Flower Picture Classification Based on Convolutional Neural Network

Ben Zhao
School of Information Technology, Shanghai Jian Qiao University, Shanghai, China

Keywords: Convolutional Neural Networks, Classification of Flower, Images, ReLu, Softmax.

Abstract: Due to changes in the ecological environment, many species of flowers are on the brink of extinction. By using scanning technology to help people quickly identify each type of flower, that can implement conservation measures more directly and effectively. In this paper, characteristics are extracted from four different types of plant pictures using a convolutional neural network (CNN) model. Model checkpoints and early stopping techniques were used to preserve the trained model during training. The trained model is used to predict a single image, classify the flower according to its characteristics, and finally output the result. However, the output results show that although the accuracy is very high, the precision is abnormally low, which indicates that the model may be overfitting. In the future, the quality of the models can be further improved by increasing the complexity of the models, or balancing the data sets to more accurately protect these endangered flower species.

1 INTRODUCTION

The application of plant species identification is found in multiple fields, such as smart agriculture, herbal research, AR, and VR. The CNN model has significant advantages in image recognition,The CNN model can be used to train on the features of plants in images (Wang, Wang, Zhang, et al., 2020)(Saba, Rehman, Jamail, Marie-Sainte, Raza & Sharif. 2021), such as recognizing the shape of flower petals, enabling the model to have the capability to identify plants,Obtain a more suitable model by using early stopping(Yan, Zhang & Wu. 2019). The specific implementation involves searching for datasets on Kaggle, and then processing a series of image data using TensorFlow 2.x(Abadi, Barham, Chen, et al., 2016). The specific implementation involves searching for datasets on Kaggle and then using TensorFlow 2.x to process a series of image data.(Abadi, Barham, Chen, et al., 2016). Save the processed image data as TFRecords files, and then use the data from the TFRecords files to train a convolutional neural network model (Liu, Zhang & Zhou. 2022), employing ReLu as the activation function and using the softmax function (Banerjee, Gupta, Vyas, et al., 2020) as the output layer to produce predicted probabilities.

However, in recent years, there have been many challenges in plant recognition technology. Factors such as lighting, soil, and other plants can affect the capture of plant images, the accuracy of recognition was significantly reduced as a result. The aim of this paper is to achieve the functionality of plant recognition, and through subsequent model optimization, it may be possible to reduce the impact of natural factors such as lighting. In complex environments, it can accurately identify plant species and also contribute to the protection of plants.

2 METHODS

When biologists study plant diversity, they can identify plant species by using drone scans to recognize plant codes. However, the accuracy of drone scanning is affected by many factors, such as sunlight, weather, and so on, facing some challenges that are difficult to resolve. This project can identify four types of flowers, including dandelions, roses, sunflowers, and tulips. However, during the experiment, due to the imbalance of data categories and the issue of overfitting, the trained model was unable to accurately identify the species.

Zhao and B.
Flower Picture Classification Based on Convolutional Neural Network.
DOI: 10.5220/0013487000004619
In *Proceedings of the 2nd International Conference on Data Analysis and Machine Learning (DAML 2024)*, pages 40-44
ISBN: 978-989-758-754-2

2.1 Data processing

During the data processing stage, the PIL library was used to load images, converting the training data into uniform photos of 64×64 pixels. These data were then transformed into TFRecords files. This approach can enhance the efficiency of data processing. TFRecords(Haloi & Shekhar. 2021) is a binary format in TensorFlow 2.x that is particularly suitable for handling large datasets. During the training process, data is read from the file, converted into image format, and then decoded and normalized. At the same time, before training the model, it is necessary to divide the dataset into a training set and a validation set. Using tf.data to handle the data can improve data processing efficiency.

