Send Orders for Reprints to reprints@benthamscience.net

Current Bioinformatics, **2019**, *14*, 1-12

RESEARCH ARTICLE

A Similarity Searching System for Biological Phenotype Images Using Deep Convolutional Encoder-decoder Architecture

Bizhi Wu[1], Hangxiao Zhang[2], Limei Lin[1], Huiyuan Wang[2], Yubang Gao[2], Liangzhen Zhao[2], Yi-Ping Phoebe Chen[1,3,*], Riqing Chen[1,*] and Lianfeng Gu[2,*]

[1]The College of Computer and Information Sciences, Fujian Agriculture and Forestry University, Fuzhou 350002, China; [2]Basic Forestry and Proteomics Research Center, Fujian Provincial Key Laboratory of Haixia Applied Plant Systems Biology, College of Forestry, Fujian Agriculture and Forestry University, Fuzhou 350002, China; and [3]Department of Computer Science and Information Technology, La Trobe University, Melbourne, Australia

ARTICLE HISTORY

Received: April 04, 2018
Revised: July 20, 2018
Accepted: August 22, 2018

DOI:
10.2174/1574893614666190204150109

Abstract: ***Background*:** The BLAST (Basic Local Alignment Search Tool) algorithm has been widely used for sequence similarity searching. Analogously, the public phenotype images must be efficiently retrieved using biological images as queries and identify the phenotype with high similarity. Due to the accumulation of genotype-phenotype-mapping data, a system of searching for similar phenotypes is not available due to the bottleneck of image processing.

***Objective*:** In this study, we focus on the identification of similar query phenotypic images by searching the biological phenotype database, including information about loss-of-function and gain-of-function.

***Method*:** We propose a deep convolutional autoencoder architecture to segment the biological phenotypic images and develop a phenotype retrieval system to enable a better understanding of genotype–phenotype correlation.

***Results*:** This study shows how deep convolutional autoencoder architecture can be trained on images from biological phenotypes to achieve state-of-the-art performance in a phenotypic images retrieval system.

***Conclusion*:** Taken together, the phenotype analysis system can provide further information on the correlation between genotype and phenotype. Additionally, it is obvious that the neural network model of image segmentation and the phenotype retrieval system is equally suitable for any species, which has enough phenotype images to train the neural network.

Keywords: Biological phenotype similarity searching, convolutional encoder-decoder architecture, segmentation of biological images, phenotypic image retrieval system.

1. INTRODUCTION

Phenotype-driven studies, such as gain-of-function and loss-of-function mutation, can provide important clues in relation to gene function, since the phenotype changes are determined by the alterations in the genotype [1]. With the growth in genotype-phenotype studies [2], it is possible to infer the potential gene function for query phenotypic images by examining public phenotypic images with a known function. The BLAST (Basic Local Alignment Search Tool) algorithm has been widely used for searching sequence similarities to study the genotype of a query sequence [3, 4], and has been cited over 67,000 times. Analogously, the lines, including similar phenotypic images, might be involved in the same pathway with the mutant-owned query phenotypic images. To the best of our knowledge, in contrast to sequence comparison, a phenotype comparison and retrieval system is still elusive at present. Therefore, it is necessary to construct a phenotype retrieval system to systematically search the phenotype database with a series of query images for the scientific community. To fill the gap in biological phenotype similarity searching, we developed a deep convolutional autoencoder architecture, which provides a way to find potential candidate genes involved in the same pathway.

Similar to other general neural network models, the autoencoder has an input layer, hidden layers and an output

*Address Correspondence to this author at Fujian Agriculture and Forestry University, Fuzhou, P.R. China; Tel: 86591-83590305; E-mail: riqing.chen@fafu.edu.cn, phoebe.chen@latrobe.edu.au or lfgu@fafu.edu.cn

1574-8936/19 $58.00+.00

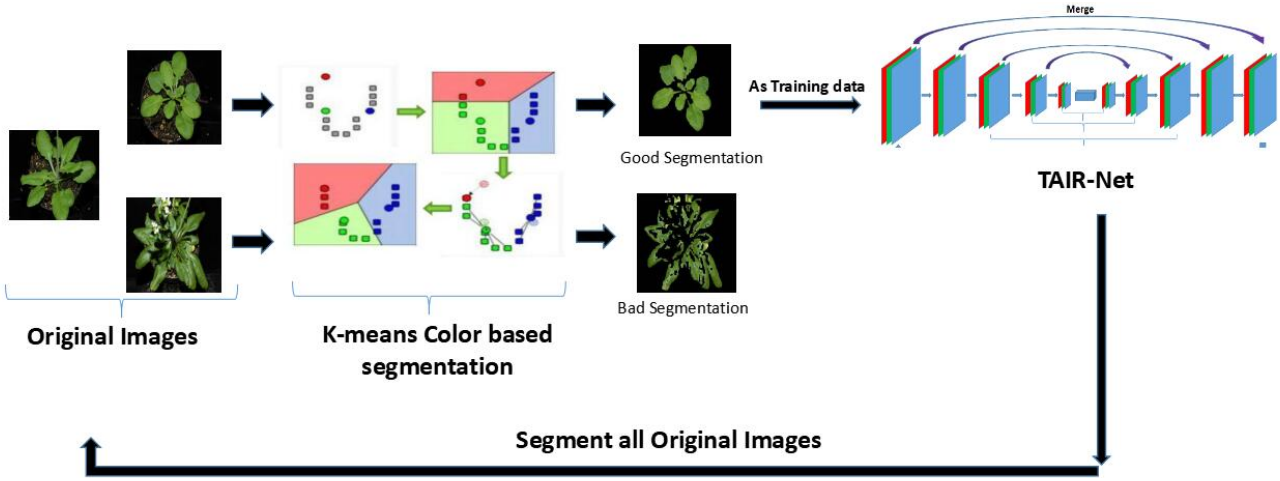

Fig. (1). The flow chart of image segmentation. Firstly, we used the K-means algorithm to perform segmentation. Then, we selected the better segmentation and the corresponding original picture to train the autoencoder. Finally, we used the autoencoder model to perform segmentation on the whole dataset.

layer [5]. The entire autoencoder is symmetrical about the innermost hidden layer in general. The middle layer is called the presentation layer. The encoder maps the input layer to the representation layer. On the contrary, the part from the representation layer to the output layer, which reconstructs the input layer is called the decoder.

