Foveated Compression for Efficient Phosphene Vision

Michelle Appel* Eleftherios Papadopoulos* Elvin Su Yüksel†
Ömer Cevdet Çalık† Yağmur Güçlütürk*
*Donders Institute for Brain, Cognition and Behaviour, Radboud University, Nijmegen, the Netherlands
†Istanbul Technical University (ITU), Istanbul, Turkey
michelle.appel@ru.nl

Abstract

Visual prostheses generate phosphene patterns to restore vision, but their success depends on encoding strategies that balance computational cost and perceptual quality. Inspired by the fovea in human vision, we evaluate foveated compression as a biologically grounded alternative to uniform downsampling. Using a differentiable phosphene simulation pipeline, we compare foveated and bilinear compression against uncompressed inputs across multiple resolutions. Results indicate that foveated compression preserves central detail while reducing peripheral redundancy, enabling efficient phosphene representations with markedly lower computational cost. In a face recognition task, performance was maintained despite this reduction in resources, suggesting that essential perceptual cues can be preserved. Faces represent one of the most critical stimulus classes for human interaction, and the focal emphasis of foveated compression makes it particularly well-suited for supporting their recognition. These findings position foveated compression as a promising, biologically inspired strategy for real-time visual implants.

Keywords: *Visual prostheses, phosphene vision, foveated compression, retinal sampling, end-to-end optimization, computational efficiency.*

1. Introduction

Blindness affects millions of people worldwide and has very important implications for the quality of life and independence of an individual. To some people, this means a hopeful solution using visual prostheses that bypass parts of the visual system no longer working properly to directly stimulate the visual cortex and evoke percepts, called phosphenes-patterns of light that represent visual input. The prosthetics aim to restore the functional sense of vision by translating camera inputs into patterns of electrical stimulation.

Given the limitations of both electrode numbers and complexity of the representation, cortical prosthetics can currently have only limited resolution. The current research focuses on optimization of the representation of visual scenes for maximum interpretability within the constraints of computational efficiency and biological plausibility [2, 3]. One popular way in which such representations are assessed for perceptual quality is through Simulated Prosthetic Vision (SPV), non-invasive testing of encoding strategies using computational models alongside sighted participants [4].

One of the biggest challenges with SPV is how to balance the various trade-offs between perceptual realism, computational load, and biological fidelity.

Some of these, such as foveated compression, have been explored as techniques inspired by human vision. Foveated compression prioritizes central visual detail by encoding the fovea-the high-acuity region of the visual field-at a higher resolution than the periphery. This biologically motivated approach aligns with the natural organization of the visual cortex and reduces computational requirements [1]. In this work, we study the benefit of incorporating foveated compression into SPV pipelines for visual prostheses. We will systematically compare different foveated and nonfoveated compression methods at various input resolutions with respect to their impact on computational efficiency and perceptual quality. We further study through behavioral experiments how these techniques affect task performance in realistic scenarios. Our findings contribute towards the establishment of efficient, biologically plausible encoding strategies that could be used in cortical prosthetics of vision and bridge existing gaps between simulations and experimental work.

2. Related Work

Efficient visual processing is a critical challenge in both biological and artificial vision systems. Foveated

vision, inspired by the human visual system, has emerged as an effective approach to address this challenge. This technique leverages the natural structure of human vision, where the central part of the visual field, corresponding to the fovea, is represented at high resolution, while the peripheral regions are progressively compressed to reduce computational load. This approach balances the need for detailed visual information in the center with the ability to process a broader field of view at lower computational cost.

Foveated Vision and Retinal Sampling: Figure 1 illustrates the concept of foveated compression, where the central visual field retains high resolution, and the peripheral regions are sampled with decreasing density. This mirrors the density distribution of photoreceptors in the human retina, which is highest in the fovea and decreases towards the periphery. By emulating this biologically inspired strategy, artificial vision systems can reduce computational complexity while preserving critical visual information.

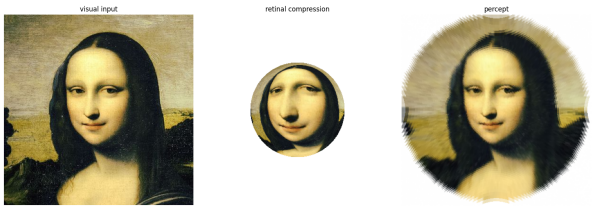

Figure 1. Illustration of foveated compression. The central part of the visual input retains high resolution, while the peripheral areas are compressed to reduce computational complexity.

Recent work by da Costa et al. [1] demonstrated how convolutional neural networks (CNNs) can replicate key organizational principles of the early visual cortex when enhanced with retinal sampling. Their study highlights that foveated sampling techniques not only improve computational efficiency but also align with biological visual processing strategies. This alignment makes these techniques particularly suitable for tasks requiring high-speed and resource-constrained computation, such as real-time visual recognition.

Visual Prostheses and Phosphene Vision: In the domain of visual prostheses, the challenge lies in creating percepts, known as phosphenes, that are interpretable and useful for individuals with severe visual impairments. The work of van der Grinten et al. [4] introduced a biologically plausible phosphene simulation framework for optimizing visual cortical prostheses. Their study integrated neural networks with cortical stimulation models to produce phosphenes that better emulate naturalistic vision. By applying differentiable optimization techniques, they were able to refine phosphene patterns to improve the usability of prosthetic devices.

This work laid a foundation for understanding how phosphene vision can be optimized by leveraging computational techniques. However, it did not fully explore the potential benefits of integrating foveated compression into the phosphene simulation pipeline, a gap that our study aims to address.

Extending Prior Work: Building upon the insights from these studies, our work uniquely integrates foveated compression into the phosphene vision pipeline. By incorporating biologically inspired compression methods, such as retinal sampling, we aim to address the trade-offs between computational efficiency and visual accuracy. Our approach evaluates the impact of foveated compression on real-time phosphene vision systems, comparing its performance to simpler methods like Nearest Neighbor (NN) compression and uncompressed inputs.

