applied sciences

MDPI

Article

An Image Edge Detection Algorithm Based on an Artificial Plant Community

Zhengying Cai *, Zhe Ma, Ziyi Zuo, Yafei Xiang and Mingtao Wang

College of Computer and Information Technology, China Three Gorges University, Yichang 443002, China
* Correspondence: caizhengying@ctgu.edu.cn

Abstract: Image edge detection is a difficult task, because it requires the accurate removal of irrelevant pixels, while retaining important pixels that describe the image's structural properties. Here, an artificial plant community algorithm is proposed to aid in the solving of the image edge detection problem. First, the image edge detection problem is modeled as an objective function of an artificial plant community searching for water sources and nutrients. After many iterations, the artificial plant community is concentrated in habitable areas that are rich in water sources and nutrients, that is, the image edges, and the nonhabitable zones that are not suitable for living are deserted, that is, the nonedges. Second, an artificial plant community algorithm is designed to solve the objective function by simulating the growth process of a true plant community. The living behavior of the artificial plant community includes three operations: seeding, growing, and fruiting. The individuals in the plant community also correspond to three forms, namely seeds, individuals, and fruit. There are three fitness comparisons in each iteration. The first fitness comparison of each iteration is carried out during the seeding operation. Only the fruit with higher fitness levels in the last iteration can become seeds, while the fruit with low fitness levels die, and some new seeds are randomly generated. The second fitness comparison is implemented in the growing operation. Only the seeds with higher fitness levels can become individuals, but the seeds with lower fitness levels will die; thus, the community size will decrease. The third fitness comparison is in the fruiting operation, where the individual with the greatest fitness can produce an identical fruit through parthenogenesis, and the individuals with higher fitness levels can learn from each other and produce more fruit, so the population size can be restored. Through the continuous cycle of these three operations, the artificial plant community will finally determine the edge pixels and delete the nonedge pixels. Third, the experiment results reveal how the proposed algorithm generates the edge image, and the comparative results demonstrate that the proposed artificial plant community algorithm can effectively solve the image edge detection problems. Finally, this study and some limitations are summarized, and future directions are suggested. The proposed algorithm is expected to act as a new research tool for solving various complex problems.

Keywords: image edge detection; image processing; artificial intelligence; plant community

check for updates

Citation: Cai, Z.; Ma, Z.; Zuo, Z.; Xiang, Y.; Wang, M. An Image Edge Detection Algorithm Based on an Artificial Plant Community. *Appl. Sci.* **2023**, *13*, 4159. https://doi.org/10.3390/app13074159

Academic Editor: Juan A. Gómez-Pulido

Received: 1 March 2023
Revised: 16 March 2023
Accepted: 23 March 2023
Published: 24 March 2023

1. Introduction

Image edge detection is an important topic in feature extraction and computer vision. It can greatly reduce the amount of data, eliminate irrelevant information, and retain the important structural attributes of an image. Image edge detection has been widely used in many areas, including spatial wave measurement [1], shape identification [2], prostate ultrasound image analysis [3], license plate recognition [4], COVID-19 detection [5], real-time moving ship detection [6,7], facial recognition [8], weak and small target infrared detection [9,10], three-dimensional microstructure reconstructions and visualizations [11], and low-light image enhancement [12]. The edge may be related to the angle of view and can vary with it. For example, different scenes and object geometries will block other objects from different angles of view. The edge may also be independent of the angle of view, depending on the properties of the object being observed, such as the surface texture and surface shape [2,13]. However, the edge of the natural image is not always the ideal ladder edge. On the contrary, it is usually affected by one or more factors [9], including focus blur caused by a limited scene depth, penumbra blur caused by shadows generated by nonzero radius light sources, shadows on the edges of smooth objects, and the local specular or the diffuse reflections near the edges of objects.

In edge detection, it is necessary to quickly search for a set of pixels with sharp changes in the surrounding pixel gray level from amongst a large number of image pixels, which is not an easy task. The edge is the most basic feature of an image. It exists between the target, background, and region [14] and is the most important basis for image segmentation. Due to the difficulty of the edge detection task, many researchers have developed algorithms to improve the solving performance. Some heuristic and artificial intelligence(AI) algorithms have been widely applied in image edge detection and image processing in recent years, such as the artificial neural network (ANN) [11], the U-net [1], the depth-perceptual network [3], the long short-term memory (LSTM) [5], the edge-net [7], the Y-net [12], the convolutional neural networks (CNNs) [10–12], deep learning (DL) [13–15], machine learning [16], fuzzy logic [8,10], particle swarm optimization (PSO) [14], ant colony optimization (ACO) [15,17], the genetic algorithm (GA) [18], the support vector machine (SVM) [18], the artificial bee colony (ABC) [5], the artificial fish swarm algorithm (AFSA) [19], the salp swarm algorithm (SSA) [20], bird swarm optimization (BSO) [21], and simulated annealing (SA) [22]. In these algorithms, each individual in the group has only simple intelligence, and complex intelligent behaviors are shown through cooperation with each other. In image edge detection, the core of swarm intelligence is that a group composed of many simple individuals can achieve a certain function and perform the edge detection task together, through simple cooperation with each other.

However, the accuracy and speed of the algorithm are often in conflict. Complex algorithms often have high accuracy in image segmentation, but they take too long. Most artificial intelligence algorithms focus on the behaviors of animals with nerves, i.e., ANN [15], CNN [14–17], DL [17–19], PSO [14], ACO [19,21], ABC [5], GA [22], AFSA [23], SSA [24], BSO [25], and naked mole-rat algorithm(NMRA) [26], but few people pay attention to the behaviors of plants. Therefore, we reviewed recent research on biological intelligence. After billions of years of evolution, plant communities can survive on the earth through their own survival strategies. Although the members of the plant community have no nerves, they can find areas rich in water and nutrients through the continuous cycle of seeding, growing, and fruiting, allowing them to survive in these areas [27,28]. If we can simulate the growth behavior of plant communities on a computer, we could also use this artificial plant community algorithm to search for the image edges on an original image.

The main contributions of this paper are as follows:

- First, the image edge is described as an area rich in water and nutrients, and the edge detection process is modeled as an objective function that can be used to search for water sources and nutrients. Hence, edges with higher habitable value are detected, and nonedges with lower habitable value are removed due to being unimportant details.
- Second, an artificial plant community (APC) algorithm is designed to solve the objective function by simulating the growth process of a true plant community. The growth behavior of an artificial plant community includes three operations: seeding, growing, and fruiting. These three operations correspond to three fitness comparisons, and the population size will also decrease or recover accordingly. Only seeds, individuals, and fruit with high fitness levels can survive, while those with low fitness levels will die. The individuals with the highest fitness levels can produce identical offspring fruit, other individuals can learn from each other to produce new fruit, and individuals with higher fitness levels can produce more fruit. Through this continuous cycle of three operations, the artificial plant community finally identifies the optimal image edge.

- Third, the proposed algorithm is verified by a series of experiments and is compared with other image edge detection algorithms. Comparative analysis results are provided and discussed.

The organization of the paper is as follows: The relevant work is briefly reviewed in Section 2. A description of the model is given in Section 3. The architecture of the proposed APC algorithm is developed in Section 4. The experimental results of the proposed scheme are presented and compared with other algorithms in Section 5. Finally, this study and some limitations are summarized, and future directions are suggested in Section 6.

2. Relevant Work

The problem with edge detection is determining how to accurately recognize significant changes in image attributes. Such changes usually reflect important events and changes in attributes, such as discontinuity of depth, discontinuity of the surface direction, changes in material attributes, and changes in scene lighting [9,12]. There are many edge detection method, and these can be divided into two main categories, namely search-based algorithms and zero-crossing-based algorithms. The search-based algorithms are mainstream and first calculate the edge strength [26], which is usually expressed by the first derivative, such as the gradient modulus. Zero-crossing-based algorithms determine the edge according to the zero-crossing point of the second derivative of the image, such as the Laplace operator.

Threshold-based edge detection is one of the most popular techniques and is the basis of many edge detection algorithms. It can be applied in many cases that emphasize computational efficiency and hardware implementation. In [26], a multilevel threshold in a hybrid transient search of a naked mole-rat optimizer was used for image segmentation.

Region-based edge detection includes two typical serial region technologies, namely region growth and decomposition. Region growth describes the continuous growth of regions based on the existing pixels until the whole region is formed so that target extraction can be carried out. Region decomposition is the inverse process of region growth, where the whole image is continuously divided into many subregions, and then the foreground regions are merged to achieve target extraction. In [14], an adaptive fuzzy-region growing fusion method was developed, and CNN-ANFIS-based automated segmentation was improved for the classification of cervical cancer. This method can be used to judge the regional computing results from the previous steps.

The feature-based clustering algorithm is a category-based edge detection method. It calculates the image space with corresponding feature space pixels, and makes segments of the feature space according to their feature aggregation. Then, it maps the feature space back onto the original image space to obtain the edge image. In [29], an image feature-based independent adversarial example detection model was developed, and in [30], an adaptive-window-based 3D feature selection method was developed for multispectral image classification, using a firefly algorithm.

Fuzzy edge detection employs fuzzy mathematics to describe a large number of uncertain concepts in order to achieve proper processing and image edge detection. Fuzzy C-means clustering (FCM) [9] and fuzzy region growing [14] are the two commonly used fuzzy detection algorithms. In [31], automobile instrument detection using prior information and fuzzy sets was illustrated, and in [32], adaptive image steganography using fuzzy enhancement and the grey wolf optimizer was developed.

In wavelet transform edge detection, the image histogram is divided into different levels of wavelet coefficients by a dyadic wavelet transform, and a threshold is then chosen according to the wavelet coefficients and given edge detection criteria. Lastly, a threshold is used to mark the region of the edge image. The calculation performance of the wavelet-transform-based edge detection algorithm varies linearly with the image size. In [33], image denoising was introduced to edge detection based on the wavelet transform.

The histogram-based algorithm uses a histogram to measure the color and strength of the image pixels. The histogram indicates the image edges, and the peaks and troughs of the histogram can be used to select the clusters in the image. In [34], a histogram-matched chest X-ray-based tuberculosis detection method using CNN was introduced. Compared with other image edge detection algorithms, the histogram-based algorithm is very effective, since it only needs a passing pixel.

The artificial neural network (ANN) and deep learning (DL) methods have undoubtedly been the biggest research topics in this area in recent years. The ANN is a nonlinear and adaptive information processing system composed of a large number of interconnected processing units [15]. Due to the surprising learning performance of the ANN and DL, many scholars have focused on this area, and many new variants have emerged, including the convolutional neural networks (CNNs) [14–17,34–37], the U-net [1], the depth-perceptual network [3], the long short-term memory (LSTM) [5], the edge-net [8], the generative adversarial network [13], the adaptive neuro fuzzy inference system (ANFIS) [14], the Y-net [16], deep learning (DL) [17–19,25,35], machine learning [20], the support vector machine (SVM) [22], the you only look once (YOLO) [35], the word embedding vector [37], and the auto-encoder [38]. The learning characteristics of a neural network can be easily integrated with other algorithms to generate new methods, i.e., [5] combined the LSTM model and the ABC algorithm, [14] incorporated the fuzzy-region and CNN, [18] merged the PSO and DL, and [25] designed a bird swarm optimization algorithm based on deep learning. However, neural networks and deep learning can only learn based on existing data; they cannot judge the correctness of the data and cannot correct the learning results unless they are retrained. In addition, complex neural network calculations are often very time-consuming.