The number of neurons is reduced from the input layer to the presentation layer, while the number of neurons from the presentation layer to the output layer is increased since the structure of the encoder is symmetrical [5]. When we apply the autoencoder method to solve image-processing problems, it usually replaces the fully connected layer of the autoencoder with convolutional neural networks. This type of autoencoder is called the convolutional autoencoder [6]. When the pooling layer is added to the convolution layer, the dimension of the input image data will be reduced. This process enables the autoencoder to obtain more efficient representation of the encoder, which generates more accurate representation of the feature space. Autoencoder uses the upsampling method and deconvolution operation in the process of decoding. In 2014, fully convolutional networks (FCN) popularized convolutional autoencoder architectures for dense predictions without any fully connected layers [7]. Most of the subsequent methods used this paradigm to perform semantic segmentation. The encoder-decoder is one of the architectures for semantic segmentation. The encoder gradually reduces the spatial dimension with pooling layers and the decoder gradually recovers the object details and spatial dimensions. Shortcut connections between the encoder and decoder ensure the better reconstruction of objects.

Though the great needs of biological phenotype similarity searching make the development of the system particularly rewarding, it faces two challenges for the phenotypic retrieval system. The first is how to effectively separate the plants from the background. In this study, we solve this problem by using a deep convolutional auto-encoder architecture. Based on the structure of SegNet [8], we propose a neural network model to segment phenotypic images in *Arabidopsis*, which is referred to as TAIR-Net. The results show that deep neural network architectures are successful in segmenting biological images. Another key challenge for the phenotypic retrieval system is to design an effective search algorithm to match the query images with databases including the most similar biological images. At present, these are two types of image retrieval technology, text-based image retrieval (TBIR) [9] and content-based image retrieval (CBIR) [10]. We adopt CBIR for content-based visual information retrieval, which can avoid the time consuming manual annotation of textual information to describe each biological image. Extracting a feature vector module is essential for CBIR. We build a library for these feature vectors and use the deep convolutional autoencoder to extract the features from the images.

In this study, we develop a phenotype retrieval system which can provide potential functional information on genes in the same pathway related to the query images. Although we have chosen *Arabidopsis* as the reference phenotype images for training, the method can be applied to any species with a large number of phenotypic images. Thus, this study provides a powerful tool to better understand large-scale phenotypic data.

2. METHODS

2.1. Data Sets for Preprocessing

In order to search for the genotype-phenotype-mapping database with a query images, we collect the loss-of function and gain-of-function line data from the well-known TAIR database based on the literature curation [2] and the *Arabidopsis* phenome database including phenotypes from *Ds* tagging line, activation tagging line, and Fox-hunting line [1] to set up a biologically similar phenotype retrieval system.

2.2. Image Segmentation by K-means in L*a*b* Color Space

Firstly, we used the K-means algorithm for pre-segmentation. Then, we selected the better segmentation and corresponding original picture to train the deep convolutional autoencoder. Finally, we used the trained autoencoder to segment the whole dataset (Fig. **1**). In a

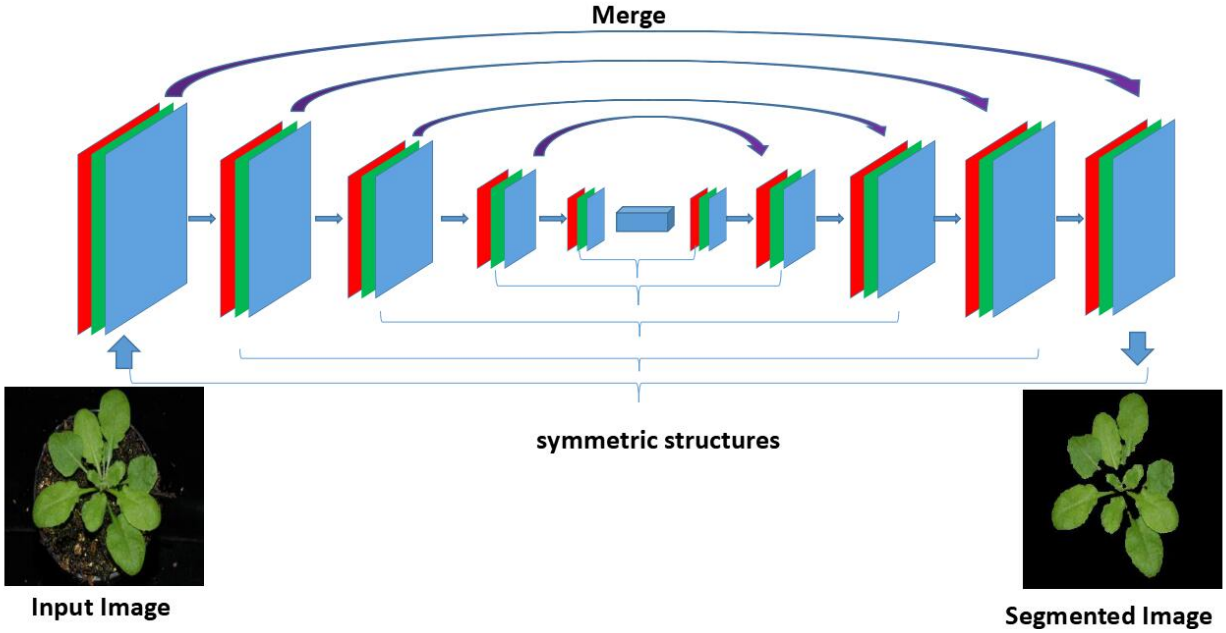

Fig. (2). Neural network model for segmentation using deep convolutional encoder-decoder architecture. The left panel is the image for segmentation. The middle panel is the neural network model for segmentation model, which is a deep convolutional autoencoder model with a symmetrical structure. The autoencoder uses the structure of skip-connect. The convolutional parameters for the first layer of the input and the last layer of the convolutional parameters were merged. The input of the second layer was fused with the parameters of the reciprocal second layer, and so on. The right panel is the segmentation.

typical phenotypic dataset on plants, the majority of the pixels can be classified into the following types: plant, soil, flowerpots or culture medium, and tags, *etc*. The image segmentation system can divide the images into a number of classes according to the defined characteristics. Hence, it is important to retain boundary information in the extracted image representation. The L*a*b* color space is a color model developed by the International Commission on Illumination (CIE) [11]. An RGB color space can be transformed into an L*a*b* color space, which makes it easier to use the clustering algorithm to classify the desired color. In the L*a*b* color space, the color information is deposited in '*a*b*' space. The input for K-means is a combination of (a*, b*). The colors in 'L*a*b*' space were classified using K-means clustering. Every pixel in the image was labeled using the results from K-means.

2.3. Deep Convolutional Autoencoder for Image Segmentation

On the basis of SegNet [8], we propose a neural network model for the segmentation of the phenotypic images, which is referred to as TAIR-Net, and the structure is shown in Fig. (**2**) and Table **1**. The training data for TAIR-Net were selected from the best segmentations using the K-means algorithm. We implement the algorithm on the Keras framework using the deep convolutional autoencoder model [12]. The initial values of the encoder and decoder are initialized with He's weight initialization (https://keras.io/). In addition, TAIR-Net uses binary cross-entropy loss as the loss function. We tested both binary cross entropy and mean square error in our model and found that using binary cross entropy as a loss function in our model results in a smaller training loss. The loss in the autoencoder should capture the discrepancy between the decoded images and the original images. Thus, we used per-pixel binary cross entropy loss in this study. Stochastic gradient descent (SGD) with a Nesterov momentum of 0.9 was used to train TAIR-Net. Its initial learning rate is 0.01 and decay of 1E-6 over each update. The mini-batch size is 16 and the total epochs are 30.

