




# Retinal spike train decoder using vector quantization for visual scene reconstruction

Kunwu Ma<sup>1</sup> · Alex Noel Joseph Raj<sup>1</sup> · Vijayarajan Rajangam<sup>2</sup> · Tardi Tjahjadi<sup>3</sup> · Minying Liu<sup>1</sup> · Zhemin Zhuang<sup>1</sup> 

Received: 20 July 2023 / Accepted: 23 December 2023  
© The Author(s) 2024

## Abstract

The retinal impulse signal is the basic carrier of visual information. It records the distribution of light on the retina. However, its direct conversion to a scene image is difficult due to the nonlinear characteristics of its distribution. Therefore, the use of artificial neural network to reconstruct the scene from retinal spikes has become an important research area. This paper proposes the architecture of a neural network based on vector quantization, where the feature vectors of spike trains are extracted, compressed, and stored using a feature extraction and compression network. During the decoding process, the nearest neighbour search method is used to find the nearest feature vector corresponding to each feature vector in the feature map. Finally, a reconstruction network is used to decode a new feature map composed of matching feature vectors to obtain a visual scene. This paper also verifies the impact of vector quantization on the characteristics of pulse signals by comparing experiments and visualizing the characteristics before and after vector quantization. The network delivers promising performance when evaluated on different datasets, demonstrating that this research is of great significance for improving relevant applications in the fields of retinal image processing and artificial intelligence.

**Keywords** Retinal impulse signal · Artificial neural network · Vector quantization · Embedded codebook

## Introduction

Retinal ganglion cells (RGCs) play a crucial role in transmitting visual information from the retina to the central nervous system [1]. Located near the inner surface of the retina, these cells receive visual inputs from photoreceptors and propagate the signals to the brain through intermediate neurons, such as bipolar cells and retinal longitudinally free cells [2]. The resulting spike trains, representing the neural activity patterns generated by visual stimuli, are of great interest in the field of retinal spike decoding [3]. The ability to accurately reconstruct visual scenes from retinal spike trains has significant implications for understanding visual perception

and developing medical interventions for visual impairments [4].

Despite recent advances in retinal imaging techniques [5], conventional spike decoding techniques remain inadequate for accurately reconstructing visual scenes from retinal data. While they can detect and analyze spike activity in the retina [6], decoding the complex relationship between retinal spikes and visual stimuli remains a challenging task. Moreover, the challenge is compounded by the specificity and information selectivity of RGCs, which have complex coding rules that are selectively computed only for specific stimulus features [7]. The highly nonlinear processing rules precisely shape retinal narratives and require more sophisticated and advanced techniques to unlock their attributes.

Artificial neural networks (ANNs) [8] such as convolutional neural network (CNN), recurrent neural network (RNN), spiking neural network (SNN), and machine learning models are inspired by the structure and function of the biological brain [9] to provide a potential solution to this problem. The machine learning models are well suited to decode retinal spike trains due to their ability to learn the complex relationship between the input signal and the target output. Additionally, vector quantization (VQ) [10]

✉ Zhemin Zhuang  
zmzhuang@stu.edu.cn

<sup>1</sup> Engineering Technology Research Center of Artificial Intelligence and Modern Ultrasonic of Guangdong Province, Department of Electronic Engineering, College of Engineering, Shantou University, Shantou, China

<sup>2</sup> Centre for Healthcare Advancement, Innovation and Research, School of Electronics Engineering, Vellore Institute of Technology Chennai, Chennai, India

<sup>3</sup> School of Engineering, University of Warwick, Coventry, UK

is an established technique for compressing data. VQ provides an efficient way to represent scalar datasets as vectors, which can then be quantized in the vector space, resulting in data compression without significant loss of information. Based on this idea, our paper presents a novel neural network framework called the vector quantization fully connected convolutional decoding network (VQ-FCDnet), designed for decoding retinal spike trains. This model achieves outstanding efficacy and superior quality in reconstructing natural visual scenes from spikes of retinal ganglion cells (RGCs) that have been recorded in the retina.

Our proposed framework comprises three major steps. First, during the encoding process, the visual scene is represented as spike trains using either retinal cells or simulation software. Second, the spike trains are converted into feature maps using a fully connected network [11], followed by convolutional operations that further extract and compress the features. Finally, during the decoding process (i.e., spikes to image), the nearest neighbour search method is employed to find the closest embedding vector in the feature maps corresponding to common embedding codebooks, and thus derive a new feature map. The feature map is then decoded using convolutional neural networks [12] and transposed convolutional neural networks [13] to reconstruct the visual scene.

The main contributions of this paper are as follows:

- We propose a new neural network model for decoding retinal spikes trains. The network has a simple structure and directly reconstructs visual scenes from retinal spike trains. It achieves higher scores than other network architectures in evaluation indicators like peak signal to noise ratio (PSNR), structural similarity index (SSIM) and mean square error (MSE).
- The proposed structure of the network based on vector quantization aggregates similar features of spike trains and distributes dissimilar features to reconstruct visual scenes, providing a better scheme for recreating visual scenes from spike trains.
- In the proposed network, the retinal spike trains are not directly decoded, but the most similar features are found in the trained embedded codebook with all the features for decoding, which increases the stability and anti-interference of the network.

The paper is organized as follows. “Related work” reviews the related work. “Proposed method” describes the detailed process of using VQ-FCDnet decoding and the loss function for training the network. “Experimental results and comparative analysis” presents the evaluation criteria, experimental configurations, data sets, and results, as well as a comparison with the first step method for reconstructing visual scenes from RGCs spike trains. “Future scope” suggests future work and “Conclusion” concludes the paper.

## Related work

In general, an ideal visual neuron decoder [14] should be able to reconstruct stimuli from neural responses to clearly restore the visual scene. However, the reconstruction of visual scenes from visual neuron spike trains is a complex and difficult task. Neurons generate noise when processing visual information [15], so each spike in a spike train contains noise. These noises make accurate reconstruction of visual information more difficult [16]. During the transmission of the spike trains, some of the spikes in the spike trains may be lost due to the complex connection between neurons. The lost spikes may contain critical information, which can also interfere with the reconstruction of visual information.

There are traditional and neural network-based methods for reconstructing visual scenes with retinal neuron decoders. Studies related to traditional methods are as follows. Pillow et al. [17] proposed a model consisting of a linearly filtered stimulus-driven leaky integrated-fire pulse generator, post-pulse currents, and Gaussian noise currents. This model can be used to derive an explicit maximum likelihood decoding rule [18] for neural spike training and primate RGC light responses with stimulus selectivity. Ariadna et al. [19] combined multi-electrode array (MEA) and a software capable of characterizing and grouping spikes based on principal component analysis (PCA) [20]. They also used different clustering algorithms to localize the response of moving stripes crossing the visual field in eight orthogonal directions. The receptive field of each cell was used to reconstruct the complex visual stimuli. However, the reconstructed scenes are very blurred, and only greyscale images can be reconstructed.

Related studies on visual scene reconstruction using neural networks for retinal pulse signals are as follows. Kim et al. [21] combined a low-pass linear decoder and a high-pass non-linear decoder to obtain preliminary reconstruction results, which were then fed into a neural network to reconstruct visual scenes. The modified approach could only reconstruct simple visual scenes such as bicycle tyres, cylinders, and simple black-and-white textures. Zhang et al. [22] designed a SID model including a spike-to-image converter and an image-to-image autoencoder to implement an end-to-end decoder from neural spikes to images to reconstruct visual scenes directly from spike signals. By combining a fully connected network (FCN), a capsule network (CapsNet) [23], Li et al. [24] designed a structural similarity index metric based on SSIM and L1 loss function for retinal spike train decoder. These images are better than those generated by previous methods but suffer from blurring. Although visual scene reconstruction has been studied for many years, the decoding performance of existing methods still needs further improvement, especially in complex visual scene reconstruc-

tion. Thus, scene neural decoding is still a challenge that needs further development and innovation.

