Institute of Imaging and Computer Vision | RWTH AACHEN UNIVERSITY

Vision Transformer-based Retinal Prosthetic Simulation with Predicted Visual Fixations

Master Thesis

presented by

Do Dinh Tan Nguyen

Supervisor:
Yuli Wu

Institute of Imaging & Computer Vision
Prof. Dr.-Ing. Johannes Stegmaier
RWTH Aachen University

Erklärung nach §18 Abs. 1 ÜPO

Hiermit versichere ich, dass ich die vorgelegte Master Thesis selbständig angefertigt habe. Es sind keine anderen als die angegebenen Quellen und Hilfsmittel benutzt worden. Zitate wurden kenntlich gemacht.

I hereby confirm that I have written this Master Thesis independently using no sources or aids other than those indicated. I have appropriately declared all citations.

Do Dinh Tan Nguyen
Aachen, 09.07.2024

Contents

List of Figures

List of Tables

1 Introduction

1.1 Context

Vision impairment is a medical condition that poses a profound challenge for over 10 million people globally, stemming from the debilitating effects of retinal degenerative diseases such as diabetic retinopathy, macular degeneration, retinitis pigmentosa, and Stargardt's disease. The irreversibility and severe consequences of these conditions have prompted the scientific community to develop a range of sight restoration technologies with the objective of mitigating the impact of vision loss.

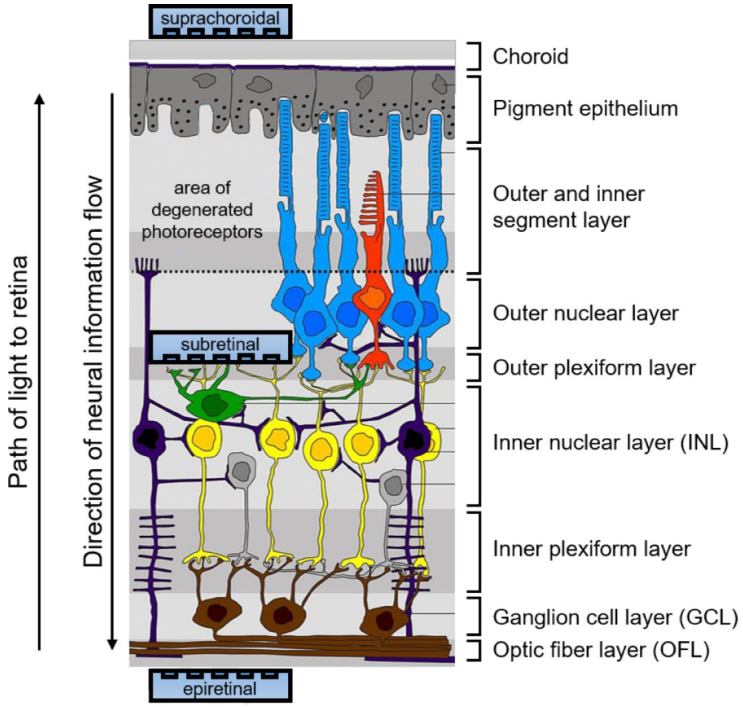


Figure 1.1: A cross-sectional diagram of the human retina [1].

As a result, retinal implants, also known as retinal prostheses or bionic eyes, have emerged as a promising solution for restoring vision, offering an alternative method to revive the gift of sight for those affected. To date, the journey of retinal implants from concept to clinical reality has seen the administration of these devices into more than 500 patients worldwide. Through electrical stimulation of surviving

retinal cells, these micro-scale prostheses invoke neuronal responses that the brain interprets as visual percepts.

Nevertheless, the technology still has a few shortcomings and potential for improvement. The visual experience produced by current retinal implants frequently fails to meet expectations, resulting in blurry or distorted imagery that impairs the user's perception and thus, quality of life. Moreover, the resolution of these devices is significantly limited. For instance, the Argus II [2] employs only 60 electrodes, arranged in a six-by-ten sparse array, to facilitate visual stimulation.

In addition, the biological mechanics of the human eye introduce further complexity to the situation. In contrast to the photographic device, the retina exhibits a markedly different underlying structure and operational mode. The retina is composed of ten distinct layers, each containing a unique subset of six different cell types [3], as illustrated in Figure 1.1. Each of these components plays a specific role and is interconnected by synapses, which facilitate the transmission of incoming photons into electrical potentials that are then processed by the brain's cortices into three-dimensional vision.

Figure 1.2: Scan patterns of fixations (circles) and saccades (lines) [4].

During the process of photon reception, or visual perception, there are multiple rapid movements, known as saccades, which are interspersed with moments of fixation. This allows for the maintenance of visual sharpness in a significant visual field, although our visual acuity is only at its peak in a narrow region of the retina, called the fovea. This relationship between the visual acuity and the distance from the fovea is illustrated in Figure 1.3. The intricate interplay between saccadic motion and focal acuity exemplifies the sophisticated nature of human vision and presents a complex challenge for existing retinal implants.

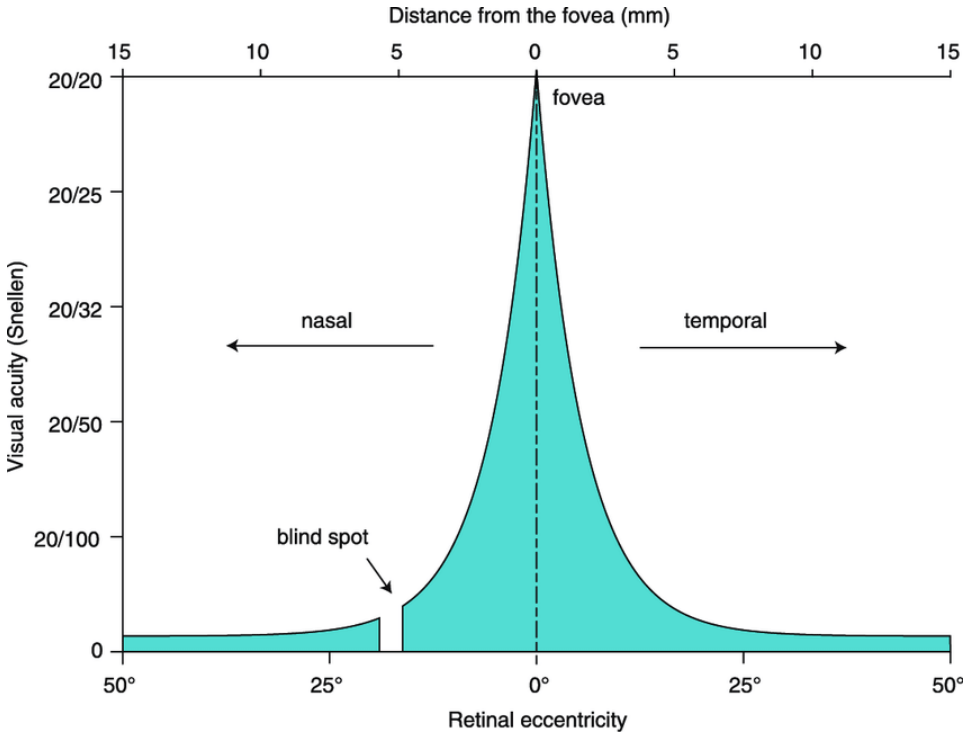


Figure 1.3: Visual acuity and the angle in degrees from the fovea relationship [5].

1.2 Thesis Objective And Outline

The discrepancies between the visual dynamics of the biological eye and the current technological limitations of retinal implants necessitate the development of a suitable solution that is attentive to these disagreements. Consequently, the objective of this work is to enhance the functionality of existing prostheses, thereby improving the level of object recognition in everyday life. Ultimately, by providing individuals afflicted by retinal degenerative diseases with a levitating visual experience, their integration into the world would be facilitated, as vision loss would not equate to a loss of independence or quality of life.

A novel approach to the objective is proposed, which involves the implementation of three distinct parts:

Firstly, the problem scope and boundaries are defined, thus enabling the presented results to be evaluated and compared in an objective manner. In light of the aforementioned objective, it is reasonable to frame the problem as an object classification task. Due to the limitations of the available hardware and the compressed timeline of a master's thesis, a subset of the Imagenet-1k dataset was utilized to reduce the computational work and speed up the training time.

Secondly, a standard pipeline is established to serve as a framework for obtaining baseline metrics and comparing further suggested improvements. The framework comprises the pre-defined subset of Imagenet-1k, a pre-trained state-of-the-art

DINOv2 model whose parameters are frozen at all times, and a process of linear probing for classification.

Finally, the proposed enhancement employs a U-net architecture as an encoder, which would be integrated into retinal implants as a vision processing unit (VPU). The encoder will optimize the image information to ensure that the retinal implants will send the most suitable electrical signals for the patient's brain to pick up on and to distinguish different daily life objects more clearly.

The following is a summary of the thesis structure. Chapter 2 provides an overview of the relevant literature in multiple fields related to this thesis topic. Chapter 3 presents the methodology employed in the study. Chapter 4 compares and discusses the results obtained. Finally, Chapter 5 draws the conclusion of the study and outlines the limitations of the research, as well as suggesting future avenues for investigation.

2 Literature Review

2.1 Deep Learning

Deep learning, a prominent subset of machine learning, has revolutionized various domains by enabling the modeling of intricate patterns in massive datasets and solving compound problems. The process entails training artificial neural networks with manifold layers, whose collective effects are hierarchical learning and the automatic extraction of valuable features from raw data. This approach has led to remarkable advancements in the domains of computer vision and natural language processing.

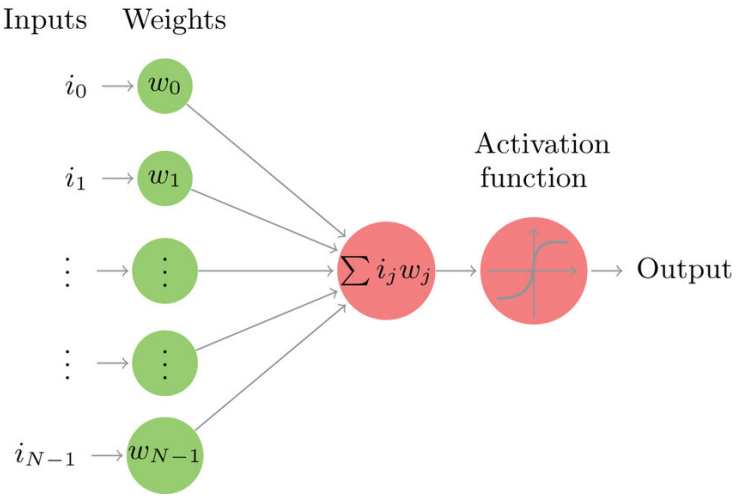


Figure 2.1: The structure of a classical perceptron [6].

The concept of artificial neural networks was first proposed in the mid-20th century, with the advent of the perceptron by Rosenblatt in 1958 [7]. Nevertheless, it was not until the 1980s, with the origination of the renowned backpropagation algorithm by Rumelhart et al. (1986) [8], that neural networks began to gather attention. In the forward phase of the backpropagation algorithm, the input data is propagated through the network to obtain the output predictions. Subsequently, the discrepancy between the predicted and actual target values, referred to as the loss, is calculated. The algorithm concludes with the backward pass, during which the gradient of the loss function is computed in accordance with the activations of the output layer. Finally, the weights and biases are adjusted using this computed gradient.

A second period of growth in the field started in the mid-2000s, driven by three key factors: increased computational power, the availability of large-scale datasets, and algorithmic improvements. Notable breakthroughs include the success of AlexNet in the ImageNet competition in 2012 [9] and the rapid emergence of sophisticated models such as Google's BERT [10] and OpenAI's GPT in recent years [11].

Deep learning is currently being employed in numerous domains across various applications, with an acclaimed impact in computer vision. Image classification, object detection, and semantic segmentation are just a few of the areas in which deep learning excels, often demonstrating significant performance gains over traditional methods. Similarly, in natural language processing (NLP), deep learning models have established new records in tasks such as machine translation, sentiment analysis, and text generation. This is evidenced by models such as BERT and GPT. Moreover, deep learning has had a profound impact on healthcare, with applications including assistance in disease diagnosis and the instigation of personalized medicine [12]. It also plays a pivotal role in the advancement of autonomous systems, including self-driving cars [13].

2.1.1 Visual Feature Learning

In the field of computer vision, visual feature learning refers to the process by which a computational system identifies and derives constructional representations of visual data that are useful for a variety of tasks, including object recognition, image classification, segmentation, and so forth. This process entails the extraction, representation, and utilization of features from images or videos, thereby enabling the system to perform complex visual tasks extensively.

Visual features can be grouped into the following categories:

1. **Low-Level Features**: These characteristics are fundamental and are derived directly from the raw pixel data, encompassing edges, corners, textures, and colors. Techniques such as edge detection (e.g., Canny edge detector), corner detection (e.g., Harris corner detector), and texture analysis (e.g., Gabor filters) are frequently employed to extract low-level features.

2. **Mid-Level Features**: This type of feature represents more complex patterns and structures in the image, such as shapes, contours, and regions. Algorithms like the Scale-Invariant Feature Transform (SIFT) and Speeded-Up Robust Features (SURF) are examples of mid-level feature extractors that search and discover local key spots or marks in the image that are invariant to scaling, rotation, and noise.

3. **High-Level Features**: These features are capable of capturing semantic information and are typically learned through deep learning models, particularly convolutional neural networks (CNNs). High-level features may be

assembled from entire objects or significant parts of objects within an image. CNNs automatically learn hierarchical structures of features from low-level edges to complex object parts through varying layers of both linear and non-linear operations.

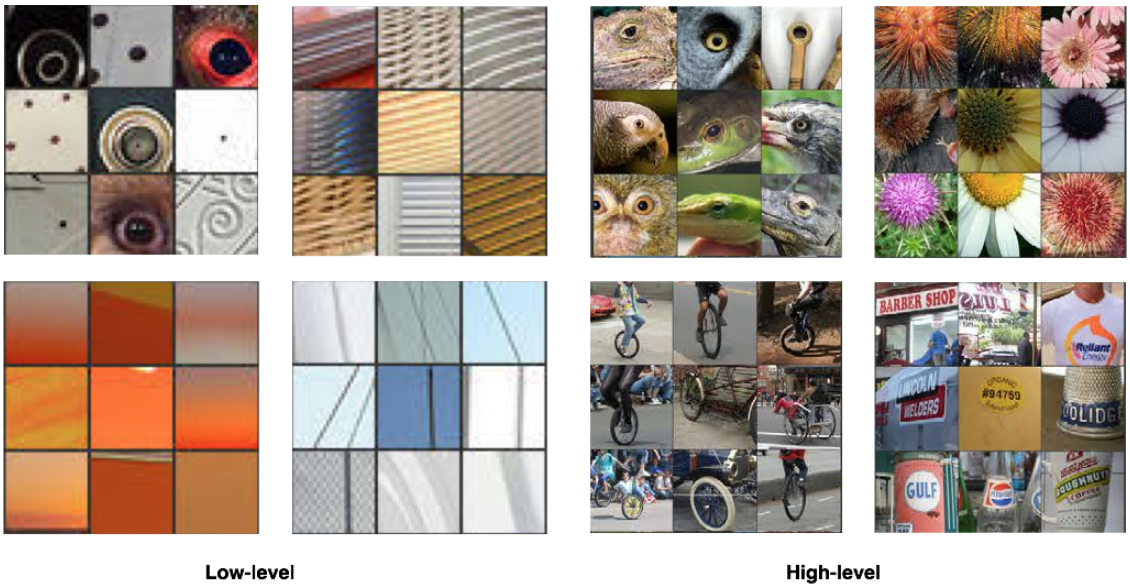


Figure 2.2: Low level vs. high level of visual features [14].

Prior to the advent of deep learning, feature extraction was predominantly reliant upon manually crafted methodologies. Techniques such as SIFT, SURF, or Histogram of Oriented Gradients (HOG) were extensively utilized. These methods necessitate domain expertise and extensive manual adjustment to implement effective feature extractors.

In the past decade, deep learning has transformed the field of feature learning by enabling the automatic extraction of hierarchical features from data. CNNs comprise a multitude of convolutional filters, which serve a function similar to traditional feature descriptors such as SIFT, SURF, and HOG. Combined with the backpropagation algorithm, the process of parameter adjustment and fine-tuning is automated computationally during the training session. This increases the capability of CNNs to identify increasingly complex features directly from the pixel data. This end-to-end learning approach enables the model to optimise feature extraction in a manner that is dependent on the specific task at hand.

The technology can be employed to achieve a number of objectives, including:

- Object Recognition: Identifying and categorizing objects within an image.
- Image Classification: Assigning a label to an entire image.
- Image Segmentation: Dividing an image into meaningful segments or regions based on semantic similarities.

- Facial Recognition: Detecting human faces by learning distinct facial features.
- Scene Understanding: Interpreting the overall context of an image by analyzing the spatial arrangement of objects and environments.

Despite the significant advances that have been made, visual feature learning still faces challenges. These include the need to handle occlusions, varying lighting conditions, and the necessity of having large, labeled datasets for training deep models. The current research directions in visual feature learning concentrate on improving unsupervised and self-supervised learning methods. This is done with the objective of reducing the dependence on labeled data, enhancing robustness to environmental variations, and developing more efficient models that can perform real-time processing on resource-constrained devices.

2.1.2 Architectures

The foundation of deep learning models rests upon artificial neural networks, which are motivated by the structural and functional organization of the human brain. Similar to the cerebrum, these networks are composed of layers of neurons, each of which manipulates the received inputs and transmits the result to subsequent layers. The architecture is typically designed with an input layer, multiple hidden layers, and an output layer. The intermediate hidden layers are of critical importance for the capture of sophisticated patterns and deep representations in data.

Two classic types of neural network architectures are convolutional neural networks (CNNs) and recurrent neural networks (RNNs). As previously stated, CNNs are renowned for their aptitude in processing image data and employ convolutional layers to facilitate the automatic and adaptive learning of spatial hierarchies. In contrast, RNNs are designed for processing sequential data or time series. Recurrent connections are utilized to retain information across time steps, rendering them suitable for tasks such as time series prediction and language modeling.

The recent breakthroughs have facilitated the accelerated development of more sophisticated architectures. Transformer models, as introduced by Vaswani et al. (2017) [16], have revolutionized NLP by enabling the processing of entire sequences simultaneously rather than sequentially. This has led to improvements in training efficiency and performance. These models have served as the foundation for state-of-the-art language models such as BERT and GPT in the following years.

2.1.2.1 U-net

Convolutional Neural Network (CNN) CNNs represent a cornerstone in the evolution of deep learning. This architecture is highly efficient in grid-like data, such as images and videos. One of the foundations of CNNs can be attributed

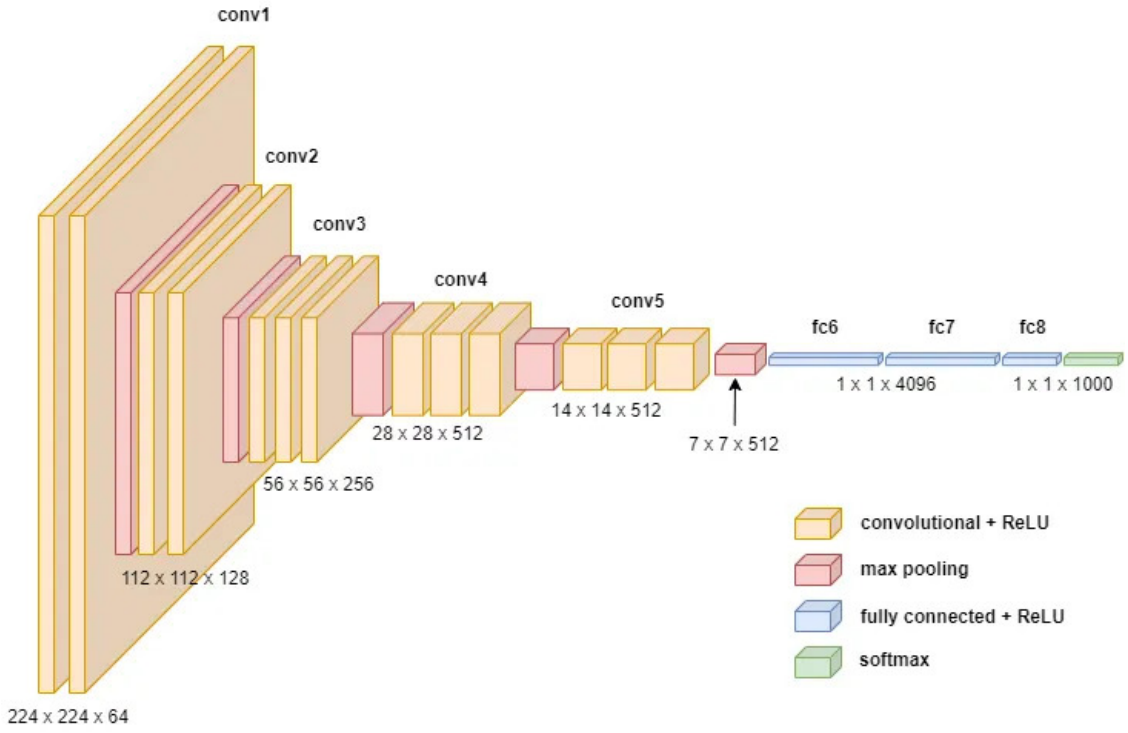


Figure 2.3: An example structure of the CNN architecture [15].

to the work on the neocognitron by Fukushima in 1980 [17], which introduced a hierarchical and multilayered structure for pattern recognition. The modern form of CNNs, however, was considerably influenced by LeCun et al. (1989) [18] through the development of LeNet-5, which demonstrated the practicality and effectiveness of CNNs in recognizing handwritten digits. The resurgence of CNNs in the 2010s was marked by the success of AlexNet in the 2012 ImageNet competition.

The fundamental building blocks of a convolutional neural network (CNN) can be typically classified into the following categories:

- **Convolutional Layer**: A set of learnable filters (or kernels) of relatively small sizes (e.g., 3x3, 5x5) are usually employed to convolve with the input data. However, in certain instances, larger-sized filters may also be utilized, contingent upon the specific CNN architecture and the specific task at hand. This process involves gliding the filters over the input data and computing the dot product between the filter parameters and the overlapping regions of the input matrices. The output of this operation is a set of feature maps, each corresponding to a different filter. The convolutional layer is designed to attend to local receptive fields, which assist the network in identifying spatially localized patterns.

- **Max Pooling**: The operation is based on the division of the input feature map into non-overlapping rectangular regions (typically 2x2 or 3x3) and the selection of the maximum value from each region. This results in a narrower

feature map that retains the most salient information while discarding less critical details. Max pooling significantly contracts the spatial dimensions and provides a degree of translation invariance, which helps the network recognize features regardless of their exact spatial position.

- **Fully Connected (FC) Layer**: In an FC (or Dense) layer, the input is transformed into the output by matrix multiplication, followed by the addition of a bias vector. Each output neuron computes a weighted sum of all input neurons, in addition to a bias term. The interconnection of neurons in a dense layer enables the capture of intricate relationships between features, rendering it an ideal choice for the final stages of a neural network, where extracted features are combined and predictions are made.

- **ReLU (Rectified Linear Unit)**: The ReLU is a widely employed activation function in CNNs. Its integration imbues the network with non-linear characteristics, enabling a more nuanced discernment of intricate patterns. By setting negative values to zero, the ReLU establishes sparsity within the network, which in turn facilitates enhanced computation efficiency and potential improvements in generalization.

- **Softmax**: The Softmax activation function is a frequent option for the output layer of classification networks. It converts the raw scores, or logits, into probabilities. This facilitates the interpretation of the model's predictions. The differentiability of the function permits the flow of gradients through it during backpropagation, thereby enabling the automated updates of the model's parameters during the training of the network.