2.4. Content-based Image Retrieval

These are four distinct parts in the image retrieval system: (i) dividing distinct areas in the image, (ii) feature extraction which determines the features that need to be extracted, (iii) feature representation which can be the largest region or a proportion of the colors in the region, (v) query processing. With the development of deep learning and deep neural networks, the features can be extracted from the data through the neural network model. In this study, we also use the convolutional autoencoder to train the image data and build the image feature library. In 2012, Krizhevsky *et al*. characterized the global image with an intermediate node of autoencoders for image retrieval [13]. In this study, following the methods proposed by Krizhevsky [13], we build a deep convolutional autoencoder model, referred to as Search-Net, to extract the features of an image after image segmentation by TAIR-Net. The feature vector is adopted to establish the phenotype feature library. The cosine similarity between the feature vectors characterizes the degree of association between the query image and the phenotype feature library. A small cosine similarity presents a similar phenotype. Fig. (**3**) and Table **2** show the architecture of Search-Net. During the training of Search-Net, the same plant might be regarded as different objects due to the angle of the camera, light intensity, growth stage, object concealment and other issues. In addition, plants with the same genetic background will show different phenotypes in

Table 1. The 38-layer architecture of TAIR-Net.

Layer	Type	Filter	Kernel Size
0	input	shape=(256,256,3)	
1	convolutional	64	3x3
2	convolutional	64	3x3
3	max pooling		pool_size(2,2)
4	convolutional	128	3x3
5	convolutional	128	3x3
6	max pooling		pool_size(2,2)
7	convolutional	256	3x3
8	convolutional	256	3x3
9	max pooling		pool_size(2,2)
10	convolutional	512	3x3
11	convolutional	512	3x3
12	Dropout		dropout_rate=0.5
13	max pooling		pool_size(2,2)
14	convolutional	1024	3x3
15	convolutional	1024	3x3
16	up sampling		size(2,2)
17	convolutional	512	2x2
18	merge[12,17]		
19	convolutional	512	3x3
20	convolutional	512	3x3
21	up sampling		size(2,2)
22	convolutional	256	2x2
23	merge[8,22]		
24	convolutional	256	3x3
25	convolutional	256	3x3
26	up sampling		size(2,2)
27	convolutional	128	2x2
28	merge[5,27]		
29	convolutional	128	3x3
30	convolutional	128	3x3
31	up sampling		size(2,2)
32	convolutional	64	2x2
33	merge[3,32]		
34	convolutional	64	3x3
35	convolutional	64	3x3
36	convolutional	3	3x3
37	convolutional	3	1x1

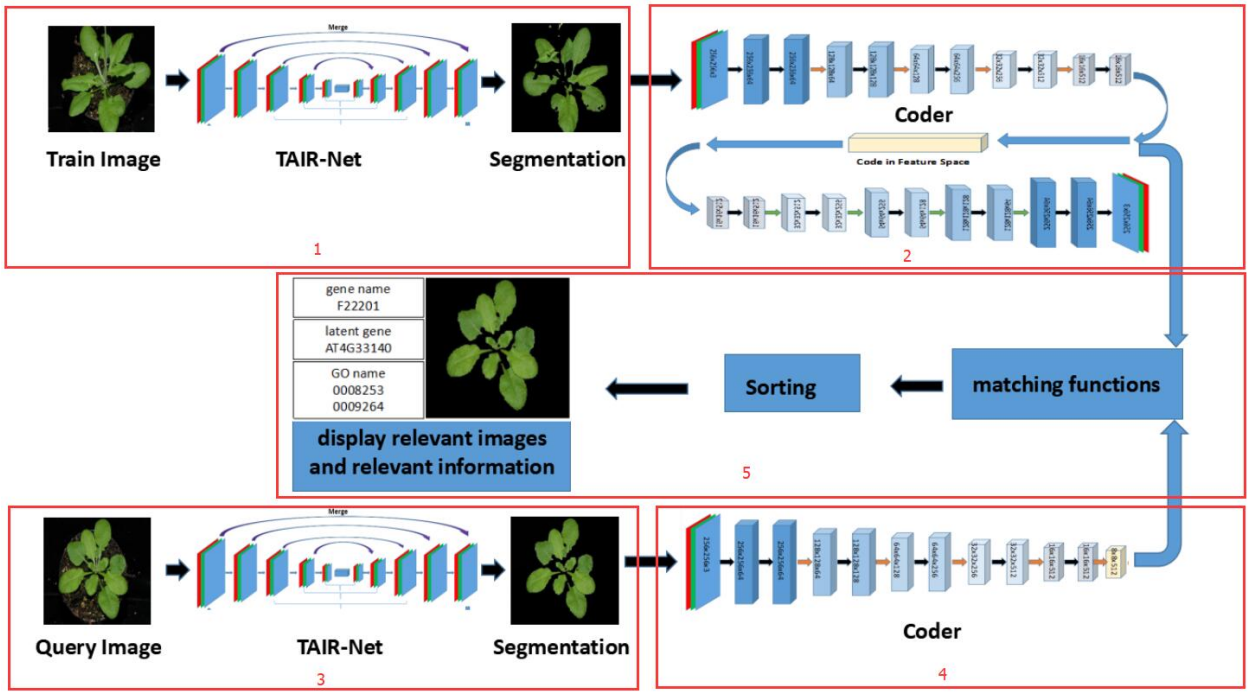

Fig. (3). The training and retrieval architecture of the whole phenotype retrieval system. Box 1 presents the TAIR-Net training process using the deep convolutional autoencoder model; Box 2 presents the Search-Net training process. Box 3 presents the query image, with the background removed using TAIR-Net. Box 4 presents the query image after segmentation was turned into a feature vector. Box 5 shows that the feature vectors from the query image were matched with the features vector in the phenotype feature library. According to cosine similarity, the possible candidates were inferred.

Table 2. The 24-layer architecture of Search-Net.