VQ is a lossy data compression method based on block coding rules. Its basic idea is to compress data without losing much information by transforming a number of scalar datasets into a vector and then performing an overall quantization in vector space. The vector quantization variant autoencoder (VQVAE) [25] based on the concept of VQ is an advanced framework for image generation similar to the traditional autoencoder and variant autoencoder (VAE) [26], but with the addition of a quantization step that maps continuous values to discrete codes. This quantization step improves the ability of the network to compress the data, while the variable component ensures that the generated output remains of high quality. By using two scales of features for quantization, VQVAE-2 [27] models the local and global features of an image, with bottom-level features used to extract local information and top-level features used to extract global information. This enables the generation of larger and clearer images. Inspired by VQVAE, we also use VQ to quantify the features extracted from RGC spike trains.

## Proposed method

Figure 1 illustrates the comprehensive process of encoding visual scene stimuli to reconstruct visual scenes. Initially, the input image is encoded into RGC Spike through an encoder, such as retinal cells or retinal simulation software (refer to “Retinal spike dataset” for more information). Subsequently, the RGC spike undergoes a flatten operation and is then fed to our novel VQ-FCDnet which produces the decoder image. In this section, we present an in-depth analysis of the architecture of our proposed VQ-FCDnet, elucidating the function of each module, the loss function, and the training method.

## Network architecture

The architecture of VQ-FCDnet is mainly composed of three blocks (as shown in Fig. 1): feature extraction and compression network (FECN); vector quantization layer; and reconstruction network (REN).

### FECN module

The FECN module, as shown in Fig. 2, performs extraction and compression of feature maps from the spike signals. It consists of fully connected feature extraction block (FEB) and convolutional feature extraction and compression block (CECB). FEB performs shallow extraction of spike signals through four fully connected layers, and its structure is shown in Fig. 3. The size of the one-dimensional features generated by these four fully connected layers is 8192, 4096, 8192, and

16,384. We employ the LeakyReLU activation function in the first three layers to enhance the nonlinearity of the feature extraction process. This function allows the network to capture more complex relationships between the input spike signals and the extracted features, thereby improving the representational power of the network.

If the size of the input spike signal after the flatten operation is 10,000, the output expression for FEB at layer  $i$  is expressed as

$$O_{FEB}^{[i]} = \psi_{(\alpha)}^{[i]} \left( W^{[i]} O_{FEB}^{[i-1]} + b^{[i]} \right), \tag{1}$$

where  $O_{FEB}^{[i-1]}$  represents the output of layer  $i - 1$  of the FEB.  $\psi_{(\alpha)}^{[i]}$  is LeakyReLU activation function of the  $i$ -th layer, where  $\alpha$  is the negative slope of LeakyReLU.  $W^{[i]}$  and  $b^{[i]}$  are, respectively, the weight matrix and bias vector of the  $i$ -th layer full connection.

As the output,  $O_{FEB(16384)}$ , of the FEB represents the output of the first four layers specifically, the fourth layer does not use the activation function. Hence,  $\psi_{(0.2)}^{[4]} = 1$ . This is mathematically represented as

$$O_{FEB(16384)} = O_{FEB}^{[4]} = \psi_{(0.2)}^{[4]} \left( W^{[4]} O_{FEB}^{[3]} + b^{[4]} \right). \tag{2}$$

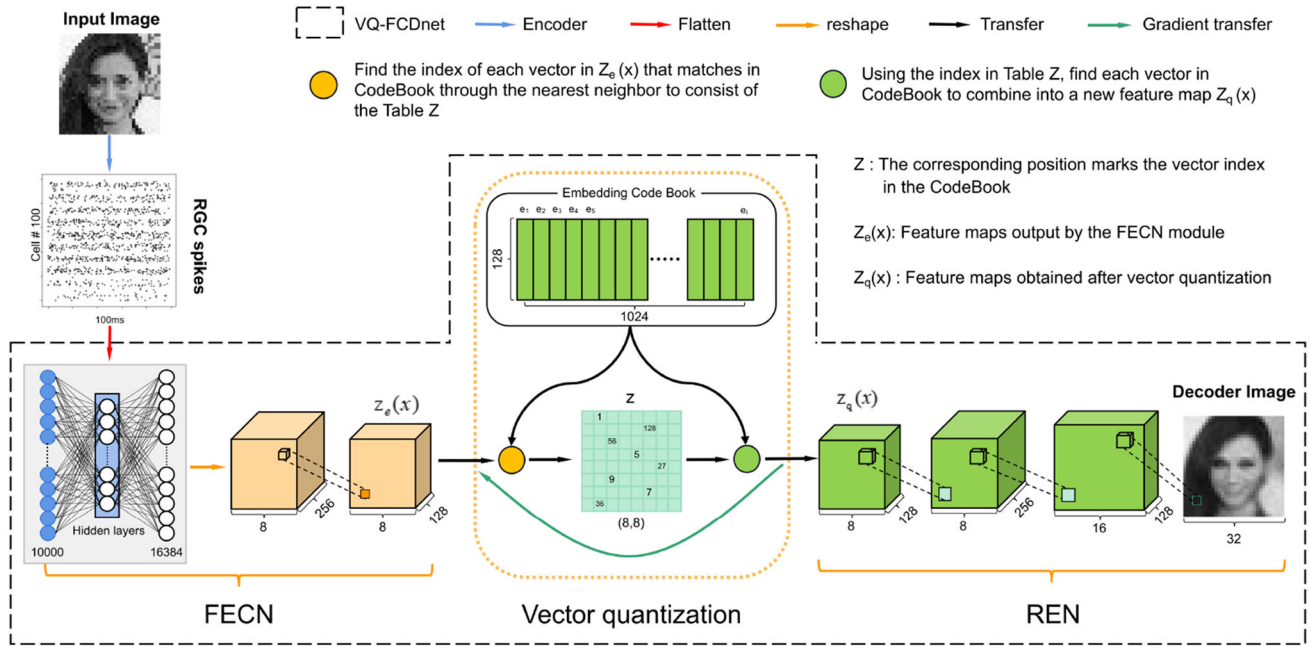
Convolutional feature extraction and compression blocks use two residual blocks (ResBlock) [28] and a channel attention module (CAM) to finely extract feature maps of size (256, 8, 8) from the feature  $O_{FEB(16384)}$ . The convolutional layer then compresses this 256 channel feature map into a 128 channel feature map. The structure of this block is shown in Fig. 4.

Residual block is a jump-connected convolutional neural network that preserves pre-convolutional features on the basis of extracted features. It consists of two convolutional modules and a hop connection, with convolutional core sizes of  $3 \times 3$  and  $1 \times 1$ , respectively. The ResBlock function is mathematically expressed as

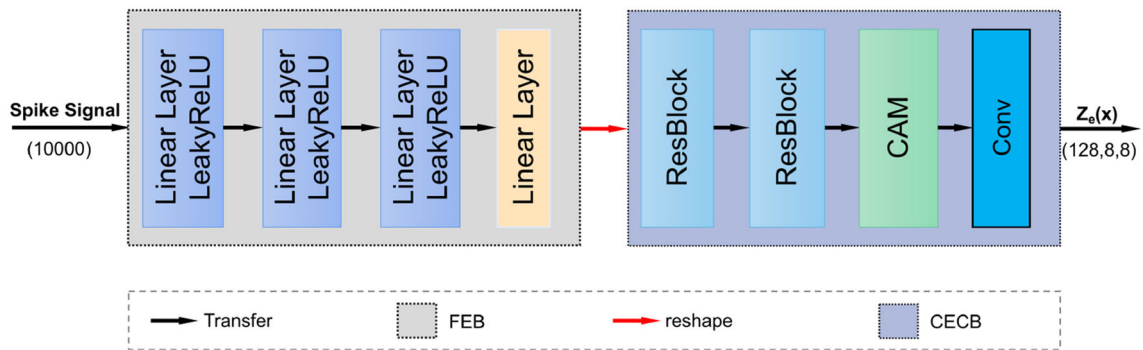
$$Res_{(c,h)}(x) = \phi(\phi(x) \odot K_{(3 \times 3,h,1,1)}) \odot K_{(1 \times 1,c,1,0)} + x, \tag{3}$$

where  $Res_{(c,h)}(x)$  represents the output of ResBlock,  $c$  is the number of channels in the input and output feature maps,  $h$  is the number of channels that hide the convolution layer, and  $x$  is the input feature maps.  $\phi$  is the activation function ReLU [29] and  $\odot$  is the convolution.  $K_{(3 \times 3,h,1,1)}$  and  $K_{(1 \times 1,c,1,0)}$  are, respectively, convolution kernels with size  $3 \times 3$  and  $1 \times 1$ , depths  $h$  and  $c$ , stride 1 and 1, padding 1 and 0.