Furthermore, our study provides a detailed analysis of the effects of foveated compression across varying resolutions, quantifying its impact on both phosphene prediction and reconstruction accuracy. Unlike previous studies, we explicitly explore the computational implications of foveated compression, including its memory and processing cost, making it a comprehensive investigation into its feasibility for real-time prosthetic applications.

Broader Implications: The integration of foveated compression into neural networks for phosphene vision has broader implications beyond visual prostheses. It represents a step towards more efficient and biologically inspired artificial vision systems, particularly for resource-constrained environments. By drawing from principles of human vision, this work contributes to a growing body of research that seeks to align artificial intelligence with biological systems to improve both performance and interpretability.

In summary, our study bridges the gap between biologically inspired compression methods and their application to phosphene vision, extending the work of da Costa et al. [1] and van der Grinten et al. [4]. This novel integration provides a new perspective on optimizing visual prostheses, offering both theoretical and practical contributions to the field.

3. Methods

3.1. Compression Methods

In this study, we explored two primary image compression techniques: bilinear sampling and retinal sampling. Each method was chosen for its ability to balance

computational efficiency with task accuracy. Retinal sampling, in particular, incorporates biologically inspired principles to emulate the non-uniform sampling of the human retina.

3.1.1 Bilinear Sampling

Bilinear sampling reduces image resolution by interpolating pixel values using a weighted average of their four nearest neighbors. For a target pixel (x', y') in the downsampled image, the value is computed as:

$$I'(x', y') = \sum_{i=0}^{1} \sum_{j=0}^{1} w_{ij} \cdot I(x_i, y_j),$$

where w_{ij} are the interpolation weights, calculated as:

$$w_{ij} = (1 - |x' - x_i|)(1 - |y' - y_j|).$$

This method results in smooth transitions and reduced aliasing artifacts, making it a common baseline for image compression.

Computational Cost: The computational cost of bilinear sampling, measured in multiply-accumulate operations (mult-adds), is given by:

$$\text{Mult-Adds}_{\text{bilinear}} = 4 \cdot H' \cdot W',$$

where H' and W' are the height and width of the output image. Each pixel computation involves four multiplications and corresponding additions.

3.1.2 Retinal Sampling (Foveated Compression)

Retinal sampling [1] is a biologically inspired compression technique designed to mimic the non-uniform distribution of retinal ganglion cells (RGCs) in the human eye. It emphasizes high-resolution detail in the central visual field (fovea), where the majority of visual information is concentrated, and reduces resolution in the periphery. This approach is particularly suitable for applications like phosphene vision, where central detail is crucial for interpretability and task performance.

Mathematical Formulation: The method is based on empirical data describing the density of RGCs as a function of eccentricity in the visual field. The density function is expressed as:

$$\rho(r_{\text{vf}}) = \rho_0 \left[\alpha \left(1 + \frac{r_{\text{vf}}}{k}\right)^{-2} + (1 - \alpha) \exp\left(-\frac{r_{\text{vf}}}{\sigma}\right) \right],$$

where:

- ρ_0: RGC density at the fovea ($r_{\text{vf}} = 0$),
- k: Eccentricity where density decreases by a factor of four,
- σ: Scale factor for the exponential decay,
- α: Weighting factor balancing the quadratic and exponential terms.

The cumulative density maps visual field radii (r_{vf}) to RGC radii (r_{rgc}):

$$r_{\text{rgc}}(r_{\text{vf}}) = \int_0^{r_{\text{vf}}} \rho(x)\, dx.$$

To remap RGC radii back to visual field radii (r_{vf}), the inverse relationship simplifies to:

$$r_{\text{vf}}(r_{\text{rgc}}) = \frac{-\rho_0 k^2}{r_{\text{rgc}} + r_0} - k,$$

with $r_0 = -\frac{\rho_0 k^2}{k}$ ensuring continuity.

Implementation: Retinal sampling involves two primary stages:

1. **Preprocessing:** Input images are resized to a uniform size of 256×256 pixels using nearest neighbor sampling. Nearest neighbor sampling assigns the value of the nearest pixel from the original image to the resized image, calculated as:

   $$I'(x', y') = I\left(\lfloor x \cdot \frac{W}{W'} \rfloor, \lfloor y \cdot \frac{H}{H'} \rfloor\right),$$

   where W, H are the dimensions of the original image, and W', H' are the resized dimensions. This step ensures uniform input for retinal mapping and incurs negligible computational cost since it relies on index lookups.

2. **Retinal Mapping:** Pixel locations are transformed based on the retinal density function. This involves:

   - Calculating the radial distance (r) of each pixel from the center.
   - Mapping r_{vf} to r_{rgc} (or vice versa, for inverse mapping) using the density function or its integral.
   - Applying these mappings to redistribute pixel values, emphasizing central details.

   In our setup, the retinal map is calculated once during preprocessing and reused for the entire dataset. This reuse significantly reduces computational overhead during subsequent image transformations. For computational cost analysis, we consider the scenario where precomputed maps are applied.

Computational Cost: The computational cost of retinal sampling can be divided into two components:

- **Nearest Neighbor Preprocessing:** This step relies solely on index lookups and incurs no mult-adds.
- **Retinal Mapping:** When using precomputed maps, the cost is proportional to the number of output pixels. The total mult-adds for this process are given by:

$$\text{Mult-Adds} = \beta H' W',$$

where β represents the average operations per pixel, and H', W' are the output image dimensions.

Advantages: Retinal sampling offers a biologically plausible method of compression, effectively reducing peripheral detail while preserving central information. By leveraging precomputed maps, it achieves high computational efficiency, making it a practical choice for real-time applications such as phosphene vision.

3.2. Experimental Setup

Our experimental setup follows an end-to-end differentiable pipeline designed to optimize visual information processing for cortical prostheses [4]. This pipeline integrates biologically plausible principles to simulate and optimize phosphene vision through three key components: an encoder, a phosphene simulator, and a decoder (see Figure 2).