Layer	Type	Filter	Kernel Size
0	input	shape=(256,256,3)	
1	convolutional	64	3x3
2	convolutional	64	3x3
3	max pooling		pool_size(2,2)
4	convolutional	128	3x3
5	max pooling		pool_size(2,2)
6	convolutional	256	3x3
7	max pooling		pool_size(2,2)
8	convolutional	512	3x3
9	max pooling		pool_size(2,2)
10	convolutional	512	3x3
11	max pooling		pool_size(2,2)
12	convolutional	512	3x3
13	up sampling		size(2,2)
14	convolutional	512	3x3
15	up sampling		size(2,2)
16	convolutional	256	3x3
17	up sampling		size(2,2)
18	convolutional	128	3x3
19	up sampling		size(2,2)
20	convolutional	64	3x3
21	up sampling		size(2,2)
22	convolutional	64	3x3
23	convolutional	3	3x3

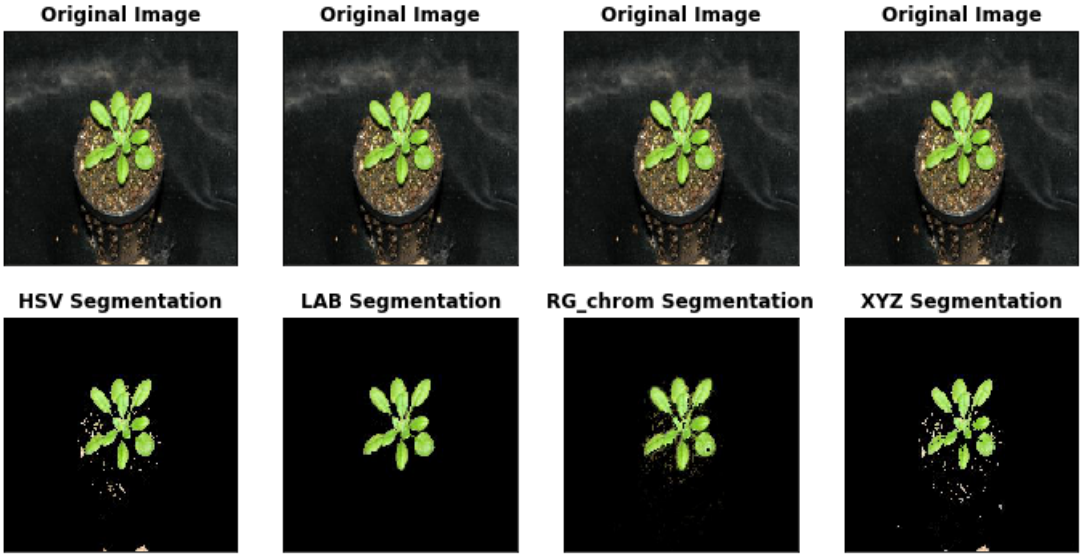

Fig. (4). Four similar color spaces, including HSV, L*a*b*, rg chromaticity and XYZ were compared using the K-means algorithm with the same parameters. According to the effect of segmentation, we conclude that the L*a*b* color space achieved the best performance.

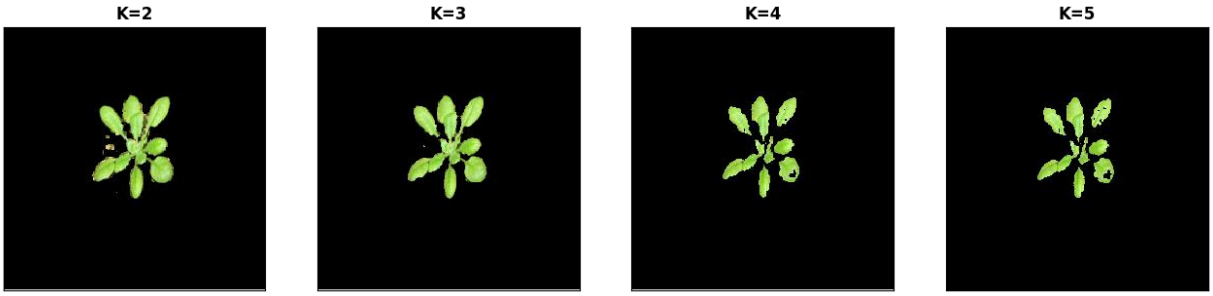

Fig. (5). Pre-segmentation performance using different clusters. Background information was introduced using K=2 for segmentation. The leaves are using K = 4 or more. K = 3 is more appropriate for image segmentation.

different physiological, biotic and abiotic conditions, which makes it difficult to match. In order to overcome these challenges, we adopt data augmen-tation technology. We use a combination of different transformations such as rotation, flip, zoom, scale, contrast, and color to increase the phenotype features, enabling the deep neural networks to learn more image invariance features during training.

3. RESULTS AND DISCUSSION

3.1. The K-means Algorithm for Phenotype Image Pre-segmentation to Train the Autoencoder

Background removal is a preprocessing task during image segmentation. If only the autoencoder model is adopted for image segmentation, the image segmentation will have to be done manually for the training data set, which is time-consuming and laborious. If only the K-means algorithm is used for segmentation, there is no need to perform manual segmentation, however the effect of the K-means algorithm is not as good as that of the autoencoder model. Therefore, we combine the advantages of the K-means algorithm and the autoencoder model in this study.

Any color information expressed in the RGB color space can be described and mapped in the L*a*b* color space. The K-means algorithm is an unsupervised learning algorithm, which assumes that the clustering structure can be characterized by a set of prototypes [11]. In this study, we compared four similar color spaces, namely HSV, L*a*b*, rg chromaticity and XYZ. Using the K-means algorithm, we obtained different segmentation results from the four color spaces using the same parameters. After manually checking a large number of segmentations, we found that the L*a*b* color space had the best performance (Fig. **4**).

We also tested different Ks and manually judged the effect and found that K = 3 is more appropriate to use for training the autoencoder. Fig. (**5**) shows a randomly selected segmentation using different clusters. As can be seen from the figure, segmentation using K=2 introduced part of the background information. However, if K = 4 or 5, the leaves will be split. Based on these results, the best pre-segmentation performance is obtained when k =3 since the culture medium, soil and label are the three major factors which can disturb the performance of segmentation.

In this study, TAIR-Net has 38 neural network layers. For a 256 x 256 image, a batch size of 32 will take up 12.6G of graphic memory. Thus, these phenotype images were reshaped into 256 x 256 sizes in this study to avoid out of memory (OOM) error due to higher resolution. Since our

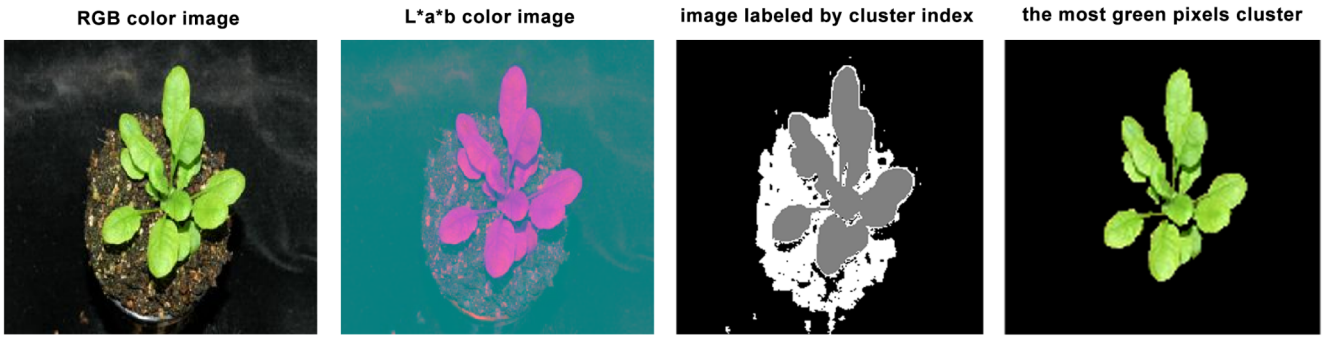

Fig. (6). Removing the background from the image using the L*a*b* color space by the K-means algorithm. The left-most image is the original image for segmentation, the second image is the image converted to the L*a*b* space. The third image is divided into three classes using the K-means algorithm. The last one shows the effect of image pre-segmentation.

