Noname manuscript No.
(will be inserted by the editor)

Fusion of ANNs as Decoder of Retinal Spike Trains for Scene Reconstruction

Wei Li[1] · **Alex Noel Joseph Raj**[1] · **Tardi Tjahjadi**[2] · **Zhemin Zhuang**[1]

Received: date / Accepted: date

Abstract The retina is one of the most developed sensing organs in the human body. However, the knowledge on the coding and decoding of the retinal neurons are still rather limited. Compared with coding (i.e., transforming visual scenes to retinal spike trains), the decoding (i.e., reconstructing visual scenes from spike trains, especially those of complex stimuli) is more complex and receives less attention. In this paper, we focus on the accurate reconstruction of visual scenes from their spike trains by designing a retinal spike train decoder based on the combination of the Fully Connected Network (FCN), Capsule Network (CapsNet) and Convolutional Neural Network (CNN), and a loss function incorporating the structural similarity index measure (SSIM) and L1 loss. CapsNet is used to extract the features from the spike trains, that are fused with the original spike trains and used as the inputs to FCN and CNN to facilitate the scene reconstruction. The feasibility and superiority of our model are evaluated on five datasets (i.e., MNIST, Fashion-MNIST, Cifar-10, Celeba-HQ andCOCO). The model is evaluated quantitatively with four image evaluation indices, i.e., SSIM, MSE, PSNR and Intra-SSIM. The results show that the model provides a new means for decoding visual scene stimuli from retinal spike trains, and promotes the development of brain-machine interfaces.

Keywords Retinal Ganglion Cells · Spike trains · Reconstruction · Decoding · Capsule Network · Artificial Neural Network · Intra-SSIM

Mathematics Subject Classification (2020) MSC code1 · MSC code2 · more

✉ Alex Noel Joseph Raj
E-mail: jalexnoel@stu.edu.cn

[1] Key Laboratory of Digital Signal and Image Processing of Guangdong Province, Department of Electronic Engineering,College of Engineering, Shantou University, China

[2] School of Engineering, University of Warwick, United Kingdom

1 Introduction

The human beings have the following five senses: eyes to see, tongue to taste, nose to smell, ears to hear, and skin to touch. However, 80% of these sensations are perceived visually. Thus, if a few of the sensations such as taste or smell are incapacitated, the eyes can still help us perceive the environment to enable us avoid any potential danger. Vision is our most important and complex sensing process [2]. Thus, it is very important to study the working principle of the human vision system as it lays the foundation for the treatment of eye diseases (e.g., helping blind people to regain their sight) and the development of visual system applications (e.g., brain-machine interfaces and neuroprosthetic devices) [1].

Visual scenes that are observed by humans are projected onto the eyes, where the photoreceptors on the retina transform the light rays into electrical signals. These electrical signals, commonly referred as "spikes" or "action potentials", are transmitted to the brain via the Retinal Ganglion Cells (RGCs), a special type of neurons in the retina and the optical nerve. The brain then decodes the signals to mentally construct the visual scenes that we perceive [3]. Interestingly, the RGCs are the only neurons capable of transmitting visual information to the brain, and the only variable encoded by RGCs is the time-varying stimuli on the retina [4]. The visual information is encoded by RGCs as a series of action potentials or spike trains that are decoded by the brain.

Although the spike trains of RGCs can be directly used to develop the model of visual scene reconstruction, compared with the encoding process, the decoding of RGCs spike trains is more complex for the following three reasons. First, since the the retina does not receive any feedback from the visual cortex or higher part of the cortex, the encoded RGCs spike trains are considered as minimal in representing the entire visual information. Second, RGCs encode the shape and colour of objects by varying their spike rates or temporal pattern of spike signals through a specific coding method rather than random activations [15]. Third, the spike trains of RGCs are the only outputs of the eyes that are transmitted to the brain that contain information of the visual scene under study [5]. Since the decoding of RGCs spike trains is complex, very few models are available that reconstruct visual scene from RGCs spike trains. In these models the decoder is considered as statistical representation in linear or nonlinear fashion [5]. Thus, the effective decoding of spikes to a scene is still a challenge.

In this paper we address the challenge by proposing a deep learning model which combines three artificial neural networks (ANNs), i.e., FCN, CapsNet and CNN as an effective RGCs spike train decoder. The novel contributions of this paper are as follows. We propose a decoder which operates as follows. First, the visual scenes are encoded into spike trains via a retinal simulation. Second, a combined FCN and CapsNet module is employed to capture the characteristics of the spike train along with their spatial relationship. FCN is used to convert the one-dimensional spike train into a two-dimensional image.

Finally, the extracted features are fused with the original RGCs spike train and introduced as input of the reconstruction module which comprises FCN and CNN or reconstructing the visual scene stimulus. We also proposed a new loss function (i.e., the combination of structural similarity index measure (SSIM) loss and L1 loss) which effectively improves the quality of the scene reconstruction. We evaluated the model on five publicly-available datasets (i.e., MNIST, Fashion MNIST, Cifar-10, Celeba-HQ and COCO). The evaluations based on SSIM, MSE, PSNR and Intra-SSIM [23] show that the model performs well both quantitatively and qualitatively.

The paper is organised as follows. Section 2 reviews the related work. Section 3 presents the proposed approach to reconstruct the visual scene stimuli from RGCs spike trains. Section 4 presents the experiments we conducted and the evaluation metrics. Section 5 presents the experimental results and comparison. Finally, Section 6 draws the conclusion from the experimental results.

2 Related Work

RGCs are the only output neurons that contain all the encoding information of a visual scene that have led to investigations to determine the encoding process, i.e., a mapping between the visual scene and RGCs spike trains via mathematical modelling [5]. These include the formulation of a closed form mathematical model of the encoding process (i.e., visual scene to the spike trains generated by the RGCs), and the decoding process (i.e., the reconstruction of visual scenes from spike trains.) Many progressive efforts have been undertaken to study the RGCs encoding process that have led to various neuroscience mechanisms for understanding the neurons and neural circuits of retinal computing in visual scenes [6–9]. Furthermore, to better understand the retinal coding principle, many retinal coding models have been developed [10–13].