U-net The U-net is a CNN architecture that has been specifically designed for biomedical image segmentation. It was first proposed by Ronneberger et al. in 2015 [19]. The distinctive symmetric U-shaped design of the U-net has contributed to its recognition and the structure is widely adopted as a compelling architecture for complex medical image analysis and robust segmentation in various contexts. Its innovative design allows for precise localization and utilization of context, rendering it highly effective in tasks where the exact portrayal of structures is of paramount importance.

As illustrated in Figure 2.4, the symmetric U-shape topology is characterized by a contracting path that is utilized for contextual feature capture and an expansive path that is employed for precise localization. The concatenation of these two paths, with a skip connection between them in each stage, enables the U-net to effectively learn hierarchical representations of medical images, thereby achieving accurate segmentation of anatomical structures and abnormalities.

The skip connections of the U-net constitute a crucial technical component. These connections bring forward a continuous fusion of high-resolution feature maps from the contracting path with the respective upsampled feature maps from the

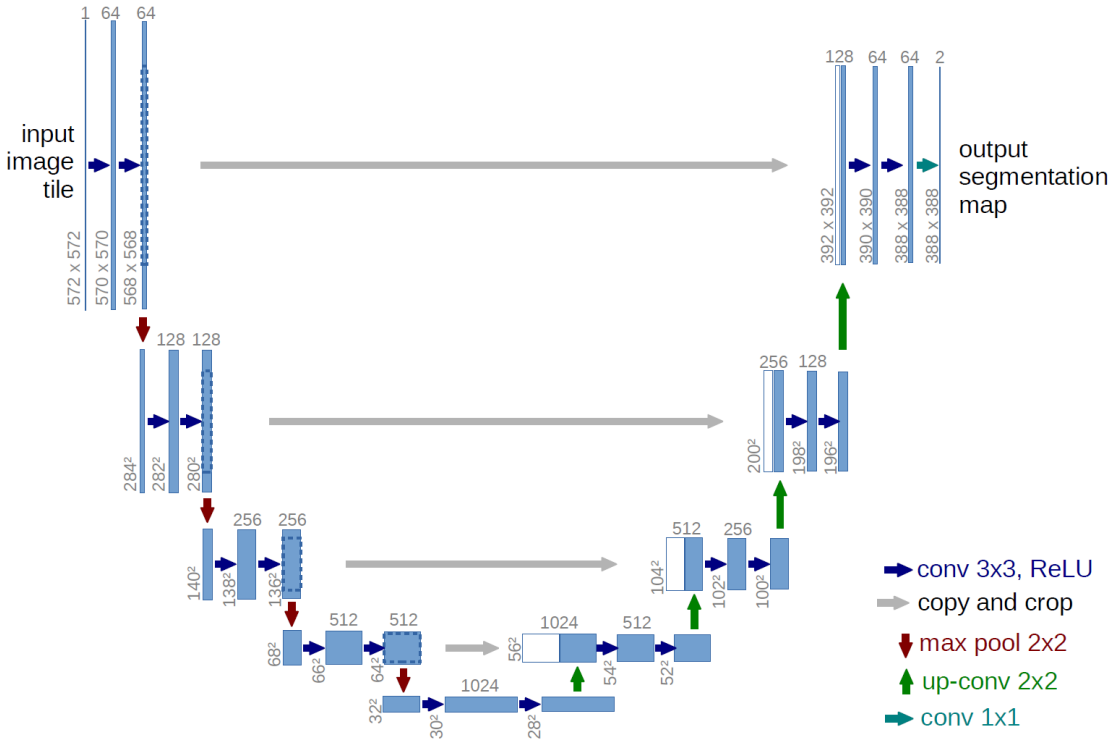


Figure 2.4: The classic U-net architecture [19].

expansive path. This fusion ensures that the high-resolution information diminished along the contracting path due to the down-convolutional layers is reintroduced after the up-convolutional layers in the expansive path. This enhances the ability of the U-net to apprehend fine-grained details, which is proven beneficial during segmentation.

2.1.2.2 Vision Transformer

Transformer The Transformer model [16] represents a significant breakthrough in the field of deep learning, particularly in the domain of NLP tasks. In contrast to traditional RNNs and their variants, which process data sequentially, the Transformer incorporates a mechanism known as self-attention with the powerful capability of processing input data in parallel. This cutting-edge technique has led to substantial improvements in both computational efficiency and model performance.

The Transformer model consists of multiple blocks of encoder-decoder structure, with each component comprising multiple layers of identical sub-layers. The encoder blocks receive the input sequence and generate a set of feature representations, while the decoder blocks use these representations to produce the output sequence.

Both the encoder and the decoder contain an essential sub-layer: a multi-head

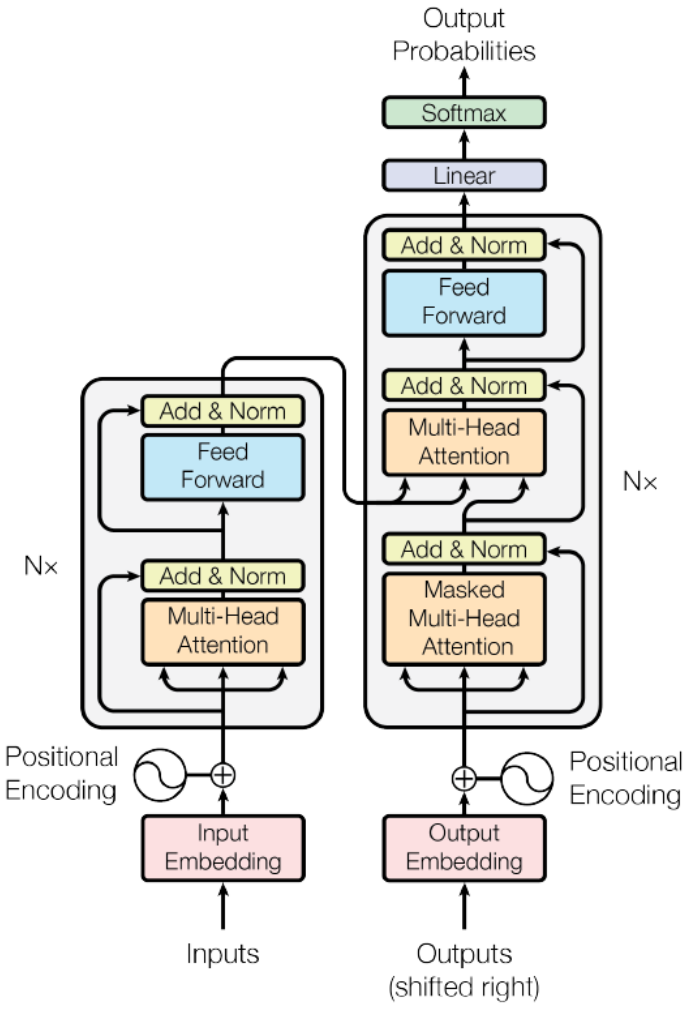


Figure 2.5: The Transformer model proposed in Vaswani et al. (2017) [16].

self-attention mechanism. This mechanics allows the model to focus on different parts of the input sequence simultaneously and to compute various attention scores in different attention heads, each representing a different knowledge subspace of the input features. As the parallel processing does not inherently capture the order of the sequence, positional encodings are introduced to the input embeddings beforehand to provide information about the related position of each token in the sequence.

The Transformer model is exceptionally notable for its outstanding ability to handle long-range dependencies. It has been applied to a diverse range of tasks, including:

- Machine Translation: The original Transformer model exhibited state-of-the-art performance on the WMT 2014 English-to-German and English-to-French translation tasks.

- Text Summarization: Models such as BERT (Bidirectional Encoder Representations from Transformers) and GPT (Generative Pre-trained Transformer) have demonstrated the efficacy of the Transformer architecture in achieving superior performance in text summarization and other language understanding tasks.
- Question Answering: Transformers have impressively enhanced the accuracy and efficiency of question answering systems by facilitating more persuasive comprehension of context and relationships within the text [20].

Vision Transformer The Vision Transformer (ViT) [21] is a pioneering model that is constructed upon the Transformer architecture, originally designed for NLP, to perform image classification tasks. This innovative approach has challenged the dominance of the convolutional layer in computer vision by leveraging the self-attention mechanism to model long-range dependencies within images. ViT has demonstrated competitive performance on image classification benchmarks, often surpassing state-of-the-art CNNs when trained on large datasets.

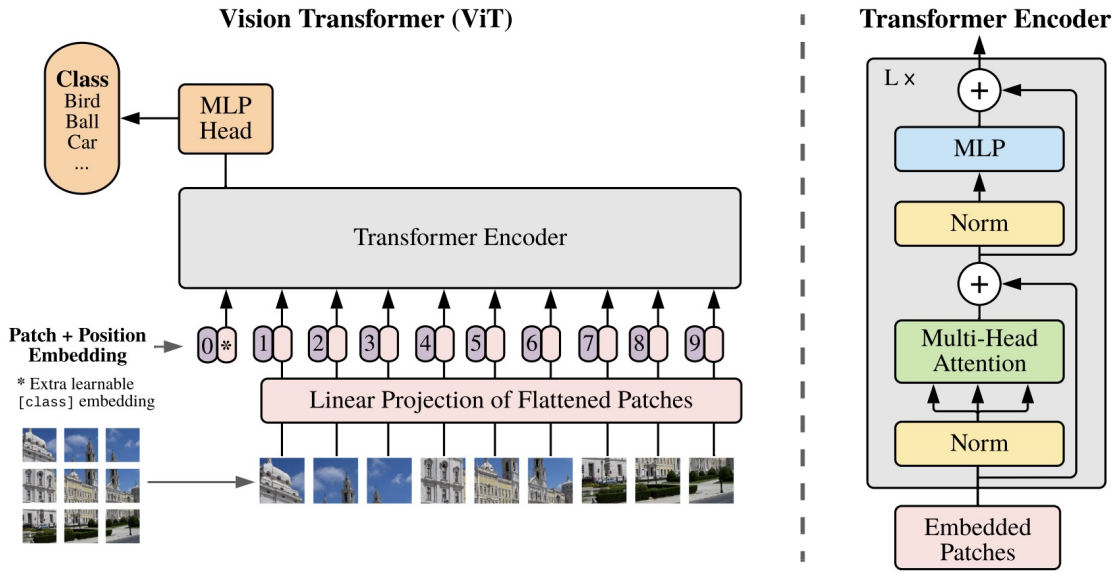


Figure 2.6: The Vision Transformer (ViT) architecture [21].

The ViT architecture modifies the Transformer model for adaptation to imagery data by segregating an image into a sequence of patches, analogous to the tokens in a sentence in NLP tasks. The overall structure of ViT can be summarized in several key components, as follows:

1. **Patch Embedding**:

   An image $\boldsymbol{I}$ of size $h \times w \times c$ (height, width, and channels) is dissected into a grid of non-overlapping squared patches, each of size $p \times p$ pixels. These

two-dimensional patches are flattened into one-dimensional vectors and then linearly embedded into a higher-dimensional space. Formally, given N patches where $N = \frac{h \times w}{p^2}$, each patch is linearly projected into an embedding vector of dimension D.

2. **Positional Encoding**:

   The encoding schemes are similar to those used in NLP Transformers, with formulas using sine and cosine functions of different frequencies:

$$PE_{(pos,2i)} = sin(\frac{pos}{10000^{2i/D}}),$$
$$PE_{(pos,2i+1)} = cos(\frac{pos}{10000^{2i/D}}),$$

   where *pos* is the patch position and i is the dimension.

3. **Encoder**:

   The core of the ViT model consists of a conventional Transformer encoder, which is composed of several identical layers. Each layer comprises two principal sub-layers: multi-head self-attention and a feed-forward neural network. Each sub-layer is followed by layer normalization and employs residual connections to stabilize the updates of learned parameters during training. The self-attention is calculated as follows:

$$\boldsymbol{Attention}(\boldsymbol{Q}, \boldsymbol{K}, \boldsymbol{V}) = softmax(\frac{\boldsymbol{Q}\boldsymbol{K}^T}{\sqrt{D}})\boldsymbol{V},$$

   where $\boldsymbol{Q}$, $\boldsymbol{K}$, and $\boldsymbol{V}$ are the query, key, and value matrices procured from the input embeddings.

4. **Classification Head**:

   Following the transformation of the input data through the Transformer encoder, the output from the classification token of the final encoder layer is utilized. This is then processed by a fully connected layer, which is subsequently followed by a softmax activation function, in order to provide the final output.

The ViT model necessitates the availability of large-scale datasets and substantial computational resources for effective training. Dosovitskiy et al. (2020) [21] demonstrated that ViT achieves state-of-the-art performance on image classification tasks when pre-trained on voluminous datasets such as JFT-300M, a proprietary dataset comprising 300 million labeled images. Following pre-training, the model can be fine-tuned on smaller datasets, such as ImageNet-1k, in order to achieve competitive results.

Moreover, the ViT model has also yielded remarkable outcomes in other image classification tasks, frequently outperforming traditional CNNs. The capacity

to model long-range correlations and process entire images in parallel confers advantages upon ViT over CNNs, particularly in the capture of global context and relationships within images. The success of ViT has prompted further research into the application of Transformer-based architectures to other computer vision tasks, including object detection, segmentation, and image synthesis. One illustrative example of this research is depicted in Figure 2.7.

Figure 2.7: A set of synthetic images generated by a Vision-Transformer-based VQGAN model trained on the ImageNet dataset [22].

2.1.3 Loss Functions

Loss functions are critical elements in the design and training of machine learning models because they provide a metric for evaluating the model's performance. By minimizing the loss function, machine learning algorithms can iteratively adjust the model parameters to enhance accuracy and generalize better to unseen data. It is therefore essential to understand and select appropriate loss functions in order to effectively train models for a range of tasks, including regression, classification, and beyond.

2.1.3.1 Mean Squared Error (MSE)

The MSE loss function, also known as the L^2 loss, is a loss function that is frequently employed in regression tasks. The MSE quantifies the average of the squared differences between the predicted values and the actual targets, thereby providing a measure of the degree to which a model's predictions align with the true data.

The mathematical equation of the mean squared error (MSE) loss function for a set of predictions is as follows:

$$MSE = \frac{1}{n}\sum_{i=1}^{n}(y_i - \hat{y}_i)^2,$$

where n is the number of observations, y_i is the desired value, and $\hat{y}_i$ is the predicted value for the i-th observation.

2.1.3.2 Cross-Entropy

The cross-entropy loss, also known as the *log* loss, is a fundamental loss function that is widely used in classification tasks. It is utilized to determine the difference between two probability distributions: the true distribution of the labels and the predicted distribution of the model. The cross-entropy loss function quantifies the degree of alignment between the model output probabilities and the actual class labels, thereby serving as a crucial metric for training models that return probabilities.

In the context of multi-class classification, the cross-entropy loss can be generalized as follows:

$$CE = -\frac{1}{n}\sum_{i=1}^{n}\sum_{c=1}^{C} y_{i,c} log(\hat{y}_{i,c}),$$

where C is the amount of classes, $y_{i,c}$ is a binary target (0 or 1) if class label c is the correct classification for observation i, and $\hat{y}_{i,c}$ is the predicted probability of observation i of class c.

2.1.4 Self-supervised Learning

The paradigm of self-supervised learning (SSL) has emerged as a compelling alternative to traditional supervised learning in machine learning. Unlike supervised learning, which relies on large amounts of labeled data, SSL exploits the intrinsic structure within the data to automatically generate supervisory signals. This approach enables models to learn useful representations from vast amounts of unlabeled data, rendering it particularly valuable in scenarios where labeled data is scarce or very expensive to acquire.

The fundamental concept of SSL is the creation of pretext tasks, which are auxiliary tasks designed to provide supervision without requiring labeled data. These tasks are formulated in such a way that solving them necessitates the learning of meaningful representations of the input data. Once a model has been pre-trained on these pretext tasks, it can be fine-tuned on downstream tasks using a smaller available collection of labeled data. The key mechanisms and methodologies employed in SSL include masked language modeling, contrastive learning, and generative modeling.

Contrastive learning is a technique that has gained considerable traction in the field of machine learning. Its objective is to facilitate the learning of representations by comparing similar and dissimilar pairs of data points. The model is trained to minimize the distance between representations of similar pairs while maximizing the distance between representations of dissimilar pairs. A prominent example of contrastive learning is the SimCLR framework [24], which employs the process of data augmentation to create positive pairs and uses other instances in the batch as

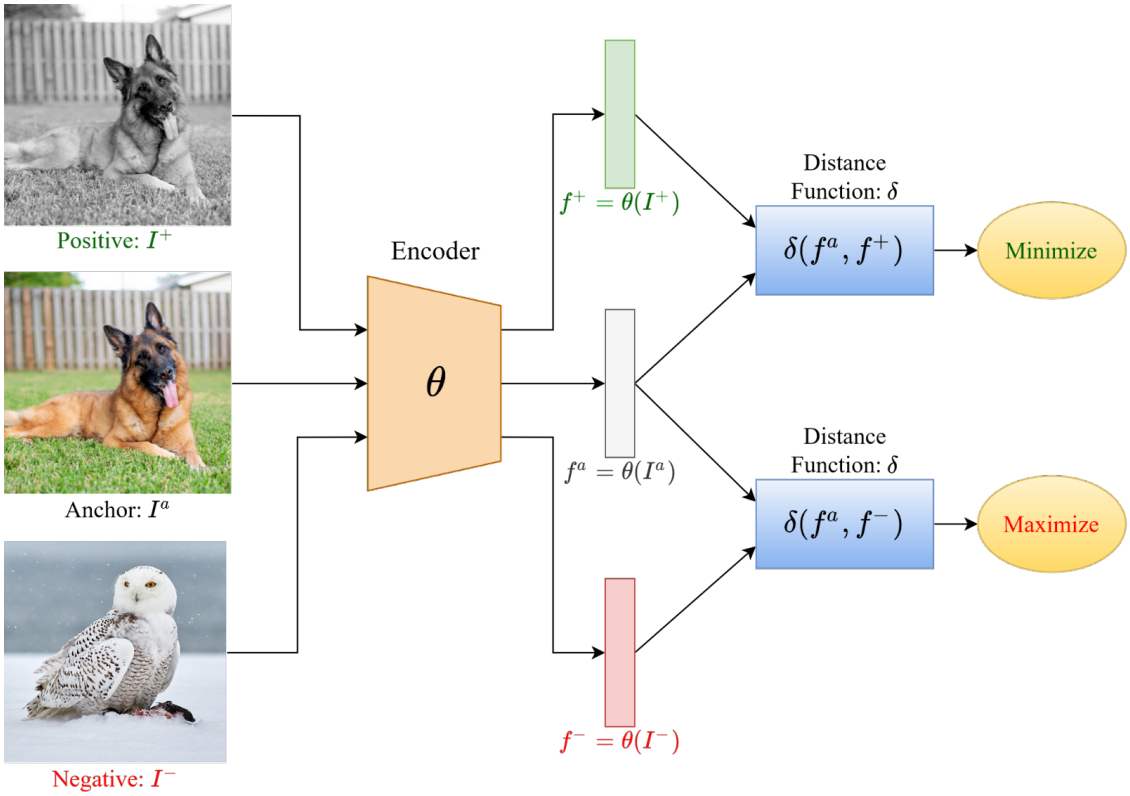


Figure 2.8: An example of contrastive learning techinque [23].

negative pairs. The loss function most commonly employed is the contrastive loss or its variants.

Masked language modeling entails the training of models with the objective of predicting the absent elements of the input data. The missing components are typically masked, compelling the model to predict the concealed tokens based on their contextual information. This approach has been demonstrated to be particularly effective in the NLP field, as evidenced by models such as BERT.

Generative models are designed to represent the underlying distribution of the data. This frequently entails training models to generate closely realistic data samples from a latent space. Examples of such models include autoencoders, variational autoencoders (VAEs), and generative adversarial networks (GANs). In the context of SSL, these models can learn vigorous representations by reconstructing the input data. For instance, VAEs are trained to encode input data into a latent space and then decode it back to the original data distribution, with the objective of optimizing a combined reconstruction and regularization loss.

The efficacy of SSL has been demonstrated across a wide range of applications, frequently achieving state-of-the-art results or providing considerable performance improvements. In addition to its applications in common computer vision and NLP tasks, SSL has also been employed in the domains of speech and audio processing.

One noteworthy illustration is the wav2vec model [25], as detailed in reference [20]. This model learns representations from raw audio signals that can be further optimized for speech recognition and other audio-related tasks.

2.1.4.1 DINOv2

DINOv2 [26] is an advanced deep learning framework designed to enhance SSL for computer vision tasks. It builds upon the foundation of the original DINO (Distillation with No Labels) [27] framework, introducing improvements that address the limitations of the previous method. DINOv2 leverages the power of self-distillation and contrastive learning to achieve superior performance in representation learning. The core innovation of DINOv2 lies in its formulation of the SSL task and optimization of the learning process, which enables the extraction of more meaningful and robust features from unlabeled data.

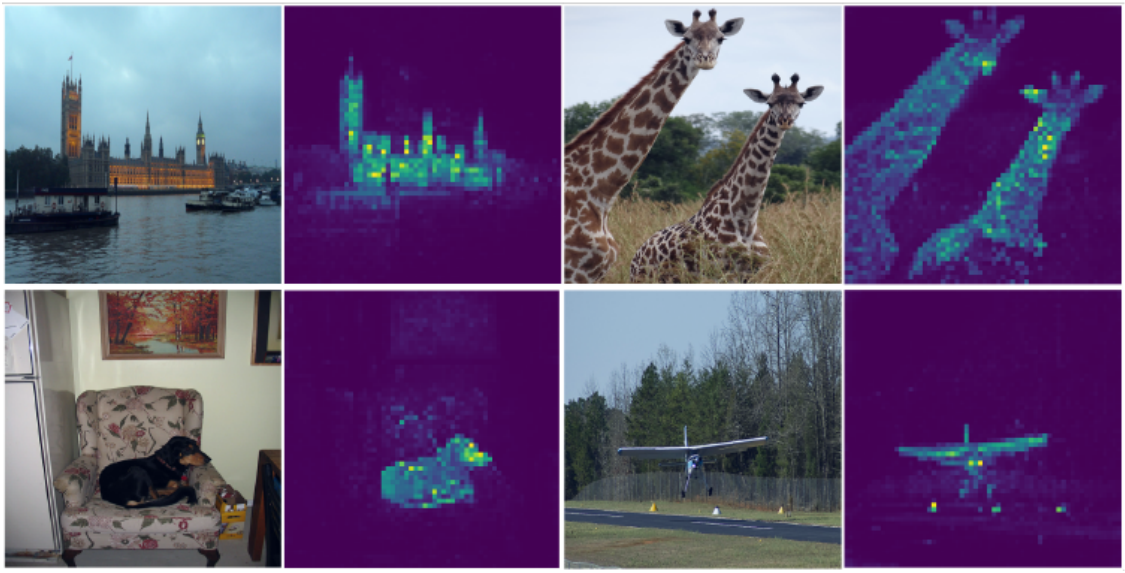


Figure 2.9: The self-attention of the [CLS] token on the heads of the last layer indicates that the model is capable of automatically learning class-specific features, which enables it to perform unsupervised object segmentations. [27].

DINOv2 has garnered attention for its effectiveness and versatility, as demonstrated by its competitive performance on benchmark vision tasks, including image and video classification, object detection, and semantic segmentation. DINOv2 excels in scenarios where labeled data is limited or unavailable, thus becoming an invaluable tool for unsupervised representation learning. The high-dimensional and high-generalized features learned by DINOv2 can be transferred to downstream applications, underscoring its significance in advancing the frontiers of SSL and providing substantial improvements over traditional methods that rely on labeled data.

Self-distillation Self-distillation is a technique that enables model refinement through the transfer of knowledge from a model onto itself. This transformative approach employs the principles of distillation, originally introduced as a learning process whereby a large, complex model (the "teacher") transfers its knowledge to a smaller, trained counterpart (the "student"). The student is typically trained to emulate the outputs of the teacher. The teacher generates target representations that the student attempts to match, and over successive iterations, the student gradually acquires better features. In self-distillation, however, the teacher and student are identical networks, but with different parameters.