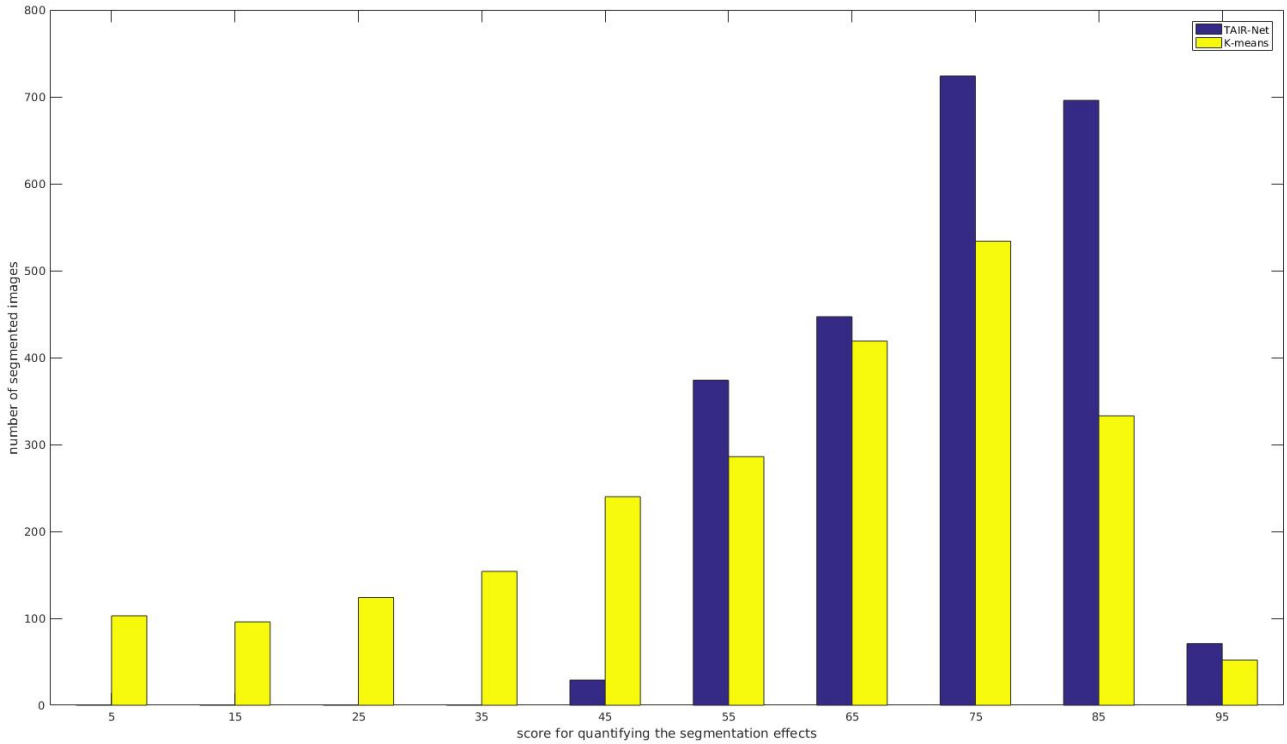

Fig. (7). A comparison between the K-means algorithm and TAIR-Net segmentation. TAIR-Net segmentation scores were significantly higher than the K-means algorithm according to the manual evaluation as shown in the histogram.

source code has been released, the user can select a higher resolution to obtain better results.

We first converted all the images in the dataset into the L*a*b* color space and clustered the objects into three clusters using the K-means algorithm, and then detected the most green area as the segmentation result of the image. Our dataset was not a benchmark dataset. Thus, we need to assess the effect of segmentation manually. We used 11 numerical intervals (0, 10, 20, 30, 40, 50, 60, 70, 80, 90, 100) to quantify the segmentation effect according to three different individuals conducting three independent rating assessments. In total, there were 1338 pictures giving a median score of 70 points or more in this study. In the original data, these are 2,341 phenotype images of different image sizes. After the initial division of the K-means algorithm, there were 1,338 images, which is a good segmentation effect. The results of image segmentation using the K-means algorithm is visualized in Fig. (**6**). Then we selected the image segmentation as training data.

3.2. Deep Convolutional Autoencoder for Segmentation

The major interference factor for biological phenotypic images includes culture medium, soil and label, *etc*. However, there are other factors which may interfere with the effect of image segmentation. Thus, part image segmentation using the K-means algorithm in the L*a*b* color space results in incomplete segmentation. The autoencoder model segments images for the training data manually. However, the disadvantage of this method is that it is time-consuming. In order to address this issue, we use the K-means algorithm to combat the disadvantages of manual segmentation. Pre-segmentation from K-means is sufficient as the training data for the autoencoder model. Finally, we used the trained autoencoder for all the image segmentation.