The recent development on RGCs spike train decoders can be summarized as follows. Botella-Soler et al [14] constructed a nonlinear (kernelized and neural network) decoders and proved that the results of nonlinear methods are better than those of linear methods. Brackbill et al. [25] analyzed the experimental records of RGCs response to natural images in macaque retina, and made a simple linear reconstruction test. They used linear regression to fit the reconstruction filter. However, the reconstruction results only display the spatial properties similar to the visual scene stimuli, but cannot completely reconstruct the visual scene stimuli. Kim et al. [26] combined a low-pass linear decoder with a high-pass nonlinear decoder to obtain preliminary reconstruction results, which are then input to a neural network to improve the clarity of the reconstruction of simple visual scenes, such as bicycle tires, cylinders and simple black-and-white textures. Zhang et al. [5] divided the reconstruction process into two stages: (a) A spike-image converter which is used to map a population of spikes to a pixel-level intermediate image; and (b) An image-image autoencoder for mapping the reconstructed results to the target pixels in the reconstructed images. The images are better than those generated by previous methods, but suffer from problems related to blurring. Although visual scene reconstruction has been studied for many years, the decoding performance of the current methods requires further improvement especially in the reconstruction of complex visual scenes [5,14,16]. This makes spikes to scene neural decoding still a challenge, requiring further development and innovation. In recent years, deep learning has been successfully used to solve many complex computer vision problems. Since deep learning models are resilient to high variability in spike trains, they can generalise the complex relationship of RGCs spike trains in space and time, providing a means to accurate scene reconstruction from spike trains.

3 Proposed Method

Fig. 1(A) illustrates the the operational processes in the proposed method from the encoding of the visual scene stimuli to the decoding to reconstruct the visual scene. These include: (1) Using a Bilinear Interpolation method, the original image is resized to 32×32; (2) Using the retinal simulation software PRANAS, the stimuli are encoded into spike trains. More details on how to simulate spike trains is provided in Section 4.2; (3) The module comprising FCN and CapsNet is used to extract the features of spike trains; and (4) The extracted features are fused with the original spike trains, and the visual scene is reconstructed by a module comprising FCN and CNN. The decoder combines FCN, CapsNet and CNN, and enables their respective advantages to overcome their individual shortcomings, providing complementary advantages between them. We introduce our decoder from five aspects: FCN, CapsNet, CNN, loss function and training model.

3.1 FCN and CapsNet as Feature Extractor

Fig. 1(B) shows the architecture which combines FCN and CapsNet as the feature extractors of the proposed method. In order to match the input shape requirement of the CapsNet, a three-layer FCN is used as a preliminary converter which converts the RGCs spike trains into an image. The first layer of the FCN receives the spike trains as input, and hence the number of neurons in the first layer corresponds to the number of RGCs used, i.e., of size of 10000 (refer to Section. 4.2 for how the spike trains are generated). The second layer, the hidden layer, is composed of 512 neurons. The third layer has 1024 neurons whose outputs are reshaped to give a feature map of 32×32 pixels, the input of the CapsNet.

CapsNet [17] is made up of capsules, each of which is a group of neurons. The activity vector of a capsule represents the instantiation parameters of a specific type of entity such as an object or an object part. The length of the activity vector represents the probability of the existence of an entity, and its

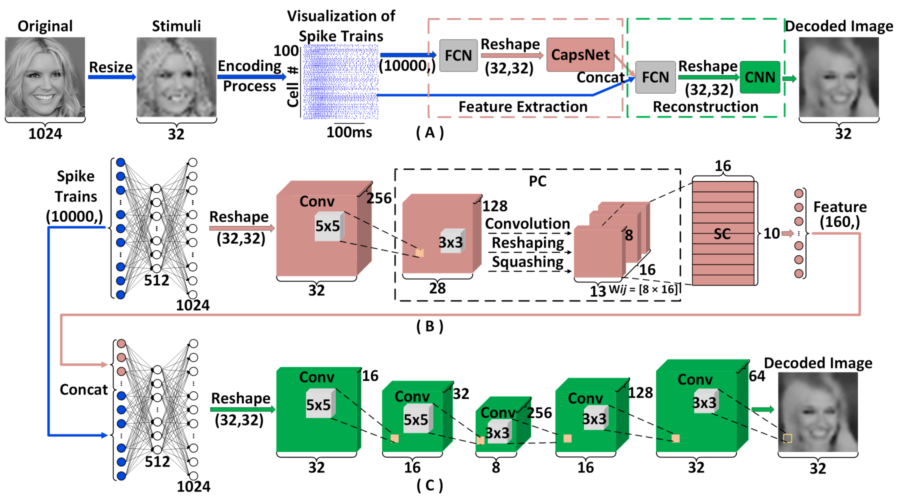

Fig. 1 Overview of the proposed method. (A) Illustration of the model operation. (B) FCN and CapsNet as spatial relationship feature extractors. (C) FCN and CNN for visual scene reconstruction.

orientation represents the instantiation parameter. In the proposed method, the CapsNet consists of three parts: the Convolutional layer, Primary Capsule (PC) and Secondary Capsule (SC). The Convolutional layer extracts the features from the 32×32 outputs of the FCN, using a kernel size 256 of 5×5 and stride 1, and outputs 256×28×28 feature maps. The PC performs three main operations: Convolution, Reshaping and Squashing. The initial convolution with the kernel size of (128,3,3) and stride 2 has 128×13×13 outputs that are reshaped into a set of vectors, each representing a unique location in the image. As the length of vectors represents the probability of the presence of a feature, the squash function (Eq. 4) is used to limit the length between 0 and 1. In total, PC has 16×13×13 capsule outputs (each output is a 8-dimenasional vector) and each capsule in the 13×13 grid shares their weight with each other. The SC layer has 10 capsules and has 10×16 capsule outputs (where each output is a 16-dimenasional vector).

The PC predicts the output of the capsules in the SC layer by Dynamic Routing algorithm. Since the output of the Convolutional layer is one dimensional, there is no orientation in its space to agree on and hence the Dynamic Routing is performed only between the PC and SC layers as follows. Let u_i denotes the output vector of a capsule in the PC and j being a capsule in the SC layer, the Dynamic Routing is formulated as

$$u'_{j|i} = W_{ij} u_i \ , \tag{1}$$

where $u'_{j|i}$ is the prediction vector of the jth capsule from the capsule i in the PC layer, and W_{ij} is the weighted transformation matrix. In order to decide whether capsule i should be routed to capsule j, a coupling coefficient c_{ij} is used, which is computed from the log prior probability of logits b_{ij} (raw values for our predictions), i.e.,

$$c_{ij} = \frac{\exp(b_{ij})}{\sum_k \exp(b_{ik})} \ . \tag{2}$$

At the first iteration, all the routing logits b_{ij} are initialized to zero. In successive iterations the dot product of the coupling coefficient c_{ij} and the predicted output vector $u'_{j|i}$ is calculated to decide if the capsules i and j agree with each other. A large dot product value denotes that capsule i is related to capsule j, and the corresponding coupling coefficient increases, and vice versa. Each capsule j in the SC layer receives a weighted sum of all individual PC predictions for j (i.e., $u'_{j|i}$ for all i in the PC layer) and is formulated as