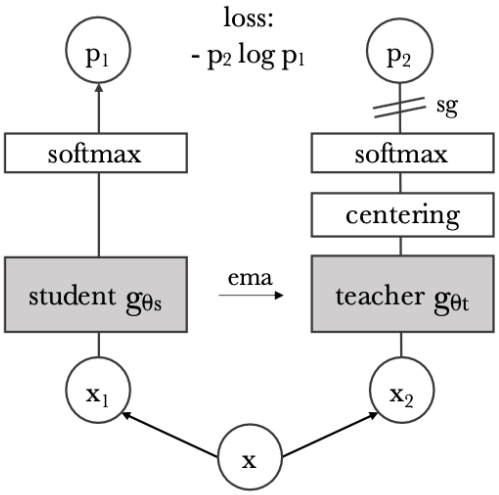


Figure 2.10: The process of self-distillation without the use of labels. [27].

In the DINO technique as illustrated in Figure 2.10, the teacher produces "soft" targets, which are then used to supervise the student. During the student's training, the teacher's parameters are periodically updated using an exponential moving average (EMA) of the student's parameters, ensuring stable and consistent learning dynamics.

The loss function employed in self-distillation is typically a mean squared error (MSE) or a Kullback-Leibler (KL) divergence between the student's outputs and the teacher's soft targets. This arrangement encourages the student to learn representations that are of a similar quality to those learned by the teacher. In addition, there are a number of other diverse supplementary strategies and techniques, including temperature scaling, feature distillation, and attention distillation, which are tailored to assist and distil specific aspects of the model's knowledge for self-improvement.

Self-distillation has emerged as a state-of-the-art technique for model optimization, performance enhancement, and knowledge consolidation in deep learning. Notably, self-distillation has demonstrated exceptional improvements in model compression, regularization, and robustness, leading to refined models with enhanced

generalization and adaptability.

Contrastive Learning In addition to self-distillation, DINOv2 employs contrastive learning to distinguish between similar and dissimilar image pairs. This mechanism facilitates the learning of invariant features that are resilient to variations in the input data. Initially, the input images are augmented to create multiple views of the same image. These augmentations encompass a diverse array of transformations, including cropping, flipping, color jittering, and Gaussian blurring. Each view is designated as a positive example for the other views of the same image, while views of different images are designated as negative examples. Subsequently, the contrastive loss function is employed to maximize the similarity between positive pairs and minimize the similarity between negative pairs. This objective encourages the model to learn discriminative features that can be generalized across different views of the same image.

The fundamental aspect of DINOv2 is the training of a ViT through a teacher-student framework, a self-distillation technique, in conjunction with the principle of contrastive learning. The objective is to enhance the learning of visual features. Other significant enhancements in DINOv2 include data augmentation strategies, an innovative momentum update mechanism, and enhanced regularization techniques. Collectively, these contribute to the efficacy and robustness of the ViT backbone model in terms of gaining knowledge of rich and diversified representations. Moreover, DINOv2 introduces novel strategies for handling large-scale and high-resolution visual data, addressing critical challenges associated with training ViTs on extensive and manifold datasets.

Linear Probing Linear probing is a technique employed in the evaluation of representation learning models, particularly in the context of self-supervised learning. The technique entails training a simple linear classifier on top of fixed, pre-trained representations with the aim of assessing the quality of the learned features. This approach provides a quantitative assessment of the degree to which the data is linearly separable in the representation space. This serves as an indicator of the usefulness and effectiveness of the learned features for downstream classification tasks [28].

The primary advantage of linear probing lies in its simplicity and interpretability. By employing a linear classifier, it is possible to isolate the quality of the learned representations from the complexity of the classifier, thereby providing a clear measure of the linear separability of the feature space. This makes it easier to directly compare different representation learning methods.

In a usual linear probing setup, a representation learning model is first pre-trained using a self-supervised learning objective. This is followed by the model encoding the input data into a lower-dimensional feature space. Once the model

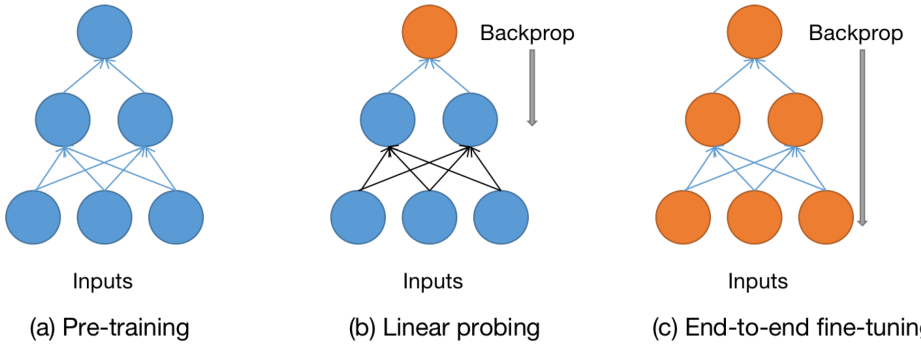


Figure 2.11: The pre-trained model parameters are retained throughout the linear probing step, during which the classifier is iteratively updated. In the context of end-to-end fine-tuning, both the model and the classifier are iteratively updated [29].

has been trained, its parameters are frozen or fixed, and a linear classifier, such as logistic regression or a single-layer neural network, is trained on these fixed representations using labeled data from a downstream task.

State-of-the-Art Results The DINOv2 technique has demonstrated efficacy in a multitude of applications within the field of SSL, including image and video classification, instance recognition, semantic segmentation, and depth estimation.

In the study by Oquab et al. (2023), the technique is applied to a range of variations of the ViT model, with the resulting performance metrics being collected and compared with those of other SSL and weakly supervised techniques in Table 2.1 and Figure 2.2.

Method	Architecture	ImageNet-1k	Kinetics-400
OpenCLIP	ViT-G/14	86.2%	78.3%
MAE	ViT-H/14	76.6%	54.2%
iBOT	ViT-L/16	82,3%	72.6%
DINOv2	**ViT-S/14**	81.1%	67.8%
	ViT-B/14	84.5%	73.2%
	ViT-L/14	86.3%	76.3%
	ViT-g/14	86.5%	78.4%

Table 2.1: Linear evaluation (top-1 accuracy) of frozen pre-trained features on the image (ImageNet-1k) and video (Kinetics-400) classification tasks [26]. *OpenCLIP is a weakly supervised method while the others are SSL. **The ViT-S/14 architecture is used in this thesis.

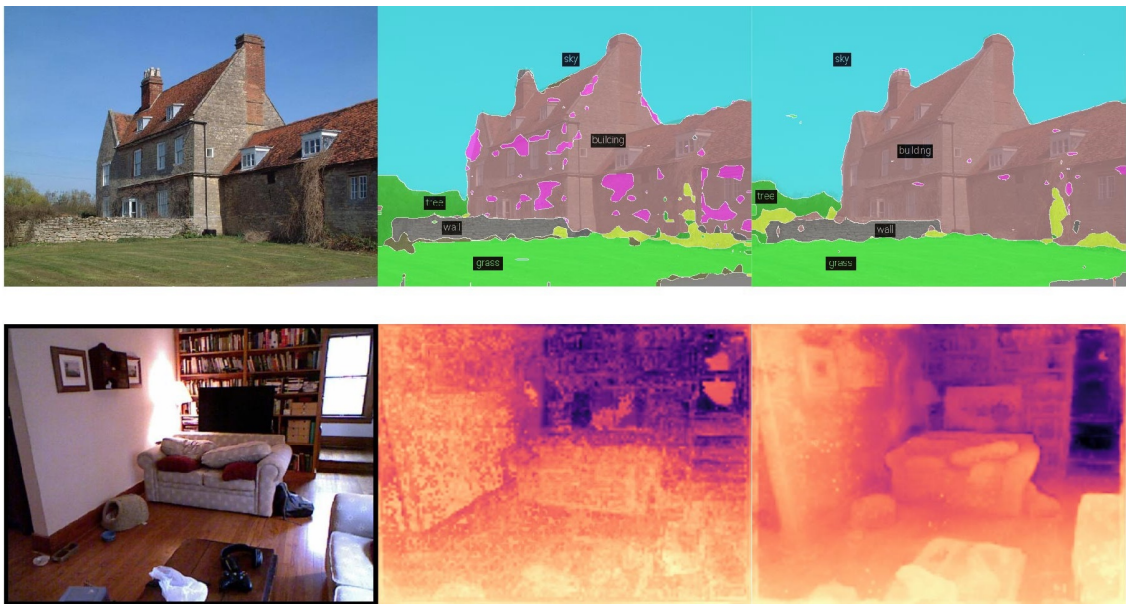


Table 2.2: Examples of semantic segmentation (top) and depth estimation (bottom) with linear probe on frozen OpenCLIP-G (middle column) and DINOv2-g (right column) features [26].

2.2 Retinal Implant Simulation

Retinal implants, or retinal prostheses, have attracted considerable attention in the field of vision restoration, particularly for patients with profound visual impairment. Retinal implants function by electrically stimulating surviving retinal cells to elicit neuronal responses, which are then interpreted by the brain as visual percepts, commonly referred to as "phosphenes." The phosphenes produced by these devices are typically grayscale, a consequence of the condition of the retinal cells and the limited capabilities of the hardware. The current landscape of retinal implants is marked by several notable aspects:

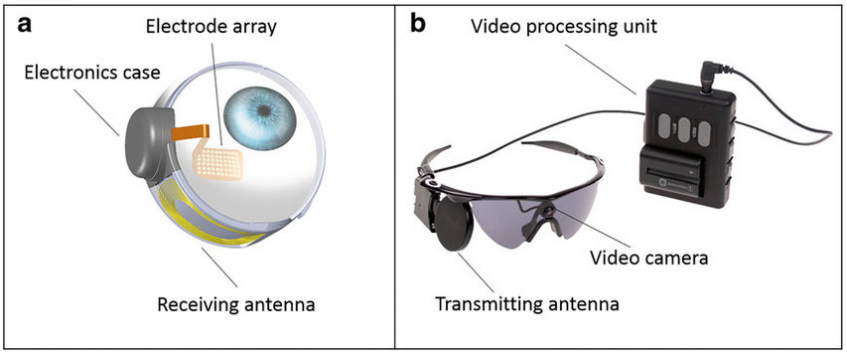


Figure 2.12: Two components of the Argus II system: the implanted hardware on the left and the external device on the right. [30].

- **Hardware Models**: Leading retinal implant hardware models, including Argus II, BVA24 and PRIMA, have been surgically implemented in over 500 patients worldwide, reflecting the continued advancement of bionic eye technology and its clinical adoption.

- **Perceptual Experience**: An area of critical concern pertains to the interactions between the electronic components of retinal implants and the underlying retinal neurophysiology, which can give rise to perceptual distortions. These distortions have the potential to significantly impact the quality of the induced visual experience, therefore warranting comprehensive investigation and mitigation strategies.

- **Computational Modeling**: Computational models, such as those provided by open-source frameworks like pulse2percept, play a pivotal role in simulating and understanding the visual experiences created by the retinal implants. These simulations, based on the hardware models and the understanding of the neurophysiology factor, are useful for providing realistic estimations of prosthetic vision, offering valuable insights that can be used to refine and optimize current and prospective retinal implant technologies.

2.2.1 Phosphene Characteristics

When the first retinal implant was designed and implemented, it was assumed that each electrode of the implant array would function as an independent light source, projecting images into the patient's brain in a manner analogous to the light bulb that comprises the scoreboard in a sports stadium. The process is analogous to that of displaying a slideshow on a projection screen or television. This was the inaugural model of a simple scoreboard that mapped the linear relationship between the electrical stimuli and the phosphenes, or visual perceptions.

However, the phosphenes evoked by the retinal implants, which are based on the Scoreboard model, are highly malformed visions and appear to become "blobs," "streaks," and "wedges" instead of sharp imagery. This phenomenon is partially attributable to the composition of the nerve fiber bundles (also referred to as axons) [31]. These bundles are organized into hierarchical structures. Each axon is an elongated part of a neuron that transmits electrical impulses to the central nervous system.

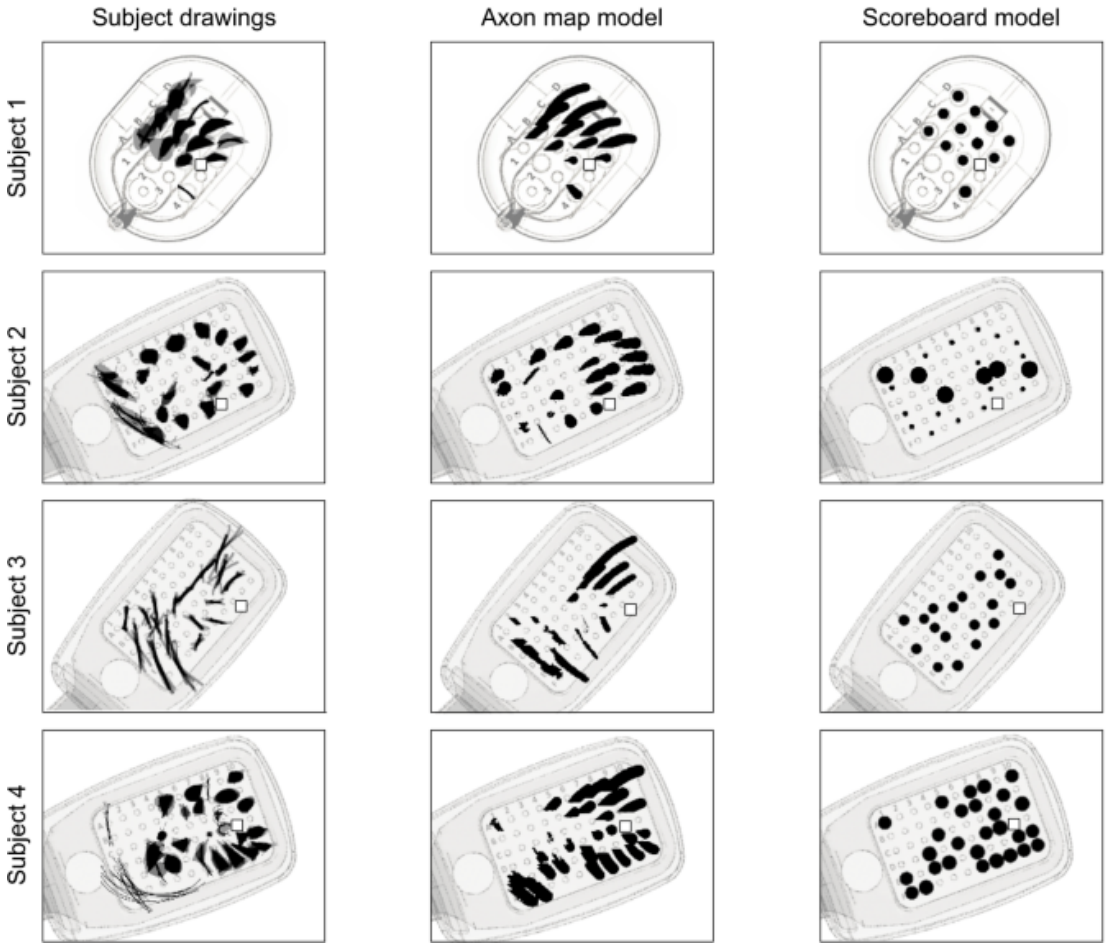


Figure 2.13: Phosphene drawings (left column) of four patients compared with the phosphene predictions of the axon map model (middle column) and the scoreboard model (right column) [31].

In the retina, these axons constitute one of the innermost layers and are composed of ganglion cells. The axons converge to form the optic nerve disc, which is the point at which these nerve fibers exit the retina to form the optic nerve. Empirically, the activation of a nerve fiber passing under a stimulating electrode is not separable from the perception that would be triggered by the activation of the corresponding ganglion cell. The phosphene may manifest at the site where the receptive cell encodes information, which could be hundreds of microns distant from the actual spot of the stimulation. To illustrate, as depicted in the upper right sketch in Figure 2.14, stimulation in the red circle could activate ganglion cells in the small black circles, resulting in a phosphene that is elongated along the orientation of the nerve fiber bundle trajectory.

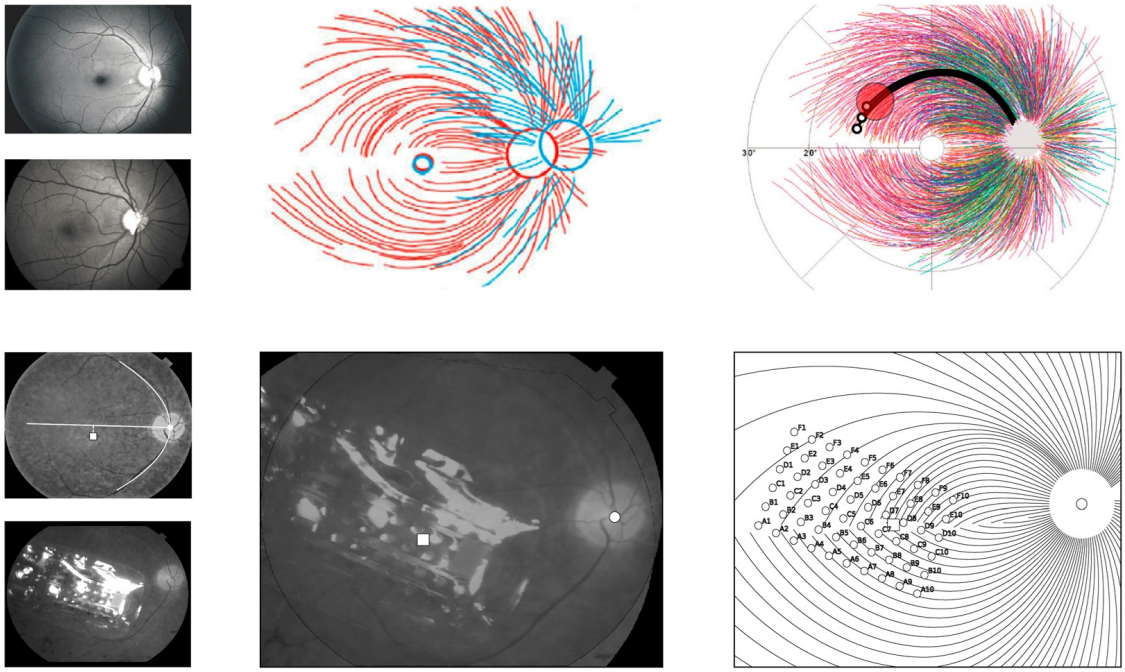


Figure 2.14: Top: The arrangement of optic nerve fiber bundles (red and blue lines) is highly universal in the human retina. Bottom: The implanted electrode array on the retinal surface (□: foveal pit, ○: optic disc) is overlaid on the axon map. [31].

This knowledge can be utilized to construct a mathematical representation of the phenomenon, based on the empirical data provided by the patients. Two parameters are introduced: ρ, which represents the decay constant for sensitivity orthogonal to the axon, and λ, which represents the decay constant for stimulation sensitivity along the axon, as demonstrated in Figure 2.15.

In addition, the distance between the electrodes and the retina affects the value of ρ, but not λ. When the electrodes are situated at a greater distance from the retinal surface, the value of ρ becomes significantly elevated, resulting in the phosphene becoming more circular in appearance (commonly referred to as a "blob"). Conversely, when $\rho < \lambda$, the phosphene becomes thin and elongated (the "streak").

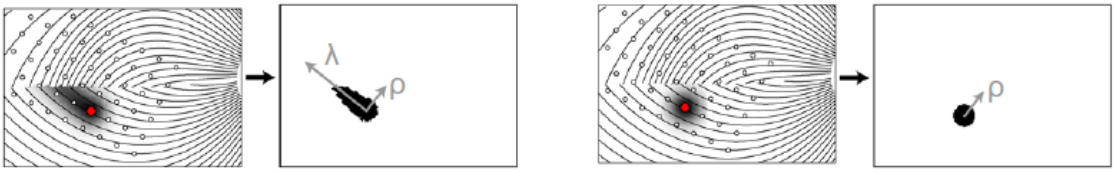


Figure 2.15: Two decay constants, ρ and λ, of the Axon Map model [31].

The relationship between electrode-retina distance and simulated phosphenes is portrayed in Figure 2.16.

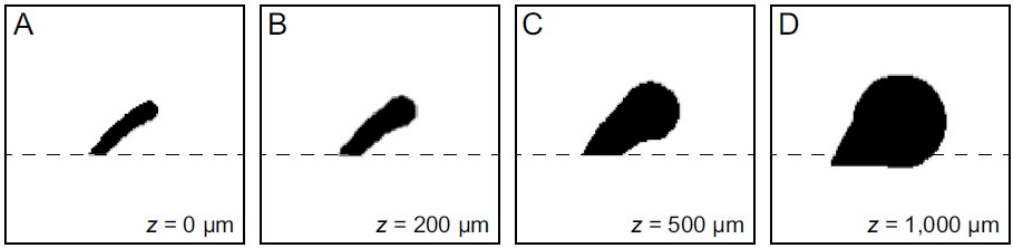


Figure 2.16: Simulated phosphenes in four cases of different distances, z, between the electrode and the retina [31].

This mathematical model is referred to as the Axon Map model. Other models have been proposed that involve only spatial information, such as the Thompson's model [32] and the Scoreboard model [31], or only temporal information, as featured in Horsager's model [33]. Alternatively, models have been developed that consider both spatial and temporal information, as evidenced by Nanduri [34] and Granley's models [35].

2.2.2 Stimulus Optimization

The perplexing and non-linear relationship between the electrical stimuli and the phosphenes causes a serious issue for individuals who have undergone implantation. Thus, the optimization of these stimuli is paramount for improving the quality of the perceived visual information and the independence of patients in daily life tasks.

The advent of recent advances in machine learning and signal processing has led to the incorporation of encoders into the stimulus optimization process. Encoders are neural networks that assist prostheses in transforming visual information into optimized electrical stimulation patterns, thereby enhancing the efficiency and effectiveness of those devices. This approach capitalizes on the capacity of neural networks to learn intricate mappings between visual stimuli and the corresponding neuronal activation patterns in the brain that give rise to visual perceptions.

One of the suggested architectures for the encoder is a CNN [36–38] due to its demonstrated efficacy in processing image data. In general, the encoder re-

ceives visual input, processes it through numerous layers of convolutional filters, and outputs a representation suitable for driving the electrodes in the implant. In particular, the encoder will be fed images and will output the values of the active electrodes, which will then be used by the computational models of the retinal implants to produce predicted perceptions. The percepts will be compared with the source images by the pixel-wise MSE function in order to iteratively refine and evaluate the encoder through backpropagation.

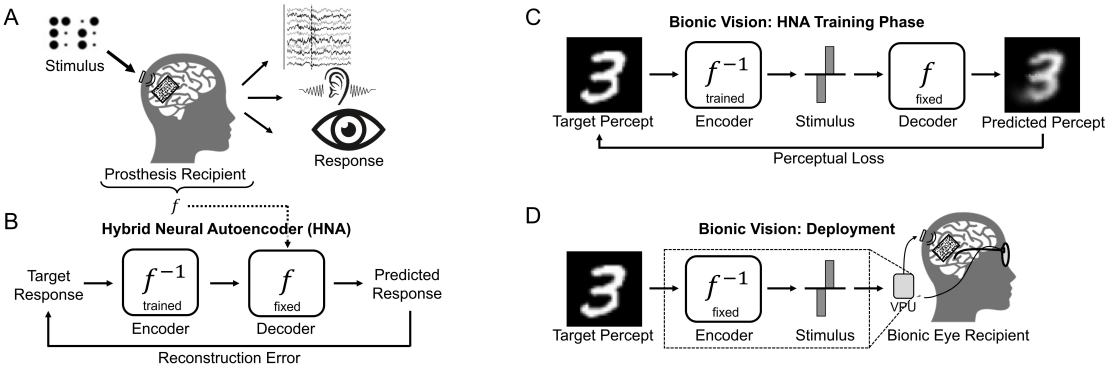


Figure 2.17: A typical process of stimulus optimization begins with a generation of the phosphene by the forward function f. The phosphene patterns are then used to train an encoder f^{-1}, which is subsequently used to predict the patterns of electrical stimulation required to elicit a phosphene as close to the target as possible [37].

