



A learning artificial visual system and its application to orientation detection

Tianqi Chen¹ · Yuki Kobayashi¹ · Chenyang Yan¹ · Zhiyu Qiu¹ · Yuxiao Hua¹ · Yuki Todo² · Zheng Tang^{3,4}

Accepted: 30 September 2024 / Published online: 14 January 2025

© The Author(s), under exclusive licence to Springer Science+Business Media, LLC, part of Springer Nature 2025

Abstract

This paper proposes a learning artificial visual system, the Learning Dendritic Model Artificial Visual System (DModel-AVS), for orientation detection inspired by biological visual mechanisms. The DModel-AVS consists of two layers: local orientation detection neurons layer and global orientation detection neurons layer. The local neurons detect local features of an image, utilizing dendrite model neurons. The global neurons are designed to implement global features of the image by summing the outputs of the local dendritic neurons. The backpropagation-based learning is performed only to the dendritic neurons. The effectiveness of the DModel-AVS is evaluated through several experiments comparing it with various convolutional neural network (CNN)-based orientation detection systems. Results show that the DModel-AVS is a more biologically plausible and effective solution to orientation detection, with higher accuracy, and lower learning costs. The proposed system has practical applications in various fields such as computer vision and robotics.

Keywords Dendritic neuron · Artificial visual system · Convolutional neural networks · Orientation detection

1 Introduction

The human brain, with its complex network of over 10^{11} neurons and 10^{15} interactions, plays a crucial role in various cognitive functions. One key function is the orientation detection, which is vital in understanding and interacting with our environment [1]. Neurons receive and react to signals and stimuli, transforming these into electronic signals, resulting in functions such as learning, memory, and feature detection [2]. In the researches of brain, the visual system is considered a key component because most of the information received by the brain is visual-resourced in nature [3]. Furthermore, visual information is given predominance in influencing the behaviors of animals and humans by providing cues such as object color, position, orientation, and movement direction, which can lead to both beneficial and adverse outcomes [4]. Hubel and Wiesel conducted animal experiments to unravel the mysteries of the retina. Their findings revealed that certain cells in the retina exhibit orientation selectivity, responding exclusively when specific stimuli occupy the receptive field and maintaining their original position in the presence of others [5]. Subsequently, these cells, categorized as simple and complex cells, undergo classification by the lateral

geniculate nucleus (LGN), which plays a crucial role in judging the detected orientation [6]. The research extended to the visual cortex of animals, where experiments recorded phenomena illustrating that neurons respond solely to light stimulation in specific orientations, demonstrating insensitivity to the stimuli's position [7, 8]. Interestingly, within their receptive field, optimal oriented light movement does not lead to neuron inactivation. In the primary cortex, a distinct category of neurons with orientation-selectivity becomes apparent [9]. Despite these discoveries, the specific mechanism governing the visual system remains elusive. In our previous papers, we proposed a novel artificial visual systems that detect object orientation with high accuracy, leveraging Hubel's theory of retinal mechanisms. These systems demonstrate superior performance in accuracy and robustness [10, 11]. These methods mentioned are limited by the biological research results previously since the visual system of creature is not a static process but a dynamic one [12]. Building bio-inspired computational visual systems using artificial intelligence solves the computer visual tasks and applications [13]. As a emerging tool, artificial neural networks-based artificial visual systems can solve various problems, such as classification, prediction, clustering and vision [14–17]. However, the lack of learning capabilities in these models limits their potential to evolve into more

Extended author information available on the last page of the article

complex systems. Blakemore and Copper observed that the animals developed neurons selectively responsive to the orientation they were exposed to [19]. The findings highlighted the plasticity of the visual system. Recently, there has been a growing interest in exploring how bio-inspired models, particularly those that mimic the structure and functionality of visual system, can be used to create more adaptive and biologically plausible artificial visual systems. LeCun et al. introduced a standard architecture (convolutional neural network (CNN)) for visual recognition task, including orientation detection [18]. Simonyan and Zisserman proposed a deep CNN architecture that achieved state-of-art performance on orientation detection and image recognition tasks [20]. He et al. introduced residual connections, allowing for the training of much deeper architectures [21]. However, these learning models need large number of parameters, require large amount of labeled data for training, and are computationally intensive limiting their application in real-time or resource-constrained environments. Additionally, they are often seen as 'black boxes', lacking interpretability and transparency. In this paper, we focus on overcoming these challenges by integrating insights from both dendritic neurons and visual systems. Our goal is to develop a new bio-inspired learning orientation detection system that exhibit the adaptability, robustness, efficiency and biological plausibility. We use dendritic neurons to realize the local orientation detection because recent studies have delved deeper into the role of dendrites in orientation detection and the dendritic neuron model is learnable [22, 23]. We designed a dendrite neuron model artificial visual system (DModel-AVS) that can detect the orientation of a single receptive field sized 3×3 . We implemented this orientation detection system using orientation selective neurons that can learn the features of each orientation through a dendritic neuron model in our computer simulation. This artificial visual system has several receptive fields sized 3×3 , each of which corresponds to one of four specific orientations. Each orientation is assigned a group of random weight and bias values, which are combined randomly in the branches. By learning from a set of images labeled with their orientations, the dendritic neuron calculates the weights and biases that can detect the features of all 4 orientations (0 degree, 45 degree, 90 degree and 135 degree), enabling the model to detect the orientation of an image. To verify the performance of our model, we designed and simulated various computer experiments. In simulations, we found that the orientation can be detected by the interaction on the dendrites of objects in our DModel-AVS, regardless of their shapes, sizes, and positions. Furthermore, as a learning orientation detection artificial visual system, it has a high accuracy and robustness outperforming simple structural CNN-based mechanisms, which are considered some of the most powerful methods in computer vision for image recognition, in several

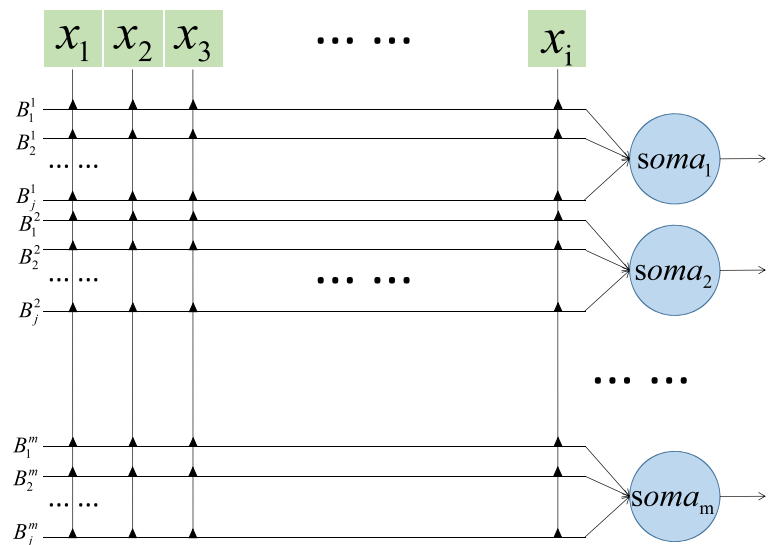
experiments. DModel-AVS also outperformed CNN-based systems in terms of both biological theory and detection accuracy, because its learning mechanism kept a black box so the biological soundness cannot be proved [26]. And our DModel-AVS acquires the shape of dendrites, the number of branches, and synapses of the local direction-selective ganglion cells through learning, this is demonstrated by the elimination of synapses and branches in our model. The simulation replicates the selectivity of neurons in the brain towards specific orientations, resulting in higher detection accuracy in orientation detection tasks. After simulations, it is proved that the DModel-AVS excels in terms of identification accuracy, noise resistance, computational and learning cost, learning data requirements, hardware implementation and overall reasonability. Additionally, advancements in improving neural network efficiency have led to the development of various systems. For instance, one study investigated the periodicity and synchronicity of inertial memristive neural networks, achieving synchronization in the drive-response system by employing matrix measure techniques and Halanay inequalities [27]. Another research explored a nonlinear network control system incorporating specific models to address output tracking control problems in the presence of input-state delays [28]. These methodologies offer insights into optimizing neural network systems for improved performance and can enhance the efficiency of the DModel-AVS by leveraging its dendritic structure. Our DModel-AVS aims to bridge the gap between biological plausibility and computational performance, addressing the limitations of current CNN-based methods. By incorporating complex dendritic structures and learning mechanisms observed in biological neurons, we strive to create a model that excels in orientation detection tasks and aligns with the principles of biological vision systems. Additionally, by testing the assumption that the ability for orientation detection is a result of learning, as proposed by Blakemore's learning theory, we further establish the biological soundness of our model. This step is crucial for demonstrating that our model not only functions effectively but also mimics the learning processes observed in biological systems.

2 Method

2.1 Dendrite neuron

The McCulloch-Pitts neuron provides a simplified representation of a biological neuron, where weighted inputs determine whether the neuron fires or remains inactive [29]. This model, foundational to modern artificial neural networks, lacks consideration for dendritic functions in the brain, which are now recognized as crucial for motion detec-

Fig. 1 A sample of a dendritic neuron model that has i inputs, m probable outputs and each soma has j branches



tion [30, 31]. In our prior work, we introduced the dendritic neuron model, emphasizing neural processing in dendrites, soma, and axon [32, 33]. The biological plausibility of the dendritic neuron model is grounded in its ability to mimic the functions of biological neurons within the visual system. The dendritic neuron model incorporates dendrites, which play a crucial role in integrating synaptic inputs and facilitating complex signal processing. Research has shown that dendrites are essential for various neuronal functions, including motion detection and orientation selectivity [23]. By simulating the detailed structure and function of dendrites, our model captures the nuanced processing capabilities observed in biological neurons. This approach not only enhances the functional accuracy of the artificial visual system but also aligns it with the biological processes underlying vision. Unlike traditional models that primarily focus on the soma and axon, our dendritic neuron model includes the intricate interactions and computations occurring in the dendritic branches, thus providing a more comprehensive and biologically realistic simulation of neuronal activity. Figure 1 depicts the dendritic neuron model with independent branches ($B_j, j = 1, 2, \dots, m$) and a soma. Inputs $x_1, x_2, \dots, x_n$ reach branches through synapses, undergo multiplication, and are weighted by branch strength, contributing to the neuron's activation level. Initially, all inputs connect to all branches via synapses, which can be either excitatory or inhibitory. The model assumes nonlinear interactions among all inputs for generality. This model processes information with a dendritic cell, providing a larger loss function but higher accuracy, reducing overfitting risk [34]. The dendritic cell is structurally simple, precise, computationally efficient, and compatible with neural networks [35]. Training loss is controllable, with the cell body's loss decreasing consistently without changing accuracy [34]. Compared to other deep learning-based orientation detection systems, dendritic

neuron model offers a more biologically plausible and intuitively oriented detection approach [36].

There are four layers of our dendritic neuron model typically referred to as the: 1. Synaptic layer represents the location of the synapses between the dendritic neuron and other neurons. It receives inputs from other neurons and is responsible for integrating the inputs. The inputs are combined and transformed into a single output that is sent to the next layer. This layer can be regarded the (1) to the input by a sigmoid function, for example, a dendritic neuron model with i inputs and j branches:

$$S_{ij}(x_i) = \frac{1}{1 + \exp(-\frac{1}{d_{ij}}(w_{ij}x_i - q_{ij}))}. \quad (1)$$

In the formula, we assume that all the inputs are nonlinear and connected to the corresponding synapses which may show the excitatory and inhibitory position by the activation function. The d_{ij} is the distance parameter, w_{ij} the weight of the synapse and q_{ij} is the bias. Since the neural system is a micro system, the d_{ij} is small enough, and the input x_i can only take on the values of 0 or 1 in a binary image, the synapses will only show 4 given position so as it is shown in Fig. 2:

1) Constant-0 position (Fig. 2 (a-1) and (a-2)): When $0 < w_{ij} < q_{ij}$ or $w_{ij} < 0 < q_{ij}$, whichever the input x_i is 0 or 1, the dendritic branch can only receive the 0 output of this synapse.