$$s_j = \sum_i c_{ij} u'_{j|i} \ , \tag{3}$$

where s_j is the total input of capsule j in the SC layer. Since the squashing function is applied to each capsule j, the output vector v_j is

$$v_j = \frac{\| s_j \|^2}{1 + \| s_j \|^2} \frac{s_j}{\| s_j \|} \ . \tag{4}$$

A CapsNet is used as an attempt to more closely mimic biological neural organization which efficiently represents its hierarchical relationships [17]. The Dynamic Routing with agreement in the CapsNet overcomes the limitations (loss of spatial information) of the max-pooling layers in CNN, thus allowing for efficient feature globalization and compression. Unlike other networks, the output of a CapsNet is a vector consisting of the probability of an observation, which represents the uniqueness of the observation. This enables the "Picasso problems" to be addressed in image recognition that cannot be solved by other networks, i.e., images that have all the right segments but do not have the correct spatial relationship among them (e.g., the positions of the mouth and one eye in a face are swapped) [18]. This distinct attribute of CapsNet can be used to extract the features of the spike train along with their spatial relationship, which are useful for reconstructing the visual scenes more accurately. Refer to [18] for more details of CapsNet.

3.2 FCN and CNN for Reconstruction

Fig. 1(C) shows the combined FCN and CNN module for scene reconstruction. A three-layer FCN is used before the CNN to match the shape of its input. The structure of FCN is the same as that of FCN for the feature extractor in Section 3.1, except that the size of the input layer is 10160 since it integrates the features extracted by CapsNet.

The CNN comprises two phases as shown in Fig. 1(C). The first phase (left side of the figure) is the contraction path (also called the encoder) which is used to capture the context in the feature map of the spike trains from FCN. This phase contains three convolutional layers. Each layer consists of the repeated application of two convolutions and followed by a 2×2 max pooling with stride 2 for downsampling, transforming feature maps from 32×32 to 8×8. The number and size of the kernels for the three layers are (16,5,5), (32,5,5), (256,3,3), where the first dimension denotes the number of kernels and the rest is their size. As the inputs pass through these layers the distinct information in the feature maps are preserved, and any noise and redundant contents are filtered [5]. The second phase (right side) is the symmetric expanding phase (also called the decoder) which provides precise spatial localization of the feature map via transposed convolutions. The decoder also contains three layers, with expanding phases that consist of a 2×2 transposed convolution and two convolutions transforming feature maps from 8×8 to 32×32. The kernel sizes of these three layers are (256,3,3), (128,3,3), (64,3,3). The final layer has 1×1 convolution followed by a tanh activation function, employed to map each 64-component feature vector to the corresponding target value. In the entire model, all layers except the last output layer use the Leaky ReLU activation function, i.e.,

$$f(x) = \text{if } (x < 0) \text{ then } ax, \text{ else if } (x > 0) \text{ then } x, \tag{5}$$

where a is a small constant. The Leaky ReLU introduces a non-linearity but prevents the vanishing gradient problem by allowing a small negative slope [22].

3.3 Loss Function

In this paper, we define a new loss function for the training model. The loss function is a weighted combination of SSIM loss and L1 loss. Our experiments show that the proposed loss function is better than the binary cross-entropy loss function in terms of generalization and overall reconstruction ability.

SSIM is introduced in detail in [19], which is summarized as follows. It is based on three comparison measurements between the samples of x and y, i.e., luminance (l), contrast (c) and structure (s). Let μ_x, σ_x^2 and σ_{xy} be the mean of x, the variance of x, and the covariance of x and y, respectively. The comparison measurements are:

$$l(x,y) = \frac{2\mu_x\mu_y + c_1}{\mu_x^2 + \mu_y^2 + c_1}, \tag{6}$$

$$c(x,y) = \frac{2\sigma_x\sigma_y + c_2}{\sigma_x^2 + \sigma_y^2 + c_2}, \tag{7}$$

$$s(x,y) = \frac{\sigma_{xy} + c_3}{\sigma_x\sigma_y + c_3}, \tag{8}$$

where c_1, c_2 and c_3 are small constants respectively given by

$$c_1 = (k_1 L)^2, \quad c_2 = (k_2 L)^2 \quad \text{and} \quad c_3 = c_2/2, \tag{9}$$

where L is the dynamic range of the pixel values ($L = 255$ for 8 bits/pixel grey scale images), and $k_1 = 0.01$ and $k_2 = 0.03$ are two scalar constants [19]. The general form of the SSIM index between signal x and y of common size $N \times N$ is defined as

$$\mathrm{SSIM}(x, y) = \left[l(x, y)^{\alpha} \cdot c(x, y)^{\beta} \cdot s(x, y)^{\gamma} \right], \tag{10}$$

where α, β and γ are parameters that define the relative importance of these three components. Specifically, we set $\alpha = \beta = \gamma = 1$, and the resulting SSlM index is

$$\mathrm{SSIM}(x, y) = \frac{(2\mu_x \mu_y + c_1)(2\sigma_{xy} + c_2)}{(\mu_x^2 + \mu_y^2 + c_1)(\sigma_x^2 + \sigma_y^2 + c_2)} \tag{11}$$

to satisfy the following conditions:

1. Symmetry: $\mathrm{SSIM}(x, y) = \mathrm{SSIM}(y, x)$;
2. Boundedness: $\mathrm{SSIM}(x, y) \leq 1$;
3. Unique maximum: $\mathrm{SSIM}(x, y) = 1$ if and only if $x = y$.

The SSIM index is used for measuring the similarity between two images (i.e., the output of the model, x, and the ground truth, y. The SSIM loss is defined as

$$\mathrm{L}_{SSIM} = 1 - \mathrm{SSIM}(x, y) \ . \tag{12}$$

Previous approaches [20] have utilised the benefits of combining targeted loss functions with traditional loss (e.g., L2 distance). In our method we employ L1 distance rather than L2, as L1 encourages less blurring [21], i.e.,

$$\mathrm{L}_{L1} = \| \, y - x \, \|_1 \ . \tag{13}$$

Our final objective is

$$\mathrm{L}_{sl} = \alpha \mathrm{L}_{SSIM} + \lambda \mathrm{L}_{L1} \ , \tag{14}$$

where α and λ denote the weights, and L_{sl} represents the new loss function that integrates the benefits of L_{SSIM} and L_{L1}.

3.4 Training model

The training model employs Adam as the optimizer and L_{sl} as the loss function. The learning rate is adjusted to 0.1 times of the previous value when the val-loss (L_{sl}) is not reduced after 5 consecutive iterations, with a lower limit set to 0.00005. Therefore, by optimizing the learning rate, the model gets generalized faster [23]. Furthermore, an early stopping mechanism which terminates the training after 20 consecutive iterations is utilised if L_{sl} is not reduced, i.e., inferring that the model has been trained to the optimal level.