Pipeline Overview: The pipeline consists of the following stages:

1. **Encoder:** The encoder transforms input images into a stimulation sequence suitable for cortical stimulation. It learns a mapping from the visual input to the corresponding stimulation signals, which are represented as 1D vectors of phosphene activations. The architecture of the encoder adapts to the input resolution by scaling down its final linear layer, which connects the flattened feature map to the stimulation output. The size of this layer is proportional to the square of the input resolution:

$$\text{Feature Size} \propto \left(\frac{\text{Input Resolution}}{\text{Base Resolution}} \right)^2,$$

where the base resolution corresponds to the input size for which the network was originally designed. This scaling ensures efficient utilization of network capacity across varying input sizes.

2. **Phosphene Simulator:** The simulator generates biologically plausible phosphene patterns from the stimulation sequence. This differentiable simulator incorporates principles of cortical neurostimulation to produce unordered phosphene representations. These representations reflect the spatial distribution and perceptual characteristics of cortical stimulation-induced phosphenes.

3. **Decoder:** The decoder reconstructs target outputs from the phosphene representations. In our experiments, the target outputs are edge maps derived from the segmentation boundaries of the input images. The decoder evaluates the interpretability of the phosphene representations by reconstructing meaningful visual features.

Encoder Training: The encoder is optimized to predict phosphene activations that align closely with ground-truth values generated by the phosphene simulator. The ground-truth activations are represented as 1D arrays corresponding to the total number of phosphenes. The loss function used for encoder training is:

$$\mathcal{L}_{\text{encoder}} = \text{MSE}(\mathbf{p}_{\text{pred}}, \mathbf{p}_{\text{gt}}),$$

where $\mathbf{p}_{\text{pred}}$ is the predicted phosphene activation vector, and $\mathbf{p}_{\text{gt}}$ is the ground truth generated by the simulator.

Decoder Training: The decoder is trained to reconstruct edge maps from the phosphene representations. The reconstruction loss ensures that the decoded output aligns with the target edge map, defined as:

$$\mathcal{L}_{\text{decoder}} = \text{MSE}(\mathbf{r}_{\text{pred}}, \mathbf{r}_{\text{gt}}),$$

where $\mathbf{r}_{\text{pred}}$ is the reconstructed edge map, and $\mathbf{r}_{\text{gt}}$ is the target edge map.

End-to-End Training: Both the encoder and decoder are trained simultaneously in an end-to-end fashion. The total loss function combines the encoder and decoder losses:

$$\mathcal{L}_{\text{total}} = \mathcal{L}_{\text{encoder}} + \mathcal{L}_{\text{decoder}}.$$

Simulation and Reconstruction: The phosphene simulator acts as an intermediary between the encoder and decoder, translating stimulation sequences into phosphene representations. These representations are unordered and approximate the perceptual effects of cortical stimulation. The decoder processes these representations to reconstruct edge maps, providing an indirect measure of the interpretability and quality of the phosphene activations.

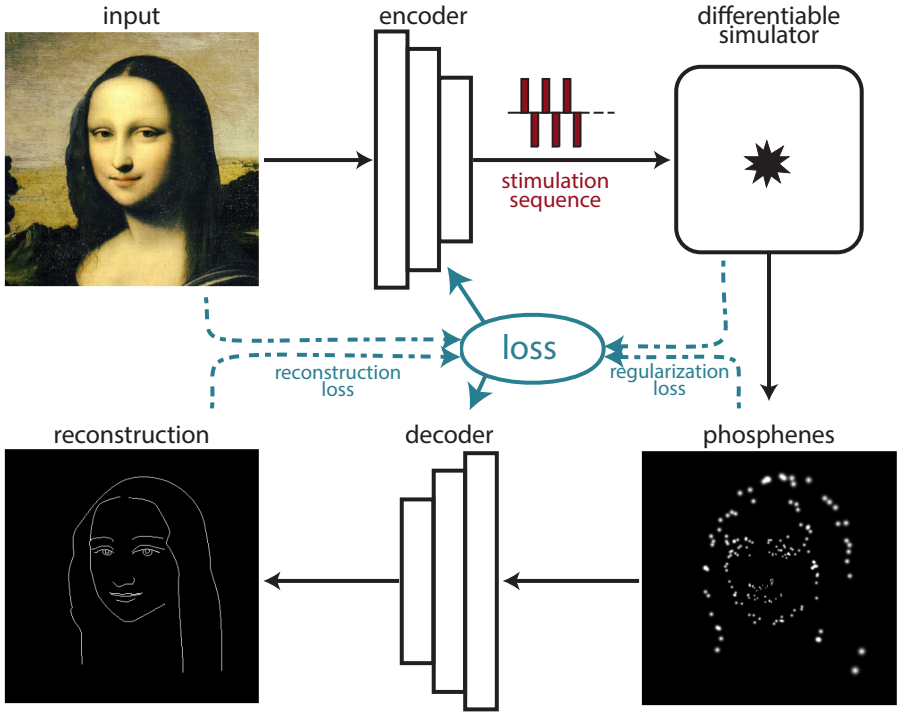

Figure 2. Illustration of the end-to-end pipeline used in phosphene vision optimization. The process begins with an input image, which is passed through an encoder to generate a feature map. This feature map is transformed into a phosphene pattern using a differentiable simulator that models the electrical stimulation of the visual cortex. The phosphene pattern is then decoded to reconstruct the image, and the system optimizes this process by comparing the reconstructed image with the original input to minimize error and improve visual interpretation.

Key Assumption: Reconstruction performance is used as a proxy for the interpretability of the phosphene representations. While this approach suggests that high reconstruction accuracy correlates with meaningful phosphene patterns, behavioral experiments are required to validate this assumption in practical settings.

This experimental setup integrates biologically plausible principles with computational efficiency, enabling the optimization of visual prostheses for real-world applications.

3.3. Evaluation Metrics

To assess the performance of our approach, we evaluated both the accuracy and computational efficiency of the proposed pipeline. The metrics are divided into two categories: accuracy metrics and computational efficiency measurements.

Accuracy Metrics: The accuracy of the pipeline is evaluated using two metrics: Mean Squared Error (MSE) and Structural Similarity Index Measure (SSIM). These metrics are applied to both the predicted phosphene activations and the reconstructed outputs to assess the quality of the encoding and decoding processes.