2) Constant-1 position (Fig. 2 (b-1) and (b-2)): When $q_{ij} < 0 < w_{ij}$ or $q_{ij} < w_{ij} < 0$, whichever the input x_i is 0 or 1, the dendritic branch can only receive the 1 output of this synapse.

3) excitatory position (Fig. 2 (c)): When $0 < q_{ij} < w_{ij}$, the output of synapse is the same with input x_i , shows 0 when x_i is 0, vice versa.

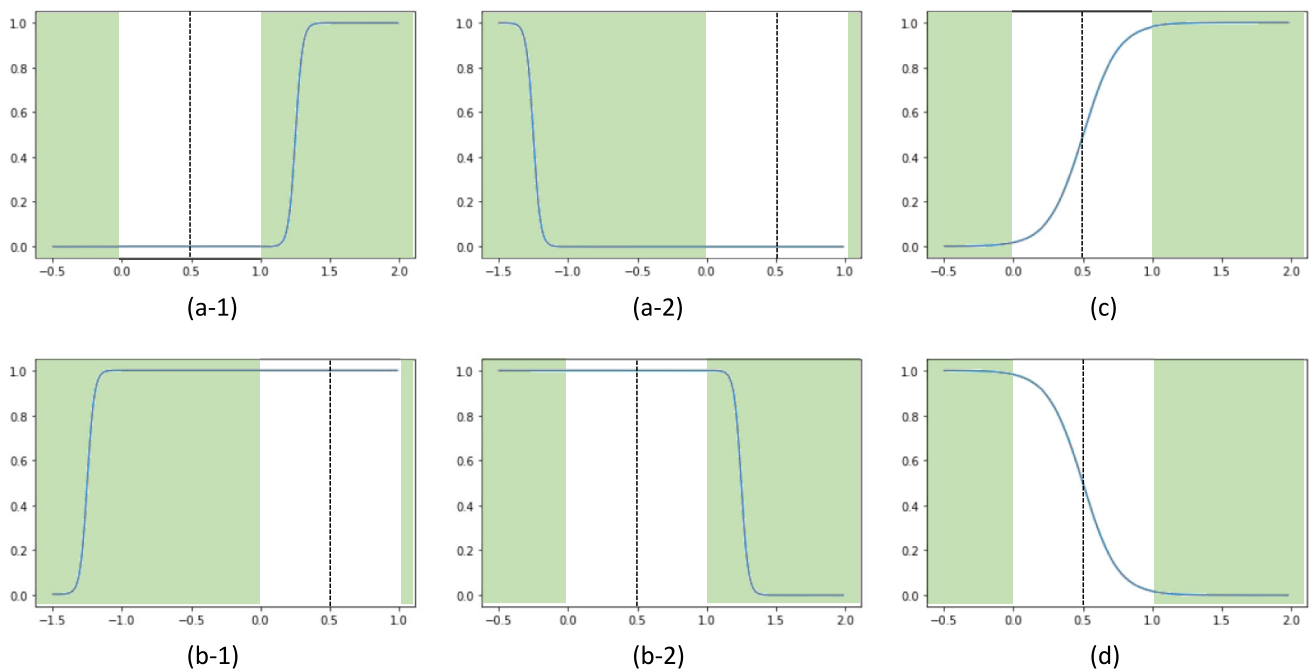


Fig. 2 (a-1) and (a-2) shows the 2 possible situations of constant-0 position, (b-1) and (b-2) shows the 2 possible situations of constant-1 position, (c) shows the excitatory position and (d) shows the inhibitory position

4) Inhibitory position (Fig. 2 (d)): When $0 > q_{ij} > w_{ij}$, the output of synapse is the completely different with input x_i , shows 1 when x_i is 0, and 0 when x_i is 1.

In this step, we can give the ability of learning to dendritic neurons. Through the learning algorithm, the model can cut the useless synapses and branches of inputs to build a detection system with much more higher accuracy.

2. Branch layer has the dendritic branches which provide a large surface area for receiving inputs, allowing the dendritic neuron to integrate many inputs from multiple neurons. This layer is responsible for distributing the inputs to the soma layer through the (2) function:

$$b_j = \prod_{i=1}^n S_{ij}(x_i). \quad (2)$$

It is evident that the output is entirely influenced by the output of the synapses and is the result of multiplication of inputs. This result will not be changed by the constant-1 position synapse and will be 0 if one of the synapses shows the constant-0 position, which is a crucial feature of the dendritic model's learning.

3. Membrane layer refers to the plasma membrane of the dendritic neuron and is responsible for regulating the flow of ions in and out of the cell. The membrane contains voltage-gated ion channels that open and close in response to changes in voltage, allowing the dendritic neuron to respond to the

inputs it receives. In our simulation system, this behavior is represented by selectively integrating the output signals from each branch and transmitting these integrated signals to the soma. This function can also be performed as a formula (3):

$$u = \frac{\sum_{j=1}^m v_j b_j}{m}. \quad (3)$$

In this layer, the information is aggregated by summing the input b_j with the connection strength v_j . The combined information is then processed by the soma for further judgment.

4. Cell body layer or soma layer is the central part of the dendritic neuron and performs non-linear processing of the inputs. It receives the integrated inputs from the branch layer and combines them in a way that produces a single output. This output is processed by (4), and the soma has its own threshold θ and the slope parameter λ :

$$O = \frac{1}{1 + \exp(-\lambda(u - \theta))}. \quad (4)$$

Using another sigmoid function, the dendritic neuron model can analyze the inputs of each branch and generate an output, which is then displayed. The soma layer can carry out various operations, such as summation, thresholding, and gain control, to process the inputs and generate a single output, which is then transmitted to the axon layer.

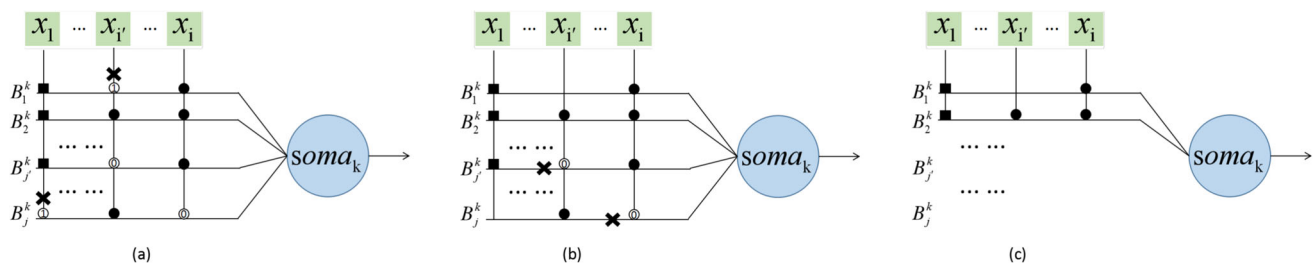


Fig. 3 (a) shows a possible result after learning of a soma, (b) cuts the part that influence by constant-1 position and (c) cuts the part that influenced by constant-0 position

These functions of the different layers of a dendritic neuron model are used to create complex processing mechanisms in the neuron, allowing it to respond to inputs in a flexible and adaptive way. The specific functions of each layer can be tailored to the needs of the application and the goals of the model.

2.2 Local orientation detection dendrite neuron model

In our model design, the decision to detect four orientations aligns with the structural inspiration drawn from the vertebrate retina. The retina serves as a crucial component in processing visual information and discerning object orientations. Specifically, the photoreceptor layer in the retina plays a pivotal role by receiving visual stimuli and translating them into electrical signals for the brain's interpretation. This layer includes specialized parts sensitive to light, capable of responding even in low-light conditions.

To mimic this biological process, we opted for a receptive field size of 3×3 in our local orientation detection dendritic neuron model. This choice ensures that each neuron processes information from a 3×3 grid of pixels. The detection of orientations relies on identifying which 2 or 3 pixels within this grid receive signals (assigned the value 1), corresponding to the orientations of 0 degrees, 45 degrees, 90 degrees, and 135 degrees. Given the four distinct orientations, we assigned four somas (neuron cell bodies) for detecting orientations, denoted as O_1 to O_4 . The signals from a central pixel and the eight surrounding pixels serve as inputs (represented as x_1 to x_9), where B_1 to B_n represent the branches connected by synapses. This structural arrangement ensures that our model emulates the biological foundation, providing a convincing basis for the simulation.

For example, a receptive field located at position (ij) exemplifies the following situation. The input x_{ij5} from photoreceptor at position (ij) serves as the center of a single receptive field, while x_{ijd} represents inputs at receptive field position d ($d = 1, 2, \dots, 9$) are applied to 4 dendritic neu-

rons: $soma_u$ ($u = 1, 2, 3, 4$) through their own synapses of four orientations. Each neuron have V_u branches. The inputs x_{ijd} are applied to the v -th branch ($v = 1, 2, \dots, V_u$) of the u -th neuron u ($u = 1, 2, 3, 4$) through synapses $S_{ij}^{uv}(x_{ijd})$ as

$$S_{ij}^{uv}(x_{ijd}) = \frac{1}{1 + \exp(-\frac{1}{d_{ij}^{uv}}(w_{ij}^{uv}x_{ijd} - q_{ij}^{uv}))}, \quad (5)$$

where w_{ij}^{uv} and q_{ij}^{uv} are synaptic parameters weight and bias to corresponding synapse. Then, the branches perform a multiplication on these synaptic signals, and produce the local potentials

$$b_{uv} = S_{ij}^{uv} \left(\prod_{d=1}^9 S_{ij}^{uv}(x_{ijd}) \right), \quad (6)$$

Each branch also has its strength ω_{uv} that can weight potential before it reaches soma. Finally, soma sums these weighted components from all branches and produces the neuron's output O_u ($u = 1, 2, 3, 4$)

$$O_u = \frac{1}{1 + \exp(-\lambda(\sum_{v=1}^{V_u} \omega_{uv} b_{uv} - \theta_u))} \quad (7)$$

where θ_u is the threshold of neuron u , and λ is the coefficient that determines the step of the sigmoid function (Fig. 3).

After learning, unnecessary synapses and branches are pruned, retaining only essential components, similar to the process of vertebrates memorizing features. For example, Fig. 4 illustrates a potential learning outcome. For the 0 degree orientation, synapse connections are cut, leaving only inputs from x_{ij4} to x_{ij6} (assuming a particular configuration). The branches are also pruned until only one remains. Consequently, the soma exhibits excitatory reactions exclusively to the inputs x_4 , x_5 and x_6 .

The learning process results in four neurons dedicated to orientation detection, each exclusively activated by a specific orientation. This aligns with Hubel and Wiesel's theory on

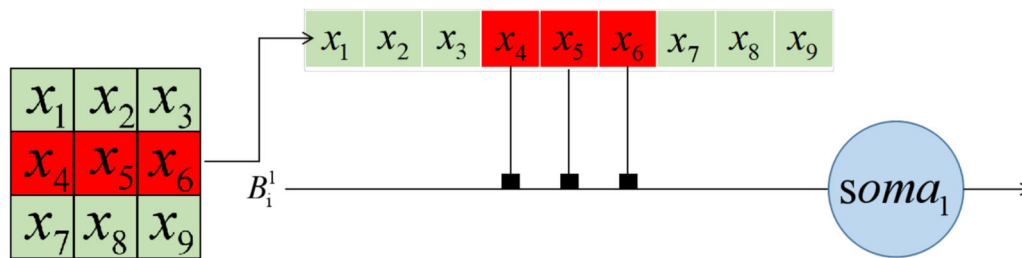


Fig. 4 An example of local orientation detection dendrite neuron model. It shows one possible result after learning

orientation-selective simple cells. Assuming the analysis of a 9-pixel receptive field using the dendrite neuron model, we derive a local orientation detection dendrite neuron model. This model facilitates the detection of an object's orientation through analysis and learning from training data. This capability stems from the specialized design of neurons in the primary cortex for learning orientation selectivity, enabling them to perform the function of orientation detection.