2.2.3 Pulse2percept

The pulse2percept module, developed in Python, presents an open-source library of a wide range of the aforementioned computational models designed for advanced visual implants, such as ArgusII, BVA24, and PRIMA. The library offers tangible methods for acquiring quantitative insights into the visual experiences of real and simulated patients implanted with such prosthetic devices, as illustrated in Figure 2.18.

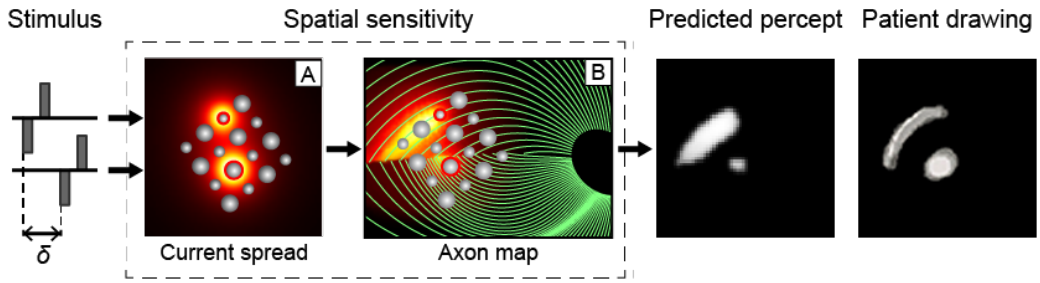


Figure 2.18: The comparison between the predicted phosphene using pulse2percept library and the actual drawing from patient for the same stimulus [1].

This assistant implementation provides researchers and developers with valuable tools for realistic assessments of prosthetic vision and determining appropriate tests to evaluate the performance of visual prostheses. It also contributes to the enhancement of existing and future technologies in this domain. The following paragraphs will explain the three fundamental concepts of this module.

- **Visual Prostheses**:

  Pulse2percept library offers a variety of prosthesis systems (or retinal implants). Each `ProsthesisSystem` is composed of an `ElectrodeArray` object and a `Stimulus` object, optionally. An `ElectrodeArray` is an assembly of `Electrode` objects, which represent the physical electrodes of the prostheses. These electrodes provide the electrical potentials for the stimuli in a specific spot within the retina.

- **Electrical Stimuli**:

  One of the functions of prostheses is to translate images and videos into electrical signals (or `Stimulus` object). The `stimuli` module provides a range of typical electrical stimulus types, which can be assigned to electrodes of a `ProsthesisSystem` object. The various types of electrical stimuli include monophasic pulse, biphasic pulse, and asymmetric biphasic pulse, or a composite of these stimuli into a pulse train.

- **Computational Models**:
  The `models` module supplies many published and verified computational models that may be employed to forecast neural responses or perceptual outcomes resulting from electrical stimulation in the visual cortex. A `Model` object is comprised of two components: a `SpatialModel`, which details the impact of electrical stimulation on the neural tissue or elicited phosphene in different spatial locations of the visual field, and a `TemporalModel`, which delineates the temporal progress of the neural tissue or elicited phosphene response.

2.2.4 Fixation / Saliency Prediction

2.2.4.1 Visual Fixation

The term "visual fixation" refers to the continuation of gaze at a single position, which is a critical function of the human visual system. It allows attention to be focused on a particular region or spot in the visual field. This process is the result of a synchronized effort of several elements, including the eye muscles, neural control, and micromovements. Each of these elements plays a important role in ensuring effective visual fixation.

The coordinated effort of the extraocular muscles is essential to maintain the

stability and alignment of the eyes with the target. Six muscles encircling each eye are in control its movement, thereby providing precise motion and stabilization during fixation [39].

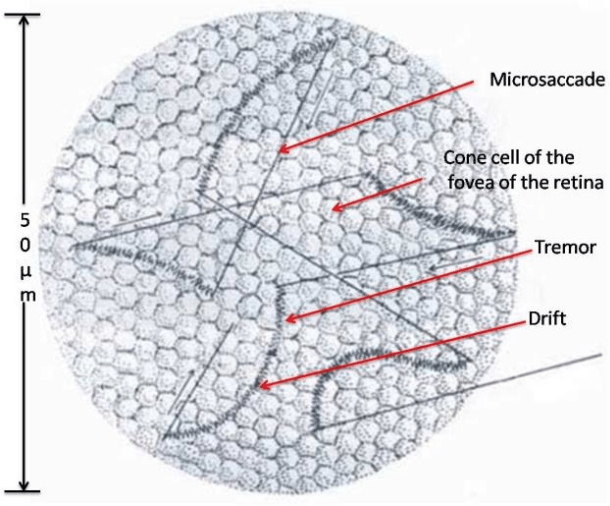


Figure 2.19: The measurement results of a variety of eye movements [40].

The neural control of visual fixation is a complex process involving interactions between several brain regions, from the frontal eye fields to the superior colliculus. The frontal eye fields, situated at the prefrontal cortex, are responsible for voluntary eye movements, while the superior colliculus, positioned in the midbrain, is in charge of the reflexive eye movements and coordination [41]. Collectively, these regions process the visual information and dictate motor responses.

Even when the eyes are fixated on a single point, they are not completely stationary. This phenomenon can be attributed to the occurrence of micromovements during the process of fixation. These micromovements, also referred to as microsaccades, are involuntary movements that serve to counteract the fading of vision, which occurs when retinal stimulation stays unchanged for an extended period and immobile objects are no longer perceived. This phenomenon is referred to as the Troxler effect. Microsaccades reanimate the view on the retina, thereby maintaining visible and stable perception [42].

The capacity to maintain visual fixation is of great importance for a multitude of activities, including reading, driving, and other tasks that require a detailed perception. Without visual fixation, significant difficulties may arise in the performance of these tasks. Therefore, an in-depth comprehension of the intrinsic mechanisms underlying visual fixation can provide valuable insights and is necessary in the optimization process of the artificial perception stimulated by the retinal implants.

2.2.4.2 Saliency Prediction

Saliency prediction is a computer vision task that aims to identify and highlight the most visually significant regions of an image. This task attempts to emulate the human visual system's capacity to direct attention to the most salient elements of a scene. Saliency prediction models generate saliency maps, which are spatial distributions that indicate the likelihood of each pixel being attended to by a human observer.

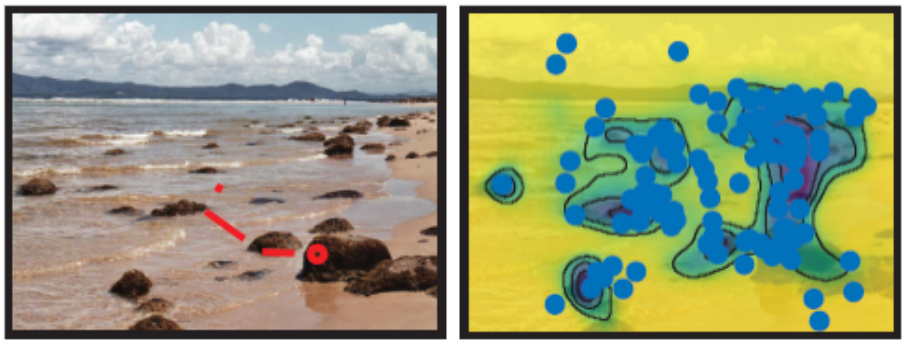


Figure 2.20: The comparison between the human eye fixations (left) and the computed saliency prediction map (right) [43].

The field of saliency prediction has its roots in psychological and neurobiological studies of human visual attention. Early studies, such as the feature integration theory proposed by Treisman and Gelade in 1980 [44], suggested that attention is determined by salient features, including color, intensity, and orientation. These ideas were subsequently formalized into computational models. For instance, Itti et al. (1998) [45] introduced one of the first computational models for saliency prediction. The model employed a multi-scale image representation and a linear combination of feature maps of color, intensity, and orientation to churn out a saliency map.

The emergence of deep learning has led to consequential changes in the field of saliency prediction. In particular, CNNs have gained notoriety for their ability to capture visual features and learn from substantial datasets. Deep learning-based saliency models typically follow an encoder-decoder architecture. The encoder network is assigned to extract hierarchical features from the input image. It frequently employs pre-trained CNNs as feature extractors, such as VGG [46] or ResNet [47]. The decoder network upsamples the encoded features to the original image resolution, producing a dense saliency map. This process entails deconvolutional layers or upsampling techniques, and can incorporate skip connections to preserve spatial information.

The development of sophisticated architectures, innovative techniques, and training strategies has enabled recent advances in saliency prediction. Notable models include:

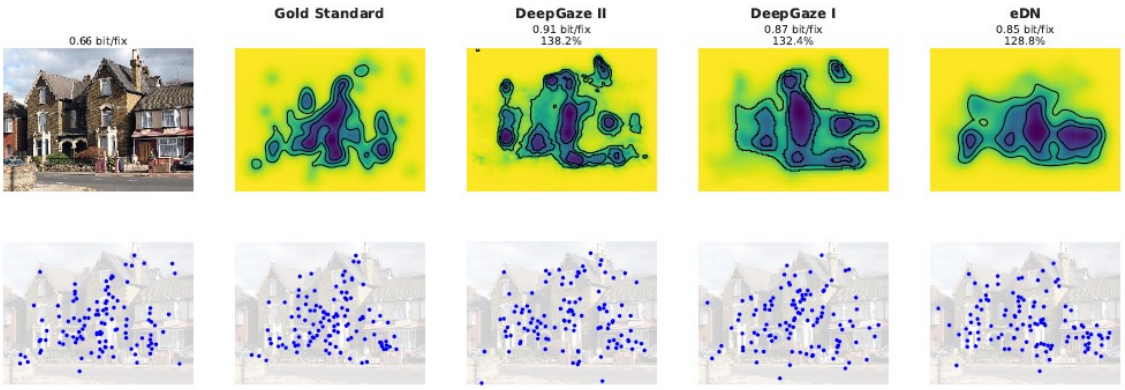


Figure 2.21: An example of the saliency probability density maps (top) and the computed fixations (bottom) of four different models: the Gold Standard, DeepGaze II, DeepGaze I, and eDN [48].

- Deep Gaze II: This model employs a pre-trained VGG network to extract features and applies a probabilistic model to predict saliency. It incorporates center bias and higher-level object features to enhance prediction accuracy. [48].
- SalGAN: The proposed adversarial network for saliency prediction comprises a generator network that predicts saliency maps and a discriminator network that distinguishes between predicted and ground truth maps. Adversarial training of this network is designed to enhance the model's ability to produce more realistic saliency maps [49].
- SAM (Saliency Attentive Model): This model incorporates attention mechanisms with deep learning, thereby enhancing the ability to focus on salient regions within an image. [50].

The applications of saliency prediction are numerous and diverse. In the field of image compression, the identification and preservation of visually important regions enables the implementation of saliency-based compression algorithms, which can reduce file sizes without a significant loss of perceptual quality. Furthermore, saliency maps can direct object detection algorithms to concentrate on the most pertinent components of an image, thereby enhancing the accuracy and efficiency of detection. In addition, saliency prediction can assist models in focusing on image regions relevant to the question in visual question answering, thereby improving the performance of the answering process. It can also assist in navigation and interaction tasks in robotics and autonomous systems.

3 Methods

3.1 Dataset

The methodology proposed in this thesis makes use of the ImageNet-1k dataset to a limited extent and primarily employs the Imagenette dataset. The Imagenette dataset is a portion of the larger and more comprehensive ImageNet-1k dataset, which has been curated with the specific intention of facilitating the development and benchmarking of image classification algorithms. Created by Jeremy Howard of Fast.ai [51], Imagenette is designed to provide a simpler, more manageable dataset for researchers and practitioners to experiment with, particularly in educational contexts or for swift prototyping of machine learning models.

Imagenette is comprised of ten distinct classes, a fragment of the ImageNet dataset. For example, ImageNet-1k contains one thousand distinct classes. These classes were selected to ensure straightforward visual differences, thereby relaxing the complexity associated with fine-grained classification. The ten classes included in Imagenette are as follows:

1. Tench (a type of fish)
2. English Springer (a breed of dog)
3. Cassette Player
4. Chain Saw
5. Church
6. French Horn
7. Garbage Truck
8. Gas Pump
9. Golf Ball
10. Parachute

Each of the two datasets is divided into two splits: training and validation. In Imagenet-1k, the training set contains a total of 1,281,167 images while the validation set has 50,000. In Imagenette, the corresponding figures are 9,469 and

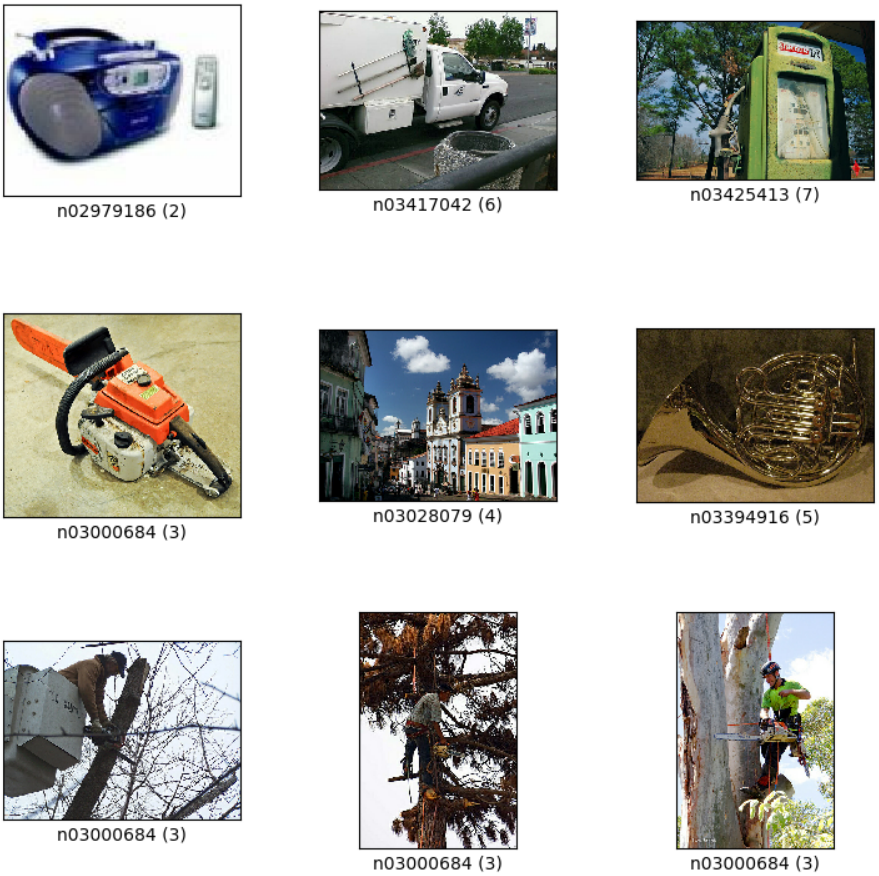


Figure 3.1: A few sample images from the Imagenette dataset [51].

3,925. In both datasets, the number of images per class is approximately equal in both splits.

The principal advantage of employing Imagenette is that it is considerably smaller than ImageNet, thereby facilitating more expeditious training and experimentation. This is particularly advantageous in view of the constraints of this thesis, both in terms of limited computational resources and a rigid timeline.

3.2 Simulated Retinal Implant

In the context of this thesis, a hypothesis is put forth regarding the potential use of a hardware configuration analogous to the Argus II prosthesis as an implant

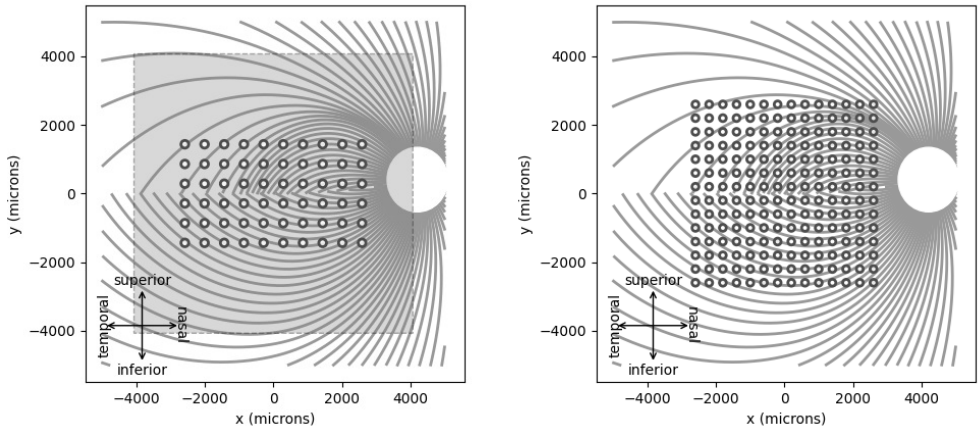
structure for two patients, Subject A and B, each with a distinct set of parameters respectively:

$$(\rho_A, \lambda_A) = (150\,\mu\text{m}, 100\,\mu\text{m}),$$
$$(\rho_B, \lambda_B) = (437\,\mu\text{m}, 1420\,\mu\text{m}).$$

The first set of values represents a hypothetical scenario, while the second reflects the actual set of values observed in an actual patient obtained from [31], which is $(437 \pm 6\,\mu\text{m},\ 1420 \pm 42\,\mu\text{m})$ to be precise.

This simulated hardware, designated Argus II*, is also designed for the epiretinal region and comprises a grid of disk electrodes, as illustrated in Figure 3.2(b). In contrast to Argus II, the number of electrodes is increased from 6×10 to 14×14. This modification is solely intended to harmonize the image resolution compatibility at the interface between the pulse2percept library and the pre-trained ViT model of DINOv2, which has a patch size of 14×14.

Accordingly, the diameter of each electrode is 200 µm, and the center-to-center spacing is 400 µm, differing from the previous specifications of 225 µm and 575 µm. This modification was made in order to accommodate the growing number of electrodes while maintaining the constant overall size of the implant.



(a) Schematic of Argus II

(b) Schematic of Argus II*

Figure 3.2: The Argus II* (right) has a greater number of electrodes, but each of them is slightly smaller and their inter-distance is reduced, resulting in an overall longer dimension that is similar to that of the Argus II (left).

Additionally, a minor alteration has been made to the Axon Map model, which is employed as a computational model. The *xystep* parameter, which defines the step size for the range of (x, y) values used to simulate the degrees of visual angle, has been modified from 0.25 to 1.

This alteration is for the purpose of simplifying the computations, which results in a reduction in memory footprint and execution time for the Axon Map model to

process batches of images when executed on a GPU node. For instance, if *xystep* is set to 0.25, the computational model on a small grayscale image (1 channel) with a resolution of 224 × 224 pixels would necessitate the allocation of more than 10 GiB of GPU RAM. A summary of the experiment on multiple *xystep* is presented in Table 3.1.

Input Shape	**Parameter** *xystep*	**GPU Memory (GiB)**	**Time (sec)**
(224, 224, 1)	*0.25*	*> 10 (out of memory)*	—
(224, 224, 1)	0.5	2.53	3.87
(224, 224, 1)	0.75	1.17	1.49
(224, 224, 1)	**1**	**0.67**	**1.00**

Table 3.1: The impact of *xystep* on GPU footprint and execution time.

3.3 Novel Approach For Stimulus Optimization

Studies such as [36, 37] consider the optimization of implant stimuli as the identification of the optimal inverse function f^{-1} of the process f in which the phosphenes are elicited based on the electrical stimuli. The objective is to achieve perceptual outcomes that are comparable to the desired perception by employing this optimization, or encoder. This traditional approach is summarized in Figure 2.17.

Given the minimal impact on the perceptual quality of those studies, particularly for patients with elevated values of ρ and λ, this thesis proposes an alternative approach to address the issue. Rather than attempting to achieve a final, perfect perception, the optimization f^{-1} could prioritize the identification of discriminating details of various daily life objects. This would then be encoded in a manner that generates a stimulus pattern allowing the patient's brain to recognize that particular object.

In other words, the problem can be reformulated as a classical classification task. In order to simulate and quantify the patient experience of this reformulation, two core components are suggested to compose the critical block that represents this simulation, as follows:

- A frozen ViT model, pre-trained with the DINOv2 method (hereafter designated as the DINOv2-ViT-S/14 model), is employed to represent the learned visual features. It can be postulated that this feature map reflects the visual

knowledge acquired by humans. Given that the knowledge base in nature is immutable, it can be assumed that the representative ViT model is also fixed.

- A linear probe of classifiers, appended to the output of the aforementioned ViT model, is used to constitute the adaptation of human brains when they are exposed to new things in the knowledge base. The probing of learned visual features is analogous to the manner in which a toddler learns to distinguish between various objects. For instance, a toddler may learn to differentiate between dogs and cats, tables and chairs, or houses and streets without explicit instruction.

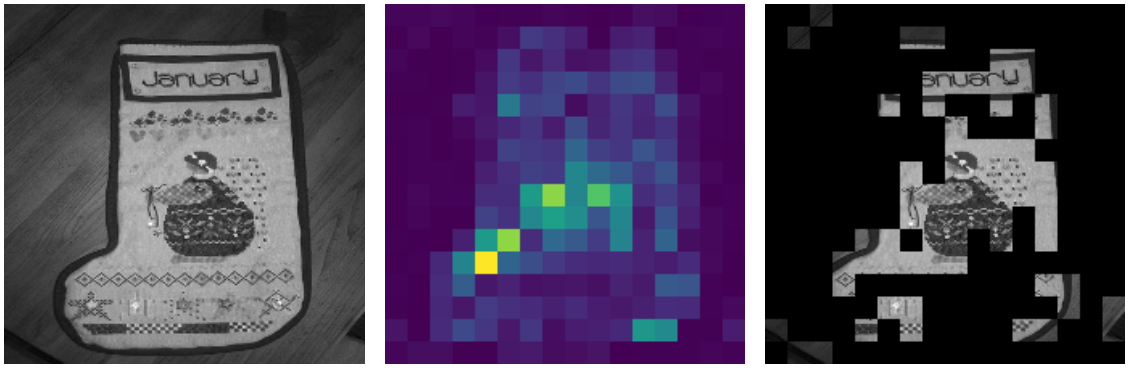
3.3.1 Modified DINOv2-ViT-S/14 Model For Fixation Classification

The anatomy and neurophysiological data of the retina have been instrumental in elucidating the fixation mechanism of humans. The current state-of-the-art technology in the field of computer vision enables the prediction of saliency probability, which mimics the tendency of psychophysical fixation. The advent of the DINOv2 technique, which trains deep learning models on extensive datasets, has demonstrated that the self-attention layer of the ViT is capable of learning transferable visual features that can be leveraged for a spectrum of downstream applications.

It is advantageous to capitalize upon the pre-trained DINOv2-ViT-S/14 model to achieve optimal results on the fixation prediction task. The self-attention map of the [CLS] token, located at the head of the final layer, can be used to derive the attention score for various regions within an input image. The areas with high attention score values also exhibit high probabilities of fixation. Therefore, the fixation prediction can be achieved without the use of additional complex deep learning networks and methods.

In order to identify the most meaningful part of the image, a threshold can be set to separate the attention score values amongst the regions by utilizing the statistical distribution. For instance, a specific value at the top one percent of the attention score for the given image can be used as a threshold. Regions exhibiting attention scores below the specified threshold will be concealed through the application of a filter operation. This process can be conceptualized as a mathematical operation, whereby those specific image regions are multiplied by zero.

Figure 3.3 demonstrates the methodology employed by the DINOv2-ViT-S/14 model to identify fixations within an image. The pre-trained version ViT-S/14 is employed in all experiments conducted within the scope of this thesis.



