X., Barnes, N., Scott, A. F., & McKone, E. (2017). Face identity recognition in simulated prosthetic vision is poorer than previously reported and can be improved by caricaturing. *Vision research*, *137*, 61–79.

Jabir, B., Falih, N., & Rahmani, K. (2021). Accuracy and efficiency comparison of object detection open-source models. *International Journal of Online & Biomedical Engineering*, *17*(5).

Killian, N. J., Vurro, M., Keith, S. B., Kyada, M. J., & Pezaris, J. S. (2016). Perceptual learning in a non-human primate model of artificial vision. *Scientific reports*, *6*(1), 1–16.

Küçükoğlu, B., Rueckauer, B., Ahmad, N., van Steveninck, J. d. R., Güçlü, U., & van Gerven, M. (2022a). Optimization of neuroprosthetic vision via end-to-end deep reinforcement learning. *bioRxiv*.

Küçükoğlu, B., Rueckauer, B., Ahmad, N., van Steveninck, J. d. R., Güçlü, U., & van Gerven, M. (2022b). Optimization of neuroprosthetic vision via end-to-end deep reinforcement learning. *bioRxiv*.

Law, H., & Deng, J. (2018). Cornernet: Detecting objects as paired keypoints. *Proceedings of the European conference on computer vision (ECCV)*, 734–750.

Lenz, G., Chaney, K., Shrestha, S. B., Oubari, O., Picaud, S., & Zarrella, G. (2021). *Tonic: Event-based datasets and transformations.* (Version 0.4.0) [Documentation available under https://tonic.readthedocs.io]. Zenodo. https://doi.org/10.5281/zenodo.5079802

Lewis, P. M., Ackland, H. M., Lowery, A. J., & Rosenfeld, J. V. (2015). Restoration of vision in blind individuals using bionic devices: A review with a focus on cortical visual prostheses. *Brain research*, *1595*, 51–73.

Li, H., Su, X., Wang, J., Kan, H., Han, T., Zeng, Y., & Chai, X. (2018). Image processing strategies based on saliency segmentation for object recognition under simulated prosthetic vision. *Artificial intelligence in medicine*, *84*, 64–78.

Li, W. (2013). Wearable computer vision systems for a cortical visual prosthesis. *Proceedings of the IEEE International Conference on Computer Vision Workshops*, 428–435.

Lichtsteiner, P., Posch, C., & Delbruck, T. (2008). A 128× 128 120 db 15 s latency asynchronous temporal contrast vision sensor. *Solid-State Circuits, IEEE Journal of*, *43*, 566–576. https://doi.org/10.1109/JSSC.2007.914337

Lieby, P., Barnes, N., McCarthy, C., Liu, N., Dennett, H., Walker, J. G., Botea, V., & Scott, A. F. (2011). Substituting depth for intensity and real-time phosphene rendering: Visual navigation under low vision conditions. *2011 Annual International Conference of the IEEE Engineering in Medicine and Biology Society*, 8017–8020.

Lomonaco, V., & Maltoni, D. (2017). Core50: A new dataset and benchmark for continuous object recognition. *Conference on Robot Learning*, 17–26.

Lomonaco, V., Maltoni, D., & Pellegrini, L. (2020). Rehearsal-free continual learning over small non-iid batches. *CVPR Workshops*, *1*(2), 3.

Lozano, A., Suárez, J. S., Soto-Sánchez, C., Garrigós, J., Martınez-Alvarez, J. J., Ferrández, J. M., & Fernández, E. (2020). Neurolight: A deep learning neural interface for cortical visual prostheses. *International journal of neural systems*, *30*(09), 2050045.

Macé, M. J.-M., Guivarch, V., Denis, G., & Jouffrais, C. (2015). Simulated prosthetic vision: The benefits of computer-based object recognition and localization. *Artificial organs*, *39*(7), E102–E113.

Niketeghad, S., & Pouratian, N. (2019). Brain machine interfaces for vision restoration: The current state of cortical visual prosthetics. *Neurotherapeutics*, *16*(1), 134–143.

Nirenberg, S., & Pandarinath, C. (2012). Retinal prosthetic strategy with the capacity to restore normal vision. *Proceedings of the National Academy of Sciences*, *109*(37), 15012–15017.

Ogawa, K., Ito, Y., & Nakano, K. (2010). Efficient canny edge detection using a gpu. *2010 First International Conference on Networking and Computing*, 279–280.

Pan, L., Scheerlinck, C., Yu, X., Hartley, R., Liu, M., & Dai, Y. (2019). Bringing a blurry frame alive at high frame-rate with an event camera. *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 6820–6829.

Pelayo, F., Martinez, A., Romero, S., Morillas, C. A., Ros, E., & Fernández, E. (2003). Cortical visual neuro-prosthesis for the blind: Retina-like software/hardware preprocessor. *First International IEEE EMBS Conference on Neural Engineering, 2003. Conference Proceedings.*, 150–153.

Pezaris, J. S., & Eskandar, E. N. (2009). Getting signals into the brain: Visual prosthetics through thalamic microstimulation. *Neurosurgical focus*, *27*(1), E6.

Posch, C. (2012). Bio-inspired vision. *Journal of Instrumentation*, *7*(01), C01054.