CAM [30] is a module that enhances the feature representation capabilities of each channel by learning the correlation between channels, thereby improving the network performance. First, it compresses the feature map into two vectors

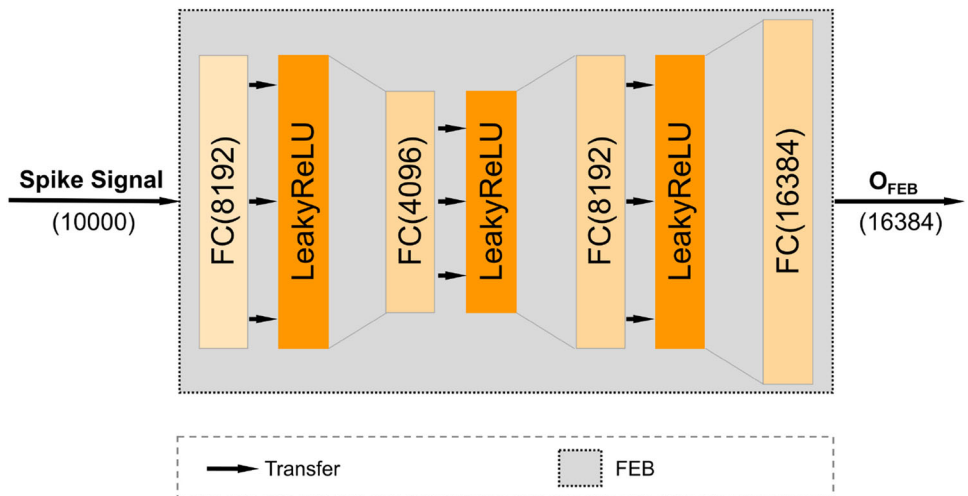


**Fig. 1** Process description: the image is encoded into a spike trains by retinal cells, and the spike trains is sent into VQ-FCDnet, where the decoded image is obtained through VQ-FCDnet decoding



**Fig. 2** FECN module

**Fig. 3** Structure of FEB



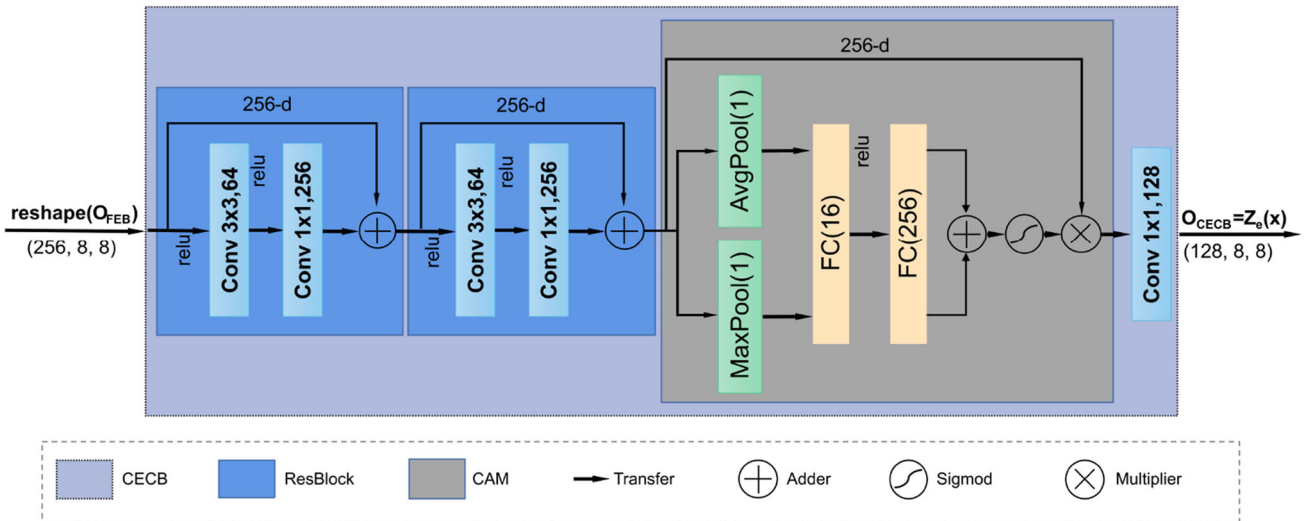


Fig. 4 Structure of CECEB

through global average pooling and global maximum pooling, respectively. A two-layer fully connected network is then used to perform a nonlinear transformation on the two vectors to obtain two weight vectors. The two weight vectors are added, and a sigmoid activation function is used to obtain the weight vectors between different channels of the feature map. Finally, the weighted feature map is obtained by multiplying the feature map with the weight vectors. The operation of CAM is mathematically expressed as

$$CAM_{(a,r)}(x) = \theta((W_2\phi(W_1\alpha(x) + b_1) + b_2) + (W_2\phi(W_1\beta(x) + b_1) + b_2)) * x, \quad (4)$$

where  $CAM_{(a,r)}(x)$  represents the output of CAM,  $a$  represents the number of channels in the input and output feature maps, and the output of the second fully connected layer.  $r$  represents the output of the first fully connected layer,  $x$  is input feature maps,  $\theta$  is the activation function sigmoid,  $\phi$  is the activation function ReLU,  $*$  is the multiplier,  $\alpha$  is average pooling and  $\beta$  is max pooling.  $W_1, W_2$  are the weight matrices, and  $b_1, b_2$  are the bias vectors of the two fully connected layers. Therefore, if  $\gamma$  is a reshape parameter, the process of CECEB is given by

$$\begin{aligned} Z_e(x) &= O_{CECB(128,8,8)} \\ &= CAM_{(256,16)}(Res_{(256,64)} \\ &\quad (Res_{(256,64)}(\gamma(O_{FEB(16384)})))) \odot K_{(1 \times 1, 128, 1, 0)}. \end{aligned} \quad (5)$$

Through these two blocks, the input spike trains is converted into a feature maps  $Z_e(x)$ .

### Vector quantization

Vector quantization is the discrete quantization of input feature maps into feature maps composed of vectors in a codebook. First, we define a latent embedding codebook of dimension  $[1024 \times 128]$  where 1024 is the number of embeddings and 128 is the dimensionality of each latent embedding vector. Thus, the codebook  $E = [e_1, e_2, \dots, e_i] \in R^{K \times D}$ .

Next, we use the feature map  $Z_e(x)$  (a 128-dimensional vector of size  $(8 \times 8)$ ) extracted from the FECN. We use the nearest neighbour search method to find the nearest embedding vector  $e_i$  in the embedded codebook for the  $8 \times 8 = 64$  128-dimensional vectors and use its index to express it, resulting in the index table  $Z$  (with a size of  $(8 \times 8)$  as shown in Fig. 1).