The edge detection problem can be seen as a combinatorial optimization problem that is a process of finding the optimal solution to the edge image. Therefore, complex algorithms, such as the CNNs [14–17,34–37] and DL [17–19,25,35], may have a high level of accuracy but take a long time to find a solution. The simulated annealing (SA) algorithm is a random optimization algorithm that can be used to solve the combinatorial optimization problem. It adopts a Monte Carlo iterative solution strategy based on the annealing process of solid materials in physics. In [26], the simulated annealing algorithm was applied to image segmentation using multilevel thresholds. However, the learning ability of the simulated annealing algorithm is not strong.

Swarm intelligence algorithms are also random optimization algorithms that simulate the foraging behavior of biological groups in nature, and can easily achieve a balance between the solving time and solving accuracy [39]. The ability or behavior rules of each individual in a swarm intelligence algorithm are very simple, so the implementation of the algorithm is convenient and simple. Common swarm intelligence algorithms include particle swarm optimization (PSO) [18,40], ant colony optimization (ACO) [19,21], the artificial bee colony (ABC) [5,41], the genetic algorithm (GA) [22,42], the artificial fish swarm algorithm (AFSA) [23], the salp swarm algorithm (SSA) [24], bird swarm optimization (BSO) [25], the naked mole-rat optimizer [26], the firefly algorithm (FA) [30,41], the grey wolf optimizer (GWO) [32], the sparrow search algorithm (SSA) [43], and the whale optimization algorithm (WOA) [44]. These algorithms provide the great advantages of swarm intelligence to the field of image edge detection and image processing.

Both neural network algorithms and swarm intelligence algorithms often use the neural learning function and swarm learning mechanism of organisms to solve problems. In fact, plants, without nerves or brains, have evolved and survived for billions of years on the earth, and they also have own special learning mechanism and evolution strategy [27]. For example, in [28], an artificial slime mold algorithm was used to solve traffic network problems by simulating the natural nerveless slime mold, and in [45], an artificial Physarum swarm algorithm was employed to solve a logistics network problem by imitating the swarm learning mechanism of the nerveless Physarum. Both methods achieved satisfactory solving results. Here, we try to build a novel swarm intelligence algorithm to solve the image edge detection problem by simulating natural plant communities [27,46]. The proposed algorithm is expected to provide us with a new research tool for solving various complex problems.

3. Model Description

3.1. Basic Structure of the APC

Based on the natural plant community described in [27,46], an artificial plant community is designed to solve the image edge detection problem. The basic structure of an artificial plant community system includes many artificial plant individuals, habitable zones, and nonhabitable zones, as shown in Figure 1. In nature, plant communities can only survive in places with water and nutrients, known as habitable zones or feasible solutions. Other places are known as nonhabitable zones, and are not suitable for living, i.e., they are infeasible solutions. The habitable zones and nonhabitable zones constitute the living area of an artificial plant community.

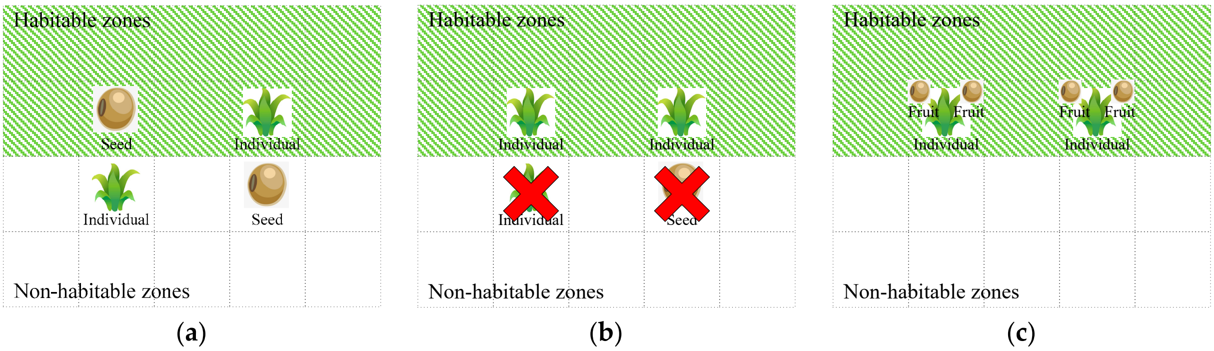

Figure 1. Artificial plant community system [27,46]: (**a**) seeding; (**b**) growing; (**c**) fruiting.

Many artificial plant individuals form an artificial plant community, and each artificial plant individual has similar characteristics and behaviors. They have three forms, namely seeds, individuals, and fruit, as shown in Figure 1. Different forms can be transformed into each other. The control of the artificial plant community is distributed, and there is no central control. Therefore, the community can better adapt to the working state of the external environment, and has strong robustness; that is, the failure of one or several artificial plant individuals will not affect the problem solution of the whole group. Each artificial plant individual in a group can change the environment, which is a method of indirect communication between artificial plant individuals, also known as stigmergy. Because artificial plant community intelligence can transmit and cooperate information through indirect communication, it has good scalability, whereby the increase in communication cost as the number of artificial plant individuals increases is small.

The solution space is divided into habitable zones and nonhabitable zones by an artificial plant community. Individual plants in habitable zones can survive, but individuals in nonhabitable zones will die. In the image edge detection problem, all feasible solutions of the edge image are equivalent to the living area of the artificial plant community. Each artificial plant individual can mark its living area as a habitable zone or a nonhabitable zone according to its learning experience, and other artificial plant individuals can learn from this experience. The complex learning behavior of an artificial plant community is an emergent intelligence that emerges through the interaction process of simple individuals in the solution space. Therefore, the artificial plant community can find habitable zones and nonhabitable zones by self-organization.

A fitness function of the image edges should be defined to select the optimal artificial plant individual, and individuals with low fitness levels will die. The artificial plant individuals are randomly distributed through the whole solution space, and separately search for feasible solutions or habitable zones according to the fitness function. The learning ability or operation behavior rules of each artificial plant individual in a group are very simple, so it is convenient to achieve artificial plant community intelligence. Each artificial plant individual marks the found edges as habitable zones and the nonedges as nonhabitable zones, according to a comparison of the fitness results. After several iterative calculations, and mutual learning of the artificial plant community, the edge image with the optimal fitness level is generated and output as a feasible solution.

3.2. Main Operations of the APC

An artificial plant community has three main operations, namely seeding, growing, and fruiting. The individuals in the plant community also correspond to three forms, namely seeds, individuals, and fruit. There are three fitness comparisons in each iteration.

The seeding operation is a process in which an artificial plant community searches for feasible solutions in the solution space, which is a probabilistic search operation, as shown in Figure 1a. The first fitness comparison of each iteration is carried out during the seeding operation. In the first iteration, the artificial plant community will randomly search for the feasible solutions in the solution space. In subsequent iterations, the fruit calculated in the previous iteration are selected as the seeds, and a small number of random seeds are generated at the same time. The seeding operation can help the artificial plant community to find new feasible solutions according to the seeding probability.

The growing operation is not only a natural selection process of an artificial plant community but also a probability search process, as shown in Figure 1b. The second fitness comparison is implemented in the growing operation. Not all artificial plant individuals can survive after randomly seeding, so the population size decreases after the growing operation. Only artificial plant individuals seeded in habitable areas will survive, and artificial plant individuals seeded in nonhabitable areas will die. The artificial plant community algorithm needs to establish a fitness function to judge the habitable area, namely the growing probability.

The fruiting operation is a swarm learning process of the artificial plant community, and is also probabilistic, as shown in Figure 1c. The third fitness comparison is completed in the fruiting operation. After seeding and growing, only a portion of the artificial plant individuals survive. The surviving plant individuals live in habitable zones, that is, the feasible solutions of the image edges. In the fruiting operation, two different plant individuals need to exchange feasible solution information to generate new plant individuals using the fruiting probability, that is, new feasible solutions. The new artificial plant individuals have the characteristics of two different parents; that is, the new feasible solutions learn the information from the previous two feasible solutions. Artificial plant individuals with higher fitness levels can produce more fruit, and the individual with the highest fitness levels can produce an identical fruit through parthenogenesis. Therefore, the population recovers to its original size.

3.3. Assumptions

To simplify the algorithm design, some assumptions are made, as follows:

i. In the artificial plant community, it is assumed that the differences in artificial plant species are not considered. Different artificial plant individuals have the same seeding, growing, and fruiting operations and can learn about the environment from each other.
ii. The survival of natural plant communities requires not only water and nutrients but also light. In the artificial plant community, it is assumed that the influence of light is not considered. Areas with water and nutrients are habitable zones for artificial plant communities. It is assumed that light does not affect the survival and solution of artificial plant individuals.
iii. The natural plant community will become lush, and the population size will become larger over time. In this algorithm, it is assumed that the population size of the seeding operation is the original one, the population size decreases during the growing operation, and the population size is restored after the fruiting operation.

iv. The seeding ranges of plant communities in nature are uncertain and are easily affected by media organisms, such as wind, water, rain, snow, fire, etc. Here, the seeding operation is assumed to be random, and is represented as a seeding probability.
v. The fitness function for natural selection is determined according to the goal of the particular edge detection problem. In the APC algorithm, the fitness function is used to measure the pixels in the digital image with obvious brightness changes and is compared three times, in the seeding, growing, and fruiting operations.
vi. The natural plant individuals are able to undergo parthenogenesis, and it is assumed that the artificial plant community will produce fruit that is identical to the optimal solution. Parents with high fitness levels can produce more fruit each time to improve their solution performance. All artificial plant individuals can perform the fruiting operation according to the proximity principle, and can implement simple spatial and temporal calculations using the fruiting probability.

3.4. Symbol Definitions

The symbol definitions used in this article are provided in this section, as shown in Table 1. The iteration counter k adds one for each iteration calculation with the maximum value being K.

Table 1. Symbol Definitions.

Symbol	Definition
I	An image
w	Image width
h	Image height
L	Grey level
Th_H	High threshold
Th_L	Low threshold
i	Pixel number
S	Population size
K	Maximum number of iterations
k	Iteration counter
x	Artificial plant individual
p_s	Seeding probability
p_g	Growing probability
p_f	Fruiting probability
$Obj\_fun()$	Value of the evaluation function
e	Error of the evaluation function
e_{th}	Error threshold

For an original image, the width is w, and the height is h. Hence, the pixel number is i, and the total number of image pixels can be attained as $w \times h$. An image $I(i,j)$ is a set of all pixel locations with the horizontal coordinate i and vertical coordinate j, and the size of the image $I(i,j)$ can be calculated as $|I(i,j)| = w \times h$. The value of the evaluation function $Obj\_fun()$ is used for the fitness evaluation, and the error threshold e_{th} is used for the end condition judgment.

The artificial plant individual can be encoded as a binary string with a length of no more than the maximum $w \times h$ value, where a binary value of 1 indicates that the pixel corresponding to the bit is an edge pixel and a binary value of 0 indicates that the pixel corresponding to the bit is a nonedge pixel.

4. Algorithm Design

4.1. Image Preprocessing

Referring to the principles of the Sobel algorithm [47] and Canny algorithm [48], four main steps are used to preprocess the image in order to provide a living environment for the APC algorithm.

First, the original image needs to be formatted, where the original color image is converted into a grayscale image. An original image $I_{\text{original}}(i,j)$ will be converted into a grayscale image $I_{\text{gray}}(i,j)$ with a gray level of L.