2.3 Global orientation detection dendrite neuron model

Considering an image composed of $M \times N$ pixels, we can employ the previously described local orientation detection neuron. The global orientation detection dendrite neuron model is based on the results of the local orientation detection dendrite neuron model's analysis of several receptive fields. In this paper, we choose to detect the global orientation of the image by scanning all the pixels, where each pixel serves as the center of a receptive field covering the entire image. Thus, each of the four local orientation detection dendrite neurons have $M \times N$ outputs O_u^k and the number of activations of a specific local motion directive detection neuron $Test_u$ can be summed as

$$Test_u = \sum_{k=1}^{MN} O_u^k, \quad (8)$$

which indicates how many local orientation detection dendrite neurons are activated at this position. For example, the result after learning may appear on the excitatory reaction to the receptive field on the pixels that received light stimuli. To process the outputs of the learned global orientation detection dendrite neuron model, specific cell calculations are necessary. A research shows that the retina can do the global detection by summing the output of the local neurons, and compare the different inside, makes it possible to do complex actions [37]. This implies the need for a sumpooling layer after the output of a local edge-orientation detection neuron to determine the correct angle in the presence of multiple detection results. The sumpooling layer works by adding up all the inputs and providing the summary output, which

aligns with observations in the primary visual cortex where similar mechanisms have been identified [38]. Therefore, following this research and supporting evidence from studies on the visual system, we adopted the sumpooling layer in our dendrite model for final data processing.

As a result, we make our model to decide the final output by judging the activated times of corresponding soma of the orientation. While the exact mechanism is not fully understood, simulations indicate that the output can be approximated by adding and comparing numbers [39].

A single 3×3 receptive field is possible to activate 0, 1, 2, 3 or 4 somas, and a single soma is also possible to be activated by any number of receptive field, the whole system diagram is presented in Fig. 5(a). As we show in the Fig. 5(b) that a 5×5 image with 3 light spots is given, it may not represent the final result of the model after learning; however, it serves as an example illustrating the mechanism of how our model works.

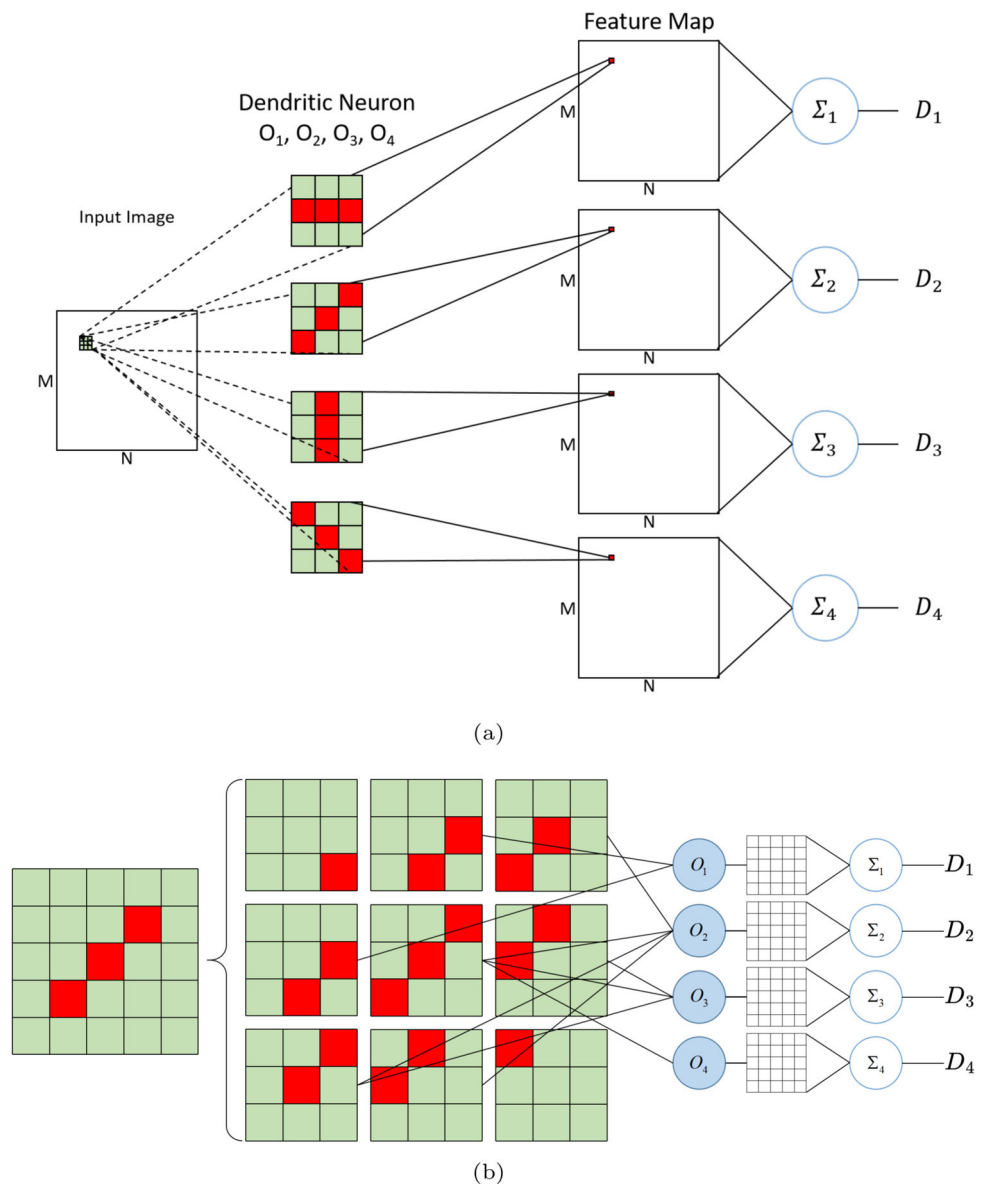
Figure 5(b) shows that the third orientation O_3 is activated four times, more than any other orientation.

In brief, the global orientation detection dendrite neuron will evaluate the information that local orientation detection dendrite neurons calculate from one to another receptive field until all the pixels in the image have become the center pixel of a receptive field in order to get all the orientation information from this image, and finally output the orientation angle that has the highest number of activated somas.

2.4 Learning algorithm of orientation detection dendrite neuron model

Blakemore's theory suggests that neurons with selectivity can develop their function due to external factors from birth [40]. To support this idea, the parameters in the synaptic connection states of the dendrites are changeable through learning. In our approach, the DModel-AVS can learn the position of connections between synapses using our learning algorithm. Given the feed-forward network architecture of our orientation detection system, we choose the widely recognized error backpropagation as our learning algorithm [41]. By regularizing the test result into $Test_u$, we aim to minimize the error E in the model with the highest accuracy using the training

Fig. 5 (a) shows The diagram of the whole system, and (b) is an example of global orientation detection dendrite neuron model, the figure shows a possible results after it learning from a 5×5 image



signal $Train_u$ orientation, and the mean square error function is

$$E = \frac{1}{2} \sum_{k=1}^4 (Train_u - \frac{Test_u}{MN})^2 = \frac{1}{2} \sum_{k=1}^4 (Train_u - \overline{Test_u})^2. \quad (9)$$

In our model, all the synapses of the 9 inputs x_{ijd} ($d = 1, 2, \dots, 9$) in the local orientation detection dendrite neuron are connected to all the branches. The changeable parameters are the weights of synapses w_{ijd} and their biases q_{ijd} , to which the error E is related. After calculation based on chain rules, it is simple to find out the parameter to minimize the error. The detailed calculation is shown in supplementary materials. The weights w_{ijd} and biases q_{ijd} are then updated iteratively to minimize the error E for a highly

accurate detection model. Finally, the entire structure of our DModel-AVS is visualized, as depicted in Fig. 6.

3 Simulation results

The DModel-AVS, an artificial neural network model inspired by biological principles, is developed to address limitations seen in CNN-based orientation detection systems. Specifically, it aims to improve the working mechanism and reduce learning costs. In this section, we will compare DModel-AVS against two kinds of established models in the field: simple structural CNNs, which have simple structure (less layers, less parameters) and novel CNNs recently worked out with stronger functions on image recognition and complex structures.

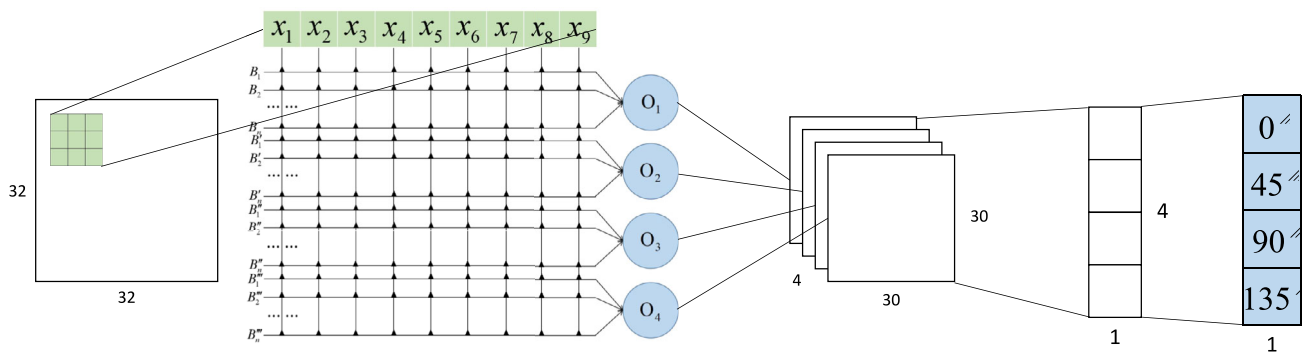


Fig. 6 The whole structure of DModel-AVS

The narrow sense CNN is widely recognized for its efficiency and accuracy in orientation detection [42]. Its capability to accurately detect orientation and handle large datasets, even with high-dimensional inputs, positions it as a leading method in the field [18, 20]. However, CNN has inherent drawbacks, including the need for substantial training data, which may not be readily available in real-world scenarios, and computational intensity that hinders broader development [24].

In contrast, DModel-AVS is designed to mitigate these limitations. To assess its performance comprehensively, we will compare it with both simple structural CNNs and complex structural CNNs-based orientation detection systems.

For the simple structural CNN model, we employ a four-layer CNN and a one-channel CNN. The one-channel CNN incorporates a softmax function after the convolutional layer and can output four orientational angles, as illustrated in Fig. 7. The four-layer CNN, depicted in Fig. 3, consists of three convolution layers with a max-pooling layer and a fully connected layer, is a little more complex than the one-channel CNN. Both simple structural CNNs use one-channel images as inputs.

EfficientNet-B0 is a sophisticated CNN-based orientation detection system designed for fast image detection with three channels. Since binary images have only one channel, we use a background value of 0 to fill the remaining two channels,

resulting in a representation resembling a black square, as shown in Fig. 3.