Finally, we find the  $8 \times 8 = 64$  embedded vectors  $e_i$  corresponding to the index table  $Z$  in the codebook, replace the 64 vectors of the feature map  $Z_e(x)$  according to the position of the index table  $Z$ , and obtain the maps  $Z_q(x)$  to be reconstructed later. The vector quantization algorithm is presented as the following pseudo code:

**Algorithm 1** Vector quantization

---

**Require:** Feature maps:  $Z_e(x)$ (128, 8, 8), Embedding Codebook:  $E$ (1024, 128)  
**Ensure:** Feature maps:  $Z_q(x)$ (128, 8, 8)  
1: New  $Z(8, 8)$ ,  $Width = 8$ ,  $Height = 8$ ,  $Book\_Length = 1024$   
2: **for**  $i = 0 \rightarrow Width - 1$  **do**  
3:   **for**  $j = 0 \rightarrow Height - 1$  **do**  
4:      $Z[i : j] = \operatorname{argmin} \|Z_e(x)[i : j] - E[k]\|_2, k \in [0, 1, 2, 3, \dots, Book\_Length - 1]$   
5:   **end for**  
6: **end for**  
7: New  $Z_q(x)$ (128, 8, 8)  
8: **for**  $i = 0 \rightarrow Width - 1$  **do**  
9:   **for**  $j = 0 \rightarrow Height - 1$  **do**  
10:      $Z_q(x)[i : j] = E[Z[i : j]]$   
11:   **end for**  
12: **end for**  
13: **return**  $Z_q(x)$

---

**REN module**

The REN module reconstructs a feature map into a visual scene. The module (shown in Fig. 5) consists of two main blocks: convolutional restore block (CRB) and transposed convolution reconstruction block (TRB).

The CRB restores the input feature map  $Z_q(x)$  to a feature map  $O_{CRB}$  through a convolutional layer and two residual blocks. It enhances the expression ability of feature maps. The steps involved can be mathematically expressed as

$$O_{CRB(256,8,8)} = Res_{(256,64)}(Res_{(256,64)}(Z_q(x) \odot K_{(3 \times 3, 256, 1, 1)})). \quad (6)$$

The TRB is the last component of our proposed VQ-FCDnet, which employs two transposed convolutions to reconstruct the visual scene. It plays a vital role in the overall architecture, as it is responsible for mapping the feature map to its corresponding target values. By doing so, it is able to provide a high-quality visual representation of the scene, thus enhancing the overall performance of the network. The attention to detail in this module ensures that the spatial location of each feature map is precisely aligned, resulting in a more accurate and realistic reconstruction of the visual scene. The role of TRB is expressed as

$$O_{image(1,32,32)} = O_{TRB(1,32,32)} = \phi(O_{CRB(256,8,8)} \odot T_{(4 \times 4, 128, 2, 1)} \odot T_{(4 \times 4, 1, 2, 1)}), \quad (7)$$

where  $\odot$  is transposed convolution, and  $T_{(4 \times 4, 128, 2, 1)}$  and  $T_{(4 \times 4, 1, 2, 1)}$ , respectively, represent convolutional kernels with size of  $4 \times 4$  and  $4 \times 4$ , depths of 128 and 1, stride of 2 and 2, and padding of 1 and 1.

**Loss function**

Similar to VQVAE, the total loss of VQ-FCDnet is due to reconstruction loss, codebook loss, and commitment loss. Reconstruction loss is used to optimize the FECN and REN modules. Due to the non-differentiability of the argmin operation, the gradient of reconstruction error cannot be transmitted to the FECN. We use a straight-through estimator to solve this problem. We directly use the gradient of  $Z_e(x)$  as the gradient of  $Z_q(x)$  and the reconstruction loss is given by

$$L_{re} = \log p(x|Z_q(x)). \quad (8)$$

Although the gradient of the reconstruction error is transmitted to the encoder by the straight-through estimation method [31], the embedding vector  $e_k$  cannot receive the gradient of the reconstruction error band, which also means that the embedding codebook cannot participate in learning.

The codebook loss is determined using a simple dictionary learning method [32], which calculates the L2 error [33] of the output  $Z_e(x)$  of the FECN and the corresponding quantized embedding vector  $e_k$ . However, to stabilize the training, improve the performance and generalization ability of the model, and prevent the occurrence of problems such as gradient explosion or disappearance during the training process, we use exponential moving average (EMA) [34] to update the codebook independently.

As the training is based on mini-batch, the updated mathematical expression for embedding the codebook is presented as

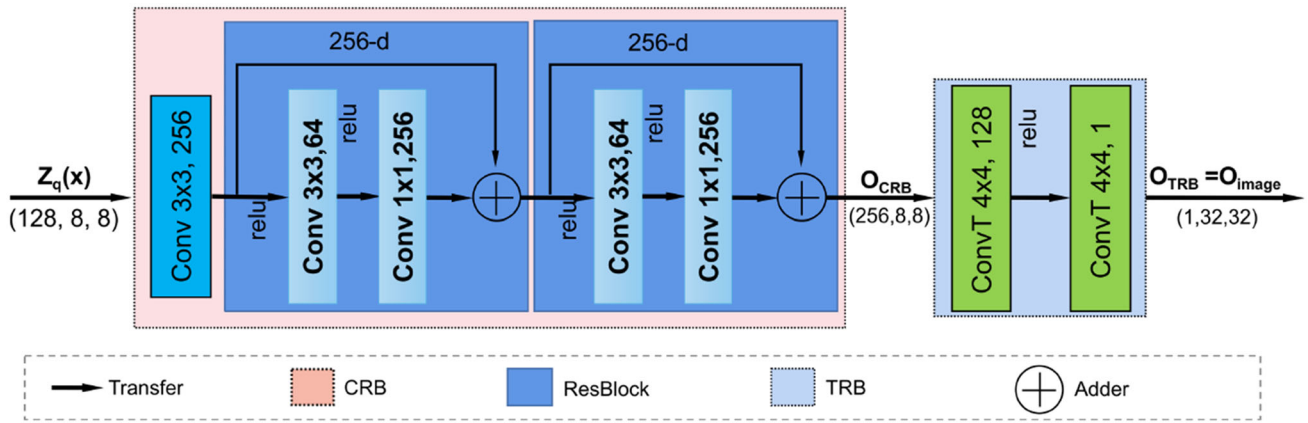


Fig. 5 REN architecture

$$e_i^{(t)} = \frac{m_i^{(t)}}{N_i^{(t)}} = \frac{\lambda m_i^{(t-1)} + (1 - \lambda) \sum_{j=1}^{n_i^{(t)}} z_{i,j}^{(t)}}{\lambda N_i^{(t-1)} + (1 - \lambda) n_i^{(t)}}, \quad (9)$$

where  $e_i$  is a vector in codebook,  $(t)$  and  $(t - 1)$ , respectively, denote the current and previous time instances,  $m_i$  is the sum of  $z_{i,1}, z_{i,2}, \dots, z_{i,n_i}$ ,  $N_i$  is the corresponding number of  $z_{i,1}, z_{i,2}, \dots, z_{i,n_i}$  elements for each embedding vector  $e_i$ , and  $\lambda$  is the decay value that takes a value of 0.99 in this experiment.

Commitment loss [35] is mainly used to constrain the consistency of the FECN output and the embedded codebook to avoid significant changes in the FECN output. Commitment loss directly computes the L2 error of the FECN output  $Z_e(x)$  and the corresponding quantized embedding vector  $e_k$ , i.e.,

$$L_{com} = ||Z_e(x) - sg[e]||_2^2, \quad (10)$$

where  $sg$  refers to a stop-gradient operation that blocks gradients from flowing through  $e$ .

Therefore, the overall training objective is

$$L = \log p(x|Z_q(x)) + \beta ||Z_e(x) - sg[e]||_2^2. \quad (11)$$

The reconstruction loss is applied to FECN and REN, and the commitment loss is used to constrain FECN. Here,  $\beta$  is the weighting coefficient, which is set as 0.25. EMA updates the codebook independently regardless of the type of optimizer and update rules.

### Training model

The training model uses Adam optimizer and L as the loss function. The learning rate is 0.0001, and the training is terminated after 50 consecutive iterations when val-loss (L) is

no longer reduced, i.e., it is inferred that the model has been trained to the optimal level.

## Experimental results and comparative analysis

This section presents the dataset, retinal spike generation software, and the experiments conducted for performance analysis.