- **Mean Squared Error (MSE):** The MSE measures the average squared difference between predicted and target values. For phosphene predictions, the MSE is defined as:

$$\text{MSE}_{\text{phosphenes}} = \frac{1}{N} \sum_{i=1}^{N} \left(p_{\text{pred},i} - p_{\text{gt},i}\right)^2,$$

where N is the number of phosphenes, $p_{\text{pred},i}$ is the predicted activation for the i-th phosphene, and $p_{\text{gt},i}$ is the corresponding ground truth value. A similar

formula applies to the reconstructed outputs:

$$\text{MSE}_{\text{reconstructions}} = \frac{1}{M} \sum_{i=1}^{M} \left(r_{\text{pred},i} - r_{\text{gt},i} \right)^2 ,$$

where M represents the number of pixels in the reconstructed image.

- **Structural Similarity Index Measure (SSIM):** The SSIM quantifies the perceptual similarity between two images, capturing luminance, contrast, and structure. For a window of size $w \times w$, SSIM is defined as:

$$\text{SSIM} = \frac{(2\mu_x\mu_y + C_1)(2\sigma_{xy} + C_2)}{(\mu_x^2 + \mu_y^2 + C_1)(\sigma_x^2 + \sigma_y^2 + C_2)},$$

where μ_x and μ_y are the mean intensities of the predicted and target images, σ_x^2 and σ_y^2 are their variances, σ_{xy} is the covariance, and C_1, C_2 are constants to stabilize the division.

Computational Efficiency Metrics: The computational efficiency of the pipeline is evaluated using three primary metrics: the number of parameters, multiply-accumulate operations (mult-adds), and memory usage.

- **Parameters:** The total number of trainable parameters in the network is calculated based on the architecture's configuration, including convolutional and fully connected layers. This provides a measure of model size and complexity.
- **Multiply-Add Operations (Mult-Adds):** Mult-adds measure the total computational operations required for a forward pass through the network and the compression methods:
  1. **Network Mult-Adds:** The mult-adds for the encoder and decoder are directly measured during execution, providing an accurate representation of computational cost based on the network's architecture.
  2. **Downsampling Mult-Adds:** The mult-adds for each compression method are calculated or measured as follows:
     - *Bilinear Sampling:* The mult-adds for bilinear sampling are proportional to the number of output pixels, as each pixel involves four interpolations:

$$\text{Mult-Adds}_{\text{bilinear}} = 4 \cdot H' \cdot W',$$

where H' and W' are the height and width of the output image.

     - *Nearest Neighbor Sampling:* Nearest neighbor sampling involves no mult-adds, as it relies purely on index lookups.
     - *Foveated Sampling:* The mult-adds for foveated sampling are measured directly during execution. This approach counts the number of multiply-accumulate operations performed based on the implementation of the retinal mapping.
- **Memory Usage:** Memory usage is a critical metric for evaluating the feasibility of applying the pipeline in real-world applications. It reflects the total memory required to store and process data during a forward pass.
  1. **Compression Memory:** The memory required for storing and applying compression operations is measured dynamically during execution. This is particularly relevant for comparing the overhead introduced by different compression techniques.
  2. **Input Memory:** This represents the memory required to store the input image after compression. It is determined by the resolution and number of channels of the processed input.
  3. **Network Memory:** This includes memory allocated for intermediate computations, activations, and model parameters during a forward pass through the network.

  Memory usage is measured in megabytes (MB), providing a quantitative understanding of resource requirements.

By combining these metrics, we evaluate the trade-offs between computational cost and performance, providing insights into the efficiency and scalability of the pipeline for visual prosthesis applications.

3.4. Experiment

This section describes the dataset, configurations, and training parameters used in our experiments, providing a comprehensive framework for evaluating the proposed pipeline.

Dataset: We utilized the LaPa dataset, a collection of 22,168 facial images with segmentation maps. The dataset was divided into:

- 18,168 images for training,
- 2,000 images for validation,
- 2,000 images for testing.

To generate the target edge maps for reconstruction, segmentation maps were processed to extract contour features. Let $S(x, y)$ represent the segmentation map of an image. The contour map $C(x, y)$ is obtained by applying an edge detection filter followed by a threshold operation:

$$C(x, y) = \begin{cases} 255, & \text{if } \nabla^2 S(x, y) > \epsilon, \\ 0, & \text{otherwise,} \end{cases}$$

where ∇^2 denotes the Laplacian operator applied to the segmentation map, and ϵ is the threshold for edge detection. This operation identifies significant transitions in pixel intensity, highlighting the boundaries of facial features.

Configurations: We evaluated the pipeline across seven configurations, varying the input resolution and the compression method. These configurations are grouped based on input resolution and summarized in Table 1.

Input Resolution	Compression	FoV
128×128	Baseline	N/A
64×64	Bilinear	N/A
	Foveated	7°
32×32	Bilinear	N/A
	Foveated	11°
16×16	Bilinear	N/A
	Foveated	14°

Table 1. Experimental Configurations

Field of View (FoV): The FoV parameter in the foveated compression configurations represents the angular size of the visual field covered by the retinal sampling. Smaller input resolutions require larger FoV values to maintain coverage of the visual field: 7° for 64×64, 11° for 32×32, and 14° for 16×16. This adjustment ensures that the retinal sampling appropriately emphasizes central visual detail while accounting for reduced resolution.

Training Parameters: The pipeline was trained using the following parameters:

- **Learning Rate:** 0.001, with weight decay of 1×10^{-6}.
- **Optimizer:** Adam optimizer, minimizing a combined loss function for phosphene prediction and reconstruction.
- **Learning Rate Scheduler:** A ReduceLROnPlateau scheduler reduced the learning rate by a factor of 0.1 after 12 epochs without improvement.
- **Early Stopping:** Training stopped early if validation performance did not improve for 15 consecutive epochs.
- **Batch Size:** 4 images per batch.
- **Number of Epochs:** Maximum of 28 epochs.
- **Simulation Output:** Phosphene activations were scaled to simulate the amplitude of electrical stimulation in a prosthetic device, with a maximum output value of 256×10^{-6} Amperes. The stimulation was constrained to 10 discrete steps within this range to ensure numerical stability.