To expand the comparison, we introduce ResNet50, a deep CNN architecture known for overcoming the vanishing gradient problem and enhancing feature representation. ResNet50 is particularly effective in maintaining accuracy even with increased network depth, reducing parameter count, and computational complexity [21]. It has much more complex structure that the depth is not limited by taking residual into computation. ResNet-50 architecture employs a 3-channel image input layer utilizing LL, LH, and HH components, featuring a hierarchical structure with $\times 2$, $\times 3$, $\times 5$ blocks, including convolution blocks for varying input dimensions and identity blocks for unchanged input dimensions as Fig. 8.

In addition to EfficientNet-B0 and ResNet50, we also employed the ConvNeXt Tiny model to detect the orientation of the dataset. ConvNeXt Tiny is a state-of-the-art convolutional network architecture that incorporates advanced techniques to achieve high accuracy and efficiency. Its design focuses on enhancing performance by incorporating principles from both traditional convolutional networks and modern advancements [43]. The choice to use the ConvNeXt Tiny model is motivated by its ability to balance computational efficiency and accuracy, which is particularly suitable given our relatively simple dataset composition. Using the tiny model instead of base model would be an unnecessary use of resources.

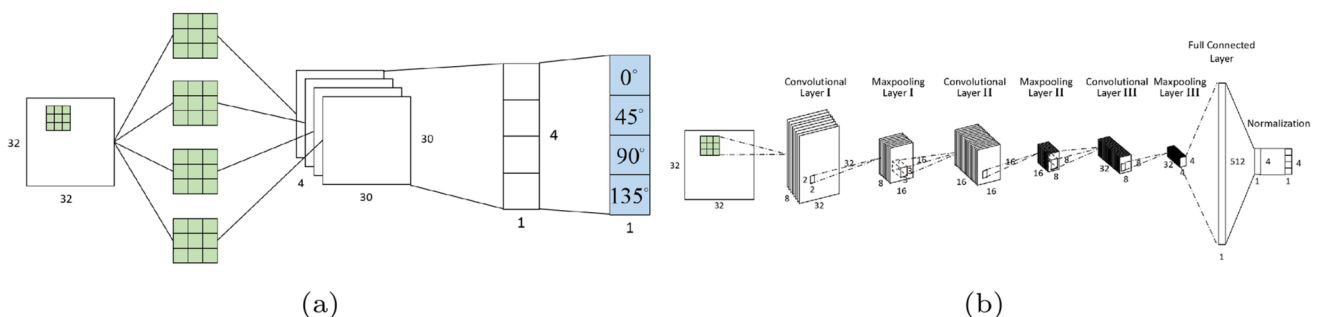
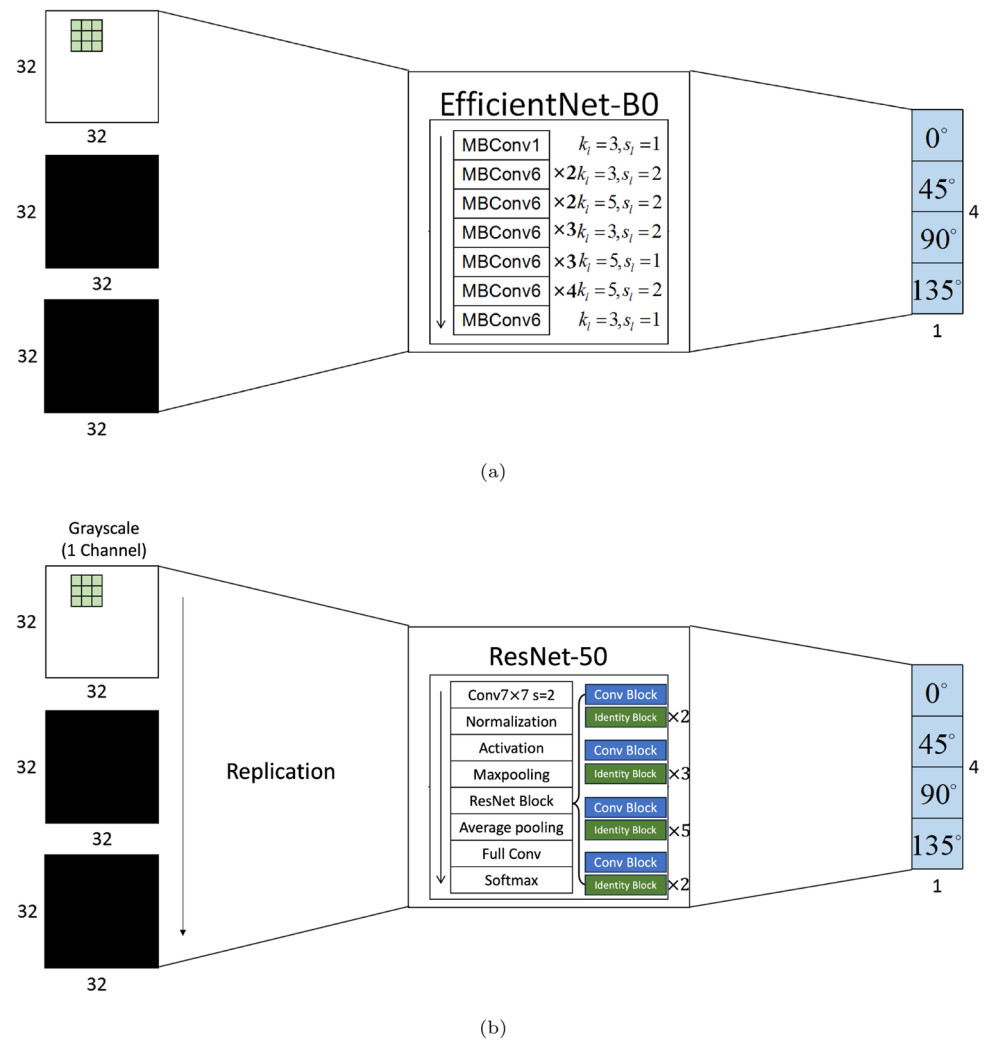


Fig. 7 Structure of (a) one channel CNN and (b) four layer CNN

Fig. 8 Structure of (a) EfficientNet-B0 and (b) ResNet50



This extended comparison aims to provide a more comprehensive evaluation of DModel-AVS against diverse CNN architectures, considering factors such as accuracy, data requirements, and computational efficiency.

3.1 Experiments data

In order to detect the orientation of objects, we designed a dataset of binary (each pixel has a value of 0 or 1) images. The binary dataset includes objects painted 1 and background painted 0. These images are sized 32×32 , all the objects are placed randomly and have different pixels, regularly shaped in order to have a clear orientation of 0° , 45° , 90° and 135° . For example, an image of the object that has 32 pixels and is set as 45° is shown in the Fig. 9. In addition, all the images have regular shape so the number of them can be calculated pixel by pixel and summed in the Table 1

Then we mixed up the size of our images, and run our program for another training and testing. What we got was

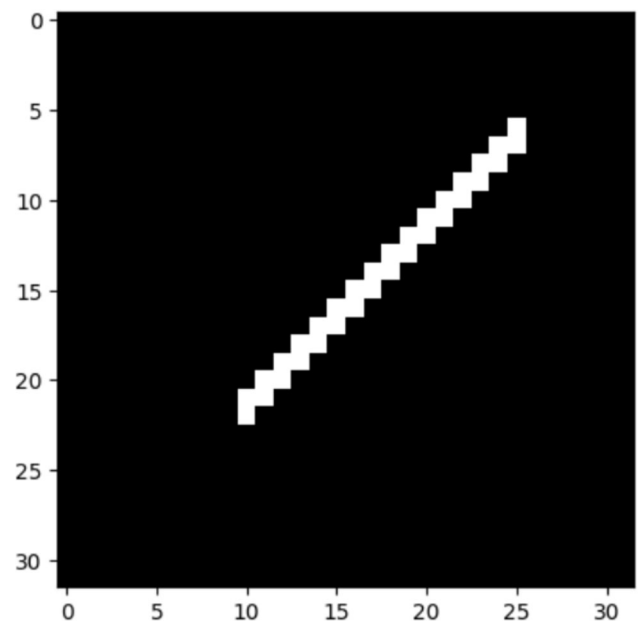


Fig. 9 Example of binary dataset

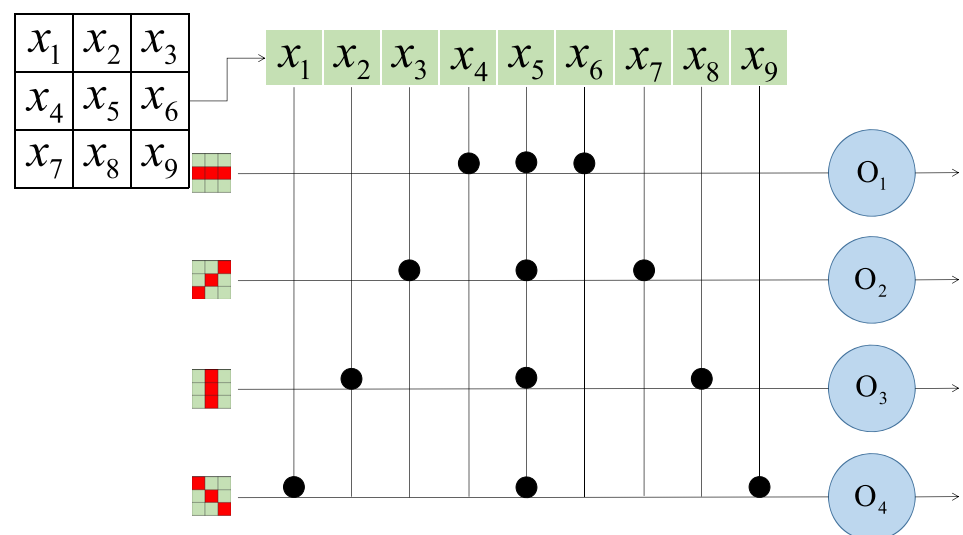
Table 1 Table caption

Object Size	0°	45°	90°	135°
3pixel	960	900	960	900
4pixel	928	841	928	841
8pixel	1699	2249	1699	2249
12pixel	2379	3411	2379	3411
16pixel	1319	1489	1319	1489
32pixel	1284	1645	1284	1645
$\geq 48pixel$	2515	1275	2515	1275

the same accuracy of 100%, and the learning result appeared as Fig. 10.

It is necessary for an orientation detection system to be able to handle noise, as not all images and objects are free from obstructions. Therefore, an important standard to define an artificial visual system is noise immunity. To assess the robustness of our model against noise, we selected three types of images with varying levels of noise, as shown in Fig. 11. Although the image dataset is binary, in order to diversify the noise from object, we painted the background as purple pixels means the 0 input, the objects are white as shown in Fig. 9 means the 1 inputs and noises are added as the 1 inputs appear on the pixel supposed to be 0. First, we added some noise that only appears in the background like Fig. 11(a), none of them are connected with each other so the noise cannot become a new shape with orientation. Another kind of images includes the noise that some of them are allowed to be connected with each other, but still only appear on the background and far from objects just like the image shown in Fig. 11(b). Last kind of images are including all the kinds of noise appearing on the background randomly, the noise can be anywhere of the image which means they have the possibility to connect with each other and the object as Fig. 11(c).

Fig. 10 The learning result of a local orientation-selective dendritic model for 4 orientations



3.2 Simulation experiments

The previous parts mentioned that there are many methods available for detecting orientations, but the most commonly used methods are based on CNN algorithms. These methods are known for their ability to learn and have the highest recognized accuracy. If our DModel-AVS can achieve higher accuracy than these methods, its results will be more reliable and believable. Therefore, it is important to compare an artificial visual system's accuracy with these established methods to assess its effectiveness. For fairness, our DModel-AVS uses the same optimization function with the CNN we take into comparison, the Adam.