### Retinal spike dataset

The dataset consists of salamander retinal ganglion cell responses to natural images. The dataset was built from Liu et al. [36] collection and contains multi-electrode array recordings of retinal ganglion cell spike activity measured in isolated salamander retinas. The stimuli were the sequences of 300 natural images plus 1 black screen (− 100% contrast), 1 grey screen (0% contrast), and 1 white screen (+ 100% contrast). The dataset contains a  $156 \times 303 \times 13 \times 300$  four-dimensional binary matrix of 0s and 1s corresponding to “spikes” (“1”) or “no spikes” (“0”). The dimensions of the matrix correspond to 156 cells, 303 images, 13 trials, and 300 time boxes. In this paper, we select 20 spike trains from different images in one of the 13 experiments as the test set, and the remaining sequences as the training set.

### Retinal simulation software: PRANAS

PRANAS [37] is a powerful retinal simulation software that provides pre-set retinal profiles to configure the corresponding retinal parameters. Once the visual scene is input into the software, the corresponding spike trains are generated from the simulated retina. In this paper, we choose the default setting. The stimulus duration for each image is 100 ms and the response of 100 neurons in 1 ms is considered.

Multiple datasets were used for experiments, each with 30,000 images corresponding to spike trains of 100 neuronal cells within 100 ms, i.e., (30000,100,100). 28,000 images and their spike trains were used as training set and 2000 were used for testing. The datasets include simple MNIST and slightly complex Fashion-MNIST, as well as complex Cifar-10, coco and Celeba-HQ.

## Configuration for experiments

We implemented the VQ-FCDnet model on PyTorch version 1.8.1 and executed Python 3.7 in the background. Experiments were performed on a workstation configured with an Intel Xeon Gold 5118 CPU and 128 G RAM. The batch size was set to 256 and the model was trained using an NVIDIA Quadro P5000 GPU.

In this study, three metrics were used to evaluate the quality of the reconstructed images: SSIM [38], MSE [39], and PSNR [40]. SSIM is a comprehensive reference metric that measures the similarity between two images based on the initial uncompressed or undistorted image (i.e., the reference in this study is the original visual scene stimulus). It is a perceptual model that views image degradation as a perceptual change in structural information, while also incorporating important perceptual phenomena such as brightness masking and contrast masking. The SSIM value ranges between 0 and 1, with a higher value indicating a higher similarity between the two images. On the other hand, MSE measures the mean square deviation between the reconstructed scene and the original visual scene. A smaller MSE value indicates a smaller mean square deviation between the two images. Finally, PSNR, which is defined through MSE, is commonly used to quantify the reconstruction quality of distorted and lossy compressed images and videos. A higher PSNR value indicates a higher reconstruction quality of the image.

## Experiments

### Ablation study of VQ

Since VQ-FCDnet is based on vector quantization, it is essential to verify whether vector quantization performs an optimization role in this experiment. In this paper, t-SNE [41] is used to visualize the two feature vectors of  $Z_e(x)$  before vector quantization and  $Z_q(x)$  after vector quantization on MNIST and Fashion MNIST datasets. The visualization of the feature diagrams are shown in Fig. 6. The figure demonstrates the effects of vector quantization on the feature points of the images. Prior to vector quantization, the feature points of similar images, such as Image 1 and Image 4 in MNIST, and Image 2 and Image 3 in Fashion-MNIST, exhibit less coincidence, while the feature points of dissimilar images, namely Image2 and Image3, exhibit more coinci-

dence. Following vector quantization, the feature points of these originally similar images show more consistency, while the feature points of dissimilar images show less consistency. Additionally, discernible differences were observed between different categories.

To sum up, the feature segmentation of different types of images before vector quantization is not obviously incomplete, and the features after vector quantization show obvious boundaries. Similar images overlap and are close to each other after quantization. The overlapping features of images with large differences are smaller. It could be observed that vector quantization separates different features and aggregates similar features.

### Experiment on noise immunity of vector quantization

In the process of acquiring pulse signals, the presence of noise poses a significant challenge to data analysis. However, by employing vector quantization, we are able to overcome this limitation and improve the anti-noise ability of our neural network. Specifically, the creation of a codebook during training enables our network to capture the essential characteristics of the spike trains. By utilizing this codebook, our network decodes features that match the characteristics of the spike trains, thereby avoiding the direct decoding of the spike trains. Through this indirect approach, our network is able to filter out noise in the signal, ultimately leading to improved performance.

To evaluate the noise immunity of our proposed model, we conducted experiments by introducing varying levels of random noise to the pulse signal. The pulse signal with noise was then fed into two networks: one with vector quantization and the other without. The results are presented in Fig. 7. The figure illustrates the proposed model exhibits robustness against random noise. As the noise ratio increases, the model with vector quantization retains the majority of the signal characteristics, whereas the model without vector quantization fails to capture the original pulse signal information in the presence of noise. These results demonstrate that vector quantization plays a crucial role in enhancing the noise immunity of our proposed model.

Figure 8 presents the experimental results of the proposed method and the method without vector quantization under various levels of noise (noise rate) on popular datasets such as MNIST, Fashion-MNIST, Celeba-HQ, Cifar-10 and Coco. The comparison curves have been drawn using reliable evaluation indicators PSNR, SSIM and MSE to assess the effectiveness of the proposed method. Our findings reveal that while the two methods exhibit similar results in low-noise scenarios, the proposed method with vector quantization outperforms the non-vector quantization methods as the noise rate increases. Specifically, we observed that the results of the non-vector quantization method exhibited a



more significant reduction in reconstruction quality across all datasets.

The outcomes of this experiment demonstrate the superior noise immunity characteristics of the proposed model and highlight the significance of vector quantization in mitigating the impact of noise. These findings are of great significance for improving the robustness of retinal visual scene reconstruction in practical applications, especially in environments where noise interference is common.

### Performance comparison with other methods

We compared the performance of the proposed network with five other methods for retinal reconstruction of visual scenes. Method I and Method II are by Zhang et al. [22] and Li et al. [24], respectively, which are considered state-of-the-art methods in the field. In addition, since the inputs are spike trains, we designed a fully connected spiking neural network based on IF neurons to evaluate the difference between the SNN [42] and the proposed network, named Method

III. Furthermore, we designed a method similar to the proposed VQ-FCDnet by combining the fully connected neural network with VQVAE [25], named Method IV, to evaluate the performance of the proposed VQ-FCDnet compared to FCN+VQVAE. Finally, to evaluate the effectiveness of VQ-FCDnet, we also removed the CECB from the model, named Method V, and performed ablation experiments. We evaluated the performance of the above five methods and the proposed method on five datasets. The reconstruction visual scene effects are illustrated in Fig. 9.

Figure 9 shows that all methods perform well in reconstructing the simple MNIST dataset. However, on more complex datasets (e.g., Fashion-MNIST and Celeba-HQ), Method I reconstructs poorly, while Method III and Method IV reconstruct blurry images, and Method II and Method V fail to capture the details in the results of the proposed methods. For the complex datasets Cifar10 and COCO, all methods produce blurred reconstruction results compared to the original images. However, in terms of contour details, the proposed method outperforms the other five methods.