4 Experiments

4.1 Implementation

We implemented the proposed reconstruction algorithm on our server with two GPUs, Tesla K40c and P4000. The Tesla K40c GPU has 12GB VRAM and 3.5 compute capability, while P4000 GPU has 8GB VRAM and 6.1 compute capability. The use of dual GPU aids in training the model to the optimal level more quickly and greatly reduce the computation time. The generation process of datasets and the evaluation of results were performed on the same server. Our code is based on the environment of tensorflow 1.13.1 and python 3.6.12. The code and results are publicly-available at `https://github.com/jalexnoel/Decoder_of_Retinal_Spike_Trains`

4.2 Simulated spike train datasets

Hitherto, we have not found a suitable real dataset for training the model. Either the dataset is too small to meet the training requirements [24]; or too large (dataset >5 TBs) requiring complex processing pipeline [25]. Therefore, we generated spike trains through retinal simulator.

In recent years, there have been more and more publications on retinal processing models [12]. The models are used to simulate the coding process of retina (i.e., the generation process of RGCs spike trains) by establishing mathematical models so as to better understand the coding principle of the retina. There are three main classes of models. The first class applies the linear-nonlinear-poisson (LNP) models [27] to simulate the spike activity of Ganglion cells (and Cortical cells) in response to synthetic or natural images. However, these models ignore the details of neuronal mechanisms and the inner details of the retina, that help in transforming the perceived image as inputs at the ganglion cell (or any type of cell) stage. The second class includes parvocellular and magnocellular pathways that use different non-separable spatio-temporal filters that are optimal for form or motion detection and serve as front-end models for subsequent computer vision task [28]. The third class represents a detailed retinal model based on circuits that predict individual or collective responses measured at the ganglion cell level [12,29]. These models have the following characteristics: a) Understand how to accurately reproduce spike activity statistics at the population level [30]; b) Coordinate connectionomics and present simple computational rules for visual motion detection [31]; and c) Investigate how such canonical microcircuits implement different retinal processing modules [32]. Since the third model can specifically explore the process of retinal image processing, we chose this model to simulate the generation of retinal RGCs spike trains.

PRANAS [12] is a retinal simulation software based on third generation model. It allows large scale simulations while keeping a strong biological plausibility and also includes a toolbox for statistical analysis of spike train population. The simulator has a good Graphical User Interface with options provided to include and tune various retinal parameters. It also provides preset retina configuration file (cat_X_cell_DLIF.xml) which we used to configure the parameters of the retina. While simulating the spikes, the default settings were chosen: stimulation duration of each image = 100ms, and the response of 100 neurons within a millisecond was considered. The software and configuration files can be found online. Refer to [12] for more information about PRANAS.

Fig. 2(A) illustrates the generation of spike trains using PRANAS. The input is a sequence of images with spikes recorded from 100 RGCs over 100 milliseconds, i.e., the length of the spike trains is 10000. The spike trains are generated from three images over a duration of 300 milliseconds. The algorithm for generating spike trains consists of three main stages. The Outer Plexiform Layer, representing the first two layers of the retina namely receptors and horizontal cells, receives the incoming light rays from a scene which are processed by spatio-temporal linear filters to present band-pass excitatory current to bipolar cells in the next stage. The second stage, a nonlinear Contrast Gain Control is applied to the bipolar cells through a variable feed back loop to enhance the influence of the contrast in the scene on the transfer properties of the retina. The third stage is the Ganglion Layer which simulates the workings of the retinal ganglion cells and presents the spike trains that are triggered from the activity of bipolar cells. Fig. 2(B) illustrates the algorithmic flow of the PRANAS simulator showing the inputs and the outputs at various stages. Further details on how to simulate the generation of spike trains can be found in [33].

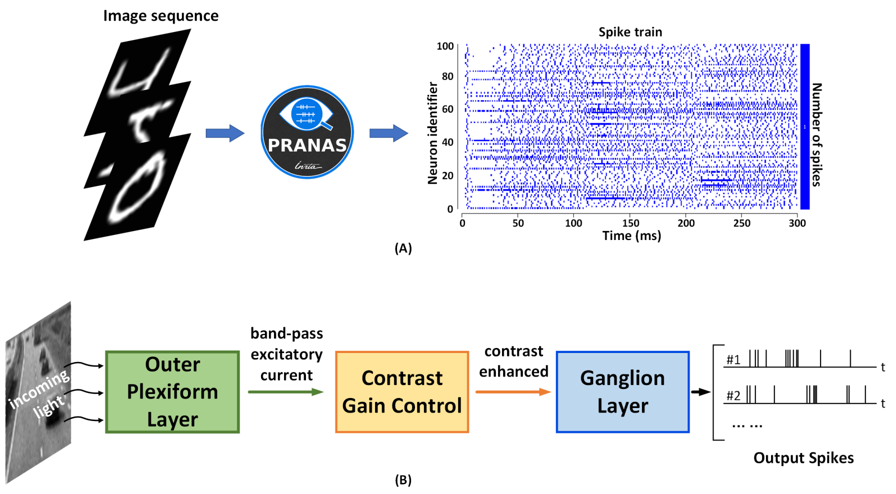

Fig. 2 (A): An example of how to generate spike trains from visual scene stimuli using PRANAS retinal simulation platform. (B): The algorithmic workflow of the simulator consists of three main stages: Outer Plexiform Layer, Contrast Gain Control and Ganglion Layer.

For our experiments we used five datasets (i.e., MNIST, Fashion-MNIST, Cifar-10, Celeba-HQ and COCO), each of which has 30000 images, including 28000 for training and 2000 for testing. The resolution of the images in all datasets is 32×32. These datasets include simple handwritten digital images and slightly more complex clothing images, as well as complex natural images and face images. Samples images are shown on the odd rows of column (a) in Fig. 4.

4.3 Reconstruction of visual scene stimuli from RGCs spike trains

In this paper, we present the reconstructed visual scenes using four different methods, and compare them quantitatively and qualitatively. The first method (Method-I) is based on the method of Zhang et al [5]. The model includes two parts and uses the binary cross-entropy loss function. Its model uses a FCN as the spike-image converter, and a CNN as the image-image autoencoder, as shown in Fig 1(C). In the second model (Method-II), we used the same architecture of Zhang et al [5] but trained the model using the proposed loss function (L_{sl}). The third model (Method-III) is based on the first model, and CapsNet is introduced as the feature extractor. The fourth model (Method-IV) represents the proposed model obtained from the combination of different artificial neural networks (i.e., FCN, CapsNet and CNN). Through the controlled variables method, we formed four groups of controlled experiments. The quantitative analysis of Method-I and -II, and Method-III and -IV (i.e., the same model structure but with different loss functions) help in demonstrating the advantages of the proposed loss function in providing better generalization. The comparison of Method-I and -III, and Method-II and -IV (where CapsNet are used in Method-III and -IV) presents the goodness of using CapsNet as an additional feature extractor. The experimental results show that the introduction of CapsNet along with the proposed loss function significantly improves the reconstruction ability of the proposed model. Table 1 shows the components of each method.