3.2.1 Different branch number

Because the number of branches determines the relative number of parameters in the DModel-AVS, we believed it can significantly affect the efficiency of the learning algorithm. In this experiment, we set a batch size at 32, and trained our model 50 epochs for learning. The training data is from our binary data and the size of objects is set at 8 pixels, 12 pixels and 16 pixels since the error of these sizes are lower and have little special situation. The training dataset had 7,500 images and the test data had 2,500 images, at the same level with our CNN comparison. We changed the branch number from 1 to 10 and tried 10 times for average and other statistical dependent variable of accuracy in Table 2, the statistical dependent variables of model performance in Fig. 12.

From our experiments, it can be found that while the branch number increases, the learning result is becoming more stable. However, the learning speed decreases at the same time. For further illustration, we showed the initial position of connections as triangles and circles for excitatory connections, numbers in circles for constant-0 or constant-1

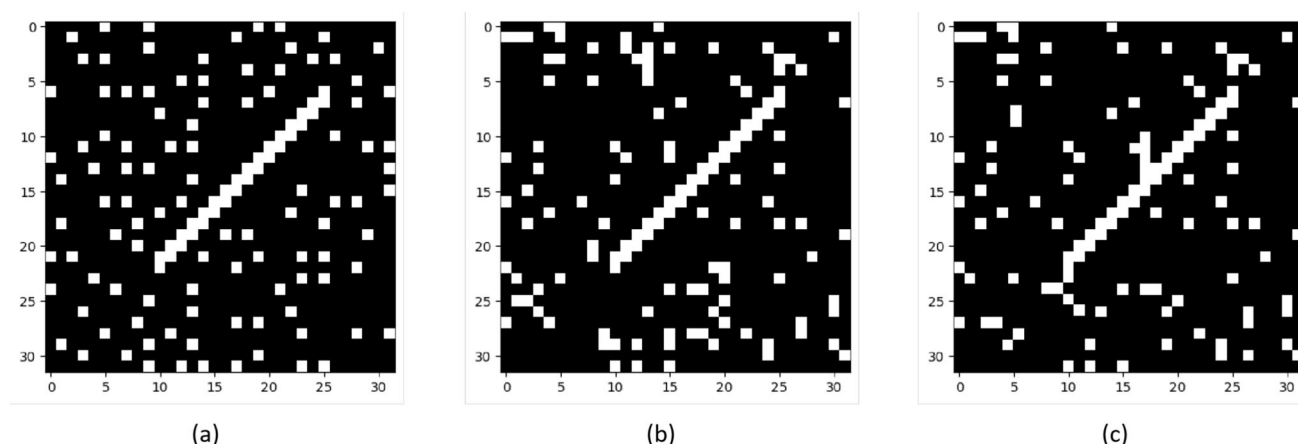


Fig. 11 The images of three types of noise encountered in our experiments on a 32-pixel object at 45°

connection in Fig. 13. We put the most extreme branches number of 10 in Fig. 13(a)(b) and 1 in Fig. 13(c)(d) as example in a certain orientation. Both of the results can be simplified as Fig. 13(e) since the membrane calculates in a linear way, which is fitting the orientation selectivity assumption of Hubel and Wiesel [7].

3.2.2 Different initial value

The initial values of parameters in a learning algorithm can have a significant impact on the performance and convergence of the algorithm. Initializing parameters with appropriate values can help the algorithm converge faster and towards a better solution, while poor initial values may lead to slow convergence or even getting stuck in local minima. Choosing the right initial values for parameters can depend on the specific learning algorithm and the problem being solved. In our DModel-AVS, we use optimization algorithms that iteratively adjust the parameter values to minimize the error or loss function as same. As a result, the initial values of parameters play an important role in determine the performance and efficiency of our model's learning algorithms, and careful consideration should be given to selecting appropriate initial values and updating them during the learning process.

To upgrade the feature sorting ability of our model, we tried to change the parameters' initial value of our model to change the initial combination position of synapses and

branches for the sake of finding out the relationship between it and the learning result. What we did in our experiments is changing all the initial combination position in our model by excitatory position, inhibitory position, constant-0 position, constant-1 position and completely random position. For the fairness of experiments, all the position has different random value generated by different seed value that can simulate the position we need for 10 times. From these experiments, we are able to know how the initial combination position change the learning results. After the experiments, we recorded the result in Table 3 and Fig. 14

3.2.3 Different noise data

We had built three kinds of datasets with noise in order to test our model's noise immunity. In these experiments, we used our model after learning which has 100 % accuracy and the branch number is 1. The percentage in the table means the number of pixels filled with noise compared to the whole image pixels.

In Table 4, we test the dataset with the noise that fits Fig. 11(a). This kind of noise means little influence to our model because the only pixel painted into 1 input in a single receptive field can only change a little local feature. The whole object receive nothing matters to orientation feature for our model but a lot to diverse CNN model.

Table 2 Learning results of different branch number

Dendrite Number	Train Mean	Test Mean	Test Std.	Test Best	Test Worst
1	100%	100%	0	100%	100%
2	100%	100%	0	100%	100%
4	100%	100%	0	100%	100%
5	100%	100%	0	100%	100%
8	100%	100%	0	100%	100%
10	100%	100%	0	100%	100%

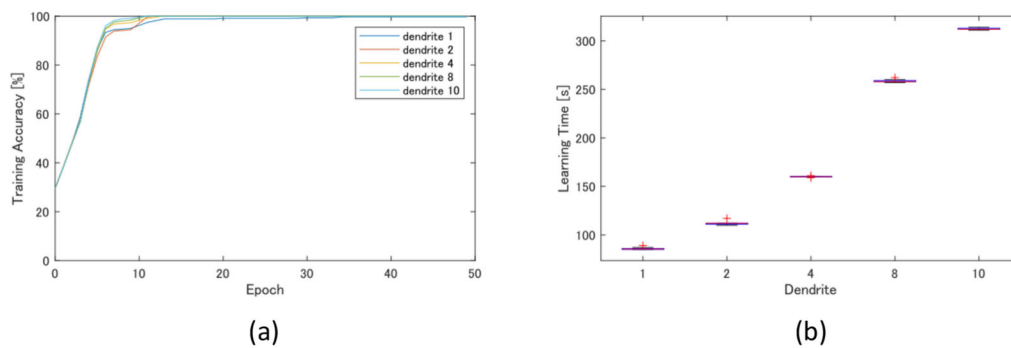


Fig. 12 (a) shows the learning speed of model, while (b) shows the computation time

In Table 5, we test the dataset with the noise that fits Fig. 11(b). This kind of noise is able to combine with each other, which means the possibility of being little size objects with specific orientation and influence the local orientation feature. This kind of noise can change both local orientation feature and global orientation feature. As a result, not only CNN's orientation detection will be influenced by this noise totally, our model is also influenced a lot.

In Table 6, we test the dataset with the noise that fits Fig. 11(c). This kind of noise is able to combine with each other, additionally, combine with the objects. This kind of noise can be found in reality easily include situations such as lighting variations, background clutter, occlusions, and other visual disturbances that are not related to the object of interest. When testing an orientation detection system, it is important to consider the impact of such noise on the system's performance. For example, if there are other objects in

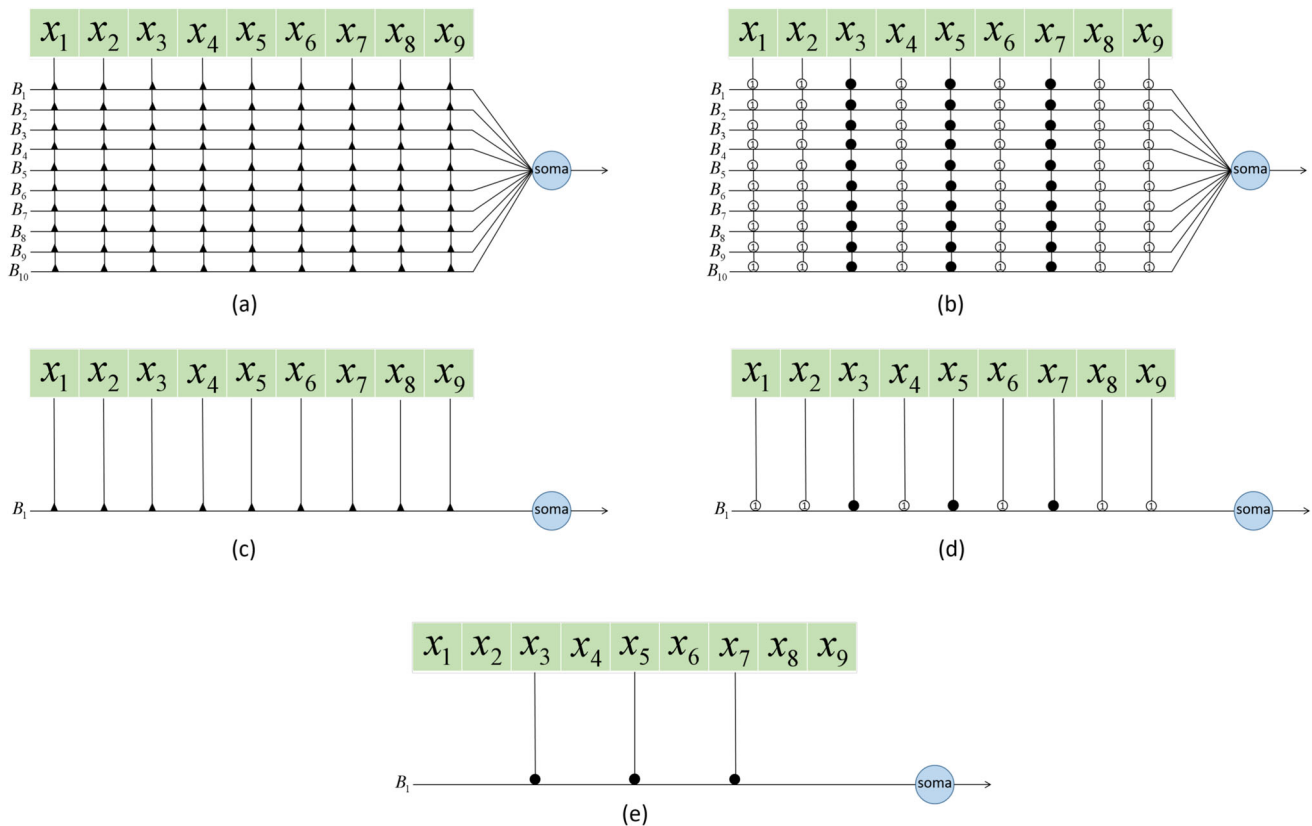


Fig. 13 (a) illustrates the initial position of a 10-branch dendritic neuron, (b) depicts the position of the 10-branch dendritic neuron after learning, (c) showcases the initial position of a 1-branch dendritic neuron, (d) displays the position of the 1-branch dendritic neuron after learning, and (e) demonstrates the simplified position after learning of (b) and (d)

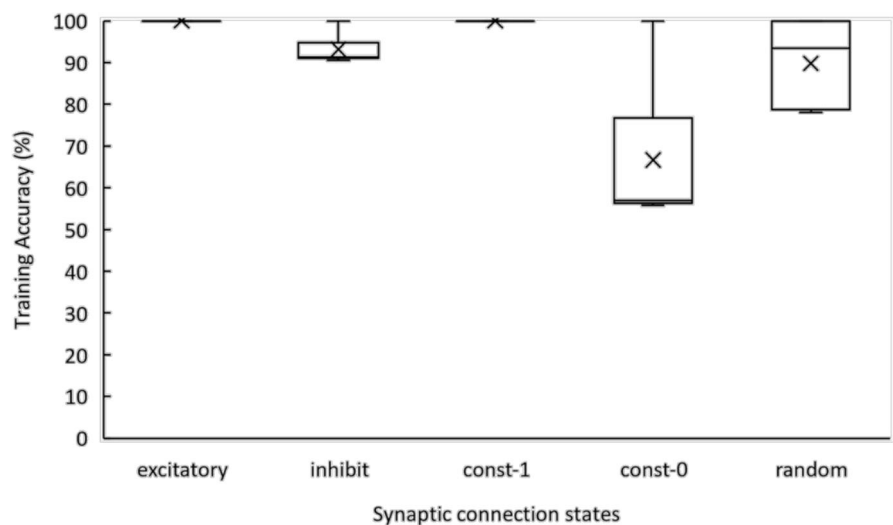
Table 3 Learning results of different initial value

Position	Mean	Std.	Best	Worst
excitatory	100%	0.000	100%	100%
inhibitory	93.2%	3.661	100%	90.6%
constant-0	66.6%	14.88	100%	55.8%
constant-1	100%	0.000	100%	100%
random	89.9%	9.941	100%	78.2%

the background that share similar features with the object of interest, the system may have difficulty accurately detecting the object's orientation. In this occasion, our model and CNN may also struggle to accurately detect the object's orientation.