Second, the noise contained in the grayscale image $I_{\text{gray}}(i,j)$ is filtered out and the level of disturbance in image edge detection is decreased. Because most of the noise in the image is Gaussian noise, the Gaussian filter can help us to obtain images with a higher signal-to-noise ratio (SNR) [7,33]. The Gaussian filter is a linear smoothing filter that is suitable for eliminating Gaussian noise and is widely used in the noise reduction process of image processing. According to the Sobel operator [47] and Canny operator [48], the Gaussian filter is used to find the weighted average pixel values of the whole image, where the value of each pixel is obtained by a weighted average of its own value and the neighbor pixels values.

A Gaussian function can be written in a separable form, so a separable filter can be used for acceleration; that is, a multidimensional convolution can be converted into multiple one-dimensional convolutions. The two-dimensional Gaussian filtering of a gray image $I_{\text{gray}}(i,j)$ involves one-dimensional convolution on the row first, and then one-dimensional convolution on the column, which greatly reduces the computational complexity [7,33].

The two-dimensional Gaussian distribution function $G(i,j)$ is used to calculate the convolution values of the grayscale image in the horizontal and vertical directions $I_{\text{gray}}(i,j)$.

$$G(i,j) = \frac{1}{2\pi\sigma^2}\exp\left(-\frac{i^2+j^2}{2\sigma^2}\right) \tag{1}$$

where σ is the standard deviation of the Gaussian smoothing filter $G(i,j)$, which controls the smoothing degree of the grayscale image I_{gray}. If the value of σ is small, it will be a precise edge but with more noise, but if the value of σ is big, it will be an imprecise edge but with less noise. After applying the Gaussian smoothing filter, the image $I_{\text{gray}}(i,j)$ will change as follows:

$$I(i,j) = G(i,j) * I_{gray}(i,j) \tag{2}$$

where $*$ is the convolution operation. Now the filtered image $I(i,j)$ can be obtained, ready to be processed for edge detection.

Third, after the convolution operation and filtering, the habitable gradient, amplitude, and direction are obtained. The traditional operator calculates the first-order partial derivative finite difference of the pixels in the 2×2 convolution template to approximately obtain the habitable gradient amplitude and direction of the image $I(i,j)$. If $H_x(i,j)$ and $H_y(i,j)$ are the horizontal and vertical partial derivative arrays, respectively, they can be calculated as follows:

$$H_x(i,j) = \frac{1}{2}[I(i+1,j) - I(i,j) + I(i+1,j+1) - I(i,j+1)] \tag{3}$$

$$H_y(i,j) = \frac{1}{2}[I(i,j+1) - I(i,j) + I(i+1,j+1) - I(i+1,j)] \tag{4}$$

Then, the habitable gradient amplitude $H(i,j)$ and habitable direction $D(i,j)$ of the image are calculated as

$$H(i,j) = \sqrt{H_x^2(i,j) + H_y^2(i,j)} \tag{5}$$

$$D(i,j) = \arctan\left[\frac{H_y(i,j)}{H_x(i,j)}\right] \tag{6}$$

Fourth, the nonmaximum algorithm is used to suppress the habitable gradient amplitude and find all potential edge pixels according to the habitable gradient amplitude $H(i,j)$. First, a 3×3 template is used to detect all pixels in $H(i,j)$, and then $H(i,j)$ is compared, in the gradient direction $D(i,j)$, with the gradient amplitudes $H_+(i,j)$, in the positive direction, and $H_-(i,j)$, in the negative direction of the adjacent pixels. When the habitable gradient amplitude of a pixel is greater than the gradient amplitudes of the adjacent gray pixels in the positive and negative gradient directions, they are potential edge pixels.

$$\begin{cases} if\ H(i,j) > H_{+}(i,j) \text{and } H_{-}(i,j), \text{then an edge pixel} \\ \text{else, then } i \text{ is a non} - \text{edge pixel} \end{cases} \quad (7)$$

Furthermore, if $H(i,j) < H_{+}(i,j)$ or $H(i,j) < H_{-}(i,j)$, and $H(i,j)$ remains unchanged, then the current adjacent pixel i is defined as a no-edge pixel, and $H(i,j)$ is set as 0. On the contrary, $H(i,j)$ is set as 1 and is defined as a potential edge pixel to achieve the nonmaximum suppressing process.

Now, all possible edge pixels of the image $I(i,j)$ have been obtained, but there are a lot of false edges. It is very difficult to determine the edge pixels and eliminate the false edges. The APC algorithm will help us to search for the true edges and delete the false edges by heuristic searching.

4.2. Initialization of the APC

In this step, the main parameters are initialized, including the population size S, maximum number of iterations K, iteration counter k, seeding probability p_s, growing probability p_g, fruiting probability p_f, and artificial plant individual x.

It is assumed that the seeding rate represents the ratio of the population size of the artificial plant community after a seeding operation to the original population size of the artificial plant community before seeding. The relationship of the seeding rate to the population size is shown in Table 2.

Table 2. Seeding rate and population size.

Seeding Rate	Population Size after 50 Iterations
0.8	0.8^{50} × original size
1.0	1 × original size
1.2	1.2^{50} × original size
1.5	1.5^{50} × original size
2.0	2^{50} × original size
5.0	5^{50} × original size

As we can see from Table 2, negative growth of the population size will cause the artificial plant community to lose its search ability quickly, and positive growth of the population size will cause the search ability of the artificial plant community to continue to increase. However, an increase in the population size will also bring about a rapid decline in the convergence speed, and the temporal and spatial performance of the algorithm will also deteriorate rapidly. Here, it is recommended that the population size S of the artificial plant community should be fixed.

The natural plant community is composed of a variety of plants, and each artificial plant has many individuals. Similarly, the artificial plant individual x is encoded into a set of possible edge pixels as a feasible solution variable. If the corresponding pixel is selected as an edge pixel, the corresponding binary bit in the artificial plant individual x is set to 1; otherwise, it is 0. To reduce the amount of computation required, pixels with habitable gradient values greater than the high threshold Th_H are marked as edge pixels, and they do not need to be calculated or solved. Hence, to obtain the feasible solution x, we only need to search the candidate edge pixels below the high threshold Th_H.

Here, double thresholds are employed in the APC to implement edge detection and recognize false edges. After image preprocessing and nonmaximum suppression, we obtain all possible edges of the image $I(i,j)$, together with many false edges. The gray value distribution of the habitable gradient edge image $H(i,j)$ produced by the bi-threshold operator detection, is uneven. If the pixels with low nonzero habitable gradient values are regarded as edge pixels, false edges will occur. Here, the use of two thresholds can help us efficiently identify the edge pixels, with a high threshold Th_H and a low threshold Th_L.

On one hand, pixels with habitable gradient amplitudes greater than Th_H must be edge pixels, not false edge pixels. On the other hand, pixels with habitable gradient amplitudes smaller than Th_L are false edge pixels, and may not be edge pixels. If the gradient amplitude of a pixel is greater than Th_L but less than Th_H, it is necessary to judge the gradient amplitude values of its neighboring pixels. Hence, if the gradient amplitude value of a neighbor pixel is greater than Th_L and less than Th_H, the APC algorithm is employed to judge a candidate edge pixel.

$$\begin{cases} if\ H(i,j) > Th_H,\ \text{then } i \text{ is an edge pixel} \\ \text{else } if\ H(i,j) < Th_L,\ \text{then } i \text{ is a false edge pixel} \\ \text{else, APC selects } i \text{ as an edge pixel by hueristic search} \end{cases} \tag{8}$$

Many potential edges identified in Equation (7) are officially recognized as edges because their gradient amplitudes are higher than the high threshold Th_H, but the connectivity is low. In Equation (8), it is still difficult to accurately determine whether the pixels below the high threshold Th_H are edges. On one hand, the edge lines under the low threshold Th_L are thick and inaccurate; on the other hand, the image edges above the high threshold Th_H are discontinuous, and their details are lost. If we want to maintain good edge connectivity, the APC should accurately select the edge pixels below the high threshold Th_H and eliminate nonedges or false edges.

The pixels below the high threshold Th_H are called candidate edge pixels, and the APC needs a fitness function to further confirm whether the candidate edge pixels are edge pixels or not. Therefore, the APC needs to connect all edges in the image into contours. When it reaches the end of the contour, it will search for the candidate edge pixels that can be connected to the contour. However, it is not easy to search all true edges and eliminate all false edges. The artificial plant community algorithm should repeatedly implement seeding, growing, and fruiting operations to aid in the search for possible edges from the candidate edge pixels to obtain a more accurate, rich, and consistent edge image.

4.3. Seeding of APC

The seeding operation is the first fitness comparison, and the population size is the original size S. The artificial plant community algorithm processes multiple individuals at the same time, and can easily achieve seeding parallelism.

In the first iteration, all artificial plant individuals are randomly selected for seeding, where each bit in the selected individual is randomly selected as an edge pixel. The smaller the seeding probability is, the greater the probability of finding a new feasible solution is, but the slower the convergence is. On the contrary, the greater the seeding probability is, the lower the probability of finding a new feasible solution is, but the faster the convergence is, and there may even be premature convergence to the local optimal solution.

The artificial plant individual x is encoded as a binary string for seeding, as shown in Equation (9):

$$x = \{x_1, x_2, x_3, \cdots, x_i, \cdots\} \tag{9}$$

Each binary bit x_i represents the selection bit of the corresponding candidate edge pixel i, that is, whether pixel i is selected as an edge pixel. x_i = 1 denotes that pixel i is selected as the possible edge pixel, and x_i = 0 denotes that pixel i is deleted as the false edge pixels.

$$\begin{cases} if\ x_i\ =\ 1,\ \text{then the pixel } i \text{ is edge pixel} \\ if\ x_i\ =\ 0,\ \text{then the pixel } i \text{ is non} - \text{edge pixel} \end{cases} \tag{10}$$

The artificial plant individual x is encoded as a binary string for seeding, as shown in Equation (9).

In the subsequent iterative calculation, the artificial plant community selects the plant individuals from the best fruit of the previous iterative calculation according to the seeding probability p_s. However, the rest of the fruit population $1 - p_s$ dies, and the APC generates a small number of random plant individuals for seeding with a probability of $1 - p_s$. On one hand, this ensures fast convergence; on the other hand, it also prevents premature attainment of the local optimal solution.

4.4. Growing of the APC

The growing operation is the second fitness comparison. The population size decreases to $p_g \times S$, while the rest of the population $(1 - p_g) \times S$ dies. The natural plant communities are selected according to the surrounding water sources and nutrients. The artificial plant community is supposed to use the evaluation function as a fitness evaluation to judge the growing probability of the individuals. The growing operation is aimed at selecting the best artificial plant individuals from the seeding population and giving them a growing probability of p_g to grow into individuals. Through the growing operation, the artificial plant individual is selected as a feasible solution by the evaluation function, and a new edge image $I'(i,j)$ is obtained. If the artificial plant individuals score highly on the evaluation function of the living area, their growing probability is greater, and if the opposite occurs, their growing probability is smaller. The scoring process of the growing operation can help the artificial plant community to converge to a feasible solution. The evaluation function for APC growth includes several indexes, including the accuracy, information entropy, standard deviation, peak signal-to-noise ratio, and degree of distortion.