Table 1 Components of four different methods.

Method	Method-I	Method-II	Method-III	Proposed
FCN+CNN	✓	✓	✓	✓
FCN+CapsNet			✓	✓
Binary_cross_entropy	✓		✓	
L_{sl}		✓		✓

4.4 Evaluation metrics

SSIM is used for measuring the similarity between two images. It is a full reference metric, i.e., the measurement of image quality is based on an initial uncompressed or distortion-free image (where in this paper the reference is the original visual scene stimuli). It is a perception-based model which considers image degradation as perceived change in structural information, while also incorporating important perceptual phenomena, including both luminance masking and contrast masking terms. Mean squared error (MSE) measures the average squared difference between the estimated values (i.e., the reconstruction) and the actual value (i.e., the original visual scene). Peak signal-to-noise ratio (PSNR) is defined via the MSE, and is commonly used to quantify the reconstruction quality of images and video that are subjected to distortion and lossy compression. Intra-SSIM [23] is a new image quality evaluation method recently proposed by us. It is different from the other three evaluation indices, since the evaluation is based only on the image under test (i.e., no reference image or ground truth is required). Higher SSIM and PSNR (i.e., lower MSE) and higher correlation coefficient (CC) of Intra-SSIM maps between the considered images indicate better quality of image reconstruction. Thus through these four evaluation metrics, we can comprehensively evaluate the reconstructed results.

Intra-SSIM is implemented based on SSIM. To compute Intra-SSIM, the image with the size of $N\times N$ pixels is divided into $n\times n$ cell blocks. Let b $= \{b_m|m = 1,2,...,n\times n-1\}$ be the cell block, and let B $= \{B(i,j)|i,j = 1,2,...,n-3\}$ represent a 3×3 cell blocks where each cell block is $(N/n)\times(N/n)$ pixels. The SSIM index between the central cell block and the remaining cell blocks in eight directions of each $B(i,j)$, i.e., SSIM_0, SSIM_1, SSIM_2, SSIM_3, SSIM_4, SSIM_5, SSIM_6, SSIM_7, are calculated, and the average SSIM values are

$$\overline{B}(i,j) = \frac{1}{8}\sum_{m=0}^{7} \text{SSIM\_}m. \tag{15}$$

Likewise, the sliding window technique with a stride of 1 along the row and column of the image is employed to calculate the Intra-SSIM of the whole image, i.e.,

$$\text{Intra-SSIM} = \frac{1}{(n-2)^2}\sum_{j=0}^{n-3}\sum_{i=0}^{n-3}\overline{B}(i,j). \tag{16}$$

To enable a realistic comparison and provide better clarity, the Intra-SSIM is computed on an image of size 256×256, where the original and reconstructed images are upsampled using bilinear interpolation. Therefore, we divide the image into 32×32 cell blocks, so that each cell block comprises 8×8 pixels. The process to compute Intra-SSIM is illustrated in Fig. 3(a) and (b), and is as follows:

```
Calculate Intra-SSIM of an image
1: Let $k = 32 \cdot j + i$
2: **For** $B(i,j)|_{i=29,j=29}$ **in** image:
3:     $\overline{B}(i,j) = \frac{1}{8}SSIM[b_{k+33},\ (b_k,\ b_{k+1},\ b_{k+2},\ b_{k+32},\ b_{k+34},\ b_{k+64},\ b_{k+65},\ b_{k+66})]$
4: Intra-SSIM = $\frac{1}{900}\sum_{j=0}^{29}\sum_{i=0}^{29}\overline{B}(i,j)$
   **return** Intra-SSIM
```

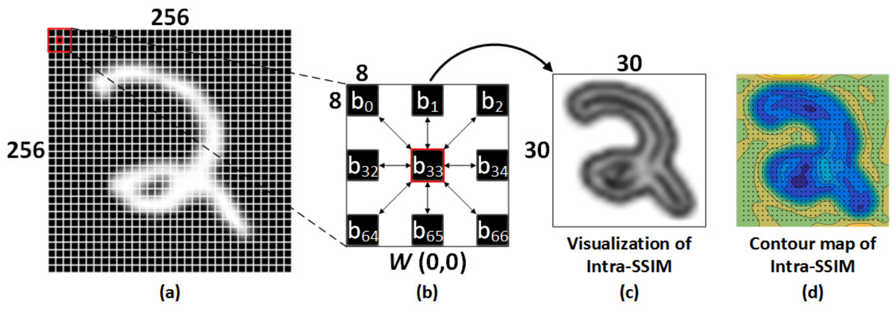

Fig. 3 Process to compute Intra-SSIM: (a) A 256×256 image divided into 32×32 cell blocks. At each step of the sliding window, the mean SSIM is computed for the central block by considering the eight neighbouring blocks as illustrated in (b). Intra-SSIM: (c) Visualization; and (d) Contour map.

The advantages of using Intra-SSIM in evaluating the images include: a) Intra-SSIM provides better understanding of the internal structure of an image (non-referenced) by calculating the similarity from its internal structures. The more complex the structure of the image, the smaller is the score, and vice versa; b) Intra-SSIM can be employed to evaluate the goodness of a non-referenced image, or to obtain a correlation between two associated images by comparing the Intra-SSIM scores of the images. The closer the scores are, the higher is the correlation between them (i.e., the reconstructions and the stimuli); and c) The Intra-SSIM contour maps (e.g., Fig. 3(d)) present visual representations for qualitative comparison of the images.

5 Results and Comparison

The performance of our method was evaluated quantitatively and qualitatively on four datasets. Compared to Method-I, Method-II and Method-III, our results are the best in all aspects of the evaluation.

5.1 Quantitative Analysis

Table 2 shows that the performance of our reconstruction method surpasses the other three methods by a large margin. On five different datasets, our method achieves the best score in each of the evaluation indices, especially for the complex datasets. On the Cifar-10 dataset, compared with Method-I, SSIM, PSNR and CC increased by 0.054, 0.48 and 0.073, respectively, and MSE decreased by 0.0037 for our method; compared with Method-II, SSIM, PSNR and CC increased by 0.037, 0.30 and 0.027, respectively, and MSE decreased by 0.0017 for our method; and compared with Method-III, SSIM, PSNR and CC increased by 0.048, 0.77 and 0.096, respectively, and MSE decreased by 0.0026 for our method. On more complex datasets (i.e., Celeba-HQ and COCO), the improvement of our method is more significant. Compared with Method-I, -II, and -III on COCO dataset, our method improves by 0.0115, 0.061 and 0.036, respectively on SSIM, and 0.181, 0.033 and 0.065 on CC, respectively.