Phosphene Simulation: The simulator was configured with 1,024 phosphenes to approximate the spatial distribution and perceptual effects of cortical stimulation. Sigmoid activation functions were used in both the encoder and decoder, ensuring bounded outputs suitable for the stimulation range.

Training Pipeline: The input images were preprocessed with a circular mask to simulate the visual field, ensuring consistency with real-world prosthetic applications. Retinal compression was applied for foveated configurations, using FoV values tailored to the input resolution as detailed in Table 1.

This experimental design systematically evaluates the proposed pipeline across diverse configurations, balancing computational efficiency with task performance.

4. Results

Configuration	MSE	SSIM
128x128	0.0734	**0.4163**
64x64 (Bilin)	0.0819	0.3024
64x64 (Fov)	**0.0722**	0.3889
32x32 (Bilin)	0.0929	0.2084
32x32 (Fov)	0.0847	0.2664
16x16 (Bilin)	0.1003	0.1454
16x16 (Fov)	0.0959	0.1784

Table 2. Performance metrics (MSE and SSIM) for phosphene predictions across different configurations. The baseline (128×128) achieves the highest SSIM, while 64×64 with foveated compression demonstrates the lowest MSE among compressed configurations. Lower resolutions show a decline in performance, reflecting the trade-off between accuracy and computational efficiency.

This section presents the results of the experiments, focusing on both model performance and computational efficiency. The results are supported by quantitative metrics presented in tables and corresponding figures for visual analysis.

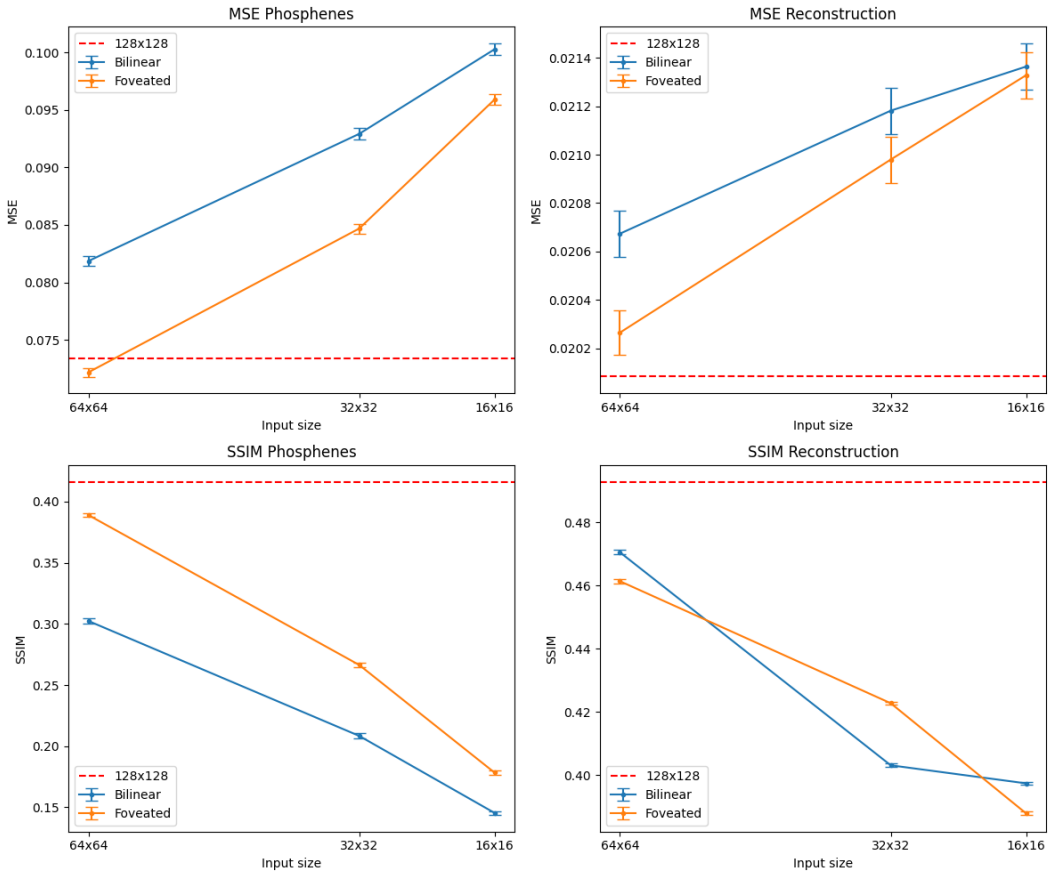

Figure 3. Quantitative comparison of phosphene and reconstruction performance across different input sizes and compression methods. The top row shows the MSE for phosphene and reconstruction performance, with lower values indicating better accuracy. The bottom row depicts SSIM for both metrics, with higher values indicating better structural similarity to the ground truth. The dashed red line represents the baseline performance of the 128x128 configuration, while the other configurations compare the effects of nearest neighbor (NN) and foveated compression.

Configuration	MSE	SSIM
128x128	**0.0201**	**0.4928**
64x64 (Bilin)	0.0207	0.4705
64x64 (Fov)	0.0203	0.4613
32x32 (Bilin)	0.0212	0.4031
32x32 (Fov)	0.0210	0.4228
16x16 (Bilin)	0.0214	0.3974
16x16 (Fov)	0.0213	0.3879

Table 3. Performance metrics (MSE and SSIM) for reconstruction outputs across different configurations. The baseline (128×128) achieves the best overall performance. Lower resolutions exhibit reduced accuracy, highlighting the impact of resolution on reconstruction quality.

4.1. Model Performance

The performance of the pipeline was evaluated using MSE and SSIM metrics for both phosphene prediction and reconstruction tasks. The quantitative results are summarized for Phosphene Performance in Table 2 and for Reconstruction performance in Table 3. Visual comparisons shown in Figure 3.