Accuracy is used to describe the precision of image edge detection, as shown in Formula (11). The greater the accuracy is, the more accurate the image edge detection is. On the contrary, the lower the accuracy is, the more inaccurate the edge detection is

$$ACC = \frac{TP + TN}{TP + TN + FP + FN} \tag{11}$$

where *TP*is the number of true positives, showing the total number of correctly detected edge pixels; *TN* is the number of true negatives, showing the total number of correctly detected nonedge pixels; *FP* is the number of false positives, showing the number of the wrongly detected edge pixels which are actually nonedge pixels; and *FN* is the number of false negatives, showing the number of the wrongly detected nonedge pixels which are actually edge pixels. The accuracy matrix defining the terms *TP*, *TN*, *FP*, and *FN* is shown in Table 3.

Table 3. Accuracy Matrix Definition.

Original Image	Edge Image	
	Edge Pixel	Non-Edge Pixel
Edge pixel	*TP*	*FP*
Non-edge pixel	*FN*	*TN*

Information entropy is an important indicator that is used to measure the richness of image information. For a grayscale image $I_{\text{gray}}(i,j)$ with a gray level of L, P_i represents the probability that the pixel gray value in the image is i, so the information entropy can be defined as

$$ENT = -\sum_{i=0}^{L-1} [P_i \times \log P_i] \tag{12}$$

where *ENT* represents the entropy of the image: the greater the entropy of the image, the more information the image has.

The standard deviation *STD* is one of the most commonly used quantization forms to reflect the degree of dispersion in the grayscale data from two images. For two images $I(i,j)$ and $I'(i,j)$ with the same size $w \times h$, the standard deviation is defined as

$$STD = \frac{1}{w \times h} \sum_{i=1}^{w} \sum_{j=1}^{h} [I(i,j) - I\prime(i,j)]^2 \quad (13)$$

The higher the *STD* is, the more discrete the data set of the image is, and the less clear the image is. On the contrary, the lower the *STD* is, the clearer the image is.

The peak signal-to-noise ratio *PSNR* of an image is an important indicator of the quality of the edge image. It is the ratio of the effective signal to the noise signal in the image, that is, the ratio of the edges to the nonedges. For two images $I(i,j)$ and $I'(i,j)$ with the same size $w{\times}h$, the mean square deviation is defined as *PSNR*.

$$PSNR = 10 \times \log(\frac{L^2}{STD}) \quad (14)$$

In general, for a given image, a high peak signal-to-noise ratio indicates high quality, and a low peak signal-to-noise ratio indicates low quality, which seriously affects image recognition.

The degree of distortion *DST* directly reflects the degree of distortion of the image. For two images $I(i,j)$ and $I'(i,j)$ with the same size $w \times h$, the distortion *DST* is defined as

$$DST = \frac{1}{w \times h} \sum_{i=1}^{w} \sum_{j=1}^{h} |I(i,j) - I\prime(i,j)| \quad (15)$$

The smaller the *DST* is, the less distortion there is. On the contrary, the greater the *DST* is, the greater the image distortion is.

In image edge detection, one or more of Equations (11)–(15) can be selected as the objective function to evaluate the growth of artificial plant individuals, where the higher the evaluation value is, the greater the growing probability of the artificial plant individual is, and the more likely it is that the optimal edges will be found. The multiobjective evaluation function for image edge detection can be built as follows:

$$\text{Obj\_fun} = \max\{ACC, ENT, PSNR\} - \min\{STD, DST\} \quad (16)$$

In each iterative computing step, an artificial plant community will compare the fitness using the evaluation function in Equation (16) to reduce the risk of falling into the local optimal solution. Thus, the artificial plant community can grow and respond to the evaluation results in the environment.

4.5. Fruiting of the APC

The fruiting operation is the third fitness comparison, and the population size recovers to the original size *S*. In nature, plant individuals often need the help of other organisms, wind, or water to complete pollination or fruit bearing. However, these conditions are not compulsory here, and natural selection is employed. The artificial plant individuals with the highest fitness levels can produce identical fruit through parthenogenesis, and those artificial plant individuals with high fitness levels are selected to produce more fruit. Through the fruiting operation, a new generation of artificial plant individuals can be obtained. These individuals have a combination of the parents' features, and the best solution is preserved. In the fruiting operation, several pixels are randomly chosen; then, the two parents' features are recombined, with the fruiting probability of p_f, to generate an offspring. The fruit are the descendants of the father's generation for the next seeding, and are the results of the evolution learning of the APC. For two artificial plant parents $x = \{x_1, x_2, x_3, \cdots, x_i, x_{i+1} \cdots\}$ and $y = \{y_1, y_2, y_3, \cdots, y_i, y_{i+1} \cdots\}$, their individual fruits x' and y' are shown as follows.

$$\begin{cases} x\prime = \{x_1, x_2, x_3, \cdots, x_i, y_{i+1}, \cdots\} \\ y\prime = \{y_1, y_2, y_3, \cdots, y_i, x_{i+1}, \cdots\} \end{cases} \tag{17}$$

After the fruiting operation, a new edge image $I''(i,j)$ is obtained, and we can also score the artificial plant individuals using Equation (16). Then, the optimal artificial individuals in $I'(i,j)$ and $I''(i,j)$ are compared, and the best solutions are selected for the next iterative computation. The fruiting probability p_f determines how much information the new feasible solution can learn from other plant individuals. The higher the fruiting probability is, the greater the information exchange between plant individuals is, and the greater the generation gap is. However, it is also easier to destroy the excellent individuals in the plant community and reduce the convergence rate. On the contrary, the smaller the fruiting probability is, the less information exchange there is between plant individuals, and the smaller the generation gap is. Nevertheless, it is also easier to protect the excellent individuals in the plant community, and improve the convergence performance.

4.6. End Judgment

The artificial plant community algorithm uses the neighbor information obtained by evolutionary learning to organize and search for feasible solutions. Among them, the artificial plant individuals with greater fitness levels have a greater survival probability, and a more adaptive plant community is gradually obtained. Through three main operations, seeding, growing, and fruiting, the solutions of the image edge detection algorithm can be judged, either by a predefined error, the maximum number of iterations, or the total computational time, so that an optimal solution will be produced.

At iteration k, the best artificial plant individual is $x(k)$ with a score of Obj_fun ($x(k)$) by Equation (16). In the previous iteration, the best artificial plant individual is $x(k-1)$ with a score of Obj_fun ($x(k-1)$). The error between the two iterations can be calculated.

$$e = |\,\text{Obj\_fun}\,(x(k)) - \text{Obj\_fun}\,(x(k-1))\,| \tag{18}$$

Subsequently, a predefined error is used to find a globally optimal solution. For example, an error threshold e_{th} can be predefined as an end judgment.

$$\begin{cases} \text{if} e < e_{th}, \text{ then go to end} \\ \text{else, preform the next iteration} \end{cases} \tag{19}$$

If the error e calculated by two iterations is less than the preset error threshold e_{th}, the calculation is completed, and the optimal solution x and the corresponding optimal edge image will be output. Otherwise, the optimal solution is taken as the seed. Then, we return to the seeding operation and perform the next iterative calculation to search for the optimal solution again.

4.7. Algorithm Flow of APC

The APC algorithm flow is shown in Figure 2, which simulates the evolution mechanism of the natural plant community and extends our knowledge about the learning behaviors of brainless living things. Based on the seeding, growing, and fruiting operations instructed by swarm learning, the APC algorithm can help us to efficiently extract the edges and delete the nonedges.

The algorithm flow is composed of two main steps and a large circulation.

The first main step is image preprocessing, including image format conversion, filtering, the calculation of habitable values, nonmaximal suppression, and double-threshold checking. These steps are performed to preprocess the image, reduce noise, identify the obvious edges, and provide candidate edges to reduce the amount of calculation required by the artificial plant community in the next stage.

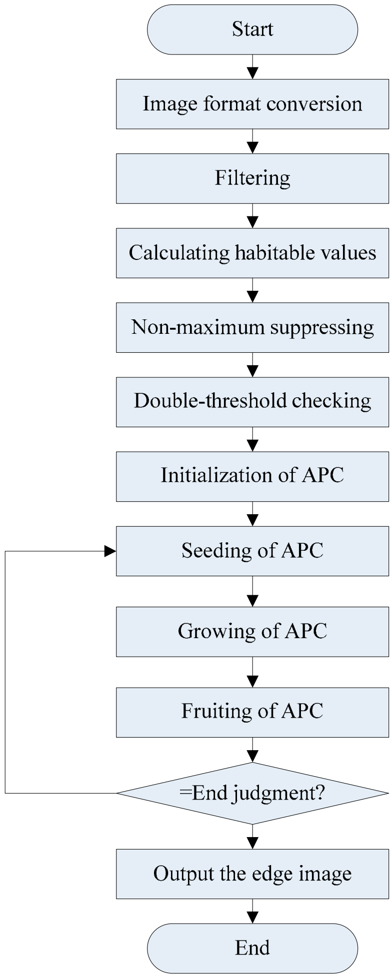

Figure 2. The APC algorithm for image edge detection.

The next major step involves the initialization, seeding, growing, and fruiting of the APC, and the end judgment. The stage includes a large circulation and double-threshold checking, where the artificial plant community randomly searches for all possible edge pixels from the candidate edges by evolution computing, and then optimizes the edge image according to the fitness function.

The APC algorithm successfully simulates the evolution process of a natural plant community through three main operations of seeding, growing, and fruiting. If the end conditions are not satisfied after the evaluation and comparison, the artificial plant community will return to the previous steps and repeatedly select the edge pixels.

5. Experimental Analysis

5.1. Experiment Results

In this section, we provide an experimental analysis of the proposed APC algorithm on edge detection problems. The experimental results, with different simulation parameters and detection effects, are further compared with other related algorithms to illustrate the application and characteristics of the APC. In these experiments, our algorithm uses several plant individuals to search for edges and nonedges, and the population size of the plant community may determine its searching capability. Three learning probabilities are used to prevent the artificial plant community from prematurely falling into local optimal solutions. Some assumptions are made before the experimental analysis.

(i) The structural complexity, topic, politics, history, ethics, ecology, and privacy protection factors of images are not considered.
(ii) The performance differences between the binarization methods and the filtering algorithms are not considered and the color differences of pixels are not compared.
(iii) It is assumed that different algorithms run the same iteration steps for the same images with the same parameters.
(iv) To simplify the analysis, we do not discuss comprehensive parameter adjustment and optimal data sets. Many algorithms have complicated improved editions, which may allow better effects on different test datasets through intelligent parameter adjustment; however, these are not studied here.

The experimental platform includes an AMD Ryzen 3 4300U with Radeon Graphics 2.70 GHz CPU, 8.00 GB RAM, a 64-bit Windows 10 operating system, and Matlab R2018a simulation software. The main simulation parameters used in our proposed algorithm are as follows: The population size of the artificial plant community is $S = 20$. For the three learning factors, the seeding probability is $p_s = 0.9$, the growing probability is set as $p_g = 0.8$, and the fruiting probability is $p_f = 0.7$. The maximum number of iterations is preset to 120, and the error threshold is less than 0.0001.

There are many test sets for image processing, so they could not all be selected for our tests. The Pascal VOC dataset and Stanford background dataset were chosen for our test, and were sufficient to verify the application of our proposed method. We provide the links to the two supporting datasets in the Data Availability Statement.

The Pascal VOC 2012 segmentation competition dataset is the ancestor of visual recognition competitions, and provides 20 categories of objects to be identified, such as people, animals, vehicles, and indoor scenes. It includes target detection, object classification, image segmentation, and other tasks. Figure 3 shows our edge detection solutions on four images from the Pascal VOC 2012 dataset. The four images have 500 × 375 pixels, 356 × 480 pixels, 500 × 375 pixels, and 314 × 186 pixels, respectively.