(a) Input: grayscale image

(b) Calculating attention score of each patch

(c) Masking patches based on the cutoff threshold

Figure 3.3: The process of obtaining predicted fixations in an image from left to right. The final masked image contains only the salient regions (patches) needed for classification.

In order to facilitate the implementation of linear probing, the [CLS] tokens present on the heads of a select few final layers are assembled. The number of final layers varies from one to four, contingent on the specific experimental design. In addition to the [CLS] tokens, the remaining head tokens are also subjected to experimentation. To provide further clarification, these tokens are concatenated and then flattened to a one-dimensional array of features. The array is shortest when the head's [CLS] tokens from the final layer are collected and longest when all the head tokens from the four final layers are assembled. The aforementioned newly introduced features are then input to a fully connected (FC) layer, which serves as a linear classifier. The number of inputs to the fully connected (FC) layer varies according to the length of the feature array, while the number of outputs is equal to the number of classes.

During linear probing, a variety of linear classifiers are tested and the classifier with the highest accuracy is selected. This classifier could be interpreted as a representative layer for the adaptation of humans to new visual experiences.

3.3.2 Modified U-net Model As An Optimization Encoder

In contrast to the original U-net architecture, which is employed for segmentation tasks, the modified U-net encoder functions as an image transformation that accentuates the dominant characteristics within the image. Although this augmentation does not increase the image definition or quality, it may improve the multitude of visual elements present in the image, which could potentially influence the phosphenes generated by the prosthesis for visually impaired individuals. The process may be referred to as image enhancement for phosphenes.

In order to accommodate the new task and to establish a compatible interface

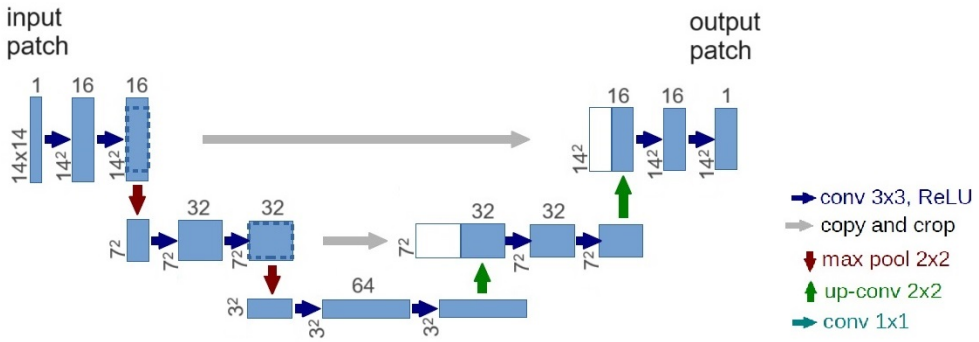


Figure 3.4: The modified U-net architecture. The input shape and output shape are identical. In contrast to the four Down blocks and four Up blocks typically found in each path, there are only two Down blocks and two Up blocks in this modified version. Consequently, the resulting bottleneck has a feature shape of (64, 3, 3).

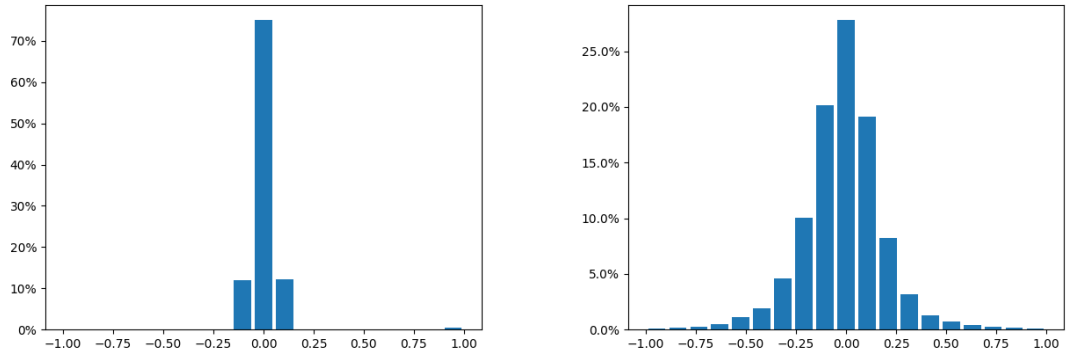
between the U-net architecture and the computational model of pulse2percept, a series of modifications have been made to the U-net, as depicted in Figure 3.4.

Firstly, the output channel is set to one, as no immediate classification information is required. In this case, segmentation is superfluous, and the new U-net output should be an enhanced version of the input grayscale image. Therefore, the output shape is maintained in accordance with the input shape, which signifies that the spatial dimensions are no longer reduced and the channel dimension is not increased as was observed in the original U-net study.

Secondly, the number of down-convolutional and up-convolutional layers is reduced in order to facilitate the processing of the down-sampled image or partial patches of the image. The processing size of 14 x 14 pixels is selected to ensure consistency with the DINOv2-ViT-S/14 model. With this much smaller size, there are not sufficient details for the long sequences of down-convolutional and up-convolutional layers to operate on. It is reasonable to reduce the length of the contracting and expansive paths in order to prevent the bottleneck from becoming overly strained. In addition, the reduction in size also results in a reduction in the training time required for the U-net encoder.

Finally, the U-net is briefly pre-trained as an identity image transformation (IIT). This means that the input and output images are very closely similar to each other. This training is regularized by the pixel-wise MSE loss function, which minimizes the difference between each pixel of the images. Subsequently, the identity transformation U-net will be employed as the initial value for the encoder, which can be referred to as the identity initialization scheme. This step is crucial for stimuli optimization, as it enables the encoder to pre-learn the visual features

that are subsequently required for the enhancement of phosphenes.



(a) During initialization **(b)** After IIT training

Figure 3.5: The distribution of weights and biases of the modified U-net during initialization vs. after IIT training.

To illustrate the final modification in greater detail, Figure 3.5 can be introduced to visualize the distributions of weights and biases of two U-net models. The former is initialized in a traditional manner, while the latter is trained on an identity transformation task. As depicted, the former's weights and biases are concentrated at 0, which makes it challenging for the model to enhance an image in the desired direction. Conversely, the more uniform distribution of the latter's weights and biases suggests the existence of a learned visual feature map, facilitating an easier transition of the image enhancement process towards the optimized direction.

3.3.3 Proposed Experiments

The experiments are divided into two categories: simulation and optimization. The objective of the simulation experiments is to provide a foundation for the latter type of experiments. In summary, the pipeline of each experiment can be described as follows:

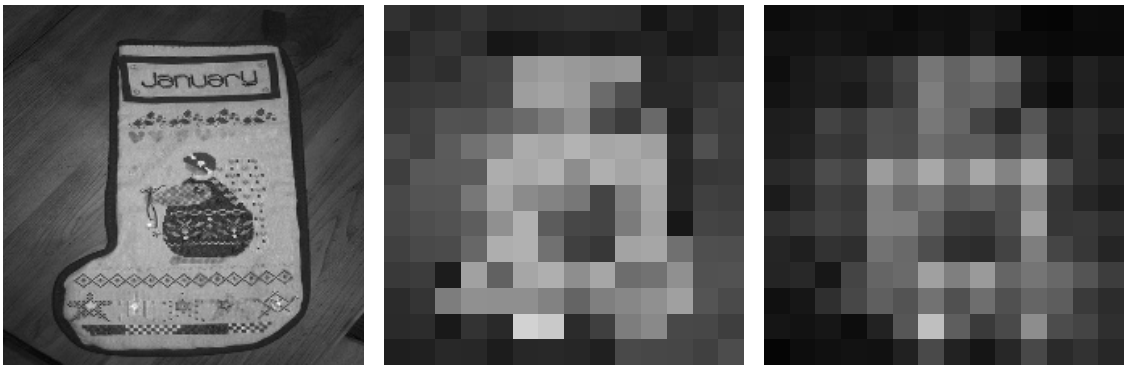
3.3.3.1 Simulation Experiments

1. **Baseline 0**: A subject with no history of ocular disease or impairment.

   It is proposed that a combined block comprising two components be utilized: the ViT model and the linear classifiers. Thereupon, the aforementioned block is fed with the dataset, after which the classification output is evaluated. This multi-class accuracy metric represents the upper baseline value against which the stimulus optimization is aimed at reducing the gap.

2. **Baseline 1**: Two subjects with specific retinal conditions.

   For the lower baseline values, two patients previously mentioned, along with the Argus II* implant, are simulated using a straightforward scoreboard approach. This process entails the reduction of image resolution (the down-sampling process) and subsequent transmission to the pulse2percept library, which serves as the hardware simulation. This library is responsible for converting the non-optimized electrical stimuli into predicted percepts.



(a) Full-size grayscale input **(b)** After down-sampling **(c)** Phosphene output

Figure 3.6: A phosphene generated by the down-sampling process.

Figure 3.6 exemplifies a percept elicited by the Argus II* using the Axon Map of pulse2percept for a subject with parameters $\rho = 150\,\mu m$ and $\lambda = 100\,\mu m$. Thereafter, these percepts will be subjected to the block of ViT and classifiers, which represent the experience and recognition of the subjects undergoing the test. A robust optimization strategy should exceed these baseline values.

3. **Baseline 2**: An alternative version of **Baseline 1**, incorporating fixation simulation.

   The succeeding phase is to conduct experiments with the concept of a saliency probability density map. In contrast to the conventional approach of indiscriminately down-sampling an image, which can result in the loss of recognizable content when the down-sampling size is insufficient, a more selective approach could be employed. This involves focusing on the distinctive meaningful positions of the image while discarding the rest. This strategy could be particularly beneficial in the context of retinal implants, such as Argus II and Argus II*, where the phosphene size is so miniature.

   This is based on the knowledge of human eye movement patterns, or fixation, which indicates that healthy individuals tend to focus on the most prominent regions of a visual field. It is reasonable to posit that for patients with retinal conditions, it would be beneficial to implement experiments where the image is divided into smaller patches (patchification) instead of naive down-sampling. These experiments serve as a second lower baseline.

The aforementioned modified DINOv2-ViT-S/14 model for fixation classification is utilized in this approach. First, all images are resized to a uniform shape of 224 x 224 pixels and divided into smaller regions, or patches, of the size of 14 x 14 pixels. Thus, each image contains a total of 256 patches. These patches are then processed by the Axon Map model individually to elicit predicted phosphenes.

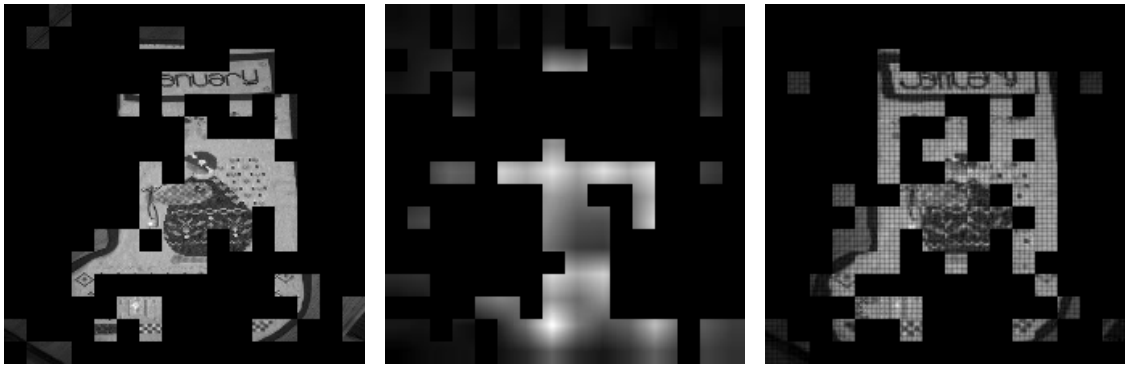
(a) Full-size grayscale input

(b) Phosphene output from divided patches of an image

(c) Fixation simulation on this phosphene

Figure 3.7: An example of fixation prediction on a generated phosphene.

Subsequently, the phosphenes undergo processing through the DINOv2-ViT-S/14 model. Only those patches that exceed the predefined fixation threshold value are retained, while the remaining patches are masked by the DINOv2-ViT-S/14 model, which represents the fixation knowledge base. This is followed by the application of learnable linear classifiers, which represent the patient's adaptability to recognize the object in the image. An illustrative example is provided in Figure 3.7, which depicts the fixation simulation on the elicited phosphene.



(a) Grayscale image

(b) Phosphene from down-sampled image

(c) Phosphene from patchified image

Figure 3.8: A visual comparison of the fixation patterns observed in different cases.

3.3.3.2 Optimization Experiments

1. **Optimization 1**: A stimuli optimization strategy for **Baseline 1**.

   A learnable U-net encoder is pre-appended to pipeline **Baseline 1**. This encoder serves as the stimuli optimizer, which assists the retinal implant in generating more suitable electrical sensations. This, in turn, allows the neurons to create better phosphenes in which the patients can recognize and distinguish different objects.

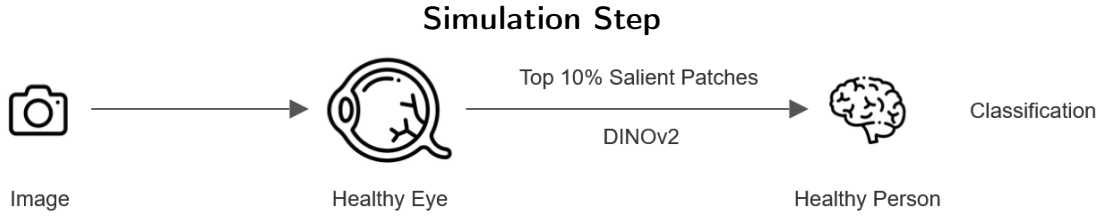
   This stimuli optimization strategy differs from the current proposed strategies, as outlined in [36, 37], in its utilization of the modified U-net architecture, as described in the previous section, to identify the inverse function, f^{-1}, which is used to elicit the desired phosphenes as closely as possible to the actual image.

2. **Optimization 2**: A stimuli optimization strategy for **Baseline 2**.

   A learnable U-net encoder is implemented similarly to the one employed in **Optimization 1**, with the exception that the pipeline **Baseline 2** is utilized. The U-net encoder will operate on the small patches of the image rather than the down-sampled image as the encoder in **Optimization 1**.

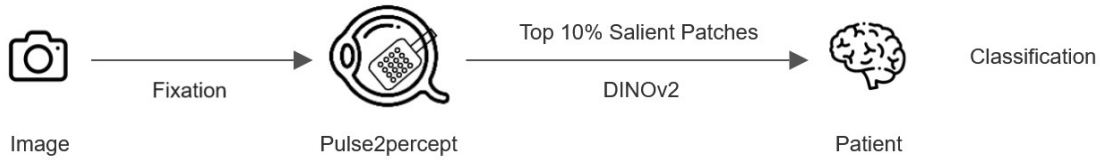
   The objective of this experiment is to demonstrate that the proposed strategy for stimuli optimization outperforms the conventional practice of image size reduction for the input of retinal implants. Furthermore, it offers a novel perspective on exploring the inverse function f^{-1}. Rather than attempting to generate closely realistic phosphenes, it would be more beneficial in terms of task-specific and item-oriented approaches to elicit phosphenes in a manner that facilitates object recognition, which is useful in real-life scenarios such as navigation or searching for keys, cards, and phones, among others.

In summary, the entire approach is encapsulated in Figure 3.9.

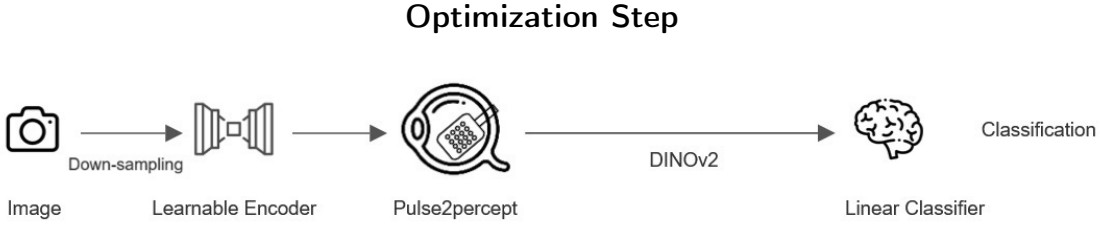


(a) Baseline 0: A subject with no history of ocular disease or impairment.

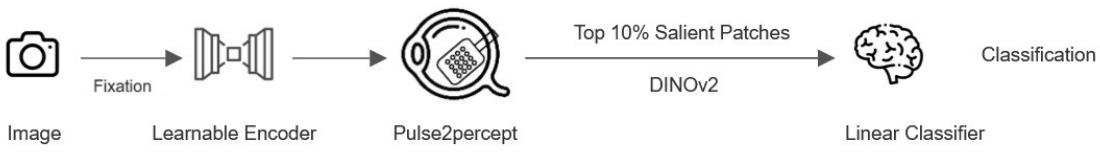
(b) Baseline 1: Two subjects with specific retinal conditions.



(c) Baseline 2: Two subjects with specific retinal conditions in conjunction with fixation simulation.



(d) Optimization 1: Optimization strategy for **Baseline 1**.



(e) Optimization 2: Optimization strategy for **Baseline 2**.

Figure 3.9: The proposed methodology.

4 Results And Discussion

4.1 Preliminary Experiments

4.1.1 A Comparative Analysis Of The Difficulty Between The Imagenet-1k And The Imagenette With Regard To The Fixation Classification Task

In this thesis, a number of sophisticated pipelines for both the simulation and optimization processes are proposed and tested. Due to the limitations of the computational and temporal resources, it is not feasible to work on a full Imagenet-1k dataset comprising more than 1.2 million training images for all the pipelines. Hence, the Imagenette sub-dataset, which comprises only 10 classes as described in Section 3.2, emerges as a suitable replacement.

The study by Oquab et al. [26] revealed that the ViT-S/14 model without registers achieved an impressive result of 81.1% on the Imagenet-1k dataset for linear probing classification, despite its relatively modest size of 21 million parameters. It can be reasonably assumed that the model would achieve a higher accuracy for the Imagenette subset. Therefore, to quantify the difficulty between these two datasets, the performance of the model can be used as a metric to infer their contingent relation. As evidenced by the results in the third row of Table 4.1 for a regular classification task, the presumption is indeed confirmed when DINOv2-ViT-S/14 reaches 99.0% accuracy for Imagenette, which is a huge increase over the results obtained on Imagenet-1k.

Furthermore, in anticipation of future pipelines that will require testing, it is necessary to conduct another experiment with a relevant task in order to ascertain the appropriate difficulty level estimation for the two datasets. The task is to classify the top 10% of fixations predicted by the modified DINOv2-ViT-S/14 model. The threshold value of 10% is selected based on the empirical observations obtained in the next section.

The images from both Imagenet-1k and Imagenette are colored samples. However, retinal implants operate on grayscale images. Therefore, it is necessary to convert these two datasets into grayscale in order to ensure compatibility with relevant future pipelines. The second and final rows of Table A illustrate the contingent relationship between Imagenet-1k and Imagenette in the context of the grayscale, top 10% fixation classification task.

Dataset	Classification	Accuracy %
Imagenet-1k	colored, 100% image	81.1
	grayscale, top 10% fixation	52.6 (after 10 epochs)
Imagenette	colored, 100% image	99.0
	grayscale, top 10% fixation	89.0 (after 10 epochs)

Table 4.1: A comparison of the difficulty of Imagenet-1k and Imagenette for two classification tasks. The first is conducted in a regular setup, while the second is a grayscale, top 10% fixation classification.

Although Imagenet-1k is considerably larger and much more challenging than Imagenette in terms of the fixation classification task, with a leading 36.4% difference, the sub-dataset still provides sufficiently complex and diverse data to avoid over-fitting. The notable decline in the multi-class accuracy, from 99.0% to 89.0% in the last two rows of Table 4.1 observed on the Imagenette validation set, further substantiates this argument.

4.1.2 Threshold Findings For Fixation Simulation Of The Modified DINOv2-ViT-S/14

In order to simulate human fixation, a series of experiments were conducted in order to determine an appropriate threshold value for the modified DINOv2-ViT-S/14's attention score, or fixation threshold. The Imagenet-1k was utilized in these experiments due to its extensive representation of visual knowledge. In conjunction with the probing of linear classifiers, as detailed in Section 3.3.1, the experiments yielded detailed results, which are presented in Table 4.2.

The first two rows of Table 4.2 demonstrate that the classification accuracy decreases by only 13.5% despite the removal of 50% of the image areas. This suggests that, in classification tasks involving fixations, the pertinent information tends to concentrate in a few salient regions, rather than evenly spreading across the image. The third row shows that a further loss of 5.3% in accuracy is observed when grayscale data is considered in place of full-color data.

Furthermore, a reduction in the threshold of the top percentage of fixation from 50% to 10% resulted in a further 9.7% decline in accuracy, from 62.3% to 52.6%. This is illustrated in the final row. This implies that for 1,000 sampled classes of the Imagenet-1k, the grayscale top 10% of its image fixation contributes more valuable details than the rest. Moreover, the pre-trained DINOv2-ViT-S/14

Model & Dataset	Image Characteristics	Accuracy (%)
DINOv2-ViT-S/14 (without registers) + linear probing Imagenet-1k	colored, 100% image	81.1
	colored, top 50% fixation	67.6 (after 10 epochs)
	grayscale, top 50% fixation	62.3 (after 10 epochs)
	grayscale, top 20% fixation	55.6 (after 10 epochs)
	grayscale, top 10% fixation	52.6 (after 10 epochs)

Table 4.2: The influence of the top percentage of fixations predicted by DINOv2-ViT-S/14 on linear classification performance.

has been sufficiently trained to learn this type of knowledge and to be able to represent the neurophysiology of human fixation. Consequently, the threshold of 10% may be employed as the cutoff range of the attention score of the modified DINOv2-ViT-S/14 model.

4.1.3 Axon Map Phosphenes And Regular Images In Fixation Prediction: A Comparative Analysis

The objective of this section is to assess the viability of utilizing phosphenes evoked by the computational model Axon Map of the pulse2percept library in conjunction with the fixation prediction of the DINOv2-ViT-S/14 model.

Quantity Evaluation This work employs the probing process of linear classifiers to quantitatively compare the fixation prediction from phosphenes with the prediction from regular images. One of the objective evaluation tools employed in the assessment of hidden features learned by deep learning models is linear probing. It is therefore reasonable to apply linear probing here to evaluate the ability of DINOv2-ViT-S/16 to predict fixation from phosphenes. Figure 4.1 depicts the outcomes across varying fixation thresholds for both conditions: phosphenes and regular images.

As illustrated in the accompanying figure, a reduction in the fixation threshold is associated with a decline in classification accuracy. It is also evident that the prediction of fixations on phosphenes is more challenging than on regular images in all values of the threshold. The observed average drop of 8.61% indicates that while the retinal implant's inherent problem in producing realistic phosphenes is

significant, it may not have the same severe degree of effect in the context of fixation prediction. A well-designed fixation prediction algorithm can offset the shortcomings of inadequate phosphenes and may even enhance the performance of optimal phosphenes in object recognition tasks.

Quality Assessment Figure 4.2 illustrates a variety of phosphene generation from disparate configurations. The top row, from left to right, depicts a regular grayscale image, a simulated phosphene from Subject A with $\rho = 150\,\mu m$ and $\lambda = 100\,\mu m$, and a simulated phosphene from Subject B with $\rho = 437\,\mu m$ and $\lambda = 1420\,\mu m$. The bottom row illustrates the fixation prediction of each image and phosphene.

The visualization indicates that while the quality of the phosphene may decline, its fixation may not necessarily deteriorate at the same rate. This suggests that the fixation approach may be a viable option for patients with appalling retinal conditions, potentially enabling them to navigate daily life with greater ease.