To comprehensively assess the performance of different models under various noise conditions, a series of simulation experiments were conducted. Table 7 provides key comparative metrics across different models, covering parameters, cost per epoch, accuracy, as well as the number of units, layers, and implementation complexity of the model structure.

The AVS model outshines in various aspects during the experiments. With a minimal 144 parameters, it achieves remarkable computational efficiency, requiring only 10 training epochs for superior performance. The model attains a flawless 100% accuracy, showcasing its effectiveness. Boasting 144 units and a single layer, the AVS model strikes a balance between representational capacity and simplicity. Its straightforward implementation further positions it as an ideal choice for resource-constrained settings. In essence, the AVS model emerges as a compelling solution, delivering competitive results against more complex counterparts while maintaining efficiency and simplicity.

Fig. 14 Accuracy stability of different initial combination

3.2.4 Different training data rate

It is well known that CNN-based orientation detection system needs to catch the features of whole image from convolution and maxpooling. This means that CNN-based orientation detection can only be able to learn a little information from a single image. However, our model is able to extract the feature from each receptive field, which means it can learn from all the receptive field and get the feature which is needed to corresponding orientation. As a result, our model can save a lot of learning cost because of its learning mechanism.

For testing this mechanism, we designed a series of experiments, especially the training data rate. In our experiments, we use 10000 images with given orientation labels as the previous experiments. Then we sorted these images into training and test data with specific percentage. First we did these experiments with 75%, 50%, 10% and 5% images as training data, then tested them with the noise-free images. The result of these experiments is shown in Table 8

Since completely noise-free datasets are uncommon, we incorporated some noise into the training set. These types of noise include the three mentioned earlier. We added various proportions of each type of noise to the original dataset. For each noisy dataset, we selected 10% and added it to the original dataset, resulting in noisy data constituting approximately 25% of the total data. From this new dataset, we randomly selected 10,000 non-repeating images, divided them into training and test sets, and conducted experiments. Similarly, we compared the performance of each model by using different training and test set proportions (75%, 50%, 10%, and 5%) to test the generalizability of our model, the testing results are shown in Table 9.

Based on the experiments, we concluded that under the same conditions, the number of epochs required for each model to learn significantly increased due to the noise added

Table 4 Testing accuracy of images with no-connection noise

Model	Training Acc	1%	2%	5%	10%
DModel-AVS	100 ± 0.00%	100 ± 0.00%	100 ± 0.00%	100 ± 0.00%	100 ± 0.00%
1-Chn CNN	100 ± 0.00%	47.9 ± 22.3%	33.3 ± 18.6%	27.7 ± 15.6%	24.2 ± 17.9%
4-Layer CNN	100 ± 0.00%	90.0 ± 15.2%	77.9 ± 23.8%	60.0 ± 27.5%	47.9 ± 23.9%
EfN-B0	100 ± 0.00%	61.9 ± 17.9%	35.8 ± 18.1%	23.5 ± 17.2%	21.5 ± 16.3%
ResNet-50	100 ± 0.00%	49.6 ± 18.8%	46.2 ± 16.4%	41.7 ± 13.3%	37.9 ± 10.5%
ConvNeXt	100 ± 0.00%	55.3 ± 3.17%	48.5 ± 3.13%	39.1 ± 4.86%	32.2 ± 6.06%

Table 5 Testing accuracy of images with noise-connection noise

Model	Training Acc	1%	2%	5%	10%
DModel-AVS	100 ± 0.00%	100 ± 0.00%	100 ± 0.00%	99.9 ± 0.01%	97.5 ± 0.67%
1-Chn CNN	100 ± 0.00%	49.9 ± 21.7%	34.4 ± 17.6%	27.5 ± 15.6%	24.7 ± 17.1%
4-Layer CNN	100 ± 0.00%	89.5 ± 14.6%	76.2 ± 23.2%	54.5 ± 21.9%	40.8 ± 27.2%
EfN-B0	100 ± 0.00%	62.6 ± 17.2%	36.7 ± 16.6%	24.8 ± 17.5%	23.1 ± 16.9%
ResNet-50	100 ± 0.00%	45.7 ± 18.9%	41.1 ± 16.5%	36.8 ± 12.3%	33.8 ± 9.00%
ConvNeXt	100 ± 0.00%	55.9 ± 3.26%	47.9 ± 3.51%	38.6 ± 4.07%	32.3 ± 3.81%

Table 6 Testing accuracy of images with completely random background noise

Model	Training Acc	1%	2%	5%	10%
DModel-AVS	100 ± 0.00%	99.9 ± 0.00%	99.9 ± 0.02%	99.3 ± 0.09%	94.4 ± 0.13%
1-Chn CNN	100 ± 0.00%	50.2 ± 21.2%	33.8 ± 16.9%	25.9 ± 16.0%	22.2 ± 17.1%
4-Layer CNN	100 ± 0.00%	86.9 ± 15.9%	72.3 ± 22.9%	49.1 ± 21.4%	37.1 ± 22.6%
EfN-B0	100 ± 0.00%	62.0 ± 16.4%	36.3 ± 15.8%	24.0 ± 17.2%	21.6 ± 16.5%
ResNet-50	100 ± 0.00%	40.1 ± 6.57%	31.3 ± 6.32%	28.1 ± 5.50%	25.9 ± 4.67%
ConvNeXt	100 ± 0.00%	34.1 ± 4.57%	28.8 ± 4.23%	25.5 ± 3.70%	24.2 ± 3.05%

Table 7 Comparison of DModel-AVS and CNNs

Attributes	AVS	1-Chn CNN	4-Layer CNN	EfN-B0	ResNet-50	ConvNeXt
Parameters	144	3604	7940	4017796	23516228	27819748
Cost(Epoch)	10	1000	100	10	10	50
Accuracy	100%	100%	100%	100%	100%	100%
Layers	1	2	4	18	50	24
Implementation	Simple	Simple	Simple	Complex	Complex	Complex
Robustness	Strong	Weak	Normal	Normal	Weak	Weak

Table 8 Accuracy of different training data ratios with noisy training sets

Train : Test Model	75:25		50:50	
	Train Acc	Test Acc	Train Acc	Test Acc
DModel-AVS	100 ± 0.00%	100 ± 0.00%	100 ± 0.00%	100 ± 0.00%
1-Chn CNN	100 ± 0.01%	100 ± 0.00%	100 ± 0.00%	100 ± 0.00%
4-Layer CNN	100 ± 0.00%	100 ± 0.00%	100 ± 0.00%	100 ± 0.00%
EfN-B0	99.9 ± 0.08%	100 ± 0.00%	99.8 ± 0.23%	99.9 ± 0.03%
ResNet-50	100 ± 0.00%	100 ± 0.00%	100 ± 0.00%	100 ± 0.00%
ConvNeXt	100 ± 0.00%	100 ± 0.05%	99.6 ± 3.10%	98.8 ± 1.32%
Train : Test	10:90		5:95	
Model	Train Acc	Test Acc	Train Acc	Test Acc
DModel-AVS	100 ± 0.00%	100 ± 0.00%	100 ± 0.00%	100 ± 0.00%
1-Chn CNN	100 ± 0.00%	99.7 ± 0.18%	100 ± 0.00%	98.4 ± 0.68%
4-Layer CNN	100 ± 0.00%	100 ± 0.00%	100 ± 0.00%	100 ± 0.00%
EfN-B0	99.8 ± 0.10%	91.4 ± 0.23%	99.9 ± 0.10%	69.2 ± 3.93%
ResNet-50	99.9 ± 0.00%	100 ± 0.00%	99.7 ± 0.01%	98.6 ± 0.02%
ConvNeXt	99.6 ± 0.17%	94.6 ± 1.86%	99.7 ± 0.38%	93.1 ± 4.09%

to the training set. Although the training accuracy was high for all models, the test accuracy notably decreased, especially when the training and test set ratio exceeded 10:90. Our model was the least affected by the noise in the training set, maintaining high accuracy across all ratios. In contrast, other models were prone to learning noise as part of the features, indicating that our model can effectively recognize the orientation of objects even in typical conditions where noisy data might be included.

For further information of our model, we also do the same experiments by testing images with noise. We saved the learning result and model in Table 8, then tested them with noised images as Fig. 11(b). Because the previous learning results show that this kind of noise did the least change to all kinds

of model. The noise pixels percentage of the whole image's pixel is 1%, 2%, 5% and 10%, and the result is Table 10.

3.2.5 Real objects image data

To comprehensively assess the efficacy of our artificial visual system in object orientation detection, we curated a novel dataset featuring real-world objects captured in diverse scenarios. The dataset comprises a broad spectrum of objects recognizable to the human eye, such as commonplace office items and scenes with a neutral background, totaling 10,000 images. These objects, including pencils, planes, and other commonly encountered items, were selected for their prevalence and detectable orientations. To enhance

Table 9 Accuracy of different training data rate when noise appears

Train : Test Model	75:25		50:50	
	Train Acc	Test Acc	Train Acc	Test Acc
DModel-AVS	96.2 ± 2.21%	97.7 ± 1.35%	96.0 ± 1.27%	97.6 ± 2.26%
1-Chn CNN	88.8 ± 7.61%	73.5 ± 1.15%	89.4 ± 7.23%	69.4 ± 1.31%
4-Layer CNN	97.2 ± 4.78%	95.2 ± 7.68%	91.8 ± 2.92%	91.6 ± 2.14%
EfN-B0	100 ± 0.02%	96.8 ± 0.44%	99.1 ± 0.22%	95.2 ± 0.68%
ResNet-50	100 ± 0.00%	95.0 ± 0.49%	98.4 ± 1.09%	93.8 ± 0.60%
ConvNeXt	99.8 ± 0.11%	90.9 ± 1.67%	99.5 ± 0.43%	90.1 ± 1.83%
Train : Test	10:90		5:95	
Model	Train Acc	Test Acc	Train Acc	Test Acc
DModel-AVS	96.3 ± 0.49%	97.5 ± 0.17%	96.4 ± 0.81%	97.4 ± 0.29%
1-Chn CNN	97.3 ± 1.12%	54.2 ± 1.63%	98.7 ± 0.53%	43.0 ± 0.53%
4-Layer CNN	91.0 ± 5.09%	86.4 ± 3.11%	95.3 ± 4.85%	84.6 ± 2.94%
EfN-B0	99.7 ± 0.14%	80.1 ± 6.98%	99.4 ± 0.34%	78.3 ± 2.52%
ResNet-50	100 ± 0.00%	76.8 ± 2.44%	100 ± 0.00%	56.5 ± 6.45%
ConvNeXt	99.5 ± 0.36%	79.4 ± 3.95%	99.5 ± 0.10%	71.1 ± 8.67%