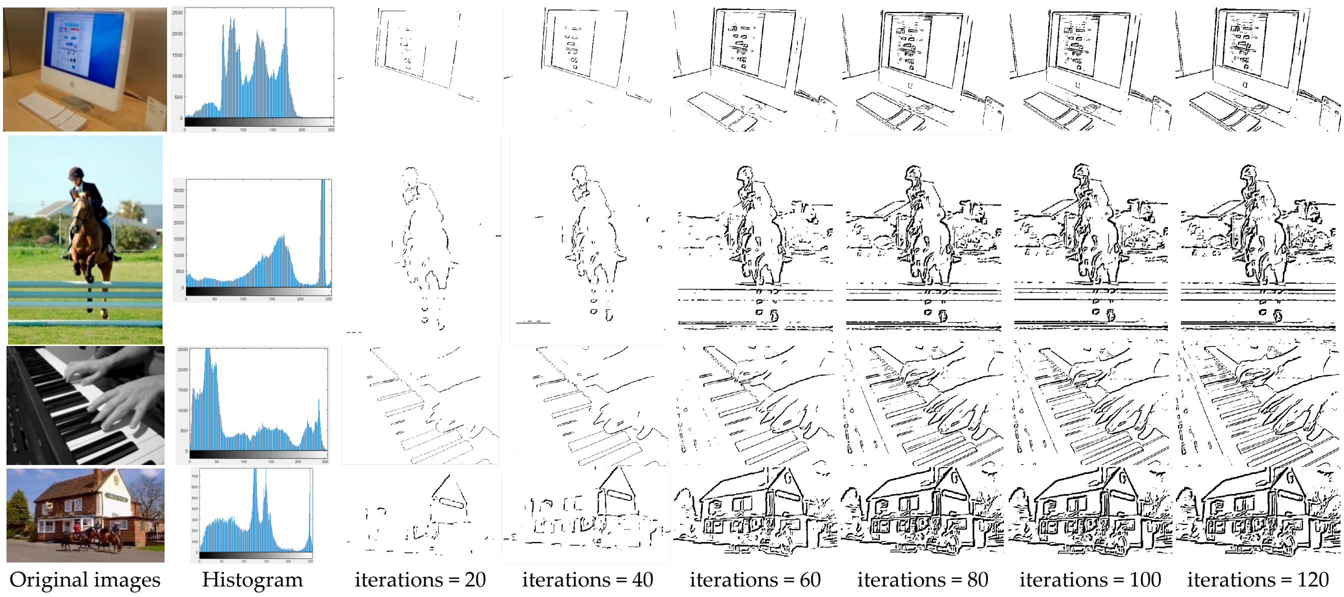

Figure 3. The test results of the proposed algorithm on the Pascal VOC dataset.

The first column in Figure 3 contains the original images, and the second column shows the histograms. Columns 3 to 8 show us the solving processes used by the artificial plant community. In the early solving stage, the artificial plant community can easily determine the edges with habitable values higher than the high threshold. The plant community will then further search for possible edge pixels to connect the image into contours. In the different test images shown in Figure 3, the artificial plant community can successfully build optimal edge images by feeding, growing, and fruiting, as shown from iterations 20 to 120 in Figure 3. In every test, the solutions for the same image may be different, because the evolution behavior of the artificial plant community is random and the pixel searching processes will also vary. These test cases, such as the personal computer, horseman, piano, and house, have different histograms, but the proposed APC algorithm shows a strong ability to connect possible edges and construct an optimal edge image. Despite the image differences and randomness in each test, the proposed APC algorithm can balance the edge detection accuracy and image noise.

The Stanford background dataset was then applied to the test. This dataset was selected from many famous databases, including LabelMe, MSRC, Pascal VOC, and Geometric Context. The Stanford background dataset has 715 outdoor images and is divided into eight categories: the sky, trees, roads, grass, water, buildings, mountains, and forest objects. Figure 4 shows our edge detection solutions and histograms on four images from the Stanford background dataset. The four images contain 320 × 214, 320 × 285, 320 × 240, and 320 × 212 pixels, respectively.

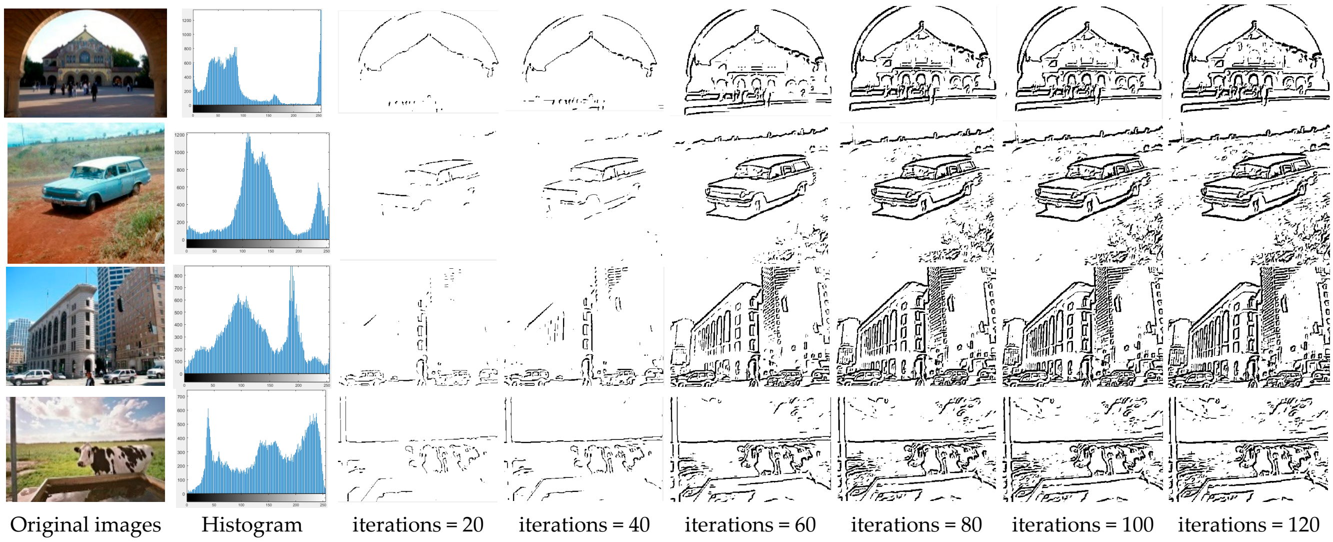

Figure 4. The test results of the proposed algorithm on the Stanford background dataset.

The first column in Figure 4 contains the original images, which have different histograms, as shown in the second column in Figure 4. Columns 3 to 8 show us the solving processes used by the artificial plant community. At the beginning of the test, the plant community is randomly distributed, and edges with habitable values higher than the high threshold are identified first, as shown in column three of Figure 4, where the number of iterations is 20. In the subsequent iterative calculation, the artificial plant community will continuously search for the edges and delete the nonedges, according to the objective function. Compared with the Pascal VOC 2012 test in Figure 3, there are fewer pixels in the Stanford background dataset, decreasing the calculation time. Despite the image differences and randomness in each test, the artificial plant community can successfully build optimal edge images by feeding, growing, and fruiting, as shown from iterations 20 to 120 in Figure 4. These edge results on the Stanford background dataset again show that the artificial plant community algorithm can be used to solve image edge detection problems. After a process of iterative computing of continuous seeding, growing, and fruiting, the proposed APC algorithm can balance the edge detection accuracy and image noise and finish the image edge detection task.

To sum up, the results presented in Figures 3 and 4 show us the evolution process of the proposed APC algorithm. The original images in the first columns on Figures 3 and 4 should be preprocessed, including image format conversion, filtering, calculating habitable values, nonmaximum suppressing, and double-threshold checking. After this, the processed images will be detected by the APC algorithm. The next major step involves the initialization, seeding, growing, and fruiting of the APC, and the end judgment. The image pixels above the high threshold Th_H are classified as edge pixels. After that, the APC repeatedly selects the edge pixels below the high threshold Th_H and eliminates nonedges or false edges until the optimal edge image is obtained.

5.2. Analysis and Discussion

In this section, the proposed APC algorithm is compared with some mainstream artificial intelligence algorithms, such as the artificial bee colony (ABC) [5,41], fuzzy C-means (FCM) [9,14,27,28], convolutional neural networks (CNNs) [14–17,34–37], particle swarm optimization (PSO) [18,40], ant colony optimization (ACO) [19,21], genetic algorithm (GA) [22,42], artificial fish swarm algorithm (AFSA) [23], and simulated annealing (SA) [26]. The algorithm comparison using the Pascal VOC 2012 dataset is shown in Figure 5, and the algorithm comparison using the Stanford background dataset is shown in Figure 6.

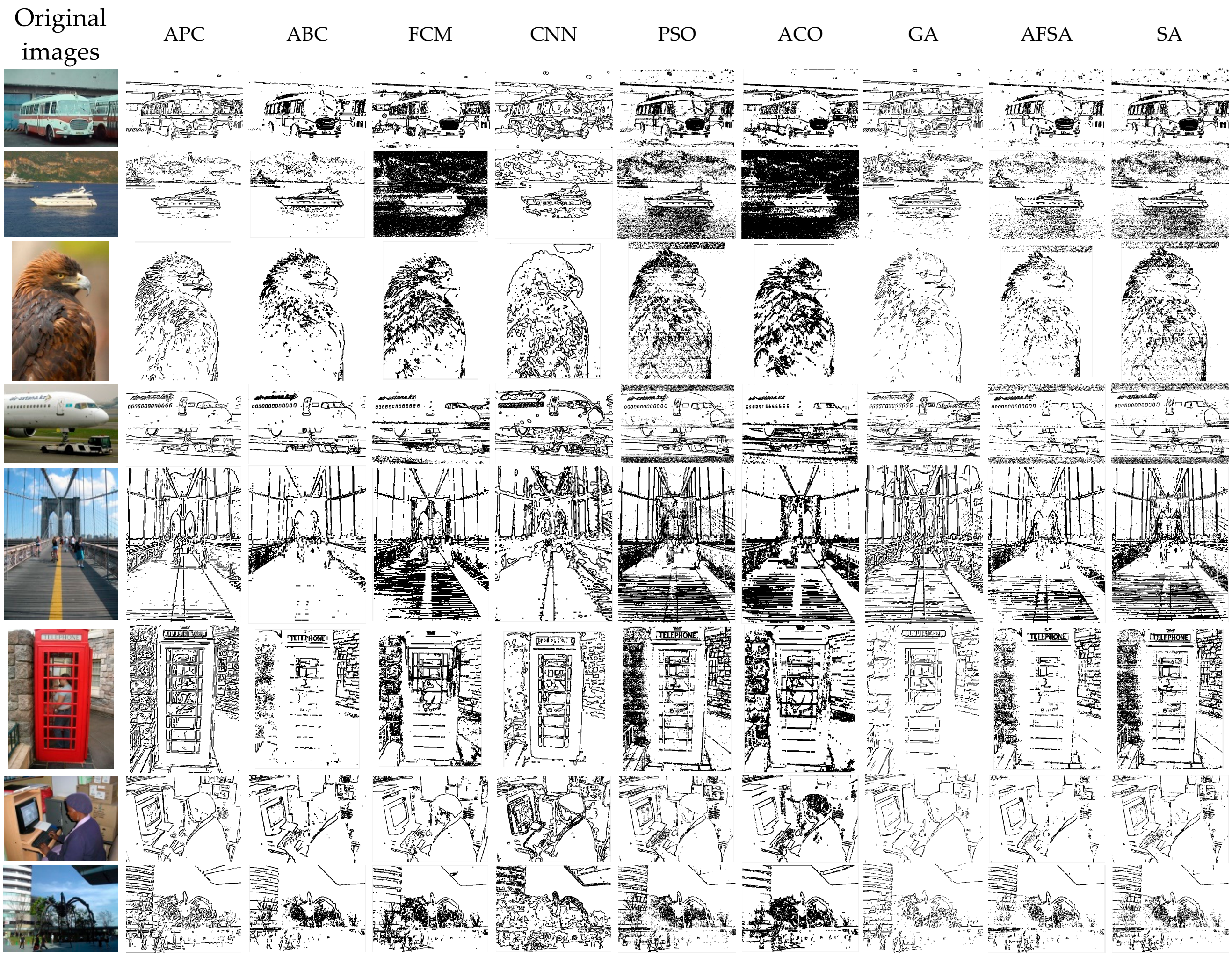

Figure 5. Algorithm comparison using the Pascal VOC 2012 dataset; ABC [5,41]; FCM [9,14,31,32]; CNN [14–17,34–37]; PSO [18,40]; ACO [19,21]; GA [22,42]; AFSA [23]; SA [26].