For phosphene prediction, the 64×64 foveated configuration achieves the best MSE score, even outperforming the baseline configuration (128×128). This result demonstrates that foveated compression can enhance accuracy by focusing on central details in the visual field. However, in all other cases, the baseline consistently achieves the best SSIM and MSE scores, indicating that the higher resolution provides more precise reconstructions.

Across all configurations, foveated compression outperforms bilinear compression for both phosphene

Configuration	Memory (Input)	Memory (Compression)	Memory (Params)	Memory (F/B Pass)	Total Memory
128x128	0.188 MB	0.094 MB	4.33 MB	12.266 MB	16.877 MB
64x64 (Bilin)	0.047 MB	0.063 MB	1.33 MB	3.072 MB	4.512 MB
64x64 (Fov)	0.047 MB	1.964 MB	1.33 MB	3.072 MB	6.413 MB
32x32 (Bilin)	0.012 MB	0.063 MB	0.58 MB	0.774 MB	1.428 MB
32x32 (Fov)	0.012 MB	1.757 MB	0.58 MB	0.774 MB	3.122 MB
16x16 (Bilin)	0.003 MB	0.063 MB	0.392 MB	0.199 MB	0.657 MB
16x16 (Fov)	0.003 MB	1.706 MB	0.392 MB	0.199 MB	2.300 MB

Configuration	Total Params	Mult-Adds (Compression)	Mult-Adds (Inference)	Total Mult-Adds
128x128 (Bilin)	1.135 M	0.111 M	122.965 M	123.075 M
64x64 (Bilin)	0.349 M	0.028 M	30.742 M	30.770 M
64x64 (Fov)	0.349 M	0.012 M	30.742 M	30.755 M
32x32 (Bilin)	0.152 M	0.007 M	7.687 M	7.694 M
32x32 (Fov)	0.152 M	0.003 M	7.687 M	7.690 M
16x16 (Bilin)	0.103 M	0.002 M	1.923 M	1.925 M
16x16 (Fov)	0.103 M	0.001 M	1.923 M	1.924 M

Table 4. Comparison of network size and computational efficiency across different configurations. The top section of the table highlights memory requirements, including compression memory, parameter memory, forward/backward pass memory, and total memory. The bottom section presents computational costs in terms of the number of parameters and Multiply-Adds (Mult-Adds), split into compression, inference, and total Mult-Adds. Results demonstrate the trade-offs between input size reduction and computational efficiency, with foveated compression showing slightly higher costs compared to bilinear compression while significantly reducing memory usage compared to the baseline 128x128 configuration.

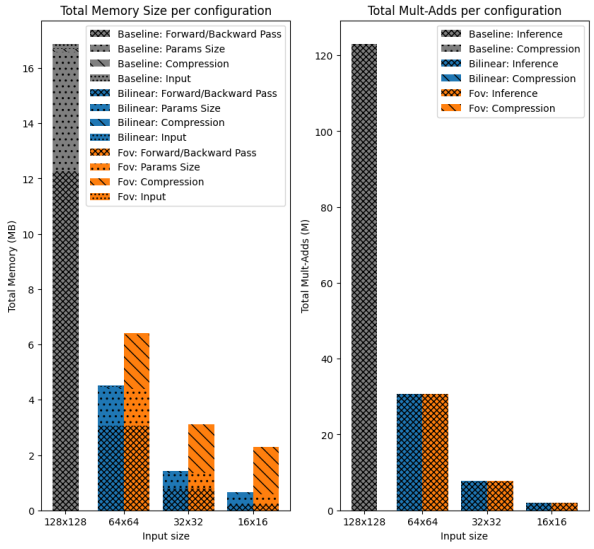

Figure 4. Comparison of memory usage and computational costs across configurations. The left bar plot illustrates the total memory usage, including input memory, compression memory, parameter memory, and forward/backward pass memory, relative to the 128×128 baseline. The right bar plot shows the total Multiply-Adds (Mult-Adds), split into compression, inference, and total computational costs.

prediction and reconstruction. Figure 5 provides qualitative support for these findings, showcasing phosphene maps and reconstructed edge maps. The visual results demonstrate that foveated compression better preserves the structural integrity of edge maps and the perceptual quality of phosphene predictions at lower resolutions.

These results highlight the trade-offs between input resolution, compression method, and performance, with 64×64 foveated emerging as a strong candidate for balancing computational efficiency and task performance.

4.2. Computational Efficiency

Computational efficiency was evaluated in terms of memory usage, mult-adds, and parameter count. Detailed results are provided in Table 4, with visual comparisons in Figure 4.

Memory usage decreases substantially as the input resolution decreases. The baseline configuration (128×128) requires the highest memory, while 16×16 configurations use a fraction of the resources. Foveated configurations consistently show slightly higher memory usage than bilinear configurations due to the overhead of precomputed maps. However, this difference is negligible compared to the overall reduction achieved by lowering input resolution.

Regarding mult-adds, bilinear sampling introduces slightly more computational operations than foveated