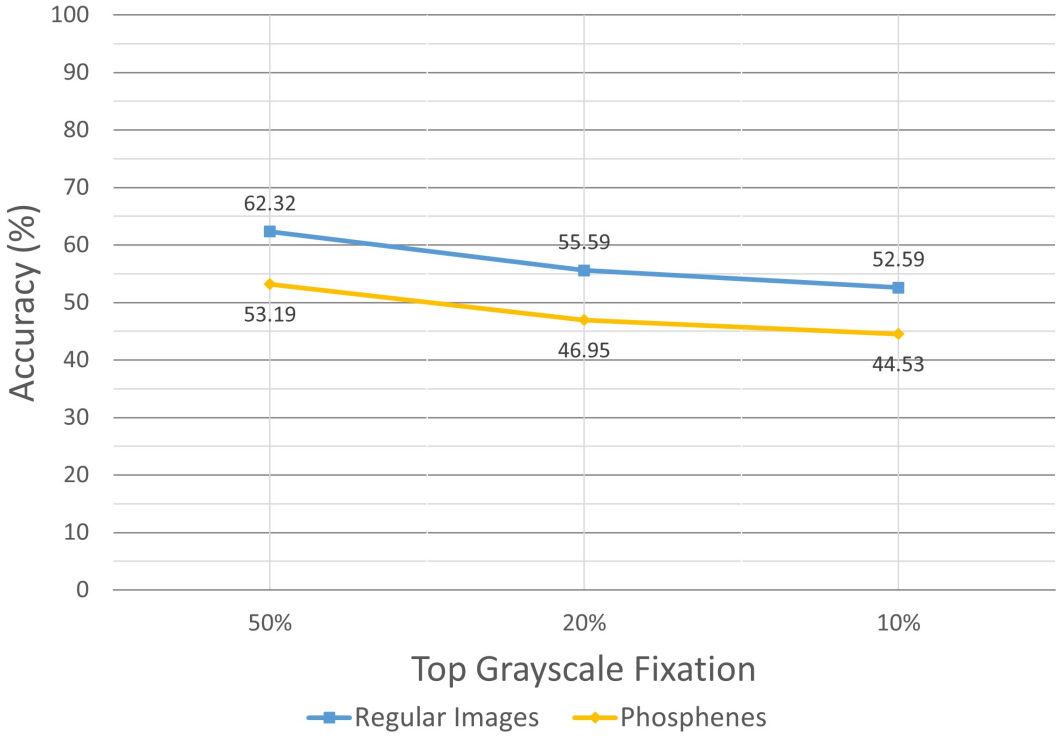


Figure 4.1: Fixation classification results at three thresholds (50%, 20%, and 10%) in two cases: regular images and phosphenes.

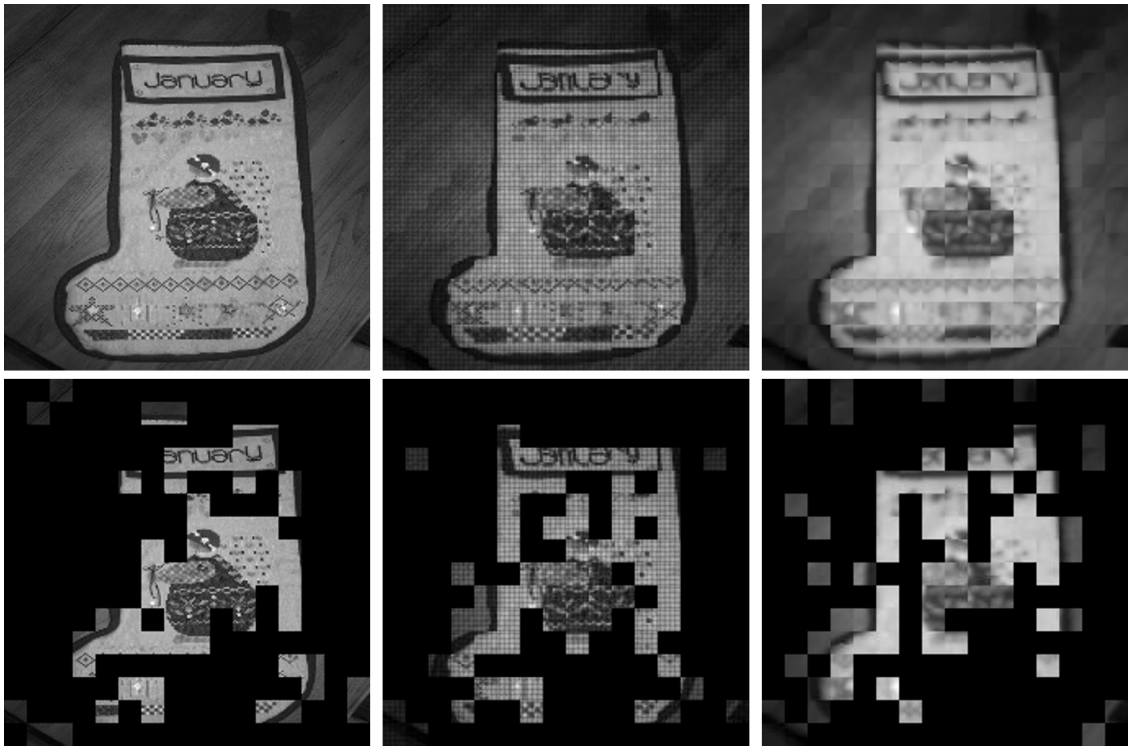


Figure 4.2: Fixation predictions in various settings. Top: a normal image (left), a phosphene of subject A (middle), and a phosphene of subject B (right). Bottom: fixation predictions of the corresponding inputs in the top row.

4.1.4 Identity Image Transformation Training Of The Modified U-net Model

In the optimization step, a U-net encoder is modified and pre-trained to generate an image identical to the input image. The pixel-wise MSE loss function is used to calculate the pixel-level discrepancies between the two images.

Quantity Evaluation A series of experiments was conducted across a range of learning rate values, with the values varying from 1×10^{-1} to 1×10^{-7}. The results obtained after 10 epochs are displayed in Table 4.3, in which it is observed that the 1×10^{-3} learning rate training session, represented by the third row, demonstrates the most encouraging outcomes.

In addition, Figure 4.4 depicts the loss curves in this training over the 20 epochs. It should be quantitatively satisfactory if the loss is below 1×10^{-7}. Therefore, from examining Figure 4.4, it is safe to use the model parameters from the 8th epoch onwards. For the sake of simplicity, the parameters in epoch 10 were selected to be the initialized values of the U-net encoder in the downstream task. The subsequent section serves to corroborate this selection by demonstrating the visual outcomes of the U-net encoder.

Learning Rate	Training Loss	Validation Loss
1×10^{-1}	2.48×10^{-6}	2.47×10^{-7}
1×10^{-2}	6.53×10^{-7}	1.50×10^{-7}
1×10^{-3}	8.66×10^{-8}	1.57×10^{-8}
1×10^{-4}	4.87×10^{-7}	1.14×10^{-7}
1×10^{-5}	6.59×10^{-7}	4.57×10^{-7}
1×10^{-6}	2.97×10^{-6}	2.48×10^{-6}
1×10^{-7}	5.15×10^{-3}	5.01×10^{-3}

Table 4.3: Training and validation loss w.r.t. learning rate at the 10th epoch.

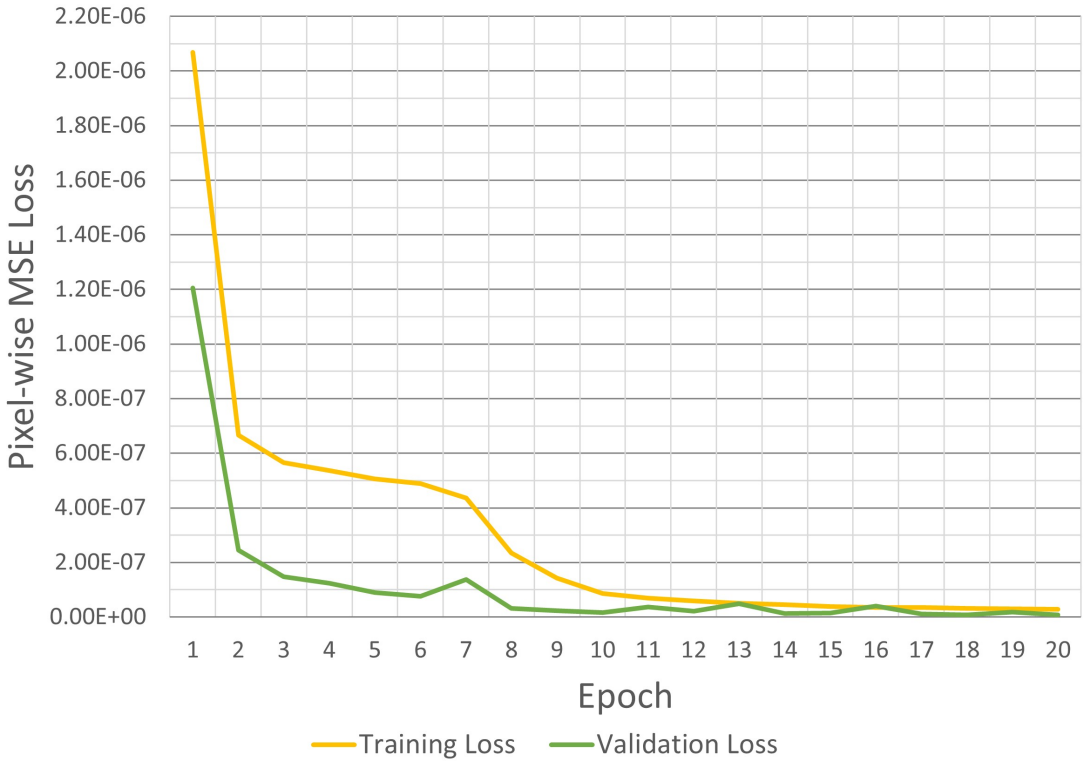


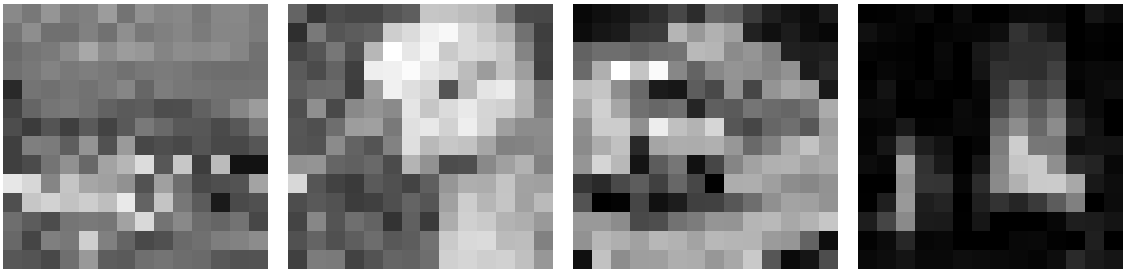
Table 4.4: Training and validation loss during an IIT training session of the U-net encoder at the learning rate of 1×10^{-3}.

Quality Assessment Due to the exceptionally low pixel-wise MSE loss value, the modified U-net is highly effective in replicating any given image. Visual representations of the aforementioned model's performance are presented in Figs 4.3 and 4.4, which demonstrate the model's ability to replicate images in various gray-scale categories and dimensions. Upon visual inspection by the human eye, it is challenging to discern any difference between the original and replicated images.

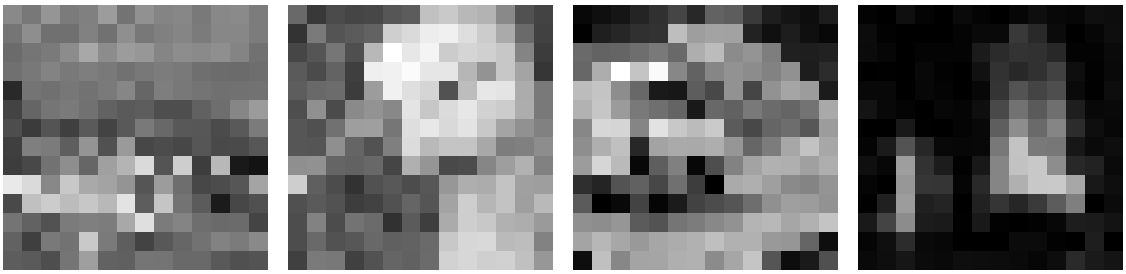
(a) Sample input 1 **(b)** Sample input 2 **(c)** Sample input 3 **(d)** Sample input 4

(e) Output 1 **(f)** Output 2 **(g)** Output 3 **(h)** Output 4

Figure 4.3: Visual example outputs of the modified U-net after IIT training. The dimensions of both the inputs and outputs are identical, measuring 224 pixels in width and height. Top: the grayscale images from the Imagenette dataset. Bottom: the images generated from the U-net.



(a) Sample input 1 **(b)** Sample input 2 **(c)** Sample input 3 **(d)** Sample input 4



(e) Output 1 **(f)** Output 2 **(g)** Output 3 **(h)** Output 4

Figure 4.4: The same examples in the same order as in Figure 4.3, but the inputs are down-sampled to 14×14 pixels. Top: the grayscale samples from the Imagenette dataset. Bottom: the images generated from the U-net.

4.2 Primary Experiments

4.2.1 Pipeline Training

4.2.1.1 The Upper Boundary

In the **Baseline 0** scenario, a healthy individual with no ocular disease or impairment is simulated by using linear probing on the fixation knowledge of the DINOv2-ViT-S/14 model. It is assumed that a normal human with full-color vision could perform well in the classification task on Imagenette in the same manner as the described pipeline.

Figure 4.5 shows the multi-class accuracy values on the Imagenette validation set of the pipeline **Baseline 0** over ten epochs. It demonstrates that the accuracy stabilizes at 92.76% at the 10th epoch, which can be considered as the upper baseline value for the thesis proposal.

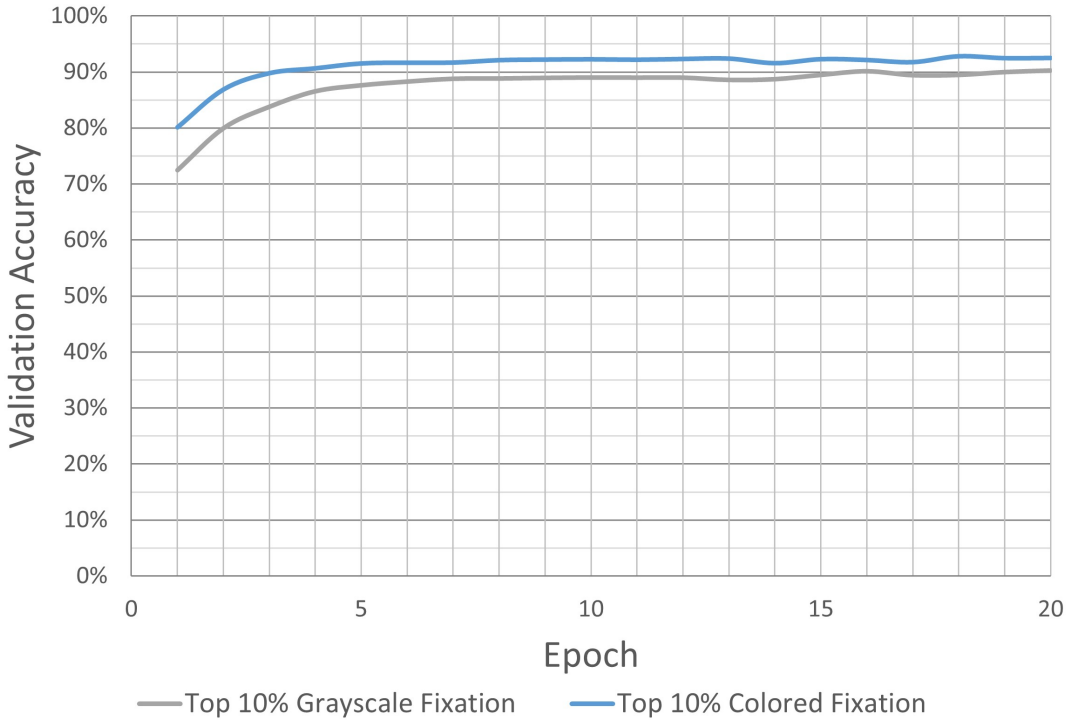


Figure 4.5: Validation Accuracy of the **Baseline 0**, which simulates a healthy individual with no ocular disease or impairment.

The pipelines **Baseline 1** and **Optimization 1** share the same characteristics, they both represent the current method where the input images are reduced in the spatial dimensions in order to generate the implanted percepts. Hence, these two pipelines can be grouped and referred to as the down-sampling pipelines. A similar grouping also happens for **Baseline 2** and **Optimization 2**, of which the group

is named the patchification pipelines.

In the first group, the input images are minimized using the `interpolation` function of PyTorch, while in the second group, the images are divided into smaller patches using the `patchify` function of the PyPatchify library.

In each pipeline of the two groups, two sets of experiments are conducted, each representing a retinal condition of a patient. One set, Subject A, of experiments employs the theoretical values of rho and lambda, which are 150 µm and 100 µm, respectively. The other set, Subject B, employs the empirical values of rho and lambda, which are 437 µm and 1420 µm, respectively, as reported in Beyeler et al. [31].

The following sections will present a comparison and discussion of the simulation and optimization results, organized according to the method groups. Figures 4.6 to 4.9 illustrate the training processes of all four pipelines. The yellow and green lines represent the performances of two Subjects A and B, respectively.

4.2.1.2 The Down-sampling Method

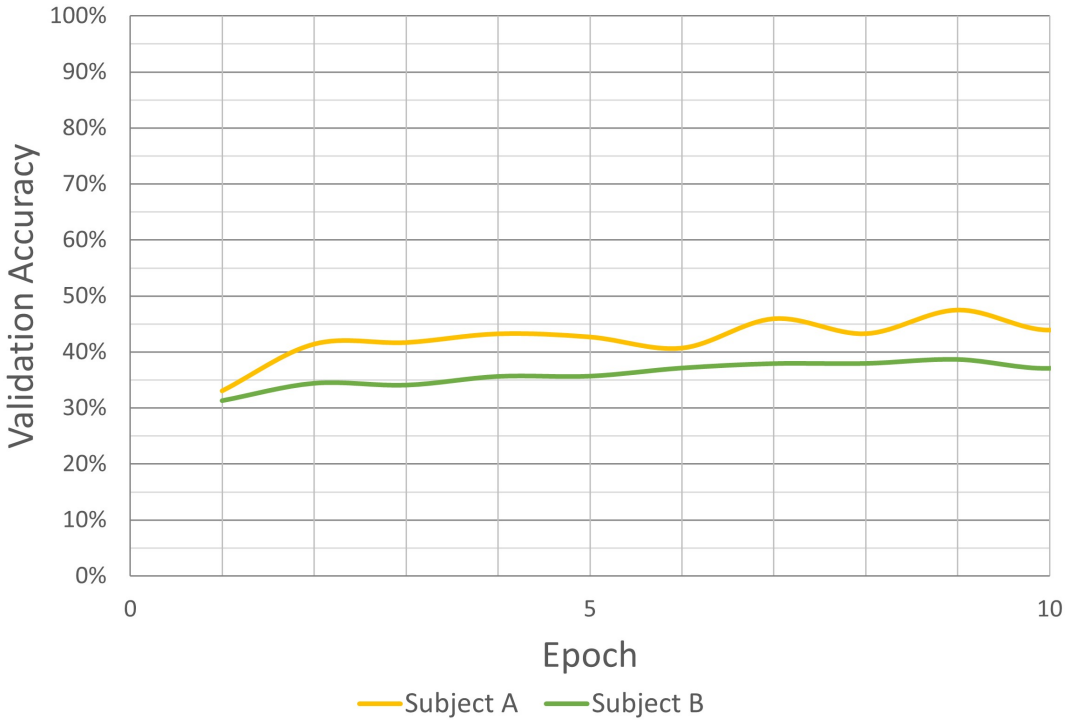


Figure 4.6: Validation accuracy of the **Baseline 1**, which simulates the down-sampling process.

As illustrated in Figure 4.6, **Baseline 1** exhibits suboptimal performance in the Imagenette benchmark, which elucidates the evident constraints of the existing retinal implant Argus II and to a certain extend, its variant Argus II*. The trial on Subject A, whose axonal parameters are very close to those of a healthy person, demonstrates that the peak accuracy is 47.46%, which is notably low in this relatively simple sub-dataset. Despite the incorporation of the U-net encoder within the pipeline, prior to the computational model Axon Map (decoder), the accuracy exhibited a marginal improvement of approximately 4.03%, reaching a value of 51.49%.

A similar situation is observed in Subject B within this group. In the simulation phase, the implant exhibited a low level of accuracy, below 40%. Following the introduction of an encoder after 10 training epochs, the metric demonstrated a slight improvement of 1.89%, rising from 38.7% to 40.59%. This is illustrated in Figure 4.7.

In comparison to the 80% accuracy mark, or 80.13% to be exact, which the same DINOv2-ViT-S/14 model can easily achieve in a regular set-up on the first epoch, the current approach, which attempts to enhance the implanted perception by emulating the realistic scenery, is a significant disappointment. This discrepancy highlights the limitations of this approach in the real-world environment.

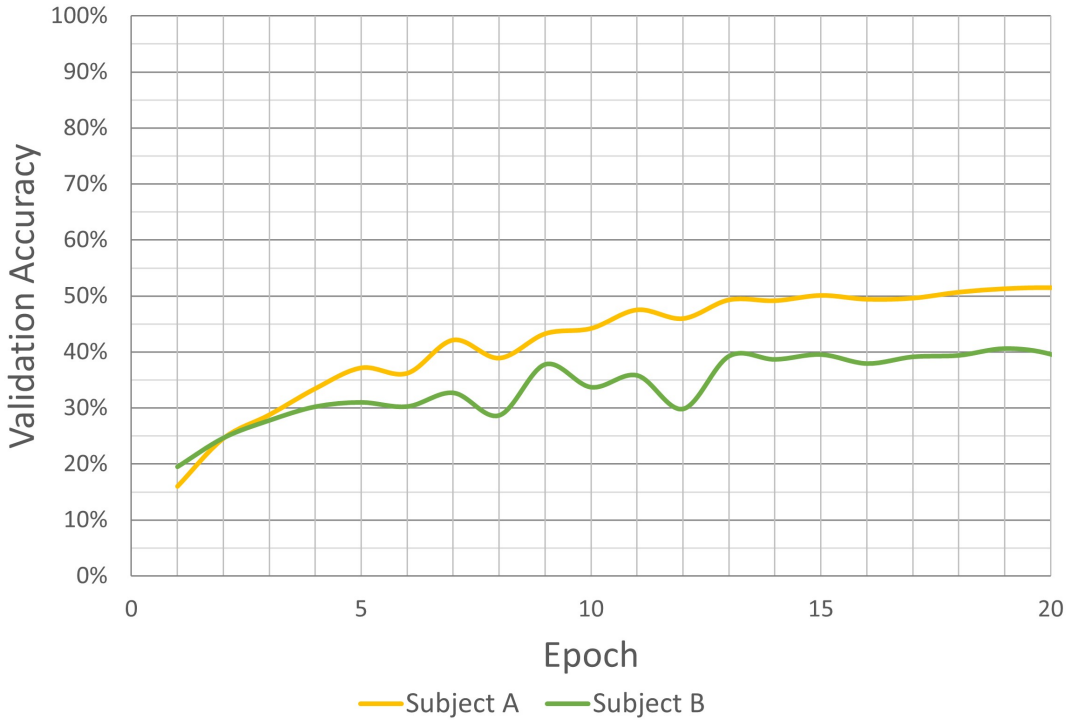


Figure 4.7: Validation accuracy of the **Optimization 1**, which optimizes the downsampling process with a U-net encoder.

4.2.1.3 The Patchification Method

By contrast, the patchification group demonstrates a more promising performance in addressing the sophisticated problem, at least in the Imagenette benchmark. The **Baseline 2** indicates that the theoretical Subject A can correctly identify images with an accuracy rate of 85.20% after a sufficient period of familiarisation. Subject B also performs well, with a peak accuracy of 81.99%, although with a somewhat longer time.