Table 2 Quantitative comparison of the reconstruction results of three methods.

Dataset	Metrics	Method-I	Method-II	Method-III	Proposed
MNIST	SSIM	0.824	0.85	0.831	0.871
	MSE	0.015	0.0134	0.0147	0.0113
	PSNR	18.74	19.32	18.81	20.0
	CC	0.871	0.916	0.890	0.929
Fashion-MNIST	SSIM	0.723	0.748	0.726	0.774
	MSE	0.0168	0.0159	0.0169	0.0143
	PSNR	18.38	18.73	18.37	19.14
	CC	0.672	0.749	0.682	0.778
Cifar-10	SSIM	0.498	0.515	0.525	0.552
	MSE	0.0326	0.0306	0.0339	0.0289
	PSNR	15.61	15.79	15.40	16.09
	CC	0.199	0.245	0.224	0.272
Celeba-HQ	SSIM	0.528	0.601	0.592	0.624
	MSE	0.035	0.0306	0.0322	0.0291
	PSNR	14.97	15.57	15.45	15.83
	CC	0.288	0.393	0.33	0.433
COCO	SSIM	0.346	0.4	0.425	0.461
	MSE	0.0433	0.0378	0.0388	0.0351
	PSNR	14.25	14.72	14.73	15.09
	CC	0.012	0.16	0.128	0.193

* CC computed between the Intra-SSIM maps of the original image and the reconstructed results from Methods I, II, III and Proposed.

Table 3 shows that the Intra-SSIM score of the proposed method is closer to the score of the original visual scene stimulation (also evident from the high CC scores in Table 2) which indicates that the structural relationship of the reconstruction results of the proposed method is indeed similar to the original image.

Table 3 On five datasets, the internal structure similarity between the reconstruction results of the four methods and the original visual scene stimuli is quantitatively compared by using Intra-SSIM.

Dataset	Original	Method-I	Method-II	Method-III	Proposed
MNIST	0.7693	0.7898	0.783	0.7887	0.7825
Fashion-MNIST	0.7216	0.8205	0.7962	0.8181	0.7854
Cifar-10	0.754	0.9773	0.9503	0.9648	0.9463
Celeba-HQ	0.6216	0.9276	0.8571	0.8988	0.833
COCO	0.6471	0.9799	0.9465	0.9755	0.9337

The above quantitative analyses show that employing the proposed loss function along with the introduction of CapsNet to improve the training and generalization ability of the model, resulted in accurate reconstruction of visual

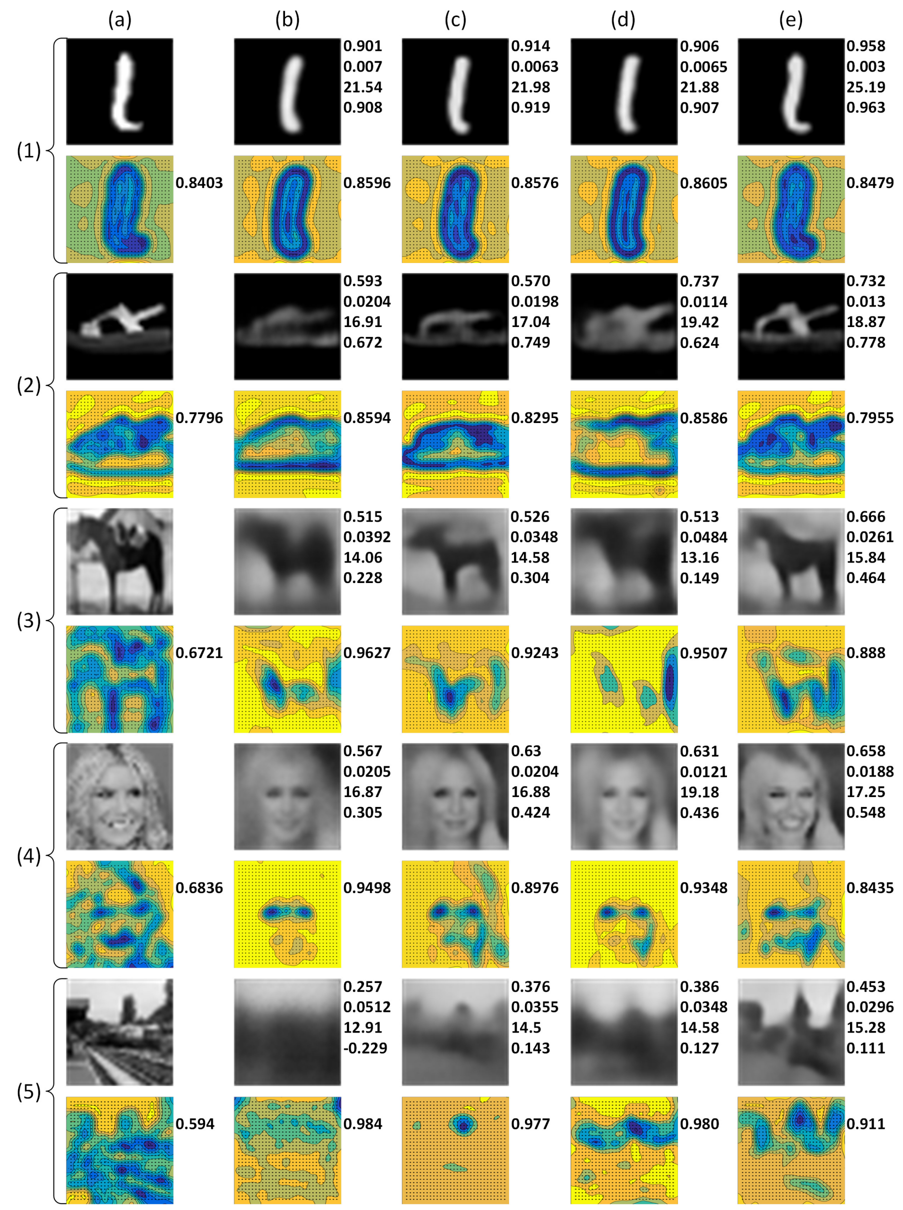

Fig. 4 Groups (1) to (5) refer to MNIST dataset, Fashion-MNIST dataset, Cifar-10 dataset, Celeba-HQ dataset and COCO datasets, respectively. The contour map of Intra-SSIM is displayed below the original images and reconstruction results. The odd rows of Column (a) are the original images; Columns (b) to (e) are the reconstruction results of Method-I, Method-II, Method-III and the proposed method, respectively. The scores of SSIM, MSE, PSNR and CC (CC computed between the Intra-SSIM maps of the original image and the reconstructed results) are indicated on the right side of the reconstructed image and the Intra-SSIM score is indicated on the right side of the contour map.

scenes. Comparing Method-I and -II, and Method-III and -IV (i.e, the model structure is the same but with different loss functions) shows that using the proposed loss function enables the model to improve in all evaluation indices. This clearly shows that the proposed loss function improves and optimizes the model parameters, so as to increase the accuracy of the visual reconstruction. Comparing Method-I and -III, with Method-II and -IV (i.e., introducing CapsNet as the feature extractor), shows that when CapsNet is used as an additional feature extractor, the indices that evaluate the structural similarity (i.e. SSIM and CC) of the reconstructed images have significantly improved. This indicates the goodness of CapsNet in increasing the overall structural accuracy of the reconstruction.