Table 10 Accuracy of different training data rate with background noise

75:25	1%	2%	5%	10%
DModel-AVS	100 ± 0.00%	100 ± 0.00%	100 ± 0.00%	100 ± 0.00%
1-Chn CNN	90.0 ± 15.2%	77.9 ± 23.8%	60.0 ± 27.5%	47.9 ± 23.9%
4-Layer CNN	47.9 ± 22.3%	33.3 ± 18.6%	27.7 ± 15.6%	24.2 ± 17.9%
EfN-B0	62.6 ± 17.2%	36.7 ± 16.6%	24.8 ± 17.5%	23.1 ± 16.9%
ResNet-50	40.1 ± 6.57%	31.3 ± 6.32%	28.1 ± 5.50%	25.9 ± 4.68%
ConvNeXt	34.1 ± 4.57%	28.8 ± 4.23%	25.5 ± 3.70%	24.2 ± 3.05%
50:50	1%	2%	5%	10%
DModel-AVS	100 ± 0.00%	100 ± 0.00%	100 ± 0.00%	100 ± 0.00%
1-Chn CNN	87.2 ± 16.8%	74.8 ± 26.4%	60.4 ± 28.7%	53.7 ± 22.6%
4-Layer CNN	52.5 ± 23.3%	35.5 ± 20.5%	28.1 ± 15.8%	24.1 ± 18.4%
EfN-B0	64.9 ± 19.5%	45.2 ± 17.9%	27.7 ± 18.83%	21.6 ± 19.2%
ResNet-50	45.0 ± 9.76%	35.7 ± 7.25%	32.1 ± 5.72%	30.2 ± 4.73%
ConvNeXt	33.3 ± 4.05%	28.3 ± 3.70%	26.5 ± 3.80%	25.2 ± 3.68%
10:90	1%	2%	5%	10%
DModel-AVS	100 ± 0.00%	100 ± 0.00%	100 ± 0.00%	100 ± 0.00%
1-Chn CNN	82.9 ± 22.5%	69.6 ± 28.5%	50.5 ± 28.2%	37.9 ± 18.6%
4-Layer CNN	55.0 ± 23.3%	35.2 ± 27.8%	23.4 ± 19.0%	20.0 ± 19.5%
EfN-B0	42.8 ± 9.27%	31.6 ± 12.7%	26.7 ± 16.9%	26.8 ± 19.2%
ResNet-50	42.7 ± 6.94%	34.2 ± 5.64%	29.9 ± 4.26%	28.0 ± 3.15%
ConvNeXt	34.4 ± 2.98%	28.7 ± 3.34%	26.8 ± 3.71%	26.4 ± 3.37%
5:95	1%	2%	5%	10%
DModel-AVS	100 ± 0.00%	100 ± 0.00%	100 ± 0.00%	100 ± 0.00%
1-Chn CNN	74.6 ± 21.6%	61.1 ± 24.3%	48.5 ± 18.4%	48.7 ± 19.1%
4-Layer CNN	55.3 ± 18.8%	34.0 ± 21.3%	20.9 ± 19.6%	16.3 ± 18.9%
EfN-B0	32.4 ± 15.7%	29.4 ± 18.0%	25.8 ± 21.9%	24.0 ± 23.4%
ResNet-50	37.2 ± 7.09%	31.2 ± 4.91%	28.5 ± 3.24%	27.2 ± 2.19%
ConvNeXt	33.2 ± 4.09%	28.2 ± 3.20%	26.3 ± 3.25%	25.4 ± 3.05%

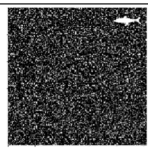
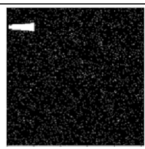
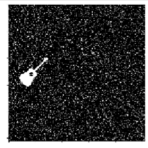
Image	Orientation	Detected	Image	Orientation	Detected
 Noised Small	0°	0°	 Noised Small	0°	0°
 Noised Small	45°	45°	 No Noise Large	135°	135°
 No Noise Small	90°	90°	 Noised Large	90°	90°

Fig. 15 Samples of detection results and corresponding reality object images

Table 11 Accuracy of real objects detection

Noise	None	1%	5%	10%	20%
DModel-AVS	100 ± 0.00%	99.9 ± 0.06%	98.5 ± 0.68%	91.6 ± 8.56%	72.5 ± 11.7%
1-Chn CNN	71.5 ± 12.6%	31.8 ± 11.6%	30.8 ± 10.4%	30.0 ± 9.26%	29.5 ± 8.54%
4-Layer CNN	30.0 ± 11.6%	26.3 ± 0.01%	25.6 ± 0.01%	24.8 ± 0.55%	25.0 ± 0.02%
EfN-B0	42.5 ± 10.8%	25.0 ± 0.64%	25.1 ± 0.52%	25.1 ± 0.53%	25.1 ± 0.53%
ResNet-50	51.6 ± 15.9%	29.4 ± 5.52%	26.4 ± 2.19%	26.5 ± 2.89%	27.8 ± 3.93%
ConvNeXt	35.6 ± 0.12%	26.9 ± 2.31%	28.8 ± 4.13%	24.2 ± 3.26%	26.3 ± 4.74%

dataset diversity, we applied preprocessing techniques, such as image resizing and data augmentation, adjusting object sizes and orientation angles. Some images, sized at 256×256 , facilitate easy comprehension of object type and orientation angle for human observers, and we labeled them accordingly through observation. This dataset poses a unique challenge for our DModel-AVS, demanding accurate orientation detection in real-world scenarios. Evaluating our system on this dataset enables a better understanding of its capabilities and limitations, facilitating further enhancements for real-world applications. The ultimate goal of an artificial visual system is practical application in reality, necessitating testing with real-world images. To address this, we utilized reality image data featuring different objects like carrots, guitars, and airplanes. The model employed in our experiments is the one saved in Table 2, achieving 100% accuracy. Similarly, CNN-based orientation detection systems utilized the saved model in the experiments. Initiating our assessment, we used our model to test specific images in Fig. 15, obtaining correct results at different position (large or small objects on the noised or not backgrounds). Subsequently, we applied the model to the entire dataset and introduced noise. The detecting results are shown in Table 11. Thankfully, this resulted in highly accurate detection. To further challenge the system, the results were compared with different kinds of CNN-based orientation detection systems. The models trained on our regular geometric (0,1) dataset showed that other models struggled to achieve accurate detection on the silhouette (0,1) data of actual objects and were easily affected by noise. It is speculated that such CNN-based models, which can only consider overall features, find it difficult to capture minor variations and tend to learn in an overly specific manner.

4 Conclusion

In our study, we introduced a novel learning artificial visual system for orientation detection based on the dendritic neuron model, achieving high accuracy. The model consists of a local orientation detection part and a global orientation detection part. The local orientation detection part can be trained

to detect the orientation of a single receptive field consisting of 9 pixels. It employs the dendritic neuron model structure and trains the synapses and branches, enabling learning from datasets using the backpropagation algorithm. During the learning process, the model calculates the relationship between the detected orientation and the labeled orientation, minimizing errors similarly to other backpropagation-based learning algorithms. The final local orientation detection results become output and be passed to the global orientation detection part in our model. The global orientation detection part aggregates the local orientation detection results, by comparing the activated neurons for corresponding orientations. The final result corresponds to the orientation with the highest number of activated neurons.

One notable merit of our model is its ability to accurately detect orientation while reducing computational complexity, meaning it efficiently uses time and resources. Furthermore, our model decreases dataset requirements and learning costs, as demonstrated through experiments. Instead of extracting features from the whole image, our model extracts them from single receptive field. This demonstrates the robustness and efficiency of our model, making it highly suitable for applications with limited data. Supporting this point, when we reduce the percentage of training data in the entire dataset, the accuracy of our model does not decrease, while CNN-based systems decrease dramatically. This is because CNNs process entire images, capturing both global and local features, whereas our model focuses on localized, biologically inspired feature extraction. Additionally, our proposed system is theoretically supported by the dendritic neuron. It not only aligns Hubel and Wiesel's theory of simple cells and complex cells' functions on orientation selectivity, but also considers the function of dendritic cells in neural networks.

Compared to our previous model, the current model possesses learning capabilities that more closely mirror human behavior. We assume that humans can judge light spots and slices from birth but require time and a learning process to detect the concept of orientation. This nuanced understanding of orientation detection is reflected in our model, which simulates this process more accurately than traditional CNNs. Our model simulates this behavior, making it more biologically based and scientifically provable in brain research. This

alignment with biological processes enhances the model's credibility and potential impact in neuroscience.

On one hand, our model is explainable, using the dendritic model, unlike the black-box CNN models. Our model is highlighted as a possible solution for situations with limited data for learning and can be applied to pathology figures, geography measures, and other scenarios requiring orientation detection of binary images. This model presents a potential avenue for researchers to discover the brain's learning mechanisms and visual system structure. The proposed mechanism successfully explains how the retina transmits visual information to the brain and how this information and stimuli are received and processed by the brain. Furthermore, our model demonstrates the potential of expanding the receptive field and exploring grayscale and colored image orientation detection in the future.

For example, we plan to incorporate additional biological neuron models from the retina into our system to realize further functions, such as using horizontal cells for grayscale detection or expanding the receptive field for detecting a wider range of angles. In summary, our model holds significance in both computer science and biological brain science, offering insights into the human brain's orientation detection mechanisms and contributing to understanding the biological structure of the visual system. This dual significance underscores the model's potential as a bridge between computational and biological research, paving the way for future innovations in artificial intelligence and neuroscience.

Appendix A: Supplementary materials

A.1 Formulation in learning algorithm

Initially, the input x_{ijd} ($d = 1, 2, \dots, 9$) are all applied to the branches of the neurons through their own independent synapses. We change the values of the synaptic connection parameters p_{ijd} and q_{ijd} so that the error E is reduced with

$$\Delta w_{ijd}^{uv} = -\eta \frac{\partial E}{\partial w_{ijd}^{uv}}, \quad (A1)$$

$$\Delta q_{ijd}^{uv} = -\eta \frac{\partial E}{\partial q_{ijd}^{uv}}. \quad (A2)$$

Following the chain rule, we can divide the relationship between err E and each parameter into

$$\frac{\partial E}{\partial w_{ijd}^{uv}} = \frac{\partial E}{\partial \overline{Test}_u} \cdot \frac{\partial \overline{Test}_u}{\partial Test_u} \cdot \frac{\partial Test_u}{\partial O_u^k} \cdot \frac{\partial O_u^k}{\partial b_{uv}} \cdot \frac{\partial b_{uv}}{\partial S_{ijd}^{uv}} \cdot \frac{\partial S_{ijd}^{uv}}{\partial w_{ijd}^{uv}}, \quad (A3)$$

$$\frac{\partial E}{\partial q_{ijd}^{uv}} = \frac{\partial E}{\partial \overline{Test}_u} \cdot \frac{\partial \overline{Test}_u}{\partial Test_u} \cdot \frac{\partial Test_u}{\partial O_u^k} \cdot \frac{\partial O_u^k}{\partial b_{uv}} \cdot \frac{\partial b_{uv}}{\partial S_{ijd}^{uv}} \cdot \frac{\partial S_{ijd}^{uv}}{\partial q_{ijd}^{uv}}. \quad (A4)$$