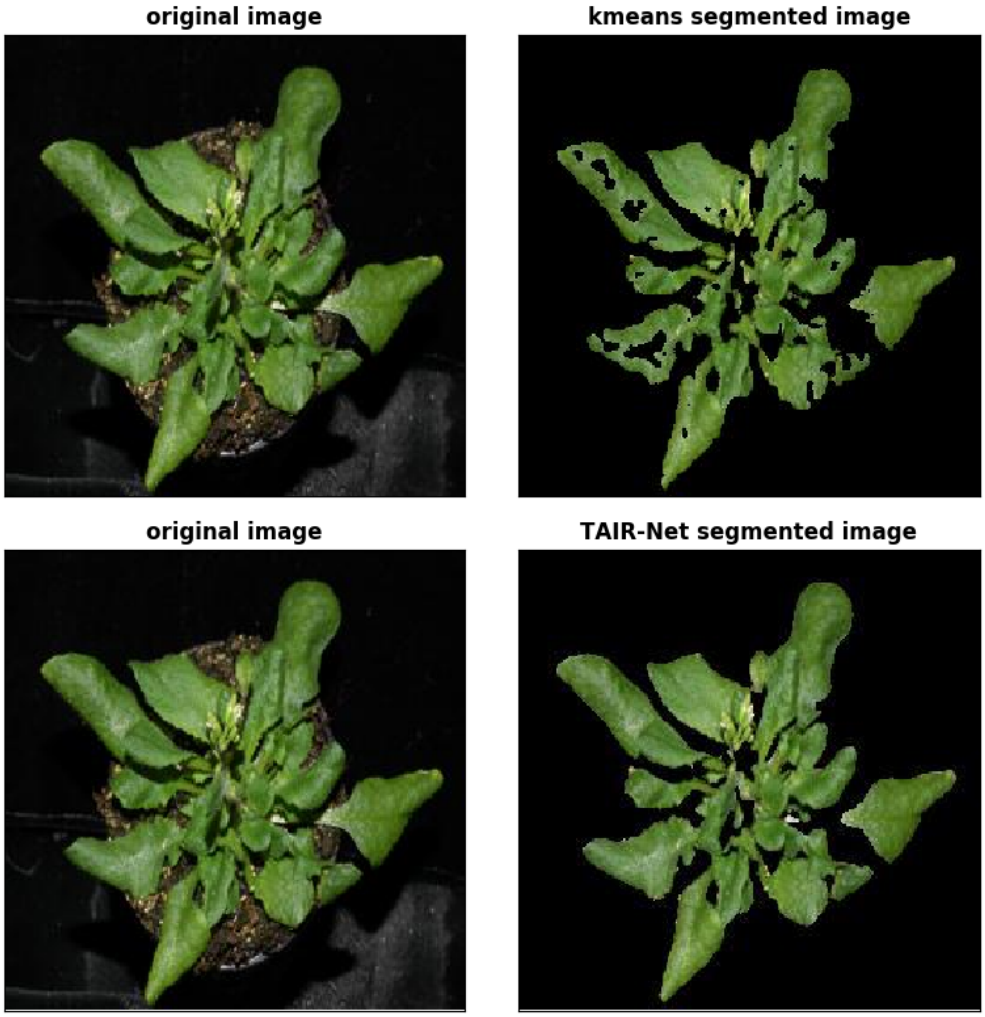

Fig. (8). A comparison of the K-means algorithm segmented image and the TAIR-Net segmented image. The images on the upper left and lower left are the original images. The upper right is the image segmentation by the K-means algorithm, and the lower right is the image segmentation using TAIR-Net.

On the basis of SegNet [8], we propose a neural network model for biological phenotypic image segmentation, which is referred to as TAIR-Net. In total, 1,338 images with the best segmentation from the K-means algorithm are used as the training sample as TAIR-Net training data. The effects of the K-means algorithm and TAIR-Net segmentation were evaluated manually. The histogram comparing the K-means algorithm and the TAIR-Net segmentation is shown in Fig. (**7**), which shows that TAIR-Net segmentation scores are significantly higher than the K-means algorithm. The results suggest that we can obtain great improvement during phenotypic image segmentation (Fig. **8**). This study addresses not only the defects of the K-means algorithm but the small amount of sample data for deep learning training. The segmented images in Fig. (**9**) indicate that TAIR-Net could achieve better results in different growth stages.

3.3. The Architecture of the Feature Extraction System

TBIR is one image retrieval technology, which has a fast retrieval speed, is convenient to operate and is easy to implement. However, TBIR cannot be updated automatically. The time consuming manual annotation also limits the application of TBIR. Thus, we use CBIR in the image retrieval system. The extracting feature vector module is essential for CBIR. After the training of TAIR-Net is completed, all the remaining images are processed by TAIR-Net and are regarded as Search-Net training data. Both TAIR-Net and Search-Net are deep convolutional autoencoder models and use binary cross-entropy loss as the loss function. We used the AdaDelta optimizer with a learning rate of 4.0 to train the Search-Net. The mini-batch size is 32 and the total epoch is 400. This kind of retrieval method extracts information from the image content, using the image feature to index and retrieve, so that the retrieval is more efficient (Fig. **10**). When an image with the same content has different forms, [such as pictures for the same object taken from different angles, using CBIR technology, these images can be retrieved as well.

3.4. Example of a Matching Reference Biological Phenotype Database

We selected 50 query images as the test dataset, which included a known gene ontology (GO). The remaining images were adopted to establish the feature extract system,

Fig. (9). The segmented images in different stages. TAIR-Net can achieve better result in different growth stages in *Arabidopsis*.

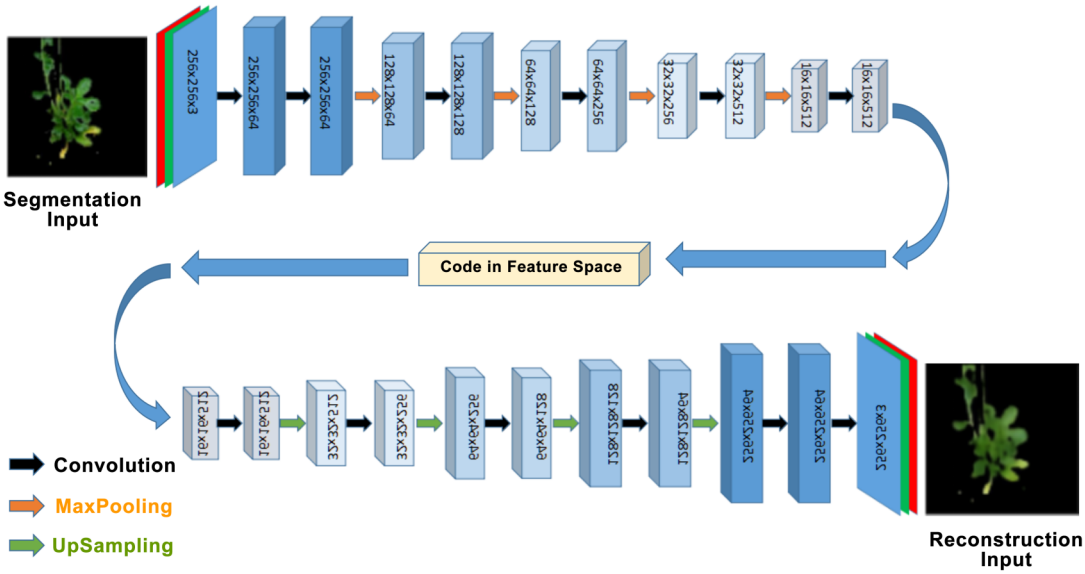

Fig. (10). The architecture of the feature extraction system. This diagram was adopted to demonstrate the Search-Net training process. Search-Net consists of encoders and decoders. The input part of Search-Net was image segmentation from TAIR-Net, which was loaded into the encoder. The output from the decoder was an image, which was similar to the original pictures. The black arrow indicates that the layer was a convolution layer, the orange color represents the MaxPooling layer. The green color represents the upper sampling layer. The feature vector was stored in the phenotype feature library as soon as the Search-Net training was completed.

as described in this study. After all the cosine similarity values were calculated and the query results were sorted in ascending order, Search_Net returned the most similar phenotypes images and corresponding GO. We regarded the first five as possible candidates. Of the 50 tested images, there were 42 query images including the same GO with respective candidate phenotypes images, which suggests the validity of the image query (Fig. **11**).