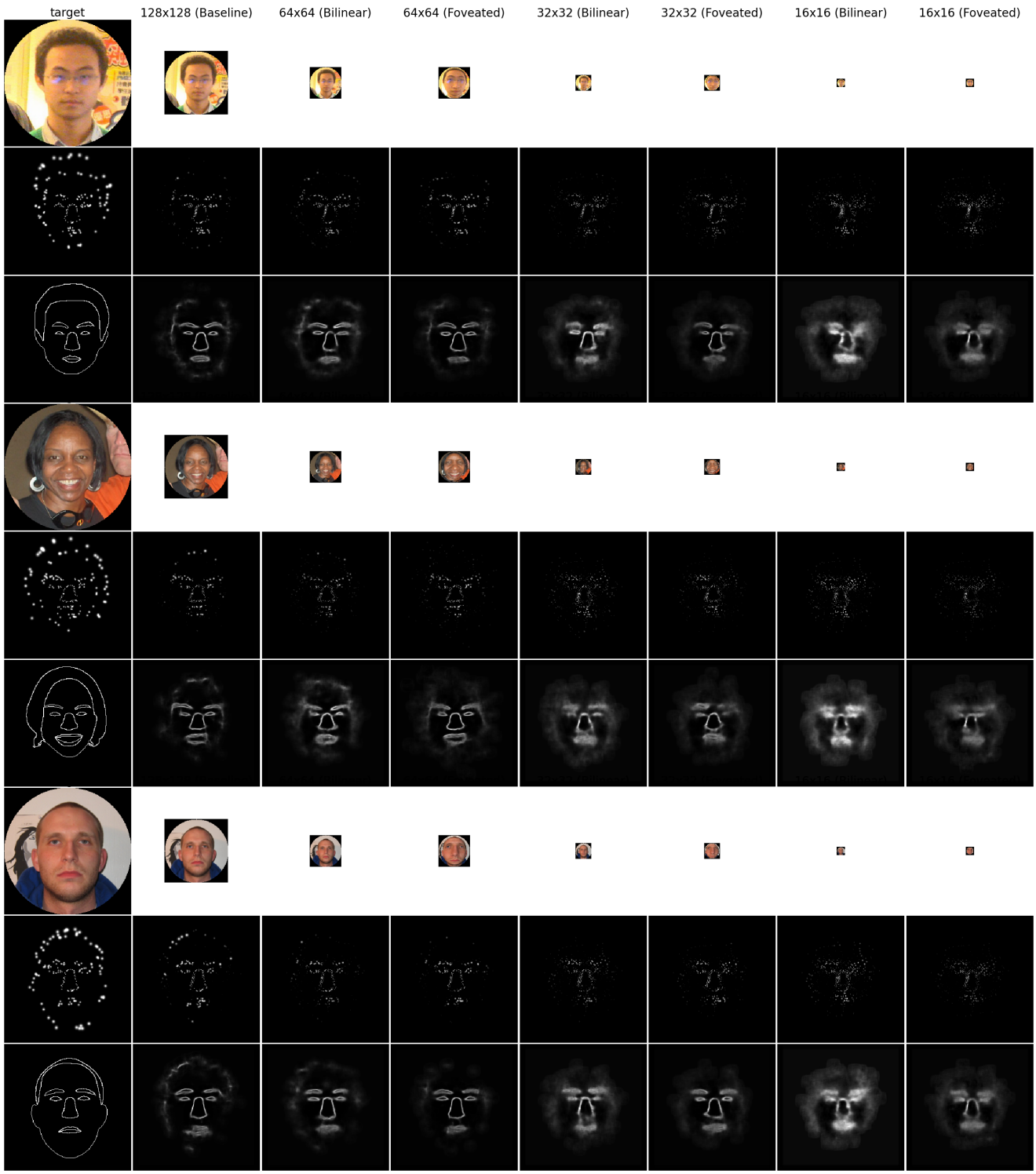

Figure 5. Visual comparison of phosphene maps and reconstructed edge maps across different configurations. The comparison highlights the effects of input resolution and compression methods (bilinear vs. foveated) on the quality of phosphene activations and reconstructed outputs. The baseline (128×128) serves as a reference, while lower resolutions (64×64, 32×32, and 16×16) demonstrate trade-offs between visual detail and computational efficiency.

compression. However, this difference is almost negligible, as the computational cost of the sampling methods is significantly overshadowed by the operations within the network itself. Both compression methods reduce the total mult-adds drastically compared to the baseline, with similar levels of efficiency at each resolution.

Figure 4 clearly visualizes these trends, emphasizing that the choice of compression method has a minimal impact on computational cost relative to the benefits gained from lowering input resolution. For example, the 64×64 foveated configuration reduces mult-adds and memory usage by more than half compared to the baseline, while maintaining competitive performance metrics.

Summary: The results demonstrate that foveated compression provides a favorable trade-off between performance and computational efficiency. The 64×64 foveated configuration is particularly notable, achieving the best phosphene prediction MSE while maintaining significant computational savings compared to the baseline. In all other cases, the baseline outperforms smaller resolutions, but foveated compression consistently provides better results than bilinear compression across metrics.

5. Behavioral Study

To validate the computational results and assess the quality of the generated phosphene representations, a behavioral study was designed and conducted. The aim was to evaluate how well participants could interpret phosphene-based stimuli and recognize corresponding visual inputs. The study builds on the premise that perceptual accuracy is a key metric for assessing the effectiveness of phosphene vision systems.

5.1. Method

Participants: The study aimed to recruit 20 healthy adult participants with normal or corrected-to-normal vision. Recruitment was conducted online through acquaintances of the researcher. All participants provided informed consent prior to the study, which was approved by the relevant ethics board.

Stimuli: Stimuli were generated using the LaPa dataset, focusing on facial images with segmentation maps. To ensure consistency, only forward-facing images were selected, based on a calculated "facing forward score." This score was derived by:

1. Cropping the face region from each image using the face skin segmentation.
2. Splitting the image in half and mirroring the right side.
3. Computing the Structural Similarity Index Measure (SSIM) between the segmentation maps of key facial features (e.g., eyes, nose, eyebrows, and lips) for the original and mirrored halves.

Images with the highest SSIM scores were deemed forward-facing. The top 30 images were selected for the experiment.

Normalization: To normalize the selected images:

- Each face was scaled to maintain a consistent skin-to-image proportion.
- Faces were centered such that the segmentation center point aligned with the image center.
- Images requiring downscaling were padded with a black background to ensure uniform size.

Phosphene representations were generated for these 30 forward-facing images using all configurations described earlier. Each phosphene representation served as the target stimulus in the behavioral task.

5.2. Experiment

Task Design: The experiment was conducted online using the Pavlovia platform, allowing participants to complete the task remotely on their own devices. Each trial consisted of the following steps:

1. A phosphene representation of a face was displayed at the center of the screen.
2. Surrounding the phosphene representation were four RGB face images: one target image and three distractor images selected randomly from the dataset.
3. The position of the target image was randomized across trials.

An example frame is shown in Figure 6.