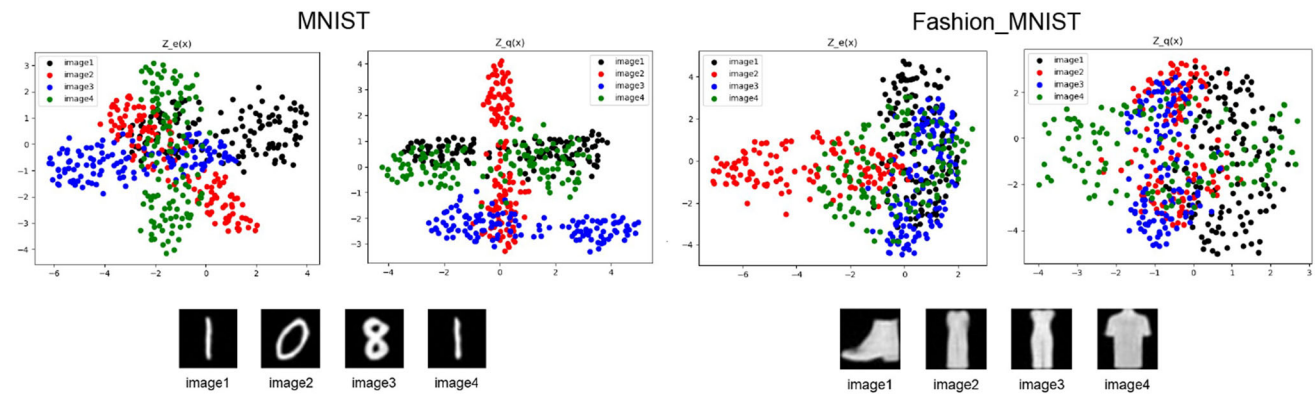
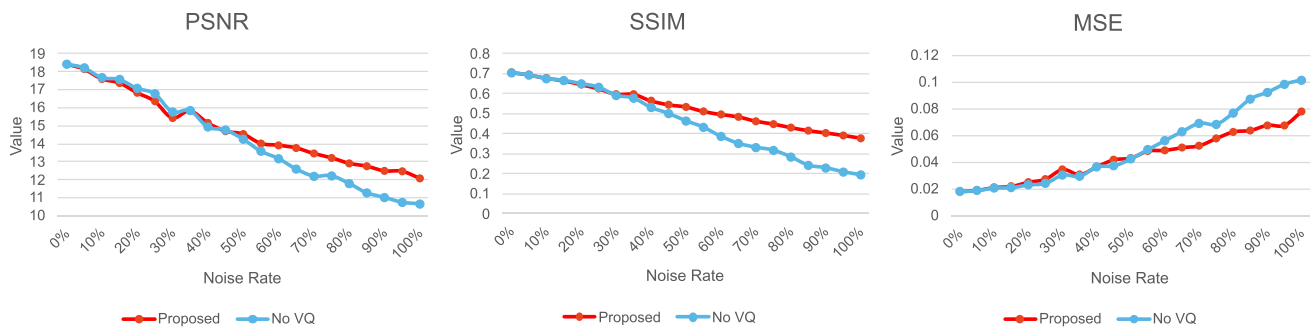


Fig. 6 Features of the MNIST and Fashion MNIST datasets before and after vector quantization

method noise rate	Proposed	No Vector quantization
0%		
10%		
15%		
20%		
25%		
50%		
75%		
100%		

Fig. 7 Differences in the presence or absence of vector quantization under different noise ratios



**Fig. 8** Effectiveness of vector quantization methods with various evaluation indicators under different noise ratios

Dataset method	MNIST	Fashion-MNIST	Cifar-10	Celeba-HQ	COCO
Source					
Method-I	 0.901 0.007 21.54	 0.593 0.020 16.91	 0.515 0.039 14.06	 0.567 0.021 16.87	 0.257 0.051 12.91
Method-II	 0.958 0.003 25.16	 0.732 0.013 18.87	 0.665 0.026 15.83	 0.658 0.019 17.25	 0.453 0.030 15.28
Method-III	 0.828 0.005 23.02	 0.689 0.013 19.01	 0.496 0.031 14.95	 0.627 0.010 19.81	 0.411 0.042 13.81
Method-IV	 0.926 0.003 25.27	 0.809 0.008 21.00	 0.566 0.031 15.08	 0.652 0.011 19.45	 0.477 0.028 15.56
Method-V	 0.958 0.002 26.40	 0.802 0.007 21.34	 0.611 0.029 15.41	 0.671 0.012 19.22	 0.540 0.026 15.85
Proposed	 0.968 0.002 28.20	 0.831 0.007 21.71	 0.669 0.023 16.39	 0.676 0.010 20.17	 0.526 0.024 16.12

**Fig. 9** Comparison of the images generated by the five methods and the proposed method with the source image, respectively, showing the SSIM, MSE, and PSNR scores of the reconstructed image with source image

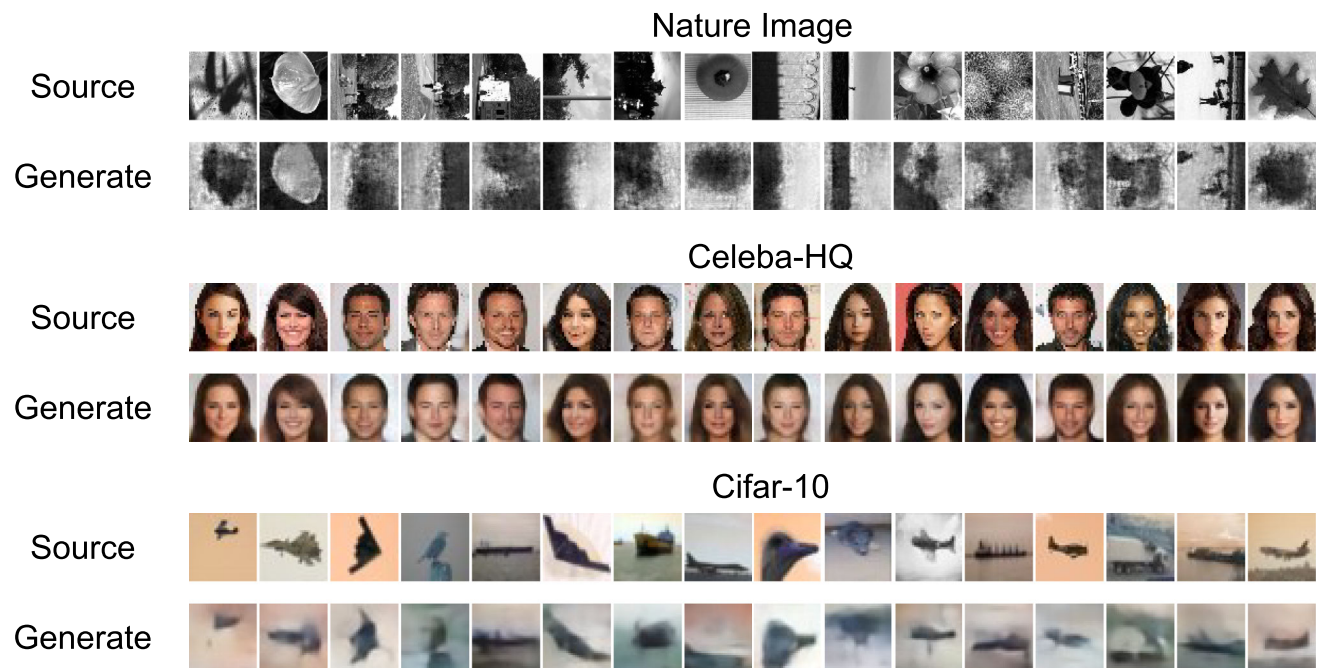
In addition, we evaluated the reconstruction results of these methods using metrics such as PSNR, SSIM and MSE. The evaluation results are shown in Table 1, where the best results are shown in red. It is observed that the performance of the proposed method is much better than the other methods.

On five different datasets, the proposed method improves PSNR and SSIM by an average of 2.016 and 0.1226, respectively, and reduces MSE by an average of 0.0109 compared to Method I proposed by Zhang et al. [22]. Compared to Method II proposed by Li et al. [24], the proposed method improves

PSNR and SSIM by an average of 1.176 and 0.05, respectively, and reduces MSE by an average of 0.0055. Thus, the proposed method outperforms the latest research methods in terms of visual reconstruction performance. Compared with Method III (i.e., SNN), the proposed method has an average improvement of 1.0596 and 0.089 in PSNR and SSIM, respectively, and an average reduction of 0.011 in MSE, which proves that the proposed method outperforms SNN in visual reconstruction. Compared with Method IV (i.e., FCN+VQVAE), which has a similar network structure