To demonstrate the application of content-based image retrieval, we used the query dataset from the publication of the *ago1-27* mutant [14], which is involved in the microRNA

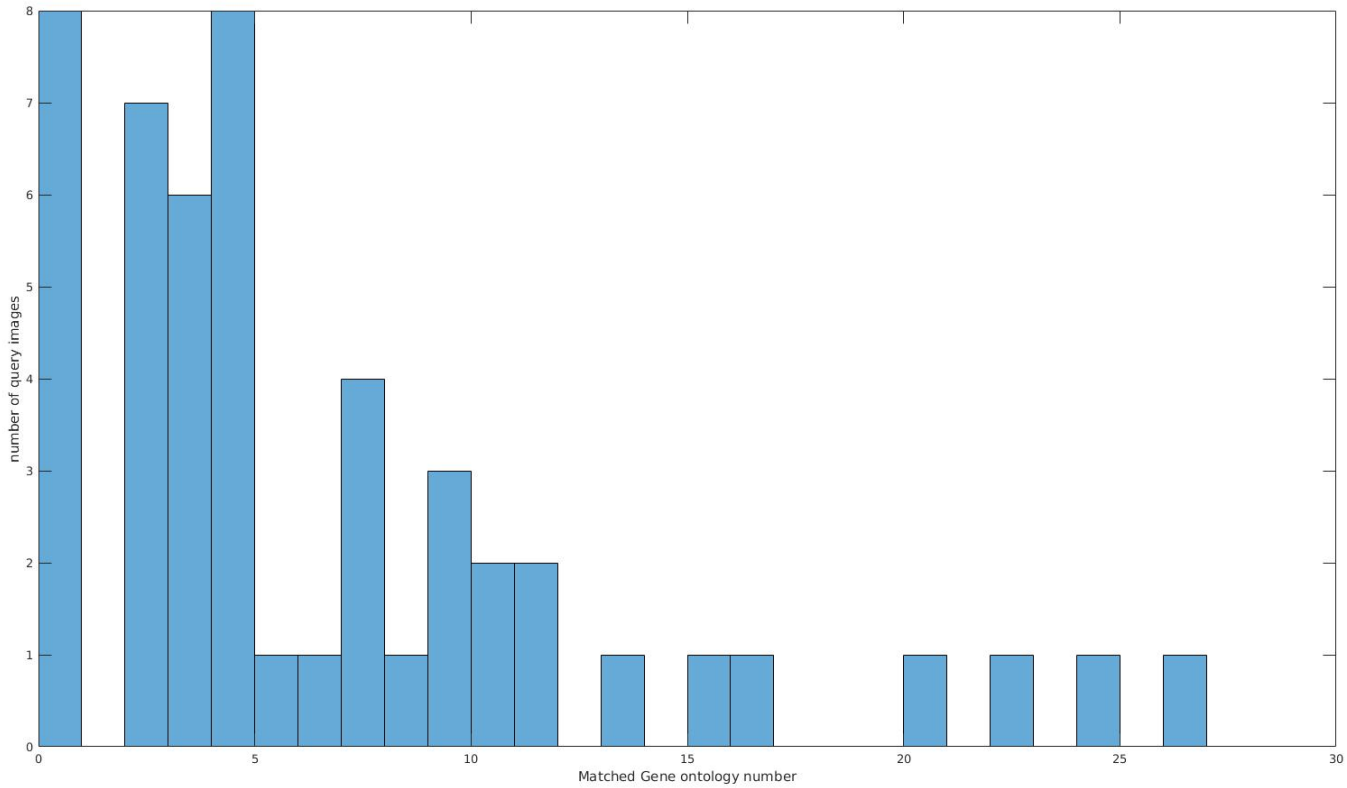

Fig. (11). Validity of the image query matching reference biological phenotype database. Of the 50 tested query images, there are 42 images including the same GO with respective candidates. Number of matched GO and images were plotted on the x-axis and y-axis, respectively.

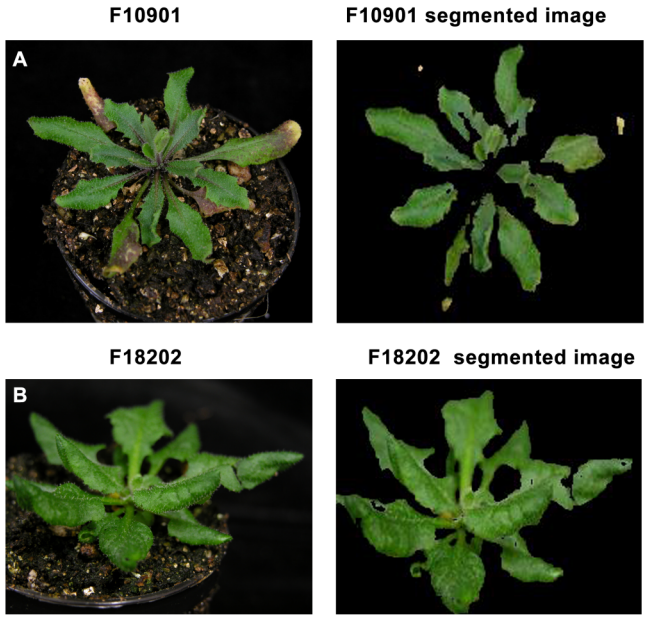

Fig. (12). Example of matching reference biological phenotype database. The image labeled F01901 was the query image and the phenotype retrieval system returned one image labeled F18202 as the potential candidate. By analyzing their corresponding GO, we found that two genes included the same GO number.

pathway. When the user submits an image of the mutant to the retrieved system, the background information from the *ago1-27* mutant will be removed by TAIR-Net and will be coded into a feature vector through the encoder of Search-Net. Lastly, the retrieval system finds the most similar feature from the reference library based on cosine similarity and returns the most related images corresponding to these features (Fig. **12**). The top match retrieved by the system includes one mutant line (Fig. **6B**), which encodes a member of the plant transcription factor (AT1G53230) which includes Teosinte branched 1, Cycloidea 1, and proliferating cell nuclear antigen (PCNA) factors. The gene is involved in the heterochronic regulation of leaf differentiation and is targeted by miRNA-319. The above example suggests that searching using the *ago1-27* mutant returns functionally related mutants. In summary, the above example shows that the retrieved system is able to detect the functional relation between similar phenotypes.