5.2 Qualitative Analysis

Fig. 4 shows the performance comparison for the proposed method with three other methods. It shows that Method-I (refer to column b) can roughly reconstruct simple visual scenes (i.e., handwritten digits and clothes), but fails to deal with complex visual scene stimuli (i.e., natural images and face images). Method-II (refer to column c) used the architecture of Method-I but replaced binary cross-entropy loss function with the proposed loss function (Eq. (14)). Its reconstruction is clearer and the details are more obvious than that of Method-I. However, its performance is still poor for stimuli from complex visual scenes and there is still room for further improvement in the reconstruction of details. Similarly, comparing the results of Method-III and -IV shows that the proposed loss function achieved better generalization during training than the traditional binary cross-entropy loss function.

Comparing Method-I and -III with Method-II and -IV clearly suggests that CapsNet as an additional feature extractor enables more accurate reconstruction, especially on the structure of the image. The reconstructions of the proposed method (Method-IV) are shown in column e of Fig. 4, where the architecture integrates CapsNet as an additional feature extractor and uses the proposed loss function (L_{sl}). Compared with the reconstructions of Method-I, -II and -III, the clarity of the reconstructed visual scenes by the proposed method is significantly improved, with less blurring than Method-I and -II, and provides more clarity than Method–III. The proposed method is excellent in presenting the reconstruction details, particularly in restoring the spatial structure of visual scenes. For example, on the face dataset (refer to Group 4 of Fig. 4), the proposed method reconstructed the facial expressions more closely to the original image. This is evident from the contour maps of Intra-SSIM that clearly show that the reconstructions of the proposed method contain more details that capture the spatial relationship between different regions more precisely. We attribute this improvement to the introduction of CapsNet and the use of L_{sl}, which aids in preserving the spatial links accurately. The proposed method (Method-IV) achieves excellent performance, especially with complex visual scenes under various image categories, such as those in Cifar-10 and COCO datasets.

6 Conclusion

In this paper, we proposed an RGCs spike trains decoder which integrates three ANNs. FCN acts as a spike trains to image converter. CapsNet acts as spatial relation feature extractor. CNN acts as an image to image autoencoder. The three networks complement each other as follows. FCN of the feature extraction module plays a preliminary role in extracting features and other information from the spike trains. To improve the feature extraction process, CapsNet is introduced to extract spatial-relationship features that neither FCN nor CNN can provide. CNN of the reconstruction module has simple encoding and decoding structure, which extracts information, filter noise and accurately reconstruct visual scenes from the spike trains. In addition, we also proposed a new loss function, which combines SSIM loss and L1 loss. The loss function provides: (a) An advantage that SSIM focuses on evaluating structural similarity; and (b) The goodness of L1 in reducing the ambiguity which in turn reduces the blurring effect.

The proposed method, Method-IV, was evaluated on five datasets, using four evaluation parameters to evaluate the reconstruction results quantitatively and qualitatively. The experimental results show that our proposed method greatly improves the reconstruction ability of the model, presenting higher clarity, richer details and more accurate spatio-structural relationship. Our contributions include:

- Combining different ANNs that achieve good results, introducing new research ideas.
- Combining SSIM loss and L1 loss to improve the generalization ability of the model, thereby improving the accuracy of the scene reconstructions.
- The proposed RGCs spike trains decoder forms a complete system that decodes the encoded spike trains, paving the way for better understanding of the working principle of the retina and the dynamics of the brian.
- The proposed method provides a new method for reconstructing visual scenes from RGCs spike trains, and promotes the development of computer vision and brain-machine interface. For example, it can improve the reconstruction quality of bionic retinal camera and helps the visually impaired recover their vision.

Although our method has achieved good results at this stage, the process of reconstructing visual scene stimuli from RGCs spike trains is still a challenge. In the future work, we will further improve the model from the following aspects to make it closer to the functioning of the human eye: a) Improve the resolution of reconstructed visual scenes and maintain the quality of the reconstruction; b) Transform grey-scale images to colour images to restore the real scenes as seen by human eyes; c) Propose models for dynamic video, achieve the reconstruction at real-time, and restore the visual scenes more realistically; d) Employ naturally recorded spikes rather than the simulated RGCs spike trains to provide model and results closer to practical applications; and e) The use of temporal filter to integrate a spatial filter to closely represent the brain's processing mechanism of spike trains. In short, we hope that our work will inspire other researchers and jointly promote the development of retinal visual scene reconstruction.

Acknowledgements This research was financially supported by the Scientific Research Grant of Shantou University, China, Grant (No. NTF17016) and National Natural Science Foundation of China (No. 82071992).

Conflict of interest

The authors declare that they have no conflict of interest.