Rueckauer, B., & van Gerven, M. (2022). Experiencing prosthetic vision with event-based sensors. *Proceedings of the International Conference on Neuromorphic Systems 2022*, 1–7.

Sanchez-Garcia, M., Martinez-Cantin, R., & Guerrero, J. J. (2018). Structural and object detection for phosphene images. *arXiv preprint arXiv:1809.09607*.

Sanchez-Garcia, M., Martinez-Cantin, R., & Guerrero, J. J. (2020). Semantic and structural image segmentation for prosthetic vision. *Plos one*, *15*(1), e0227677.

Shim, S., Eom, K., Jeong, J., & Kim, S. J. (2020). Retinal prosthetic approaches to enhance visual perception for blind patients. *Micromachines*, *11*(5), 535.

Srikanth, A., Srinivasan, A., Indrajit, H., & Venkateswaran, N. (2021). Contactless object identification algorithm for the visually impaired using efficientdet. *2021 Sixth International Conference on Wireless Communications, Signal Processing and Networking (WiSPNET)*, 417–420.

Tan, M., Pang, R., & Le, Q. V. (2020). Efficientdet: Scalable and efficient object detection. *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, 10781–10790.

Thompson, R. W., Barnett, G. D., Humayun, M. S., & Dagnelie, G. (2003). Facial recognition using simulated prosthetic pixelized vision. *Investigative ophthalmology & visual science*, *44*(11), 5035–5042.

van der Grinten, M. L., van Steveninck, J. d. R., Lozano, A., Pijnacker, L., Rueckauer, B., Roelfsema, P., van Gerven, M., van Wezel, R., Guclu, U., & Gucluturk, Y. (2022). Biologically plausible phosphene simulation for the differentiable optimization of visual cortical prostheses. *bioRxiv*.

van Steveninck, J. d. R., Güçlü, U., van Wezel, R., & van Gerven, M. (2022). End-to-end optimization of prosthetic vision. *Journal of Vision*, *22*(2), 20–20.

van Steveninck, J. d. R., van Gestel, T., Koenders, P., van der Ham, G., Vereecken, F., Güçlü, U., van Gerven, M., Güçlütürk, Y., & van Wezel, R. (2022). Real-world indoor mobility with simulated prosthetic vision: The benefits and feasibility of contour-based scene simplification at different phosphene resolutions. *Journal of Vision*, *22*(2), 1–1.

Veraart, C., Raftopoulos, C., Mortimer, J. T., Delbeke, J., Pins, D., Michaux, G., Vanlierde, A., Parrini, S., & Wanet-Defalque, M.-C. (1998). Visual sensations produced by optic nerve stimulation using an implanted self-sizing spiral cuff electrode. *Brain research*, *813*(1), 181–186.

Vergnieux, V., Macé, M. J.-M., & Jouffrais, C. (2014). Wayfinding with simulated prosthetic vision: Performance comparison with regular and structure-enhanced renderings. *2014 36th Annual International Conference of the IEEE Engineering in Medicine and Biology Society*, 2585–2588.

Vergnieux, V., Macé, M. J.-M., & Jouffrais, C. (2017). Simplification of visual rendering in simulated prosthetic vision facilitates navigation. *Artificial organs*, *41*(9), 852–861.

Wandell, B. A., Brewer, A. A., & Dougherty, R. F. (2005). Visual field map clusters in human cortex. *Philosophical Transactions of the Royal Society B: Biological Sciences*, *360*(1456), 693–707.

Wandell, B. A., Dumoulin, S. O., & Brewer, A. A. (2007). Visual field maps in human cortex. *Neuron*, *56*(2), 366–383.

Xu, Z., Baojie, X., & Guoxin, W. (2017). Canny edge detection based on open cv. *2017 13th IEEE international conference on electronic measurement & instruments (ICEMI)*, 53–56.

Zhao, Y., Lu, Y., Tian, Y., Li, L., Ren, Q., & Chai, X. (2010). Image processing based recognition of images with a limited number of pixels using simulated prosthetic vision. *Information Sciences*, *180*(16), 2915–2924.

Zhao, Y., Lu, Y., Zhou, C., Chen, Y., Ren, Q., & Chai, X. (2011). Chinese character recognition using simulated phosphene maps. *Investigative ophthalmology & visual science*, *52*(6), 3404–3412.

Zhao, Y., Xu, D., Wang, T., & Ren, Y. (2020). Dynamic action recognition under simulated prosthetic vision. *2020 International Conference on Networking and Network Applications (NaNA)*, 417–421.

Zhao, Z.-Q., Zheng, P., Xu, S.-t., & Wu, X. (2019). Object detection with deep learning: A review. *IEEE transactions on neural networks and learning systems*, *30*(11), 3212–3232.

Zhu, A. Z., Yuan, L., Chaney, K., & Daniilidis, K. (2018). Ev-flownet: Self-supervised optical flow estimation for event-based cameras. *arXiv preprint arXiv:1802.06898*.

Zou, Z., Shi, Z., Guo, Y., & Ye, J. (2019). Object detection in 20 years: A survey. *arXiv preprint arXiv:1905.05055*.

Zrenner, E. (2002). Will retinal implants restore vision? *Science*, *295*(5557), 1022–1025.