In the previous formulations (5) (6) (7) (9), some parts of the differential chain is easy to be calculated:

$$\frac{\partial E}{\partial \overline{Test}_u} = \overline{Test}_u - Train_u, \quad (A5)$$

$$\frac{\partial \overline{Test}_u}{\partial Test_u} = \frac{1}{NM}, \quad (A6)$$

$$\frac{\partial Test_u}{\partial O_u^k} = 1, \quad (A7)$$

$$\frac{\partial O_u^k}{\partial b_{uv}^k} = \frac{w_{uv}^k \lambda \exp(-\lambda(w_{uv} - \theta_u))}{(1 + \exp(-\lambda(w_{uv} - \theta_u)))^2}, \quad (A8)$$

$$\frac{\partial b_{uv}^k}{\partial S_{ijd}^{uv}} = \frac{S_{ij}^{uv} \prod_{s=1}^9 S_{ijs}^{uv}}{S_{ijd}^{uv}}, \quad (A9)$$

$$\frac{\partial S_{ijd}^{uv}}{\partial w_{ijd}^{uv}} = \frac{\frac{1}{d_{ijd}^{uv}} x_{ijd} \exp(-\frac{1}{d_{ijd}^{uv}} (x_{ijd} w_{ijd}^{uv} - q_{ijd}^{uv}))}{(1 + \exp(-\frac{1}{d_{ijd}^{uv}} (x_{ijd} w_{ijd}^{uv} - q_{ijd}^{uv})))^2}, \quad (A10)$$

$$\frac{\partial S_{ijd}^{uv}}{\partial p_{ijd}^{uv}} = \frac{-\frac{1}{d_{ijd}^{uv}} \exp(-\frac{1}{d_{ijd}^{uv}} (x_{ijd} p_{ijd}^{uv} - q_{ijd}^{uv}))}{(1 + \exp(-\frac{1}{d_{ijd}^{uv}} (x_{ijd} p_{ijd}^{uv} - q_{ijd}^{uv})))^2}. \quad (A11)$$

Acknowledgements This work was supported by JST SPRING, Grant Number JPMJSP2135 and JSPS KAKENHI, Grant Number 23K11261.

Author Contributions Tianqi Chen: Conceptualization, Methodology, Investigation, Software, Writing - Original Draft Preparation. Yuki Kobayashi: Data curation, Formal analysis, Methodology, Writing - Review & Editing. Chenyang Yan: Investigation, Software, Visualization. Zhiyu Qiu: Project administration, Resources, Supervision. Yuxiao Hua: Validation, Writing - Review & Editing, Funding acquisition. Yuki Todo: Conceptualization, Methodology. Zheng Tang: Conceptualization, Data curation, Writing - Review & Editing.

Data Availability Statement The data used in this study are limited access but available for reasonable requirement. Interested parties may request access by contacting corresponding author at ykudo@se.kanazawa-u.ac.jp. Access is subject to approval and compliance with confidentiality and ethical guidelines. The authors are committed to facilitating access within legal and ethical boundaries. ResNet-50 Result: [Simulation_Result](#); Efn-B0 Result: [Simulation_Result](#); DModel-AVS Result: [All_Result](#); Added Experiments(noised training data): [Mixed_Training_Data](#)

Declarations

Conflict of Interest There is no conflict of interest in this paper.

Ethical and informed consent for data used There is no ethical or informed consent for data used in this paper.

References

1. Todo Y, Tang Z, Todo H, Ji J, Yamashita K (2019) Neurons with multiplicative interactions of nonlinear synapses. *Int J Neural Syst* 29(08):1950012

2. Laughlin SB (2001) The metabolic cost of information-a fundamental factor in visual ecology. In: *Ecology of Sensing*, Springer, pp 169–185
3. Medina J (2011) *Brain Rules: 12 Principles for Surviving and Thriving at Work, Home, and School*. ReadHowYouWant. com, ???
4. Vanston JE, Strother L (2017) Sex differences in the human visual system. *J Neurosci Res* 95(1–2):617–625
5. Hubel DH, Wiesel TN (1959) Receptive fields of single neurones in the cat's striate cortex. *J Physiol* 148(3):574
6. McLaughlin D, Shapley R, Shelley M, Wielaard DJ (2000) A neuronal network model of macaque primary visual cortex (v1): Orientation selectivity and dynamics in the input layer 4c α . *Proceed National Academy Sci* 97(14):8087–8092
7. Hubel DH, Wiesel TN (1963) Shape and arrangement of columns in cat's striate cortex. *J Physiol* 165(3):559
8. Hubel DH, Wiesel TN (1962) Receptive fields, binocular interaction and functional architecture in the cat's visual cortex. *J Physiol* 160(1):106
9. Hubel DH (1982) Exploration of the primary visual cortex, 1955–78. *Nature* 299(5883):515–524
10. Li B, Todo Y, Tang Z (2022) Artificial visual system for orientation detection based on hubel-wiesel model. *Brain Sci* 12(4):470
11. Chen T, Li B, Todo Y (2022) Orientation detection system based on edge-orientation selective neurons. *Electronics* 11(23):3946
12. Johnson SP (2020) Chapter 16 - development of the visual system. In: Rubenstein J, Rakic P, Chen B, Kwan KY (eds.) *Neural Circuit and Cognitive Development* (Second Edition), Second edition edn., pp. 335–358. Academic Press, ??? <https://doi.org/10.1016/B978-0-12-814411-4.00016-0> <https://www.sciencedirect.com/science/article/pii/B9780128144114000160>
13. Xu Yuelei LS, Chao LV (2017) Review of development of visual neural computing. *Comput Eng Appl* 53(24):30–34
14. Castellani M, Rowlands H (2009) Evolutionary artificial neural network design and training for wood veneer classification. *Eng Appl Artif Intell* 22(4–5):732–741
15. Du K-L (2010) Clustering: A neural network approach. *Neural Netw* 23(1):89–107
16. Taormina R, Chau K-W, Sethi R (2012) Artificial neural network simulation of hourly groundwater levels in a coastal aquifer system of the venice lagoon. *Eng Appl Artif Intell* 25(8):1670–1676
17. Weimer D, Thamer H, Scholz-Reiter B (2013) Learning defect classifiers for textured surfaces using neural networks and statistical feature representations. *Procedia CIRP* 7:347–352
18. LeCun Y, Bengio Y, Hinton G (2015) Deep learning. *Nature* 521(7553):436–444
19. Blakemore C, Cooper GF (1970) Development of the brain depends on the visual environment. *Nature* 228(5270):477–478
20. Jaderberg M, Simonyan K, Zisserman A, et al (2015) Spatial transformer networks. *Adv Neural Inf Process Syst* 28
21. He K, Zhang X, Ren S, Sun J (2016) Deep residual learning for image recognition. In: *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp 770–778
22. Tao S, Zhang X, Hua Y, Tang Z, Todo Y (2023) A novel artificial visual system for motion direction detection with completely modeled retinal direction-selective pathway. *Mathematics* 11(17):3732
23. Chen T, Kobayashi Y, Todo Y, Tang Z (2024) A learning artificial visual system for motion direction detection. *Neural Comput Appl* 1–17
24. Krizhevsky A, Sutskever I, Hinton GE (2017) Imagenet classification with deep convolutional neural networks. *Commun ACM* 60(6):84–90
25. Zhu MM, Xu YL, Ma HQ (2018) Edge detection based on the characteristic of primary visual cortex cells. *J Phys: Conference Series* 960:012052
26. Maweu BM, Dakshit S, Shamsuddin R, Prabhakaran B (2021) Cefes: a cnn explainable framework for ecg signals. *Artif Intell Med* 115:102059
27. Rakkiyappan R, Kumari EU, Chandrasekar A, Krishnasamy R (2016) Synchronization and periodicity of coupled inertial memristive neural networks with supremums. *Neurocomputing* 214:739–749
28. Aslam MS, Radhika T, Chandrasekar A, Zhu Q (2024) Improved event-triggered-based output tracking for a class of delayed networked t-s fuzzy systems. *Int J Fuzzy Syst* 1–14
29. Marsalli M (2006) Mcculloch-pitts neurons. In: *The 2008 Annual meeting of the consortium on cognitive science instruction (ccsi)*, vol. 1161, pp 1162
30. Rossi LF, Harris KD, Carandini M (2020) Spatial connectivity matches direction selectivity in visual cortex. *Nature* 588(7839):648–652
31. Vlasits A, Baden T (2019) Motion vision: A new mechanism in the mammalian retina. *Current Biol* 29(19):933–935
32. Hamburger C (2016) Active dendritic properties and local inhibitory input enable selectivity for object motion in mouse superior colliculus neurons. *J Neurosci* 36(35):9111–9123
33. Koch C, Poggio T, Torre V (1982) Retinal ganglion cells: a functional interpretation of dendritic morphology. *Philosophical Transactions of the Royal Society of London. B, Biological Sciences* 298(1090):227–263
34. Liu G, Wang J (2021) Dendrite net: a white-box module for classification, regression, and system identification. *IEEE Trans Cybernet* 52(12):13774–13787
35. Gerstner W, Kempter R, Van Hemmen JL, Wagner H (1996) A neuronal learning rule for sub-millisecond temporal coding. *Nature* 383(6595):76–78
36. Wang R-L, Lei Z, Zhang Z, Gao S (2022) Dendritic convolutional neural network. *IEEJ Trans Electrical Electron Eng* 17(2):302–304
37. Henning M, Ramos-Traslosheros G, Gür B, Silies M (2022) Populations of local direction-selective cells encode global motion patterns generated by self-motion. *Sci Adv* 8(31):7112
38. Carandini M, Heeger DJ (1994) Summation and division by neurons in primate visual cortex. *Science* 264(5163):1333–1336
39. Antinucci P, Suleyman O, Monfries C, Hindges R (2016) Neural mechanisms generating orientation selectivity in the retina. *Current Biol* 26(14):1802–1815
40. Blakemore C, Cooper GF (1970) Development of the brain depends on the visual environment. *Nature* 228(5270):477–478
41. Rumelhart DE, Durbin R, Golden R, Chauvin Y (1995) Backpropagation: The basic theory. *Backpropagation: Theory, architectures and applications* 1–34
42. Tan M, Le Q (2019) Efficientnet: Rethinking model scaling for convolutional neural networks. In: *International Conference on Machine Learning*, PMLR, pp 6105–6114
43. Liu Z, Mao H, Wu C-Y, Feichtenhofer C, Darrell T, Xie S (2022) A convnet for the 2020s. In: *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pp 11976–11986

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

Springer Nature or its licensor (e.g. a society or other partner) holds exclusive rights to this article under a publishing agreement with the author(s) or other rightsholder(s); author self-archiving of the accepted manuscript version of this article is solely governed by the terms of such publishing agreement and applicable law.

Authors and Affiliations

Tianqi Chen¹ · Yuki Kobayashi¹ · Chenyang Yan¹ · Zhiyu Qiu¹ · Yuxiao Hua¹ · Yuki Todo² · Zheng Tang^{3,4}

✉ Yuki Todo
yktdo@se.kanazawa-u.ac.jp

Tianqi Chen
chentianqi@stu.kanazawa-u.ac.jp

Yuki Kobayashi
bard8196@stu.kanazawa-u.ac.jp

Chenyang Yan
yy772013879@gmail.com

Zhiyu Qiu
qiuzy1916@stu.kanazawa-u.ac.jp

Yuxiao Hua
sh13818971028@gmail.com

Zheng Tang
tang@iaii.ac.cn

¹ Division of Electrical, Engineering and Computer Science, Graduate School of Natural Science & Technology, Kanazawa University, Kakuma, Kanazawa, Ishikawa 9201192, Japan

² Faculty of of Electrical, Information and Communication Engineering, Kanazawa University, Kakuma, Kanazawa, Ishikawa 9201192, Japan

³ Institute of AI for Industries, Chinese Academy of Sciences, 168 Tianquan Road, Nanjing, China

⁴ School of Computer Engineering and Science, Shanghai University, Shanghai 200444, China