**Table 1** Performance comparison with other methods

Dataset	Evaluation	Method I	Method II	Method III	Method IV	Method V	Proposed
MNIST	PSNR	18.74	20.0	19.61	20.374	20.891	21.323
	SSIM	0.824	0.871	0.806	0.866	0.893	0.903
	MSE	0.015	0.0113	0.0121	0.0103	0.0093	0.0084
Fashion_MNIST	PSNR	18.38	19.14	19.63	19.392	20.392	20.608
	SSIM	0.723	0.774	0.731	0.759	0.801	0.808
	MSE	0.0168	0.0143	0.0126	0.0129	0.0105	0.01
Cifar10	PSNR	15.61	16.09	16.13	16.61	17.195	17.31
	SSIM	0.498	0.552	0.506	0.543	0.625	0.628
	MSE	0.0326	0.0289	0.0276	0.0249	0.0219	0.0214
Celeba-HQ	PSNR	14.97	15.83	16.06	16.416	16.683	16.857
	SSIM	0.528	0.624	0.612	0.632	0.661	0.668
	MSE	0.035	0.0291	0.0269	0.0248	0.0233	0.0224
COCO	PSNR	14.25	15.09	15.3	15.313	15.842	15.93
	SSIM	0.346	0.461	0.433	0.484	0.524	0.525
	MSE	0.012	0.0351	0.0327	0.033	0.0293	0.0289

**Fig. 10** Comparison of reconstructed images with original images from other datasets

to the proposed method, there is an average increase of 0.785 and 0.049 on PSNR and SSIM, respectively, and an average reduction of 0.003 on MSE. The metrics prove that the structural improvement of the proposed method improves the visual reconstruction. Compared with Method V of the ablation experiment, the proposed method has an average increase of 0.205 and 0.005 on PSNR and SSIM, respectively, and an average decrease of 0.0006 on MSE, demonstrating that CECB improves the visual reconstruction.

### Other experiments

To verify the effectiveness of our proposed method on a broader range of image datasets, we conducted experiments on nature image datasets as well as colour image datasets such as Celeba-HQ and Cifar10. The results are presented in Fig. 10 and the corresponding evaluation metrics are presented in Table 2.

Based on the restored images and evaluation metrics, it is evident that the proposed method is capable of reconstruct-

**Table 2** Evaluation of other datasets

Dataset	Evaluation method	Proposed
Nature image	PSNR	13.809
	SSIM	0.349
	MSE	0.046
Celeba-HQ(rgb)	PSNR	15.828
	SSIM	0.634
	MSE	0.029
Cifar10(rgb)	PSNR	15.515
	SSIM	0.559
	MSE	0.032

ing visual scenes with strong contrast even in the presence of a small amount of data, while also effectively reconstructing complex visual scenes in the colour image dataset. The experimental results highlight the efficacy of applying our method to a wide range of applications, such as image and video compression, image restoration and scene reconstruction. Furthermore, the performance of the proposed method on diverse datasets suggests its robustness and generalizability, paving the way for its potential integration into various real-world scenarios.

## Future scope

Although our method has achieved good results at its current stage of development, the process of reconstructing visual scene stimuli from RGC spike trains is still a challenge. In future work, we will further improve the network model in the following aspects to make it closer to the function of the human eye:

- Improve the quality of the reconstructed visual scene and ensure that the reconstructed image is similar to the nature scene in terms of clarity and colour recovery.
- Improve the reconstruction ability of the model for small datasets.
- Propose a model for a dynamic video to realize real-time reconstruction and restore the visual scene more realistically.

Finally, we hope that our work can inspire other researchers and jointly promote the development of retinal visual scene reconstruction.

## Conclusion

In this paper, we have proposed a deep network VQ-FCDnet for retinal spike signal reconstruction of visual scenes based on vector quantization. We first built a FECN module composed of multi-layer fully connected neural networks and convolutional neural networks to extract and compress the feature information of pulse signals. The nearest neighbour search method is then used to distribute the feature information into multiple vectors of potential codebooks. These vectors are recombined into new feature maps and sent to the REN module composed of convolutional neural networks and transposed convolutional neural networks to reconstruct the visual scene. In structural analysis experiments, it has been shown that vector quantization has a significant impact on the aggregation of similar features and the dispersion of different features of retinal pulse signals. By comparing the impact of networks with or without vector quantization under different levels of noise, it has been verified that vector quantization has a significant impact on the immunity of the network to noise, providing a new method for reconstructing retinal visual scenes. The proposed method was evaluated on multiple datasets and the reconstruction results were evaluated using five evaluation parameters. Experimental results show that the proposed method is superior to other methods, with higher clarity, richer details, and more accurate spatial structure relationships.

**Acknowledgements** This research was financially supported by the Scientific Research Grant of Shantou University, China, Grant (No. NTF17016), National Natural Science Foundation of China (No. 82071992) and Basic and Applied Basic Research Foundation of Guangdong Province [No. 2020B1515120061].

**Data availability** Data supporting the results of this study can be obtained from GitHub at <https://github.com/jalexnoel/Retinal-spike-train-decoder.git>.

## Declarations

**Conflict of interest** The authors declare that the research was conducted in the absence of any commercial or financial relationships that could be construed as a potential conflict of interest.

**Open Access** This article is licensed under a Creative Commons Attribution 4.0 International License, which permits use, sharing, adaptation, distribution and reproduction in any medium or format, as long as you give appropriate credit to the original author(s) and the source, provide a link to the Creative Commons licence, and indicate if changes were made. The images or other third party material in this article are included in the article's Creative Commons licence, unless indicated otherwise in a credit line to the material. If material is not included in the article's Creative Commons licence and your intended use is not permitted by statutory regulation or exceeds the permitted use, you will need to obtain permission directly from the copyright holder. To view a copy of this licence, visit <http://creativecommons.org/licenses/by/4.0/>.