References

1. Schwemmer MA, Skomrock ND, Sederberg PB, Ting JE, Sharma G, Bockbrader MA, Friedenberg DA (2018) Meeting brain-computer interface user performance expectations using a deep neural network decoding framework. Nature medicine 24(11):1669-76 https://doi.org/10.1038/s41591-018-0171-y
2. Hutmacher F (2019) Why is there so much more research on vision than on any other sensory modality? Frontiers in psychology. https://doi.org/10.3389/fpsyg.2019.02246
3. Epstein, Russell A and Baker, Chris I (2019) Scene perception in the human brain. Annual review of vision science 5:373-97. https://doi.org/10.1146/annurev-vision-091718-014809
4. Langer, Kirstin B and Ohlemacher, Sarah K and Phillips, M Joseph and Fligor, Clarisse M and Jiang, Peng and Gamm, David M and Meyer, Jason S (2018) Retinal ganglion cell diversity and subtype specification from human pluripotent stem cells. Stem cell reports 10(4):1282-1293. https://doi.org/10.1016/j.stemcr.2018.02.010
5. Zhang Y, Jia S, Zheng Y, Yu Z, Tian Y, Ma S, Huang T, Liu JK (2020) Reconstruction of natural visual scenes from neural spikes with deep neural networks. Neural Networks 125:19-30. https://doi.org/10.1016/j.neunet.2020.01.033
6. Grimes WN, Songco-Aguas A, Rieke F (2018) Parallel processing of rod and cone signals: retinal function and human perception. Annual review of vision science 4:123-41. https://doi.org/10.1146/annurev-vision-091517-034055
7. O'Brien J, Bloomfield SA (2018) Plasticity of retinal gap junctions: roles in synaptic physiology and disease. Annual review of vision science 4:79-100. https://doi.org/10.1146/annurev-vision-091517-034133
8. Rivlin-Etzion M, Grimes WN, Rieke F (2018) Flexible neural hardware supports dynamic computations in retina. Trends in neurosciences 41(4):224-37. https://doi.org/10.1016/j.tins.2018.01.009
9. Demb JB, Singer JH (2015) Functional circuitry of the retina. Network: computation in neural systems 12(2):199. https://doi.org/10.1146/annurev-vision-082114-035334
10. Chichilnisky EJ (2001) A simple white noise analysis of neuronal light responses. Frontiers in systems neuroscience 10:109.
11. Pillow JW, Shlens J, Paninski L, Sher A, Litke AM, Chichilnisky EJ, Simoncelli EP (2008) Spatio-temporal correlations and visual signalling in a complete neuronal population. Nature 454(7207):995-9. https://doi.org/10.1038/nature07140
12. Cessac B, Kornprobst P, Kraria S, Nasser H, Pamplona D, Portelli G, Viéville T (2017) PRANAS: a new platform for retinal analysis and simulation. Frontiers in Neuroinformatics 11:49. https://doi.org/10.3389/fninf.2017.00049

13. Meyer AF, Williamson RS, Linden JF, Sahani M (2017) Models of neuronal stimulus-response functions: elaboration, estimation, and evaluation. Frontiers in systems neuroscience 10:109. https://doi.org/10.3389/fnsys.2016.00109
14. Botella-Soler V, Deny S, Martius G, Marre O, Tkačik G (2018) Nonlinear decoding of a complex movie from the mammalian retina. PLoS computational biology 14(5):e1006057. https://doi.org/10.1371/journal.pcbi.1006057
15. Pillow JW, Paninski L, Uzzell VJ, Simoncelli EP, Chichilnisky EJ (2005) Prediction and decoding of retinal ganglion cell responses with a probabilistic spiking model. Journal of Neuroscience 25(47):11003-13. https://doi.org/10.1523/JNEUROSCI.3305-05.2005
16. Parthasarathy N, Batty E, Falcon W, Rutten T, Rajpal M, Chichilnisky EJ, Paninski L (2017) Neural networks for efficient bayesian decoding of natural images from retinal neurons. Advances in Neural Information Processing Systems 30:6434-45. https://doi.org/10.1101/153759
17. Sabour S, Frosst N, Hinton G.E (2017) Dynamic routing between capsules. Advances in Neural Information Processing Systems pp. 3859-3869.
18. Roy K, Roy P, Chaudhuri SS (2021) Capsule Neural Network Architecture Based Multi-class Fruit Image Classification. In Advances in Smart Communication Technology and Information Processing: OPTRONIX 2020 165:171-180.
19. Wang Z, Simoncelli EP, Bovik AC (2003) Multiscale structural similarity for image quality assessment. In The Thrity-Seventh Asilomar Conference on Signals, Systems Computers,2003 2:1398-1402.
20. Pathak D, Krahenbuhl P, Donahue J, Darrell T, Efros AA (2016) Context encoders: Feature learning by inpainting. In Proceedings of the IEEE conference on computer vision and pattern recognition pp. 2536-2544.
21. Isola P, Zhu JY, Zhou T, Efros AA (2017) Image-to-image translation with conditional adversarial networks. InProceedings of the IEEE conference on computer vision and pattern recognition pp. 1125-1134.
22. Gadirov H (2018) Capsule architecture as a discriminator in generative adversarial networks. MS thesis.
23. Li W, Raj AN, Tjahjadi T, Zhuang Z (2021) Digital hair removal by deep learning for skin lesion segmentation. Pattern Recognition 117:107994. https://doi.org/10.1016/j.patcog.2021.107994
24. Onken A, Liu JK, Karunasekara PC, Delis I, Gollisch T, Panzeri S (2016) Using matrix and tensor factorizations for the single-trial analysis of population spike trains. PLoS computational biology 12(11):e1005189. https://doi.org/10.1371/journal.pcbi.1005189
25. Brackbill N, Rhoades C, Kling A, Shah NP, Sher A (2020) Reconstruction of natural images from responses of primate retinal ganglion cells. Elife 9:e58516. https://doi.org/10.7554/eLife.58516
26. Kim YJ, Brackbill N, Batty E, Lee J, Mitelut C, Tong W, Chichilnisky EJ, Paninski L (2021) Nonlinear decoding of natural images from large-scale primate retinal ganglion recordings. Neural Computation 33(7):1719-50. https://doi.org/10.1162/neco_a_01395
27. Odermatt B, Nikolaev A, Lagnado L (2012) Encoding of luminance and contrast by linear and nonlinear synapses in the retina. Neuron 73(4):758-73. https://doi.org/10.1016/j.neuron.2011.12.023
28. Benoit A, Caplier A, Durette B, Hérault J (2010) Using human visual system modeling for bio-inspired low level image processing. Computer vision and Image understanding 114(7):758-73. https://doi.org/10.1016/j.cviu.2010.01.011
29. Martinez-Alvarez A, Olmedo-Payá A, Cuenca-Asensi S, Ferrández JM, Fernandez E (2013) RetinaStudio: A bioinspired framework to encode visual information. Neurocomputing 114:45-53. https://doi.org/10.1016/j.neucom.2012.07.035
30. Nasser H, Kraria S, Cessac B (2013) EnaS: a new software for neural population analysis in large scale spiking networks. BMC Neuroscience 14(1):1-2. https://doi.org/10.1186/1471-2202-14-S1-P57
31. Cofre R, Cessac B (2014) Exact computation of the maximum-entropy potential of spiking neural-network models. Physical Review E 89(5):052117. https://doi.org/10.1103/PhysRevE.89.052117
32. Gollisch T, Meister M (2010) Eye smarter than scientists believed: neural computations in circuits of the retina. Neuron 65(2):150-64. https://doi.org/10.1016/j.neuron.2009.12.009

33. Wohrer A, Kornprobst P (2009) Virtual retina: a biological retina model and simulator, with contrast gain control. Journal of computational neuroscience, 26(2), 219-249. https://doi.org/10.1007/s10827-008-0108-4