The first column in Figure 5 contains the original images from the Pascal VOC 2012 dataset. The eight images have 500 × 340 pixels, 500 × 375 pixels, 350 × 500 pixels, 500 × 343 pixels, 375 × 500 pixels, 375 × 500 pixels, 500 × 375 pixels, and 500 × 375 pixels, respectively. The second column in Figure 5 shows the edge detection results for our proposed APC algorithm, and columns 3~10 in Figure 5 are the results of the ABC [5,41], FCM [9,14,31,32], CNN [14–17,34–37], PSO [18,40], ACO [19,21], GA [22,42], AFSA [23], and SA [26]. Figure 5 presents eight cases, including the bus, yacht, eagle, plane, bridge, telephone booth, computer operator, and building, and they shows that the proposed APC algorithm can achieve similar effects in image edge detection to other artificial algorithms. The 5th column in Figure 5 presents the results of a multi-layer convolutional neural network [14–17,34–37]. This algorithm can achieve a good edge connection effect, but it is the most time-consuming of all algorithms. Other artificial algorithms can achieve good edge detection results with shorter calculation times, and are more suitable for fast and high-quality processing of image data. The comparative results, using the Pascal VOC 2012 dataset, show that our algorithm can successfully build a high-quality edge image, that is not inferior to other AI algorithms, through the evolution mechanism of an artificial plant community.

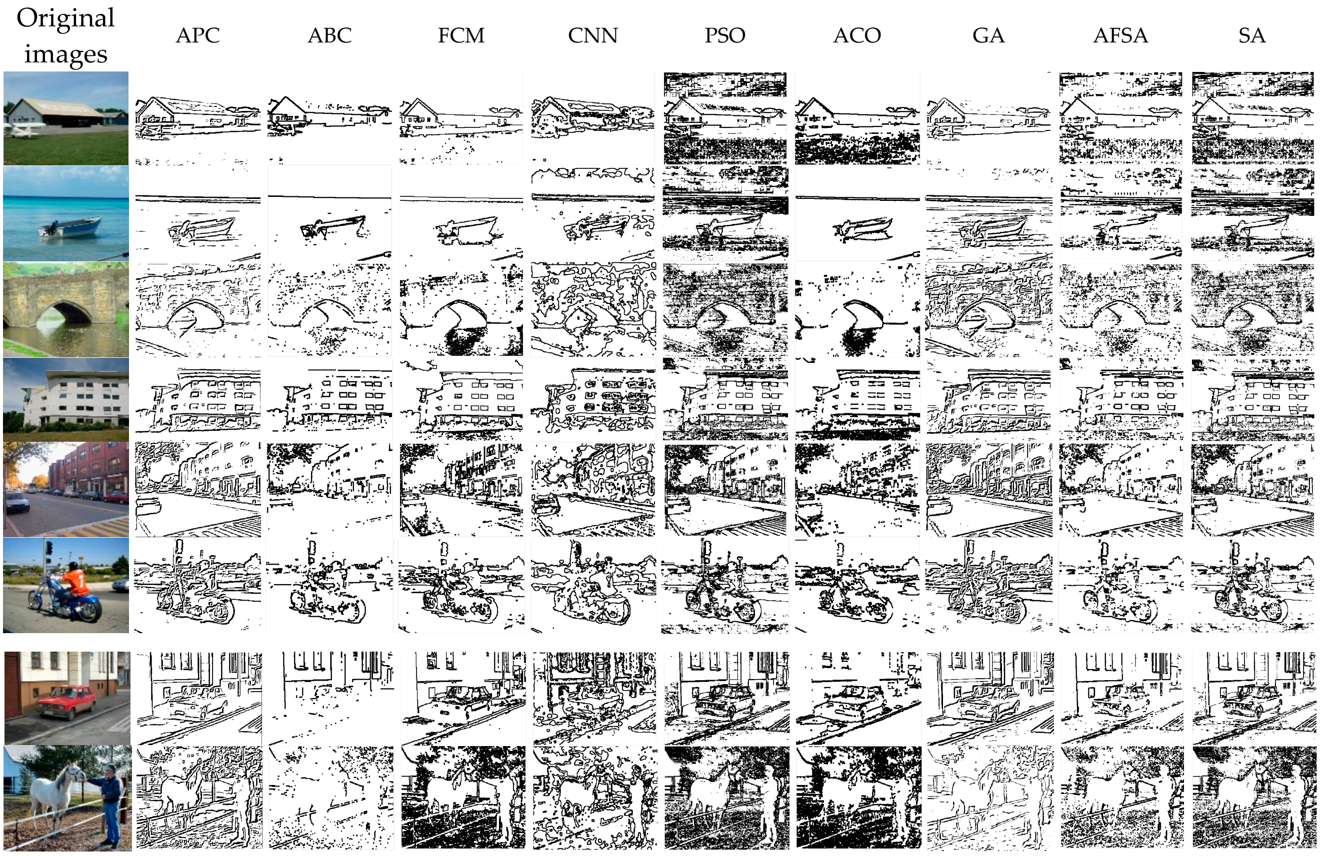

Figure 6. Algorithm comparison using the Stanford background dataset; ABC [5,41]; FCM [9,14,31,32]; CNN [14–17,34–37]; PSO [18,40]; ACO [19,21]; GA [22,42]; AFSA [23]; SA [26].

Figure 6 shows more cases from the Stanford background dataset to provide a further comparison between the proposed algorithm and other artificial intelligence algorithms. Figure 6 gives eight cases, including the bungalow, yacht, Si-o-Seh Bridge, apartment house, streetscape, motorcyclist, red car, and horse feeder. The first column in Figure 6 contains the original images from the Stanford background dataset. The eight images have 320 × 240 pixels, 320 × 240 pixels, 320 × 240 pixels, 320 × 213 pixels, 320 × 240 pixels, 320 × 240 pixels, 320 × 239 pixels, and 320 × 264 pixels, respectively. Columns 3–10 of Figure 6 show the edge detection solutions of the different algorithms, including our proposed APC algorithm, ABC [5,41], FCM [9,14,31,32], CNN [14–17,34–37], PSO [18,40], ACO [19,21], GA [22,42], AFSA [23], and SA [26].

The results presented in Figure 6 again verify that the proposed algorithm can achieve similar effects in image edge detection to other artificial intelligence algorithms. Because these test images from the Stanford background dataset have fewer pixels than those from the Pascal VOC 2012 dataset, the computational workload is greatly reduced, and all algorithms can complete the edge detection task in a shorter period of time. Among them, the computing time required for the CNN [14–17,34–37] algorithm is significantly reduced, but it is still the most time-consuming algorithm. Additionally, our APC algorithm can successfully complete the edge detection tasks and achieve a good and timely performance.

Furthermore, the quantitative comparison results of the different detection models on the Pascal VOC dataset and Stanford background dataset are shown in Tables 4 and 5, respectively. There are six quantitative indexes, namely the average accuracy (*ACC* in Equation (11)), entropy (*ENT* in Equation (12)), standard deviation (*STD* in Equation (13)), peak signal-to-noise ratio (*PSNR* in Equation (14)), distortion (*DST* in Equation (15)), and average solving time (ms).

Table 4. Comparison of results of different models on the Pascal VOC dataset.

Detection Models	Average Accuracy (*ACC*)	Entropy (*ENT*)	Standard Deviation (*STD*)	Peak Signal-to-Noise Ratio (*PSNR*)	Distortion (*DST*)	Average Solving Time (ms)
APC	0.902	0.384	61.72	18.48	14.65	813
ABC [5,41]	0.791	0.286	62.03	18.62	15.27	858
FCM [9,14,31,32]	0.825	0.322	83.35	14.77	14.53	739
CNN [14–17,34–37]	0.914	0.391	58.67	19.36	13.73	3022
PSO [18,40]	0.859	0.325	80.91	12.59	15.18	806
ACO [19,21]	0.846	0.318	70.54	13.61	15.09	1496
GA [22,42]	0.873	0.323	82.49	12.42	15.21	847
AFSA [23]	0.790	0.307	69.26	13.34	14.84	1305
SA [26]	0.787	0.299	71.60	12.78	15.12	714

Table 5. Comparison of results of different models on Stanford background dataset.

Detection Models	Average Accuracy (*ACC*)	Entropy (*ENT*)	Standard Deviation (*STD*)	Peak Signal-to-Noise Ratio (*PSNR*)	Distortion (*DST*)	Average Solving Time (ms)
APC	0.913	0.457	55.31	19.42	14.53	649
ABC [5,41]	0.822	0.305	57.54	18.49	15.16	724
FCM [9,14,31,32]	0.817	0.381	61.29	15.38	14.25	692
CNN [14–17,34–37]	0.926	0.502	56.48	20.23	14.51	2107
PSO [18,40]	0.864	0.413	74.60	12.81	14.97	706
ACO [19,21]	0.857	0.398	63.07	14.84	13.82	1138
GA [22,42]	0.889	0.409	73.16	12.90	14.68	785
AFSA [23]	0.828	0.326	59.53	13.76	14.64	1140
SA [26]	0.811	0.320	65.72	13.05	14.79	653

In terms of the average performance, the proposed APC and CNN [14–17,34–37] can achieve the highest accuracy levels, but the proposed APC needs less time than the latter. In terms of the average detection time, the proposed APC and SA [26] can achieve the fastest detection rates, but the proposed APC has an advantage in terms of its average accuracy. The performance test results using the Pascal VOC dataset are worse than those using the Stanford background dataset, due to the increased complexity of the problems to be solved. In both test datasets, the heuristic algorithms, including the proposed APC, ABC [5,41], FCM [9,14,31,32], PSO [18,40], ACO [19,21], GA [22,42], AFSA [23] and SA [26], have more advantages than the CNN [14–17,34–37], as they have shorter solving times. Compared with other heuristic algorithms, the proposed APC can solve the image edge detection problem with a better average accuracy, entropy, and peak signal-to-noise ratio, but a lower standard deviation and level of distortion.