## References

1. Kim US, Mahroo OA, Mollon JD, Yu-Wai-Man P (2021) Retinal ganglion cells-diversity of cell types and clinical relevance. *Front Neurol*. <https://doi.org/10.3389/fneur.2021.661938>
2. Masland RH (2001) The fundamental plan of the retina. *Nat Neurosci* 4:877–886. <https://doi.org/10.1038/nn0901-877>
3. 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* 454:995–999. <https://doi.org/10.1038/nature07140>
4. Weiland JD, Humayun MS (2014) Retinal prosthesis. *IEEE Trans Biomed Eng* 61(5):1412–1424. <https://doi.org/10.1109/TBME.2014.2314733>
5. Grimes WN, Songco-Aguas A, Rieke F (2018) Parallel processing of rod and cone signals: retinal function and human perception. *Ann Rev Vis Sci* 4:123–141. <https://doi.org/10.1146/annurev-vision-091517-034055>
6. Gütig R, Gollisch T, Sompolinsky H, Meister M (2013) Computing complex visual features with retinal spike times. *PLoS ONE* 8(1):1–15. <https://doi.org/10.1371/journal.pone.0053063>
7. Gollisch T, Meister M (2010) Eye smarter than scientists believed: neural computations in circuits of the retina. *Neuron* 65(2):150–164
8. Gershenson, C.: Artificial neural networks for beginners (2003). CoRR [arXiv:cs.NE/0308031](https://arxiv.org/abs/cs.NE/0308031)
9. Rivest F, Bengio Y, Kalaska J (2004) Brain inspired reinforcement learning. In: Saul L, Weiss Y, Bottou L (eds) *Advances in neural information processing systems* 17. [https://proceedings.neurips.cc/paper\\_files/paper/2004/file/d37b3ca37106b2bfdeaa12647e3bb1c9-Paper.pdf](https://proceedings.neurips.cc/paper_files/paper/2004/file/d37b3ca37106b2bfdeaa12647e3bb1c9-Paper.pdf)
10. Wu Z-B, Yu J-Q (2019) Vector quantization: a review. *Front Inf Technol Electron Eng* 20(4):507–524. <https://doi.org/10.1631/FITEE.1700833>
11. Li J, Li B, Xu J, Xiong R, Gao W (2018) Fully connected network-based intra prediction for image coding. *IEEE Trans Image Process* 27(7):3236–3247. <https://doi.org/10.1109/TIP.2018.2817044>
12. Gu J, Wang Z, Kuen J, Ma L, Shahroudy A, Shuai B, Liu T, Wang X, Wang G, Cai J et al (2018) Recent advances in convolutional neural networks. *Pattern Recogn* 77:354–377. <https://doi.org/10.1016/j.patcog.2017.10.013>
13. Gao H, Yuan H, Wang Z, Ji S (2020) Pixel transposed convolutional networks. *IEEE Trans Pattern Anal Mach Intell* 42(5):1218–1227. <https://doi.org/10.1109/TPAMI.2019.2893965>
14. Graf AB, Kohn A, Jazayeri M, Movshon JA (2011) Decoding the activity of neuronal populations in macaque primary visual cortex. *Nat Neurosci* 14(2):239–245. <https://doi.org/10.1038/nn.2733>
15. Hinrikus H, Karai D, Lass J, Rodina A (2010) Effect of noise in processing of visual information. In: *Nonlinear biomedical physics*, vol 4. Springer, pp 1–7. <https://doi.org/10.1186/1753-4631-4-S1-S5>
16. Rumyantsev OI, Lecoq JA, Hernandez O, Zhang Y, Savall J, Chrapkiewicz R, Li J, Zeng H, Ganguli S, Schnitzer MJ (2020) Fundamental bounds on the fidelity of sensory cortical coding. *Nature* 580(7801):100–105. <https://doi.org/10.1038/s41586-020-2130-2>
17. Pillow JW, Paninski L, Uzzell VJ, Simoncelli EP, Chichilnisky E (2005) Prediction and decoding of retinal ganglion cell responses with a probabilistic spiking model. *J Neurosci* 25(47):11003–11013. <https://doi.org/10.1523/JNEUROSCI.3305-05.2005>
18. Nee R, Zelst A, Awater G (2000) Maximum likelihood decoding in a space division multiplexing system. In: *VTC2000-Spring. 2000 IEEE 51st vehicular technology conference proceedings* (Cat. No.00CH37026), vol 1, pp 6–101. <https://doi.org/10.1109/VETECS.2000.851407>
19. Díaz-Tahoces A, Martínez-Alvarez A, García-Moll A, Humphreys L, Bolea JÁ, Fernández E (2015) Towards the reconstruction of moving images by populations of retinal ganglion cells. In: *Artificial computation in biology and medicine: international work-conference on the interplay between natural and artificial computation, IWINAC 2015, Elche, Spain, June 1–5, 2015, Proceedings, Part I* 6. Springer, pp 220–227
20. Kurita T (2019) Principal component analysis (pca). In: *Computer vision: a reference guide*, pp 1–4. [https://doi.org/10.1007/978-3-030-03243-2\\_649-1](https://doi.org/10.1007/978-3-030-03243-2_649-1)
21. Kim YJ, Brackbill N, Batty E, Lee J, Mitelut C, Tong W, Chichilnisky E, Paninski L (2021) Nonlinear decoding of natural images from large-scale primate retinal ganglion recordings. *Neural Comput* 33(7):1719–1750. [https://doi.org/10.1162/neco\\_a\\_01395](https://doi.org/10.1162/neco_a_01395)
22. 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 Netw* 125:19–30. <https://doi.org/10.1016/j.neunet.2020.01.033>
23. Xi E, Bing S, Jin Y (2017) Capsule network performance on complex data <https://doi.org/10.48550/arXiv.1712.03480>
24. Li W, Joseph Raj AN, Tjahjadi T, Zhuang Z (2022) Fusion of ANNs as decoder of retinal spike trains for scene reconstruction. *Appl Intell* 52(13):15164–15176. <https://doi.org/10.1007/s10489-022-03402-w>
25. Van Den Oord A, Vinyals O et al (2017) Neural discrete representation learning. In: Guyon I, Von Luxburg U, Bengio S, Wallach H, Fergus R, Vishwanathan S, Garnett R (eds) *Advances in neural information processing systems*, vol 30. [https://proceedings.neurips.cc/paper\\_files/paper/2017/file/7a98af17e63a0ac09ce2e96d03992fbc-Paper.pdf](https://proceedings.neurips.cc/paper_files/paper/2017/file/7a98af17e63a0ac09ce2e96d03992fbc-Paper.pdf)
26. Kingma DP, Welling M (2022) Auto-encoding variational bayes. [arXiv:1312.6114](https://arxiv.org/abs/1312.6114) [stat.ML]
27. Razavi A, Oord A, Vinyals O (2019) Generating diverse high-fidelity images with vq-vae-2. *Advances in neural information processing systems*, vol 32. [https://proceedings.neurips.cc/paper\\_files/paper/2019/file/5f8e2fa1718d1bbcadf1cd9c7a54fb8c-Paper.pdf](https://proceedings.neurips.cc/paper_files/paper/2019/file/5f8e2fa1718d1bbcadf1cd9c7a54fb8c-Paper.pdf)
28. Tai Y, Yang J, Liu X (2017) Image super-resolution via deep recursive residual network. In: *Proceedings of the IEEE conference on computer vision and pattern recognition (CVPR)*
29. Chen Y, Dai X, Liu M, Chen D, Yuan L, Liu Z (2020) Dynamic relu. In: *Computer vision—ECCV 2020: 16th European conference, Glasgow, UK, August 23–28, 2020, proceedings, Part XIX* 16, pp 351–367
30. Lee H, Park J, Hwang JY (2020) Channel attention module with multiscale grid average pooling for breast cancer segmentation in an ultrasound image. *IEEE Trans Ultrason Ferroelectr Freq Control* 67(7):1344–1353. <https://doi.org/10.1109/TUFFC.2020.2972573>
31. Liu Z, Cheng K-T, Huang D, Xing EP, Shen Z (2022) Nonuniform-to-uniform quantization: towards accurate quantization via generalized straight-through estimation. In: *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pp 4942–4952
32. Tošić I, Frossard P (2011) Dictionary learning. *IEEE Signal Process Mag* 28(2):27–38. <https://doi.org/10.1109/MSP.2010.939537>
33. Lee C-P, Lin C-J (2013) A study on l2-loss (squared hinge-loss) multiclass svm. *Neural Comput* 25(5):1302–1323. [https://doi.org/10.1162/NECO\\_a\\_00434](https://doi.org/10.1162/NECO_a_00434)
34. Haynes D, Corns S, Venayagamoorthy GK (2012) An exponential moving average algorithm. In: *2012 IEEE Congress on evolutionary computation*. IEEE, pp 1–8. <https://doi.org/10.1109/CEC.2012.6252962>
35. Tjandra A, Sisman B, Zhang M, Sakti S, Li H, Nakamura S (2019) VQVAE unsupervised unit discovery and multi-scale code2spec inverter for zerospeech challenge 2019. CoRR [arXiv:1905.11449](https://arxiv.org/abs/1905.11449)

36. Liu JK, Karamanlis D, Gollisch T (2022) Simple model for encoding natural images by retinal ganglion cells with nonlinear spatial integration. *PLoS Comput Biol* 18(3):1009925. <https://doi.org/10.1371/journal.pcbi.1009925>
37. Cessac B, Kornprobst P, Kraria S, Nasser H, Pamplona D, Portelli G, Viéville T (2017) Pranas: a new platform for retinal analysis and simulation. *Front Neuroinform* 11:49. <https://doi.org/10.3389/fninf.2017.00049>
38. Brunet D, Vrscaj ER, Wang Z (2012) On the mathematical properties of the structural similarity index. *IEEE Trans Image Process* 21(4):1488–1499. <https://doi.org/10.1109/TIP.2011.2173206>
39. Schluchter MD (2005). Mean square error. <https://doi.org/10.1002/0470011815.b2a15087>
40. Johnson DH (2006) Signal-to-noise ratio. *Scholarpedia* 1(12):2088. <https://doi.org/10.4249/scholarpedia.2088>
41. Arora S, Hu W, Kothari PK (2018) An analysis of the t-sne algorithm for data visualization. In: *Conference on learning theory*. PMLR, pp 1455–1462. <https://proceedings.mlr.press/v75/arora18a.html>
42. Tavanaei A, Ghodrati M, Kheradpisheh SR, Masquelier T, Maida A (2019) Deep learning in spiking neural networks. *Neural Netw* 111:47–63. <https://doi.org/10.1016/j.neunet.2018.12.002>

**Publisher's Note** Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.