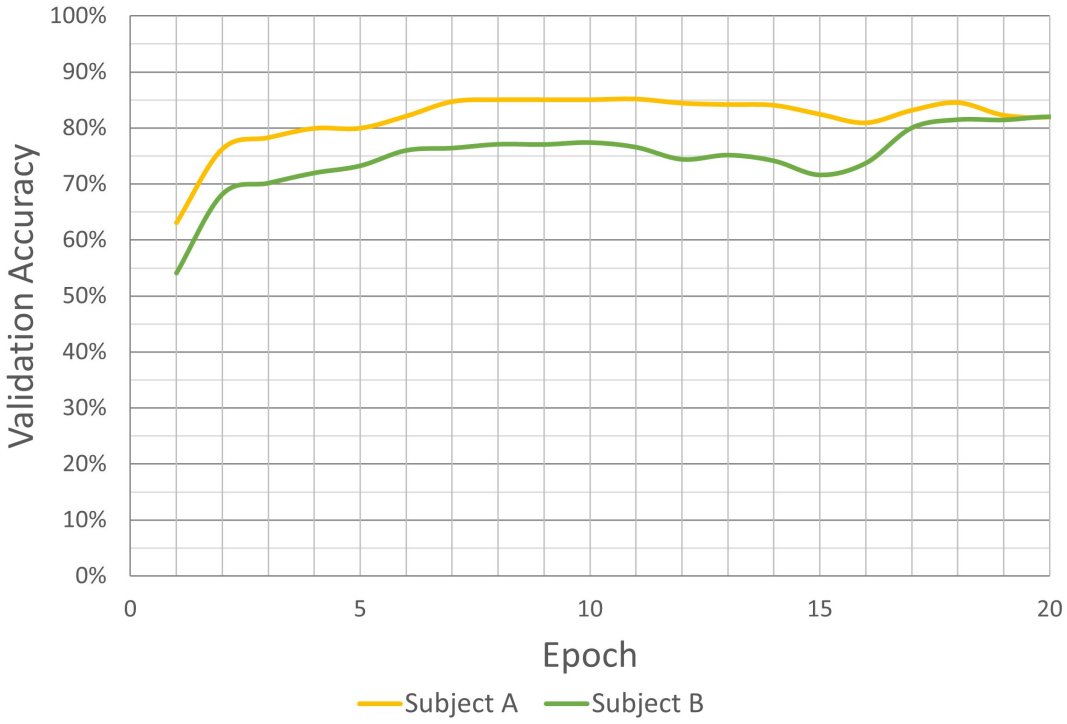


Figure 4.8: Validation accuracy of the **Baseline 2**, which simulates the patchification process.

Both metrics fall short of 7.56% and 10.77%, respectively, from the one of the simulated individual with perfect vision (accuracy of 92.76%). The two metrics of **Baseline 2** are nearly double those of **Baseline 1**, with an average difference of 40.52%. This is depicted in 4.8.

It is anticipated that the situation will improve with the assistance of the U-net model. Indeed, both Subjects A and B are benefiting from the auxiliary component, with the top performances settling at 90.85% and 87.72%, respectively, as shown in Figure 4.9. In this pipeline, the performance of Subject B suffers a significant deficit at the beginning, remaining below 50% for the first five epochs. Nevertheless, the training session subsequently accelerates, reaching approximately 80% in the five following epochs. This could be indicative of the fact that for individuals with large axonal parameters, an encoder may initially underperform. However, with

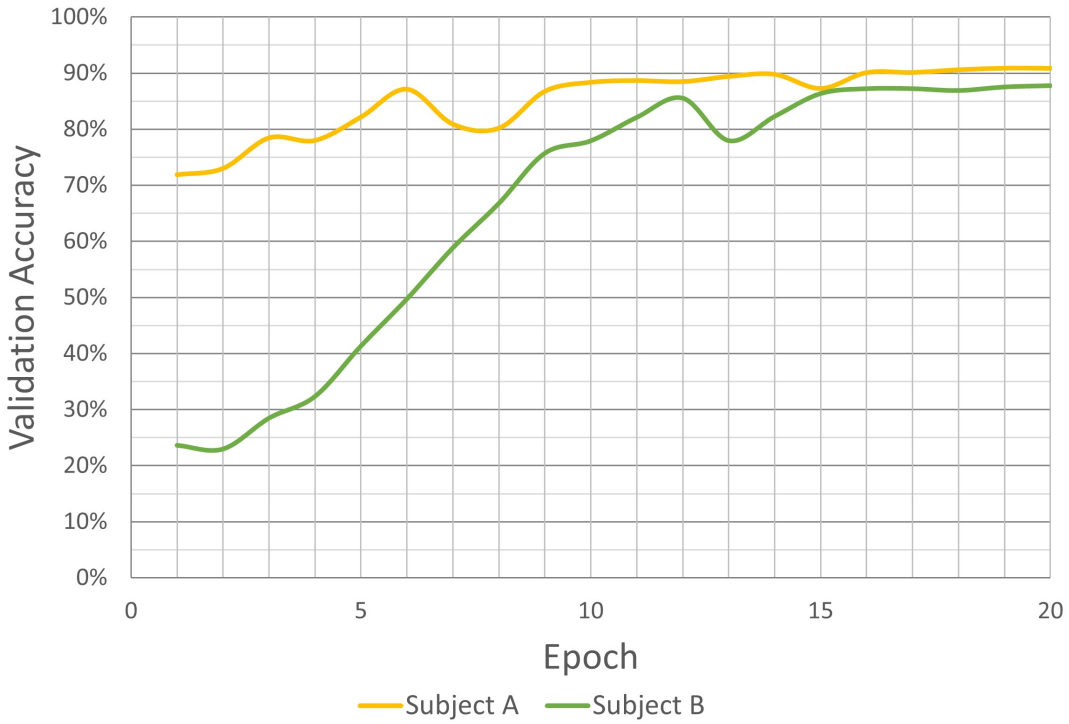


Figure 4.9: Validation accuracy of the **Optimization 2**, which optimizes the patchification process with a U-net encoder.

the application of patience and a period of adaptation, the encoder and the patient would ultimately achieve optimal outcomes.

Moreover, the enhancement in patchification resulting from **Optimization 2** in comparison to **Baseline 2** is 5.65% and 5.73%. These figures are all superior to the improvement in down-sampling resulting from **Optimization 1** relative to **Baseline 1**, which stands at 4.03% and 1.89%. The aforementioned improvements in patchification are particularly noteworthy when one considers the narrow gap between **Baseline 2** and Baseline 0, which averages only 9.17%. This is in stark contrast to the considerable distance between **Baseline 1** and Baseline 0, which averaged 49.68%.

This noteworthy observation serves to substantiate two key assertions.

- Firstly, it is evident that the proposed patchification approach outperforms the conventional method in terms of object classification.
- Secondly, the additional encoder, at least with the U-net architecture, is more effective when used in the novel method than it was in the previous one.

Further exploration of deep learning models may yield insights into the efficacy of different encoder algorithms across the two methods.

The outcomes of all the experimented pipelines are summarized in Table 4.5. Additionally, a few illustrative examples of the implanted perceptions of the two subjects in those trials are also presented for the visual assessment in Figures 4.10 - 4.11 (Subject A) and 4.12 - 4.13 (Subject B). In both figures, the first column depicts the input images, the second column illustrates the output phosphenes in the absence of the pre-appended encoder (baseline pipelines), the third column depicts the intermediate images generated by the U-net encoder, and the final column depicts the final output phosphenes produced by the encoder-decoder procedure (optimization pipelines).

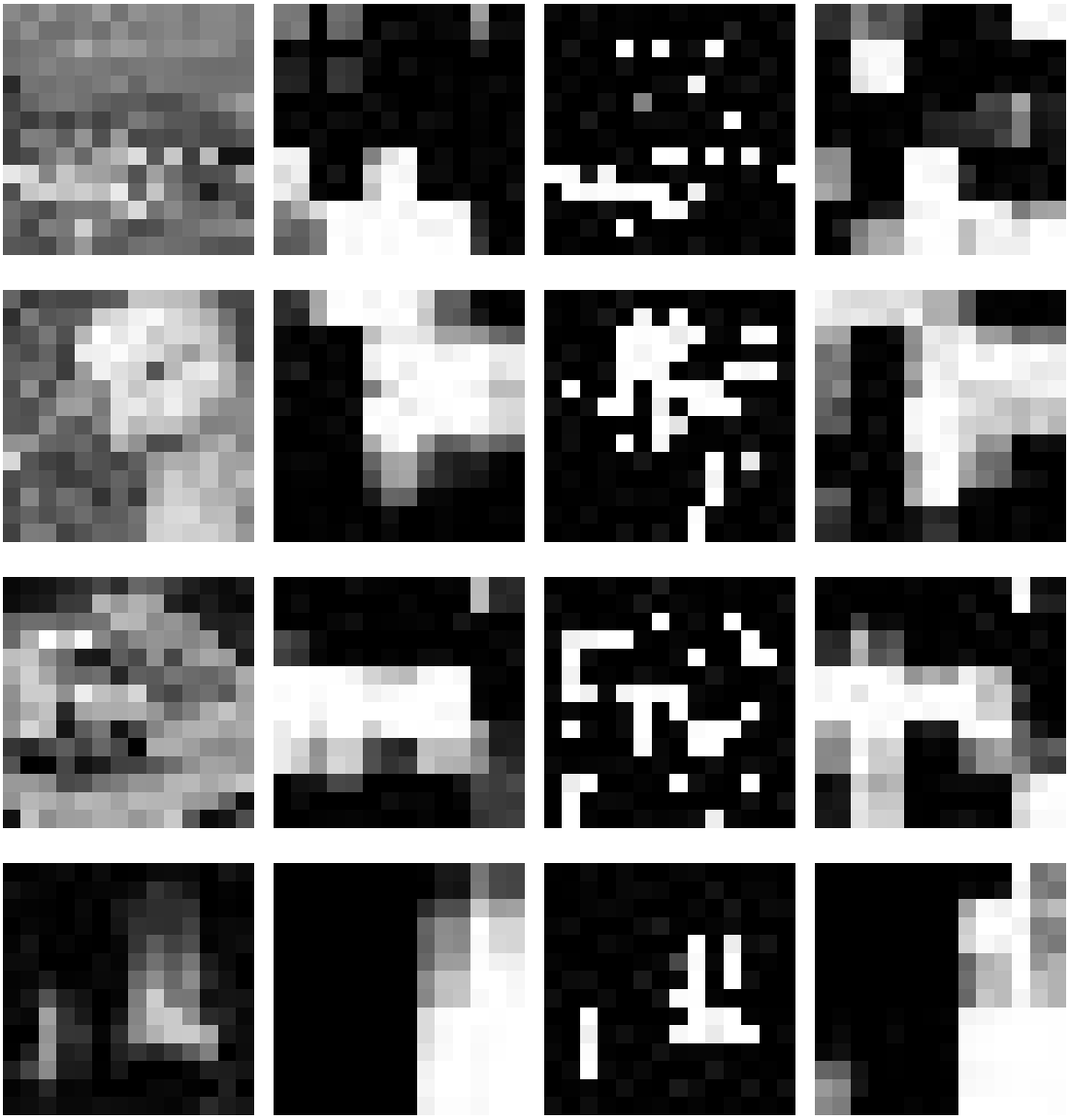


Figure 4.10: Phosphenes of Subject A in the down-sampling pipelines.
The first column depicts the input images, the second column illustrates the output phosphenes in the absence of the pre-appended encoder (baseline pipelines), the third column depicts the intermediate images generated by the U-net encoder, and the final column depicts the final output phosphenes produced by the encoder-decoder procedure (optimization pipelines).

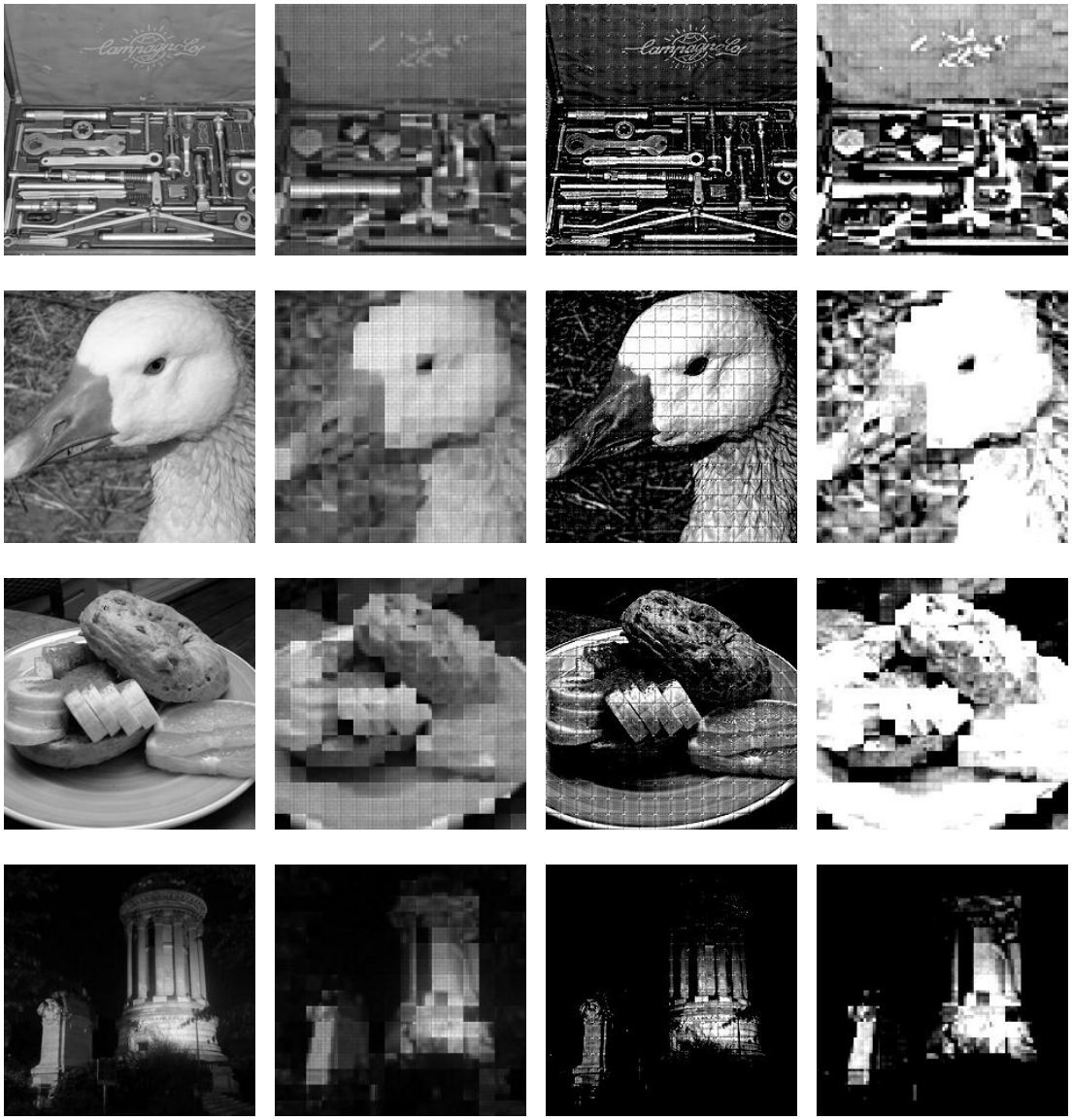


Figure 4.11: Phosphenes of Subject A in the patchification pipelines.
The first column depicts the input images, the second column illustrates the output phosphenes in the absence of the pre-appended encoder (baseline pipelines), the third column depicts the intermediate images generated by the U-net encoder, and the final column depicts the final output phosphenes produced by the encoder-decoder procedure (optimization pipelines).

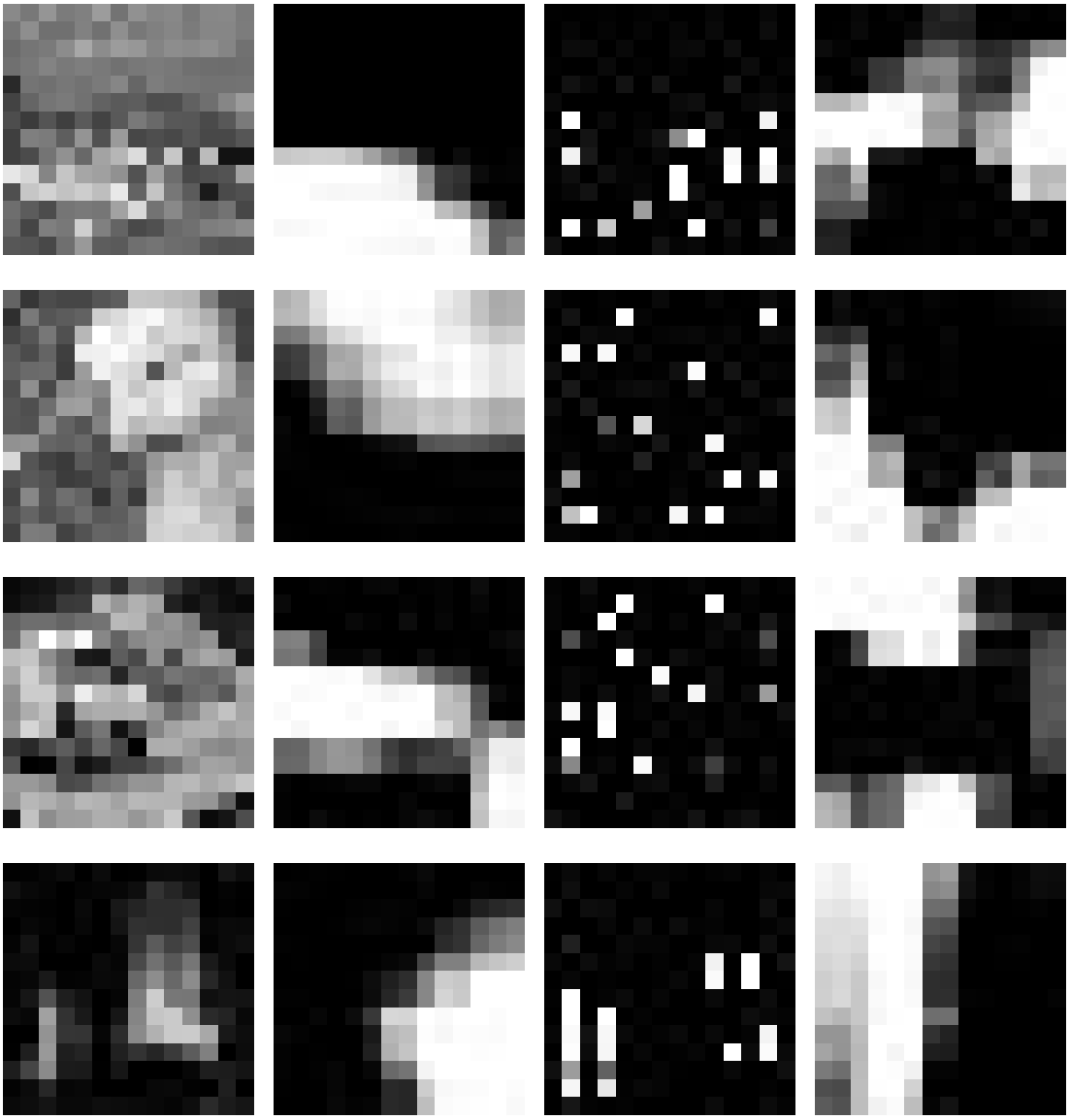


Figure 4.12: Phosphenes of Subject B in the down-sampling pipelines.
The first column depicts the input images, the second column illustrates the output phosphenes in the absence of the pre-appended encoder (baseline pipelines), the third column depicts the intermediate images generated by the U-net encoder, and the final column depicts the final output phosphenes produced by the encoder-decoder procedure (optimization pipelines).

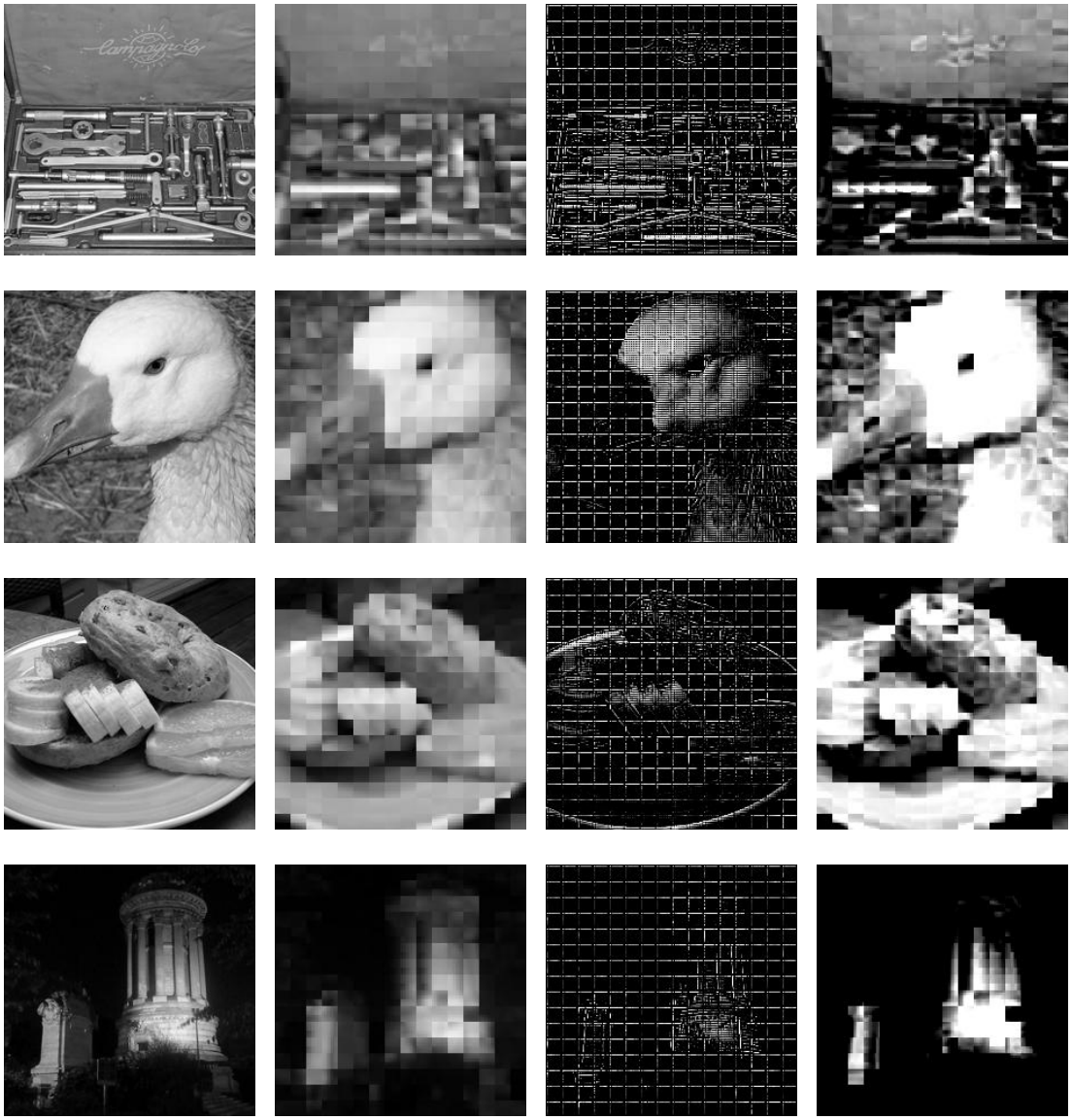


Figure 4.13: Phosphenes of Subject B in the patchification pipelines.
The first column depicts the input images, the second column illustrates the output phosphenes in the absence of the pre-appended encoder (baseline pipelines), the third column depicts the intermediate images generated by the U-net encoder, and the final column depicts the final output phosphenes produced by the encoder-decoder procedure (optimization pipelines).