By comparing the experimental results presented in Figures 5 and 6, Tables 4 and 5, it is clear that the proposed APC algorithm can successfully solve image edge detection problems through a probabilistic, parallel, and distributed evolution mechanism that differs from traditional algorithms. For both the Pascal VOC 2012 dataset and the Stanford background dataset, the artificial plant individuals can cooperate and evolve to search for the optimal edge pixels with accurate solutions in a shorter period of time. Unlike the traditional AI algorithms, such as the ABC [5,41], FCM [9,14,31,32], CNN [14–17,34–37], PSO [18,40], ACO [19,21], GA [22,42], AFSA [23] and SA [26], the proposed APC uses artificial plant individuals and a variable population size, and can search for the edge pixels through three fitness comparisons of seeding, growing, and fruiting in each iteration. Finally, the APC will converge on the optimal solution with the highest fitness level and output it as an optimal edge image.

5.3. Performance Summary and Internal Cause Analysis

From Figures 5 and 6, and Tables 4 and 5, it is apparent that the APC algorithm has some advantages for edge detection. In this section, we try to reveal the internal reasons for the experimental results presented in Sections 5.1 and 5.2. The performance comparison summary of different artificial intelligence algorithms is shown in Table 6, where the main indexes include the artificial swarm, time performance, fitness comparison, and variable population size. It is assumed that all algorithms use the same the population size m and iteration steps t to solve the edge detection problem in an image with n pixels. The artificial algorithms include particle swarm optimization (PSO) [18,40], ant colony optimization (ACO) [19,21], the artificial bee colony (ABC) [5,41], the genetic algorithm(GA) [22,42], the artificial fish swarm algorithm (AFSA) [23], the salp swarm algorithm (SSA) [24], bird swarm optimization (BSO) [25], the naked mole-rat algorithm (NMRA) [26], simulated annealing (SA) [26], the firefly algorithm (FA) [30,41], the grey wolf optimizer (GWO) [32], the sparrow search algorithm (SSA) [43], and the whale optimization algorithm (WOA) [44].

Table 6. Performance comparison summary of different artificial intelligence algorithms.

Detection Algorithms	Artificial Swarm	Time Performance	Fitness Comparison	Variable Population Size
APC	Artificial plant community	$O(mnt)$	Three times	$m \to m \times p_g \to m$
ABC [5,41]	Artificial bees	$O(mnt)$	Often once	Fixed m
FCM [9,14,31,32]	Fuzzy classes	$O(mnt)$	Often once	Fixed m
CNN [14–17,34–37]	Artificial neurons	$O(n^2m^2c_{in}c_{out})$	Often once	Fixed m
PSO [18,40]	Artificial particles	$O(mnt)$	Often once	Fixed m
ACO [19,21]	The ants	$O(mnt)$	Often once	Fixed m
GA [22,42]	Chromosomes	$O(mnt)$	Often once	Fixed m
AFSA [23]	Artificial fish	$O(mnt)$	Often once	Fixed m
SSA [24]	Salp swarm	$O(mnt)$	Often once	Fixed m
BSO [25]	Bird swarm	$O(mnt)$	Often once	Fixed m
NMRA [26]	Naked mole-rats	$O(mnt)$	Often once	Fixed m
SA [26]	Null	$O(nt)$	Often once	Null
FA [30,41]	Fireflies	$O(mnt)$	Often once	Fixed m
GWO [32]	Grey wolves	$O(mnt)$	Often once	Fixed m
SSA [43]	Sparrows	$O(mnt)$	Often once	Fixed m
WOA [44]	Whales	$O(mnt)$	Often once	Fixed m

Table 6 shows that the time performance of the swarm learning algorithms can be described as $O(mnt)$, which is nearly linear, and is related to the image scale n, the population size m, and the number t of iterations. Swarm learning algorithms with similar time performances include the proposed APC, the ABC [5,41], FCM [9,14,31,32], PSO [18,40], ACO [19,21], GA [22,42], AFSA [23], SSA [24], BSO [25], FA [30,41], GWO [32], SSA [43], and the WOA [44]. The time performance of the CNN [14–17,34–37] is related to the scale n of the image, the population size m of the convolution cores, the number c_{in} of input channels, and the number c_{out} of output channels. However, there is no swarm learning mechanism in SA [26], and its time performance is decided by the image scale n and the number t of iterations.

To sum up, some special characteristics can give the APC some advantages in terms of its solving performance.

First, the three fitness comparisons conducted per iteration in the APC algorithm; this is higher than in most AI algorithms, which often only include one fitness comparison. The seeding, growing, and fruiting operations all compare fitness levels and select individuals with high fitness levels to survive, while individuals with low fitness levels die. In this sense, one round of iterative computation of the APC algorithm is equivalent to three rounds of iterative computation of the other AI algorithms.

Second, the variable population size can improve the convergence of the APC. In the seeding operation, previous fruit with high fitness levels can be seeds with a probability of p_s, while some new seeds are randomly generated to form an original population size m. In the growing operation, the seeds with high fitness levels can survive with a probability of p_g, and the population size decreases. In the fruiting operation, individuals with a high fitness level can produce more fruit with a probability of p_f, and the population size recovers to m.

Third, the optimal solution in the APC can be well preserved by parthenogenesis, while the optimal solutions in other artificial intelligence algorithms are randomly changed in each iteration. Furthermore, the optimal solution, in the APC algorithm, is given the highest priority to be learnt by other individuals, in order to produce more fruit. This avoids the loss of the optimal solution in the solution process, and can also allow the search for new feasible solutions near to the optimal solution.

Fourth, the APC algorithm has a good global searching capability. In the seeding operation, a batch of random seeds will be generated for the global search. The growing operation then helps to screen out the seeds with low fitness levels. In the fruiting operation, the individuals surviving from the growing operation will learn from each other to produce new fruit, and those individuals with high fitness levels can produce more fruit.

Finally, the time performance of APC is equivalent to that of other AI algorithms with $O(mnt)$. With an increase in the population size m, the APC algorithm will also increase its search ability, as in many AI algorithms, making it possible to find the optimal solution with fewer iterative steps, but the calculation time of each iteration will increase linearly. With an increase in the image size, the time and difficulty required for the APC algorithm to provide a solution will increase each round, like many AI algorithms, and the solving accuracy will also be affected.

Additionally, enhanced editions and improved parameters will strengthen the global performance of all artificial intelligence algorithms. Here, we only focus on the feasibility of the proposed APC algorithm and its potential application, but we do not intend to denigrate other artificial algorithms. In fact, other intelligence algorithms, including the ABC [5,41], FCM [9,14,31,32], CNN [14–17,34–37], PSO [18,40], ACO [19,21], GA [22,42], AFSA [23], SSA [24], BSO [25], NMRA [26], SA [26], FA [30,41], GWO [32], SSA [43], and WOA [44] have been verified as effective tools for image edge detection.

6. Conclusions and Future Work

Here, an artificial plant community (APC) algorithm was presented to solve the image edge detection problem. The proposed algorithm can simulate the evolutionary learning and swarm intelligence of a natural plant community without requiring professional biological laboratories or biological operations [19,30]. The growth behavior of an artificial plant community includes three operations: seeding, growing, and fruiting. Three operations correspond to three fitness comparisons, and the population size will also decrease or recover accordingly. Only seeds, individuals, and fruit with high fitness levels can survive, while those with low fitness levels will die. The individuals with the highest fitness levels can produce identical offspring fruit, other individuals can learn from each other to produce new fruit, and the individuals with high fitness levels can produce more fruit. The experimental results on the Pascal VOC 2012 dataset and Stanford background dataset verify that the proposed artificial plant community algorithm can obtain an average accuracy of above 0.9 and a solving time of less than 800 ms. This balance in performance seems to be better than that of most other artificial intelligence algorithms, such as the ABC [5,41], FCM [9,14,31,32], CNN [14–17,34–37], PSO [18,40], ACO [19,21], GA [22,42], AFSA [23] and SA [26]. Different from traditional artificial intelligence algorithms, our algorithm employs many anencephalic and nerveless plant individuals to optimize the edge images and obtain a good image detection performance. The proposed APC algorithm may provide us with a new research tool for solving similar image processing problems and multi-objective optimization problems through a series of computing iterations.

The limitations of our study mainly focus on the fact that the parallelism of iterative computing is different from a natural plant community. Real plant communities can produce more and more plant individuals and seeds through continuous evolution, which will increase their ability to search the environment. However, on a personal computer, the expansion of artificial plant populations will worsen the computing performance of the computer system. In addition, more data tests and filters were not implemented here due to restrictions on time, space, and resources; specifically, only the Pascal VOC 2012 and the Stanford background datasets were tested, and only the Gaussian filter was employed. In addition, many powerful improved versions of artificial intelligence algorithms were not tested in this article, and the proposed APC algorithm was only compared with mainstream artificial intelligence algorithms, such as the ABC [5,41], FCM [9,14,31,32], CNN [14–17,34–37], PSO [18,40], ACO [19,21], GA [22,42], AFSA [23], and SA [26]. Today, there are so many state-of-the-art techniques that we cannot grasp all of them and cannot reproduce them one by one, so this study could only preliminarily prove the feasibility of the proposed algorithm.

In the future, the proposed APC algorithm will be further improved and compared with more state-of-the-art techniques. From there, more parameter adjustments, different population sizes, more datasets, more filters, and more algorithm comparisons will be conducted, together with many powerful improved versions of artificial intelligence algorithms. Furthermore, the proposed APC algorithm should be extended to solve more research problems, such as handwriting recognition, face recognition, license plate recognition, image enhancement, and multi-objective optimization problems.

Author Contributions: Conceptualization, Z.C. and Z.M.; methodology, Z.C., Z.M. and Z.Z.; validation, Z.Z. and Y.X.; writing—original draft preparation, Z.M., Z.Z., Y.X. and M.W.; writing—review and editing, Z.C., Z.M., Z.Z., Y.X. and M.W.; supervision, Z.C. and Z.M.; project administration, Z.C. and Z.M.; funding acquisition, Z.C. All authors have read and agreed to the published version of the manuscript.

Funding: This work was supported, in part, by the National Natural Science Foundation of China (No. 71471102), the Major Science and Technology Projects in Hubei Province of China (Grant No. 2020AEA012), and the Yichang University Applied Basic Research Project in China (Grant No. A17-302-a13).

Institutional Review Board Statement: Not applicable.

Informed Consent Statement: Not applicable.

Data Availability Statement: The data associated with this manuscript are contained in a data repository. Links to the supporting data are available: (1) Pascal VOC 2012 dataset https://pjreddie.com/projects/pascal-voc-dataset-mirror/ (accessed on 18 May 2012). (2) Stanford background dataset http://dags.stanford.edu/projects/scenedataset.html (accessed on 8 July 2009).

Acknowledgments: The authors would like to thank all the anonymous reviewers for their hard work to improve the quality of this submission.

Conflicts of Interest: The authors declare no conflict of interest.