Importantly, the user should be aware of the limitations of this method. Firstly, to obtain meaningful results, a deluge of high-resolution reference phenotype images is essential for the training data. Advances in automated high-throughput imaging technologies can solve this challenge, such as unmanned aerial vehicles [15, 16], LemnaTec (http://www.lemnatec.com/) and PSI (http://plantphenotyping.com/), which can generate a massive amount of phenotype images. In the future, reference images which come from natural genetic variation [17] can also be included in this study. Another limitation is the randomness for the query image, such as different environmental effects, varying growth stages, different scales and different capture perspectives which can disturb the accuracy of the comparison. For the present, we have adopted data augmentation, which resulted in substantial improvements. However, data augmentation requires more computer resources and time for training.

3.5. Computation Time for the Framework

TAIR-Net was trained on a computer with a dual GTX1080 graphics card, i7-6700k, 32G RAM. The batch_size of TAIR-Net is set to 16, it trained 200 epochs and took about 5 hours. The batch_size of Search_net was set to 32 and it trained 200 epochs, taking about 3 hours. The K-means algorithm, TAIR-Net and Search_Net were tested on a single GTX1070 graphics card, i7-3770, 24 GRAM computer. K-means took around 320 seconds to segment all the 2341 pictures. TAIR-Net took about 2s to segment a single picture and occupied 2.0G of RAM; TAIR-Net took about 500s to segment the 2341 pictures, which is approximately an average of 4.7 pictures per second. The time to retrieve a single picture (including the model loading time) was 3.5s, and it occupied 5.6G of RAM.

CONCLUSION

In this study, we provided a powerful framework using deep convolutional autoencoder architecture to overcome the bottleneck of searching for large biological phenotype datasets. This system contains only reference images from *Arabidopsis thaliana*, which includes many phenotype resources. It will be interesting to see if our method can be expanded to other species with a large number of phenotypic images from mutant or overexpressing lines.

Taken together, this phenotype retrieval system is especially important as it provides further information on the correlation between genotype and phenotype. In order to provide the user with easier installation without the need to deal with dependencies, we deployed the retrieval system by Docker which is freely available at http://www.bioinfor.org/tool/TAIR-Seq-Net. The source code can be found at https://github.com/wubizhi/Phenotypic-images-retrieval-system so the user can set up their own neural network training.

LIST OF ABBREVIATIONS

BLAST	=	Basic Local Alignment Search Tool
FCN	=	Fully convolutional networks
TBIR	=	Text-based image retrieval
CBIR	=	Content-based image retrieval
SGD	=	Stochastic gradient descent

ETHICS APPROVAL AND CONSENT TO PARTICIPATE

Not applicable.

HUMAN AND ANIMAL RIGHTS

No Animals/Humans were used for studies that are base of this research.

CONSENT FOR PUBLICATION

Not applicable.

AVAILABILITY OF DATA AND MATERIALS

Not applicable.

FUNDING

None.

CONFLICT OF INTEREST

The authors confirm that this article content has no conflict of interest.

ACKNOWLEDGEMENTS

This work was supported by International Science and Technology Cooperation and Exchange Fund from Fujian Agriculture and Forestry University (KXGH17016), the National Key Research and Development Program of China (2016YFD0600106), the National Natural Science Foundation of China Grant (Grant No. 31570674 and 61702100).

REFERENCES

[1] Akiyama K, Kurotani A, Iida K, Kuromori T, Shinozaki K, Sakurai T. RARGE II: an integrated phenotype database of Arabidopsis mutant traits using a controlled vocabulary. Plant and Cell Physiology. 2013;55(1):e4-e.

[2] Swarbreck D, Wilks C, Lamesch P, Berardini TZ, Garcia-Hernandez M, Foerster H, *et al*. The Arabidopsis Information Resource (TAIR): gene structure and function annotation. Nucleic acids research. 2007;36(suppl_1):D1009-D14.

[3] McGinnis S, Madden TL. BLAST: at the core of a powerful and diverse set of sequence analysis tools. Nucleic acids research. 2004;32(suppl_2):W20-W5.

[4] Altschul SF, Gish W, Miller W, Myers EW, Lipman DJ. Basic local alignment search tool. Journal of molecular biology. 1990;215(3):403-10.

[5] Bengio Y. Learning deep architectures for AI. Foundations and trends® in Machine Learning. 2009;2(1):1-127.

[6] Masci J, Meier U, Cireşan D, Schmidhuber J. Stacked convolutional autoencoders for hierarchical feature extraction. Artificial Neural Networks and Machine Learning–ICANN 2011. 2011:52-9.

[7] Long J, Shelhamer E, Darrell T, editors. Fully convolutional networks for semantic segmentation. Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition; 2015.

[8] Badrinarayanan V, Kendall A, Cipolla R. Segnet: A deep convolutional encoder-decoder architecture for scene segmentation. IEEE transactions on pattern analysis and machine intelligence. 2017.

[9] Li W, Duan L, Xu D, Tsang IW-H, editors. Text-based image retrieval using progressive multi-instance learning. Computer Vision (ICCV), 2011 IEEE International Conference on; 2011: IEEE.

[10] Singhai N, Shandilya SK. A survey on: content based image retrieval systems. International Journal of Computer Applications. 2010;4(2):22-6.

[11] Luccheseyz L, Mitray S. Color image segmentation: A state-of-the-art survey. Proceedings of the Indian National Science Academy (INSA-A). 2001;67(2):207-21.

[12] He K, Zhang X, Ren S, Sun J, editors. Delving deep into rectifiers: Surpassing human-level performance on imagenet classification. Proceedings of the IEEE international conference on computer vision; 2015.

[13] Krizhevsky A, Hinton GE, editors. Using very deep autoencoders for content-based image retrieval. ESANN; 2011.

[14] Morel J-B, Godon C, Mourrain P, Béclin C, Boutet S, Feuerbach F, *et al*. Fertile hypomorphic ARGONAUTE (ago1) mutants impaired in post-transcriptional gene silencing and virus resistance. The Plant Cell. 2002;14(3):629-39.

[15] Singh A, Ganapathysubramanian B, Singh AK, Sarkar S. Machine learning for high-throughput stress phenotyping in plants. Trends in plant science. 2016;21(2):110-24.

[16] Peña JM, Torres-Sánchez J, Serrano-Pérez A, de Castro AI, López-Granados F. Quantifying efficacy and limits of unmanned aerial vehicle (UAV) technology for weed seedling detection as affected by sensor resolution. Sensors. 2015;15(3):5609-26.

[17] Seren Ü, Grimm D, Fitz J, Weigel D, Nordborg M, Borgwardt K, *et al*. AraPheno: a public database for Arabidopsis thaliana phenotypes. Nucleic acids research. 2017;45(D1):D1054-D9.