Participants were instructed to select the RGB image they believed corresponded to the phosphene representation by clicking on it. No time limit was imposed, but participants were encouraged to respond as quickly and accurately as possible.

Data Collection: The following data points were recorded for each trial:

- Response accuracy (correct/incorrect),
- Response time (in milliseconds),
- Mouse click coordinates,
- Selected stimulus.

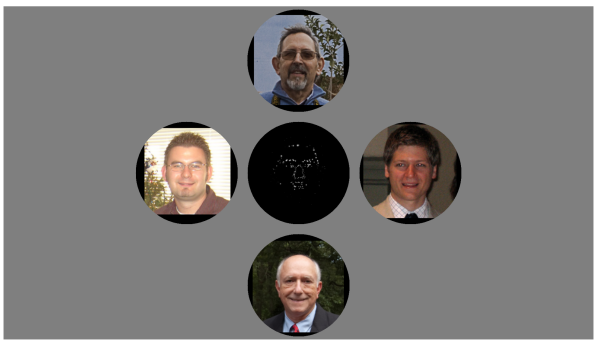

Figure 6. Example trial setup for the behavioral experiment. A phosphene representation of a face is displayed at the center of the screen, surrounded by four RGB face images. One of the RGB images corresponds to the target image represented by the phosphene, while the other three are randomly selected distractor images. Participants are instructed to select the RGB image they believe matches the phosphene representation by clicking on it. The experiment is conducted online, and response accuracy, response time, and mouse click coordinates are recorded.

Ethics: Participants were informed about the nature of the study and their rights before beginning the experiment. Ethical approval for the study was obtained from the appropriate review board.

5.3. Results

The results of the behavioral study are currently pending and will be analyzed to assess:

- Participant accuracy in identifying the correct image corresponding to the phosphene representation.
- Response times across different configurations to determine how resolution and compression methods influence interpretability.

These results will provide insight into the perceptual quality of the generated phosphene representations and their potential applicability in real-world visual prosthesis systems.

6. Discussion and Conclusion

This study explored the integration of foveated compression into phosphene vision systems for visual prostheses, comparing its performance to bilinear compression and uncompressed inputs across multiple configurations. By systematically evaluating computational efficiency and perceptual accuracy, we have gained insights into the trade-offs inherent in optimizing phosphene representations for real-time prosthetic applications.

Foveated Compression in Phosphene Vision: The results demonstrate that foveated compression, inspired by human visual processing, can provide a viable alternative to uniform compression methods. Notably, the 64×64 resolution with foveated compression consistently achieved the lowest Mean Squared Error (MSE) for phosphene prediction among compressed configurations, while also maintaining a competitive Structural Similarity Index Measure (SSIM). This suggests that foveated compression preserves critical central details required for accurate phosphene representations, aligning with the biological principles of the visual system.

However, the computational cost of foveated compression was slightly higher than bilinear sampling, as shown by memory usage and Multiply-Adds (Mult-Adds). This additional cost arises from the non-uniform nature of the compression, which prioritizes central detail over peripheral regions. Despite this, the computational overhead remains manageable, making foveated compression suitable for real-time applications in visual prostheses.

Limitations and Challenges: While the computational results are promising, the behavioral study results are still pending. The perceptual quality of phosphene representations, as judged by human participants, will provide crucial insights into the practical usability of these methods. Furthermore, our current setup assumes a static foveation point, which may not fully reflect the dynamic nature of human vision where the foveation point shifts with gaze. Future work could explore dynamic foveated compression methods that adapt in real-time to user gaze direction.

Another limitation lies in the trade-off between resolution and interpretability. While lower resolutions reduce computational cost, they may hinder perceptual accuracy. The behavioral study will help clarify how this trade-off impacts real-world usability.

Broader Implications: The findings of this study have implications beyond visual prostheses. The integration of foveated compression into neural networks could benefit other resource-constrained applications, such as augmented reality, robotics, and autonomous systems. By leveraging biologically inspired principles, these systems can achieve a balance between computational efficiency and task performance.

6.1. Conclusion

In this study, we evaluated the use of foveated compression for optimizing phosphene vision systems in visual prostheses. Our results highlight that foveated compression provides a biologically inspired method for balancing computational efficiency and visual accuracy. By preserving central detail while reducing peripheral

resolution, this approach aligns with the natural principles of human vision and demonstrates potential for real-time prosthetic applications.

The 64×64 foveated configuration emerged as a promising candidate, offering strong performance in phosphene prediction with manageable computational costs. However, further evaluation through the behavioral study is required to confirm its perceptual benefits.

Overall, this work contributes to the growing body of research on biologically inspired visual processing, offering new insights into the optimization of phosphene vision. Future research should focus on incorporating dynamic foveation and evaluating the long-term usability of these methods in real-world prosthetic systems. By continuing to align artificial vision systems with biological principles, we can enhance both their performance and interpretability, paving the way for improved assistive technologies.

References

[1] Danny da Costa, Lukas Kornemann, Rainer Goebel, and Mario Senden. Convolutional neural networks develop major organizational principles of early visual cortex when enhanced with retinal sampling. *Scientific Reports*, 14, 12 2024. 1, 2, 3

[2] Richard A. Normann, Bradley A. Greger, Paul House, Samuel F. Romero, Francisco Pelayo, and Eduardo Fernandez. Toward the development of a cortically based visual neuroprosthesis. *Journal of Neural Engineering*, 6, 2009. 1

[3] Melani Sanchez-Garcia, Ruben Martinez-Cantin, and Jose J. Guerrero. Semantic and structural image segmentation for prosthetic vision. *PLoS ONE*, 15, 1 2020. 1

[4] Maureen van der Grinten, Jaap de Ruyter van Steveninck, Antonio Lozano, Laura Pijnacker, Bodo Rueckauer, Pieter Roelfsema, Marcel van Gerven, Richard van Wezel, Umut Güçlü, and Yağmur Güçlütürk. Towards biologically plausible phosphene simulation for the differentiable optimization of visual cortical prostheses. *eLife*, 13, 2 2024. 1, 2, 4