References

1. Chen, J.; Hu, Y.; Chen, S.; Ren, Z.; Arikawa, T. Spatial Wave Measurement Based on U-net Convolutional Neural Network in Large Wave Flume. *Water* **2023**, *15*, 647. [CrossRef]
2. Kong, X.; Yi, J.; Wang, X.; Luo, K.; Hu, J. Full-Field Mode Shape Identification Based on Subpixel Edge Detection and Tracking. *Appl. Sci.* **2023**, *13*, 747. [CrossRef]
3. Lu, X.; Wu, S.; Xiao, Z.; Huang, X. An enhanced multiscale generation and depth-perceptual loss-based super-resolution network for prostate ultrasound images. *Meas. Sci. Technol.* **2023**, *34*, 024002. [CrossRef]
4. Adel, A.; Anis, K.; Wadii, B.; Bilel, B.; Yasser, A. A multi-stage deep-learning-based vehicle and license plate recognition system with real-time edge inference. *Sensors* **2023**, *23*, 2120.
5. Er, M.B. COVID-19 detection based on pre-trained deep networks and LSTM model using X-ray images enhanced contrast with artificial bee colony algorithm. *Expert Syst.* **2022**, *40*, e13185. [CrossRef]
6. Yu, J.; Huang, D.; Shi, X.; Li, W.; Wang, X. Real-Time Moving Ship Detection from Low-Resolution Large-Scale Remote Sensing Image Sequence. *Appl. Sci.* **2023**, *13*, 2584. [CrossRef]
7. Jiang, X.Q.; Xie, H.T.; Chen, J.X.; Zhang, J.; Wang, G.Q.; Xie, K. Arbitrary-oriented ship detection method based on long-edge decomposition rotated bounding box encoding in SAR images. *Remote Sens.* **2023**, *15*, 673. [CrossRef]
8. Alharbi Amal, H.; Karthick, S.; Venkatachalam, K.; Abouhawwash, M.; Khafaga, D.S. Spoofing face detection using novel edge-net autoencoder for security. *Intell. Autom. Soft Comput.* **2023**, *35*, 2773–2787. [CrossRef]
9. Yang, D.; Bai, Z.; Zhang, J. Infrared Weak and Small Target Detection Based on Top-Hat Filtering and Multi-Feature Fuzzy Decision-Making. *Electronics* **2022**, *11*, 3549. [CrossRef]
10. Sun, T.; Xiong, Z.; Yin, J.; Wu, Y.; Wang, Z. Gradient-Constraint Super-Resolution Reconstruction Method Serving for Infrared Target Detection. *IEEE Consum. Electron. Mag.* **2021**, *12*, 14–25. [CrossRef]
11. Ullah, A.; Liu, G.Q.; Wang, H.; Khan, M.; Khan, D.F.; Luan, J.H. Optimal approach of three-dimensional microstructure reconstructions and visualizations. *Mater. Express* **2013**, *3*, 109–118. [CrossRef]
12. Cai, R.T.; Chen, Z.K. Brain-like retinex: A biologically plausible retinex algorithm for low light image enhancement. *Pattern Recognit.* **2023**, *136*, 109195. [CrossRef]
13. Zhu, C.; Xu, J.; Feng, D.; Xie, R.; Song, L. Edge-based video compression texture synthesis using generative adversarial network. *IEEE Trans. Circuits Syst. Video Technol.* **2022**, *32*, 7061–7076. [CrossRef]
14. Jeyshri, J.; Kowsigan, M. Adaptive fuzzy-region growing fusion and improved CNN-ANFIS-based automated segmentation and classification of cervical cancer. *Concurr. Comput. Pract. Exp.* **2022**, *35*, e7493. [CrossRef]
15. Deng, Y.; Meng, Y.; Chen, J.; Yue, A.; Liu, D.; Chen, J. TChange: A Hybrid Transformer-CNN Change Detection Network. *Remote. Sens.* **2023**, *15*, 1219. [CrossRef]
16. Fekri, P.; Khodashenas, H.; Lachapelle, K.; Cecere, R.; Zadeh, M.; Dargahi, J. Y-net: A deep convolutional architecture for 3D estimation of contact forces in intracardiac catheters. *IEEE Robot. Autom. Lett.* **2022**, *7*, 3592–3599. [CrossRef]
17. Alqahtani, T.M. Big Data Analytics with Optimal Deep Learning Model for Medical Image Classification. *Comput. Syst. Sci. Eng.* **2022**, *44*, 1433–1449. [CrossRef]
18. Manikandakumar, M.; Karthikeyan, P. Weed classification using particle swarm optimization and deep learning models. *Comput. Syst. Sci. Eng.* **2023**, *44*, 913–927. [CrossRef]
19. Mehnatkesh, H.; Jalali, S.M.J.; Khosravi, A.; Nahavandi, S. An intelligent driven deep residual learning framework for brain tumor classification using MRI images. *Expert Syst. Appl.* **2023**, *213*, 119087. [CrossRef]
20. Alkhaldi, N.A.; Halawani, H.T. Intelligent machine learning enabled retinal blood vessel segmentation and classification. *CMC Comput. Mater. Contin.* **2023**, *74*, 399–414.
21. Ticala, C.; Pintea, C.-M.; Matei, O. Sensitive Ant Algorithm for Edge Detection in Medical Images. *Appl. Sci.* **2021**, *11*, 11303. [CrossRef]
22. Zhang, S.; Huang, H.; Huang, Y.; Cheng, D.; Huang, J. A GA and SVM Classification Model for Pine Wilt Disease Detection Using UAV-Based Hyperspectral Imagery. *Appl. Sci.* **2022**, *12*, 6676. [CrossRef]
23. Gu, M.; Xie, R.; Xiao, L. Two-Step Inversion Method for NMR Relaxometry Data Using Norm Smoothing and Artificial Fish Swarm Algorithm. *Appl. Magn. Reson.* **2021**, *52*, 1615–1634. [CrossRef]
24. Nasef Mohammed, M.; Eid Fatma, T.; Amin, M.; Sauber, A.M. An efficient segmentation technique for skeletal scintigraphy image based on sharpness index and salp swarm algorithm. *Biomed. Signal Process. Control.* **2023**, *79*, 104046. [CrossRef]

25. Omana, S.P.; Dar, J.A.; Kumar, T.R.; Sampath, A.K.; Sharma, S. Henry gas bird swarm optimization algorithm-based deep learning for brain tumor classification using magnetic resonance imaging. *Concurr. Comput. Pract. Exp.* **2023**, *35*, e7541.
26. Singh, S.; Mittal, N.; Nayyar, A.; Singh, U.; Singh, S. A hybrid transient search naked mole-rat optimizer for image segmentation using multilevel thresholding. *Expert Syst. Appl.* **2023**, *213*, 119021. [CrossRef]
27. Luo, Y.; He, J.; Long, Y.; Xu, L.; Zhang, L.; Tang, Z.; Li, C.; Xiong, X. The Relationship between the Color Landscape Characteristics of Autumn Plant Communities and Public Aesthetics in Urban Parks in Changsha, China. *Sustainability* **2023**, *15*, 3119. [CrossRef]
28. Cai, Z.; Xiong, Z.; Wan, K.; Xu, Y.; Xu, F. A node selecting algorithm for traffic network based on artificial slime mold. *IEEE Access* **2020**, *8*, 8436–8448. [CrossRef]
29. Kim, M.H.; Yun, J. AEGuard: Image feature-based independent adversarial example detection model. *Secur. Commun. Netw.* **2022**, *2022*, 3440123. [CrossRef]
30. Rajakani, M.; Kavitha, R.J.; Ramachandran, A. Adaptive window based 3-D feature selection for multispectral image classification using firefly algorithm. *Comput. Syst. Sci. Eng.* **2023**, *44*, 265–280. [CrossRef]
31. Zhang, J.; Liu, Y.; Yu, J.; Yang, X.; Yu, X.; Rodriguez-Andina, J.J.; Gao, H. Automobile instrument detection using prior information and fuzzy sets. *IEEE Trans. Ind. Electron.* **2021**, *69*, 13524–135342022. [CrossRef]
32. Xie, J.; Wang, H.; Wu, D. Adaptive Image Steganography Using Fuzzy Enhancement and Grey Wolf Optimizer. *IEEE Trans. Fuzzy Syst.* **2022**, *30*, 4953–4964. [CrossRef]
33. You, N.; Han, L.; Zhu, D.; Song, W. Research on Image Denoising in Edge Detection Based on Wavelet Transform. *Appl. Sci.* **2023**, *13*, 1837. [CrossRef]
34. Ignatius, J.L.P.; Selvakumar, S.; Louis, P.K.G.J.; Kailash, A.B.; Keertivaas, S.; Prajan, S.A.J.A.R. Histogram matched chest X-rays based tuberculosis detection using CNN. *Comput. Syst. Sci. Eng.* **2023**, *44*, 81–97. [CrossRef]
35. Kao, S.-P.; Chang, Y.-C.; Wang, F.-L. Combining the YOLOv4 Deep Learning Model with UAV Imagery Processing Technology in the Extraction and Quantization of Cracks in Bridges. *Sensors* **2023**, *23*, 2572. [CrossRef]
36. Feng, W.; Lan, L.; Buro, M.; Luo, Z. Online multiple-pedestrian tracking with detection-pair-based graph convolutional networks. *IEEE Internet Things J.* **2022**, *9*, 25086–25099. [CrossRef]
37. Lalitha, B.; Gomathi, V. Image captioning using detectors and swarm based learning approach for word embedding vectors. *Comput. Syst. Sci. Eng.* **2023**, *44*, 173–189. [CrossRef]
38. Shi, C.; Lu, J.; Sun, Q.; Zhou, J.; Huang, W.; Xia, R. Multi-scale auto-encoder for edge detection. *IEEE Access* **2022**, *10*, 116253–116260. [CrossRef]
39. Cai, Z.; Zhang, Y.; Wu, M.; Cai, D. An entropy-robust optimization of mobile commerce system based on multi-agent system. *Arab. J. Sci. Eng.* **2016**, *41*, 3703–3715. [CrossRef]
40. Dumitru, D.; Dioşan, L.; Andreica, A.; Bálint, Z. A Transfer Learning Approach on the Optimization of Edge Detectors for Medical Images Using Particle Swarm Optimization. *Entropy* **2021**, *23*, 414. [CrossRef]
41. Giuliani, D. Metaheuristic Algorithms Applied to Color Image Segmentation on HSV Space. *J. Imaging* **2022**, *8*, 6. [CrossRef]
42. Guidio, B.; Nam, B.H.; Jeong, C. Multilevel Genetic Algorithm–Based Acoustic–Elastodynamic Imaging of Coupled Fluid–Solid Media to Detect an Underground Cavity. *J. Comput. Civ. Eng.* **2023**, *37*, 04022047. [CrossRef]
43. Xiong, Q.; Zhang, X.; He, S.; Shen, J. Data augmentation for small sample iris image based on a modified sparrow search algorithm. *Int. J. Comput. Intell. Syst.* **2022**, *15*, 110. [CrossRef]
44. Bhuvaneswari, A.; Devi, T.M. Contrast enhancement of retinal images using green plan masking and whale optimization algorithm. *Wirel. Pers. Commun.* **2022**, *125*, 1047–1073. [CrossRef]
45. Cai, Z.; Yang, Y.; Zhang, X.; Zhou, Y. Design a Robust Logistics Network with an Artificial Physarum Swarm Algorithm. *Sustainability* **2022**, *14*, 14930. [CrossRef]
46. Gu, X.; Xiong, K.; Wu, C.; Luo, D. Study on Soil Moisture Characteristics in Southern China Karst Plant Community Structure Types. *Forests* **2023**, *14*, 384. [CrossRef]
47. Alphonse, A.S.; Abinaya, S.; Arikumar, K.S. A novel monogenic Sobel directional pattern (MSDP) and enhanced bat algorithm-based optimization (BAO) with Pearson mutation (PM) for facial emotion recognition. *Electronics* **2023**, *12*, 836. [CrossRef]
48. Guo, L.; Wu, S. FPGA Implementation of a Real-Time Edge Detection System Based on an Improved Canny Algorithm. *Appl. Sci.* **2023**, *13*, 870. [CrossRef]