From the depictions, it is evident that the object classification accuracy is significantly compromised by the suboptimal-quality phosphenes elicited by the downsampling process. For Subject B with large axonal parameters, the phosphenes are diverged from the input images to the extent that they are no longer recognizable. Nevertheless, in the patchification pipeline, the encoder is still able to retain essential features for the Axon Map model to construct comprehensible perceptions

comparable to those of Subject A. For instance, when examining the top-right percept of Figure 4.13, all background information is lost. Only the details of a few wrenches are reasonably preserved.

Pipeline	**Subject A** $\rho = 150\,\mu m$, $\lambda = 100\,\mu m$	**Subject B** $\rho = 437\,\mu m$, $\lambda = 1420\,\mu m$
Primary Experiments		
Baseline 1 Linear classification on phosphene elicited from down-sampled images	47.46%	38.70%
Optimization 1 Linear classification on phosphene elicited from down-sampled images, with the proposed encoder	51.49%	40.59%
Baseline 2 Linear classification on phosphene elicited from fixation	85.20%	81.99%
Optimization 2 Linear classification on phosphene elicited from fixation, with the proposed encoder	90.85%	87.72%
Ablation Study		
Optimization 2*a* Frozen linear classification on phosphene elicited from fixation, with the proposed encoder	85.94%	82.62%
Optimization 2*b* Linear classification on phosphene elicited from fixation, with the proposed encoder, but the encoder is not pre-trained in IIT	53.81%	68.79%
Optimization 2*c* Linear classification on phosphene elicited from fixation, with a simple encoder that is a single linear layer	21.12%	21.43%

Table 4.5: A summary of the accuracy results of all of the experiments.

4.2.2 Ablation Study

Further experiments were conducted to examine additional aspects of the patchification pipeline. For instance, the *Optimization 2a* pipeline tests the hypothesis of a static fixation knowledge base in lieu of a fluid one. In this configuration, the final linear classifier was held constant, and its parameters were extracted from the optimal iteration of **Baseline 2**. Therefore, it is hypothesized that the encoder must attempt to improve the quality of the phosphene to align as closely as possible with the predefined collection of learned patterns that subjects utilize to recognize arbitrary fixations. In **Optimization 2**, the collection can be expanded during training in conjunction with the encoder.

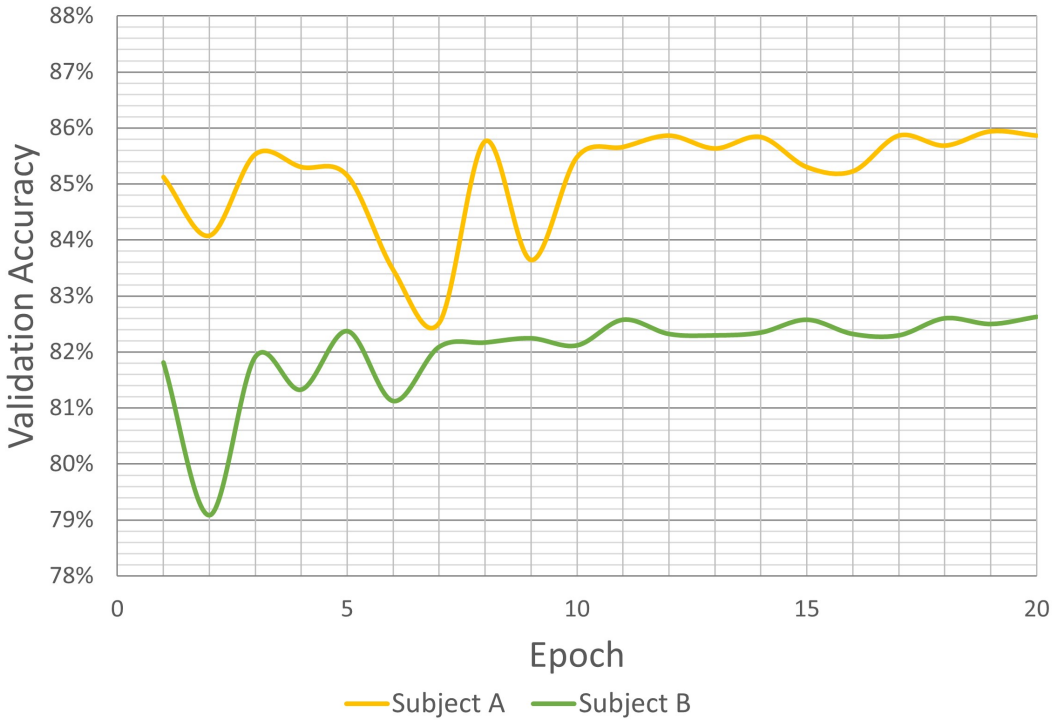


Figure 4.14: Validation accuracy of the *Optimization 2a*. Fixed linear classifiers at the end of the pipeline constrains the range of improvement in which the U-net coder can work to optimize the phosphenes.

As anticipated, the highest accuracy values of 85.94% and 82.62% for Subjects A and B, respectively, were observed, representing a mere 0.69% increase on average from **Baseline 2**. This suggests that the fixation phosphenes generated by the encoder-decoder process may possess unusual visual characteristics that require patients to adjust to and learn new connections between these features and the real objects.

The results of the *Optimization 2b* experiment demonstrate that when the pre-

trained U-net is not employed for the downstream task, the obtained results are significantly unsatisfactory. The model is unable to identify the optimal solution for minimizing the loss function, potentially leading to a lack of progress or even a complete stall in the optimization process. For example, in the case of Subject A, the accuracy achieved was only 42.01%.

In addition, the thesis presents an alternative to the U-net model utilized in **Optimization 2** by replacing it with a single linear layer in the *Optimization 2c* pipeline. The results demonstrate that the use of an inadequate deep learning architectural model in the optimization process leads to inferior outcomes.

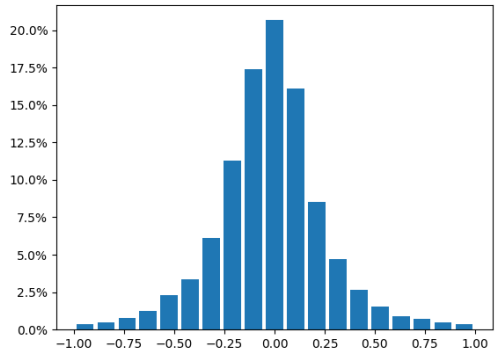
4.2.3 Examining The U-net Model As An Encoder

Table B presents a statistical analysis of the weights and biases of all layers in the U-net encoder across multiple phases. Upon examination, a discernible pattern emerges, whereby the maximum, minimum, and standard deviation exhibit an increase following each stage. In other words, the final sets of U-net parameters exhibit a broader distribution than the set after IIT training, and the set after IIT training exhibits a broader distribution than the initial set. This further elucidates the significance of IIT as a catalyst for identifying optimal encoder values. The pipeline training process would be considerably more challenging and time-consuming in the absence of IIT.

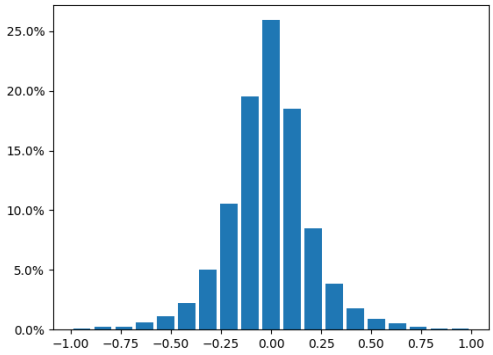
Figure G depicts the histograms of several final parameter sets of the U-net obtained in numerous optimization experiments. With the exception of a shared similar distribution shape, the parameters exhibit considerable variability across trials. The final values appear to be random, as there is no discernible pattern in the value changes between Subject A and B, or between the down-sampling and patchification methods. This indicates that there is no linear correlation between the sets of axonal parameters. Each patient is an independent entity that requires optimization on an individual basis in order to identify their optimal encoder parameters.

Experiments		Max	Min	Mean	Standard Deviation
Default initialization		1.0	-0.3303	0.0049	0.0844
IIT training		1.9469	-1.5498	-0.0134	0.2087
Optim. 1	Sub. A	10.5217	-9.7963	-0.0114	0.4293
	Sub. B	3.0298	-2.7436	-0.0125	0.2344
Optim. 2	Sub. A	2.0253	-1.9446	-0.0139	0.2177
	Sub. B	2.2595	-2.0136	-0.0085	0.2297

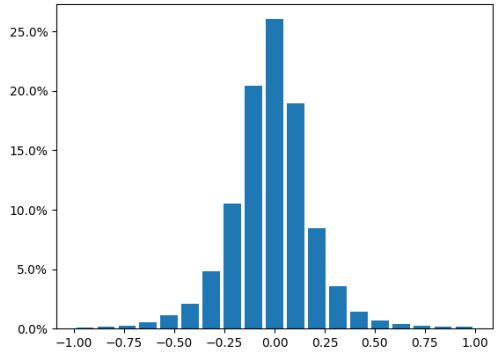
Table 4.6: Statistics of the resulting U-net model over the optimization trials.



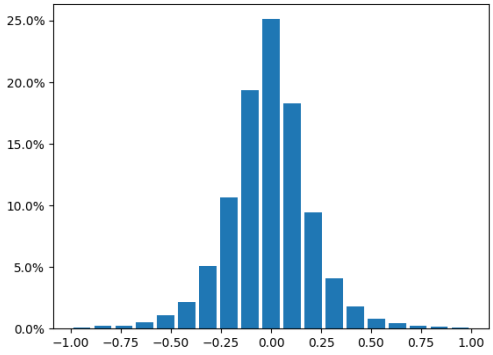
(a) Optim. 1, Sub. A



(b) Optim. 1, Sub. B



(c) Optim. 2, Sub. A



(d) Optim. 2, Sub. B

Figure 4.15: The histograms of the parameters of the U-net as an encoder in four optimization experiments.

5 Conclusion

5.1 Summary

This thesis presents an innovative approach to the optimization process of the phosphenes elicited from the retinal implant Argus II*, which is a theoretical replacement of Argus II. This is accomplished by adopting a novel perspective to examine the issue.

The two-step proposal with various pipelines is explained in detail in the preceding chapters and can be summarized as follows:

- The first simulation step transforms the problem into a conventional classification task and incorporates an additional element of fixation prediction. This element is based on the integration of knowledge regarding the fixation mechanism of the human eye and the image saliency probability density map. A well-calibrated fixation prediction algorithm, exemplified by the pre-trained DINOv2-ViT-S/14 model in the context of this study, enables the retention of the majority of pertinent information even when 90% of the image is discarded.

- The second optimization step proposes a deep learning architecture as an encoder to augment the output of the decoder, which is the Axon Map model. The proposed deep learning architecture is a modified U-net model with shorter contracting and expansive paths. The model has been pre-trained on the IIT task before being used in the optimization process.

The results of the experiments, derived from multiple pipelines, provide evidence of the potential of this innovative perspective and demonstrate both objective and subjective measures of improvement in using a U-net as an encoder to generate better phosphenes, which are more desired and suitable for the suggested classification task on top fixation of images.

5.2 Outlook

Nevertheless, it should be noted that this work is not without limitations. Further research and implementation of improvements could be conducted to enhance the thesis results and facilitate a more comprehensive examination of the problem.

Firstly, the dataset could be replaced with a more extensive one that includes a greater number of classes and samples per class. Such an expansion would provide a more comprehensive representation of the types of objects encountered in daily life and would facilitate a more nuanced understanding of the true capabilities of the proposed approach.

Secondly, alternative algorithms for predicting fixation or computing saliency probability density maps could be tested and employed. In addition, different architectures could be investigated to compare with the well-known U-net model.

Finally, this thesis is focused on item-oriented tasks with the purpose of improving the visual experience in straightforward scenarios such as navigation or object recognition in an efficient manner. The work could be extended to other applications. One potential avenue for future research could be semantic segmentation, which is currently undergoing rapid development. This field could potentially be leveraged to enhance the outcome of retinal implants, thereby facilitating more sophisticated artificial perception.

Bibliography

[1] M. Beyeler, G. M. Boynton, I. Fine, and A. Rokem, "pulse2percept: A python-based simulation framework for bionic vision," *BioRxiv*, p. 148015, 2017.

[2] L. Yue, V. Wuyyuru, A. Gonzalez-Calle, J. D. Dorn, and M. S. Humayun, "Retina–electrode interface properties and vision restoration by two generations of retinal prostheses in one patient—one in each eye," *Journal of Neural Engineering*, vol. 17, no. 2, p. 026020, 2020.

[3] N. Mahabadi and Y. Al Khalili, "Neuroanatomy, retina," 2019.

[4] N. Scott, R. Zhang, D. Le, and B. Moyle, "A review of eye-tracking research in tourism," *Current Issues in Tourism*, vol. 22, no. 10, pp. 1244–1261, 2019.

[5] S. Lambertus, N. M. Bax, A. Fakin, J. M. Groenewoud, B. J. Klevering, A. T. Moore, M. Michaelides, A. R. Webster, G. J. van der Wilt, and C. B. Hoyng, "Highly sensitive measurements of disease progression in rare disorders: Developing and validating a multimodal model of retinal degeneration in stargardt disease," *PLoS One*, vol. 12, no. 3, p. e0174020, 2017.

[6] S. Mangini, F. Tacchino, D. Gerace, C. Macchiavello, and D. Bajoni, "Quantum computing model of an artificial neuron with continuously valued input data," *Machine Learning: Science and Technology*, vol. 1, no. 4, p. 045008, 2020.

[7] F. Rosenblatt, "The perceptron: a probabilistic model for information storage and organization in the brain.," *Psychological review*, vol. 65, no. 6, p. 386, 1958.

[8] D. E. Rumelhart, G. E. Hinton, and R. J. Williams, "Learning representations by back-propagating errors," *nature*, vol. 323, no. 6088, pp. 533–536, 1986.

[9] A. Krizhevsky, I. Sutskever, and G. E. Hinton, "Imagenet classification with deep convolutional neural networks," *Advances in neural information processing systems*, vol. 25, 2012.

[10] J. Devlin, M.-W. Chang, K. Lee, and K. Toutanova, "Bert: Pre-training of deep bidirectional transformers for language understanding," *arXiv preprint arXiv:1810.04805*, 2018.

[11] T. Brown, B. Mann, N. Ryder, M. Subbiah, J. D. Kaplan, P. Dhariwal, A. Neelakantan, P. Shyam, G. Sastry, A. Askell, *et al.*, "Language models are

few-shot learners," *Advances in neural information processing systems*, vol. 33, pp. 1877–1901, 2020.

[12] S. Zhang, S. M. H. Bamakan, Q. Qu, and S. Li, "Learning for personalized medicine: a comprehensive review from a deep learning perspective," *IEEE reviews in biomedical engineering*, vol. 12, pp. 194–208, 2018.

[13] M. Bojarski, D. Del Testa, D. Dworakowski, B. Firner, B. Flepp, P. Goyal, L. D. Jackel, M. Monfort, U. Muller, J. Zhang, *et al.*, "End to end learning for self-driving cars," *arXiv preprint arXiv:1604.07316*, 2016.

[14] E. Shelhamer, "Developing an intuition for better understanding of convolutional neural networks," *url: https://developer.nvidia.com/blog/deep-learning-computer-vision-caffe-cudnn/. (accessed: 07.07.2024)*, 2014.

[15] K. Leung, "How to easily draw neural network architecture diagrams," *url: https://towardsdatascience. com/how-to-easily-draw-neural-network-architecture-diagramsa6b6138ed875. (accessed: 18.06.2024)*, 2022.

[16] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, Ł. Kaiser, and I. Polosukhin, "Attention is all you need," *Advances in neural information processing systems*, vol. 30, 2017.

[17] K. Fukushima, "Neocognitron: A self-organizing neural network model for a mechanism of pattern recognition unaffected by shift in position," *Biological cybernetics*, vol. 36, no. 4, pp. 193–202, 1980.

[18] Y. LeCun, B. Boser, J. S. Denker, D. Henderson, R. E. Howard, W. Hubbard, and L. D. Jackel, "Backpropagation applied to handwritten zip code recognition," *Neural computation*, vol. 1, no. 4, pp. 541–551, 1989.

[19] O. Ronneberger, P. Fischer, and T. Brox, "U-net: Convolutional networks for biomedical image segmentation," in *Medical image computing and computer-assisted intervention–MICCAI 2015: 18th international conference, Munich, Germany, October 5-9, 2015, proceedings, part III 18*, pp. 234–241, Springer, 2015.

[20] Z. Yang, Z. Dai, Y. Yang, J. Carbonell, R. R. Salakhutdinov, and Q. V. Le, "Xlnet: Generalized autoregressive pretraining for language understanding," *Advances in neural information processing systems*, vol. 32, 2019.

[21] A. Dosovitskiy, L. Beyer, A. Kolesnikov, D. Weissenborn, X. Zhai, T. Unterthiner, M. Dehghani, M. Minderer, G. Heigold, S. Gelly, *et al.*, "An image is worth 16x16 words: Transformers for image recognition at scale," *arXiv preprint arXiv:2010.11929*, 2020.

[22] J. Yu, X. Li, J. Y. Koh, H. Zhang, R. Pang, J. Qin, A. Ku, Y. Xu, J. Baldridge,

and Y. Wu, "Vector-quantized image modeling with improved vqgan," *arXiv preprint arXiv:2110.04627*, 2021.

[23] R. Kundu, "The beginner's guide to contrastive learning," *url: https://www.v7labs.com/blog/contrastive-learning-guide. (accessed: 18.06.2024)*, 2022.

[24] T. Chen, S. Kornblith, M. Norouzi, and G. Hinton, "A simple framework for contrastive learning of visual representations," in *International conference on machine learning*, pp. 1597–1607, PMLR, 2020.

[25] S. Schneider, A. Baevski, R. Collobert, and M. Auli, "wav2vec: Unsupervised pre-training for speech recognition," *arXiv preprint arXiv:1904.05862*, 2019.

[26] M. Oquab, T. Darcet, T. Moutakanni, H. Vo, M. Szafraniec, V. Khalidov, P. Fernandez, D. Haziza, F. Massa, A. El-Nouby, *et al.*, "Dinov2: Learning robust visual features without supervision," *arXiv preprint arXiv:2304.07193*, 2023.

[27] M. Caron, H. Touvron, I. Misra, H. Jégou, J. Mairal, P. Bojanowski, and A. Joulin, "Emerging properties in self-supervised vision transformers," in *Proceedings of the IEEE/CVF international conference on computer vision*, pp. 9650–9660, 2021.

[28] G. Alain and Y. Bengio, "Understanding intermediate layers using linear classifier probes," *arXiv preprint arXiv:1610.01644*, 2016.

[29] Y. Gao, X. Sun, and C. Liu, "A general self-supervised framework for remote sensing image classification," *Remote Sensing*, vol. 14, no. 19, p. 4824, 2022.

[30] G. Dagnelie, P. Christopher, A. Arditi, L. da Cruz, J. L. Duncan, A. C. Ho, L. C. Olmos de Koo, J.-A. Sahel, P. E. Stanga, G. Thumann, *et al.*, "Performance of real-world functional vision tasks by blind subjects improves after implantation with the argus® ii retinal prosthesis system," *Clinical & experimental ophthalmology*, vol. 45, no. 2, pp. 152–159, 2017.

[31] M. Beyeler, D. Nanduri, J. D. Weiland, A. Rokem, G. M. Boynton, and I. Fine, "A model of ganglion axon pathways accounts for percepts elicited by retinal implants," *Scientific reports*, vol. 9, no. 1, p. 9199, 2019.

[32] R. W. Thompson, G. D. Barnett, M. S. Humayun, and G. Dagnelie, "Facial recognition using simulated prosthetic pixelized vision," *Investigative ophthalmology & visual science*, vol. 44, no. 11, pp. 5035–5042, 2003.

[33] A. Horsager, S. H. Greenwald, J. D. Weiland, M. S. Humayun, R. J. Greenberg, M. J. McMahon, G. M. Boynton, and I. Fine, "Predicting visual sensitivity in retinal prosthesis patients," *Investigative ophthalmology & visual science*, vol. 50, no. 4, pp. 1483–1491, 2009.

[34] D. Nanduri, I. Fine, A. Horsager, G. M. Boynton, M. S. Humayun, R. J. Greenberg, and J. D. Weiland, "Frequency and amplitude modulation have different effects on the percepts elicited by retinal stimulation," *Investigative ophthalmology & visual science*, vol. 53, no. 1, pp. 205–214, 2012.

[35] J. Granley and M. Beyeler, "A computational model of phosphene appearance for epiretinal prostheses," in *2021 43rd Annual International Conference of the IEEE Engineering in Medicine & Biology Society (EMBC)*, pp. 4477–4481, IEEE, 2021.

[36] L. Relic, B. Zhang, Y.-L. Tuan, and M. Beyeler, "Deep learning–based perceptual stimulus encoder for bionic vision," in *Proceedings of the Augmented Humans International Conference 2022*, pp. 323–325, 2022.

[37] J. Granley, L. Relic, and M. Beyeler, "Hybrid neural autoencoders for stimulus encoding in visual and other sensory neuroprostheses," *Advances in neural information processing systems*, vol. 35, pp. 22671–22685, 2022.

[38] Y. Wu, I. Karetić, J. Stegmaier, P. Walter, and D. Merhof, "A deep learning-based in silico framework for optimization on retinal prosthetic stimulation," in *2023 45th Annual International Conference of the IEEE Engineering in Medicine & Biology Society (EMBC)*, pp. 1–4, IEEE, 2023.

[39] R. J. Leigh and D. S. Zee, *The neurology of eye movements*. Oxford University Press, USA, 2015.

[40] J. Li and X. Zhang, "The performance evaluation of a novel methodology of fixational eye movements detection," *International Journal of Bioscience, Biochemistry and Bioinformatics*, vol. 3, no. 3, p. 262, 2013.

[41] D. L. Sparks, "The brainstem control of saccadic eye movements," *Nature Reviews Neuroscience*, vol. 3, no. 12, pp. 952–964, 2002.

[42] S. Martinez-Conde, S. L. Macknik, and D. H. Hubel, "The role of fixational eye movements in visual perception," *Nature reviews neuroscience*, vol. 5, no. 3, pp. 229–240, 2004.

[43] M. Kümmerer and M. Bethge, "Predicting visual fixations," *Annual Review of Vision Science*, vol. 9, pp. 269–291, 2023.

[44] A. M. Treisman and G. Gelade, "A feature-integration theory of attention," *Cognitive psychology*, vol. 12, no. 1, pp. 97–136, 1980.

[45] L. Itti, J. Braun, D. Lee, and C. Koch, "Attentional modulation of human pattern discrimination psychophysics reproduced by a quantitative model," *Advances in neural information processing systems*, vol. 11, 1998.

[46] K. Simonyan and A. Zisserman, "Very deep convolutional networks for large-scale image recognition," *arXiv preprint arXiv:1409.1556*, 2014.

[47] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 770–778, 2016.

[48] M. Kümmerer, T. S. Wallis, and M. Bethge, "Deepgaze ii: Reading fixations from deep features trained on object recognition," *arXiv preprint arXiv:1610.01563*, 2016.

[49] J. Pan, C. C. Ferrer, K. McGuinness, N. E. O'Connor, J. Torres, E. Sayrol, and X. Giro-i Nieto, "Salgan: Visual saliency prediction with generative adversarial networks," *arXiv preprint arXiv:1701.01081*, 2017.

[50] M. Cornia, L. Baraldi, G. Serra, and R. Cucchiara, "Predicting human eye fixations via an lstm-based saliency attentive model," *IEEE Transactions on Image Processing*, vol. 27, no. 10, pp. 5142–5154, 2018.

[51] J. Howard, "imagenette."