2.2 Model building

The convolutional neural network (CNN)(Yan, Guo, Xiao & Zhang. 2020)architecture shown in Figure 1 is used in model building; it is specifically designed to handle data with a grid structure, and images are a prime example of this. It consists of regularization, fully linked, pooling, and convolutional layers. Its filters are only connected to a small portion of the input, allowing it to capture local features. Simultaneously, the same filter at many locations might share the same weights, which lowers the number of parameters. In CNNs, filters have the characteristic of translational invariance, meaning that no matter where the recognized features appear subsequently, they can still be identified. For example, the shape of the petals can accurately capture their form, regardless of how they appear in the image. In the construction of the model, the ReLu(Zhao, Zhang, Guan, Tang & Wang. 2018) activation function(1) and softmax(2) output layer(Lee, Wang & Cho. 2022) were also used to predict the probability of each category. ReLu is an activation function commonly used in CNNs(Zheng, Han & Soomro. 2020)(Demir, Abdullah & Sengur. 2020).

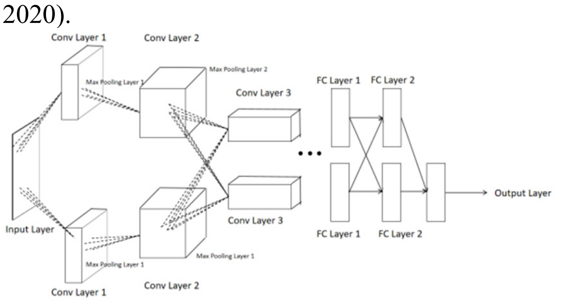

Figure 1: The architecture of convolutional neural networks (Picture credit : Original)

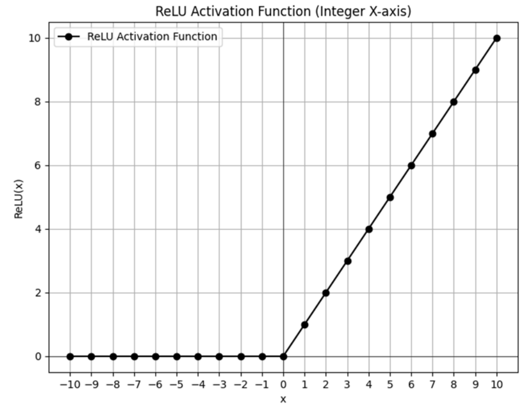

Figure 2: ReLu Activation Function (Picture credit : Original)

$$f(x) = \max(0,x) \tag{1}$$

Just as shown in Figure 2.When $x > 0$, output x; otherwise, output 0. The output of this function is greater than or equal to 0. ReLu(Agarap. 2018) is a nonlinear activation function where any negative input results in an output of 0. This creates sparse activation, which can reduce the risk of overfitting. Moreover, if the input of a neuron is 0, then its corresponding weight will also be 0, which can speed up the training process.

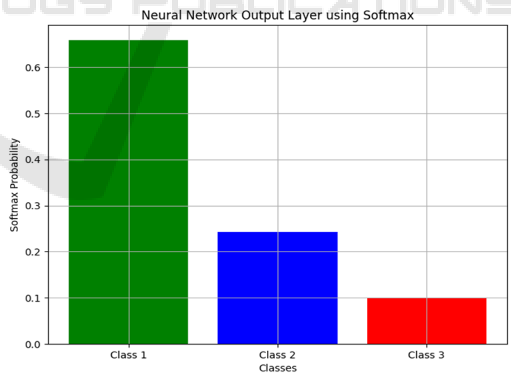

Figure 3: Neural Network Output Layer using Softmax (Picture credit : Original)

$$\sigma(z_i) = \frac{e^{z_i}}{\sum_{j=1}^{n} e^{z_j}} \tag{2}$$

Figure 3 shows the outputs when the input values to the softmax function are 2, 1, and 0.1. The softmax function can convert different logits into probabilities, helping to select the most likely category, with the sum of all output probabilities equal to 1.

In formula (2). z_i represents the i-th element in the input vector, usually the logits or the output values of the last layer of a neural network. N is the number of elements in the input vector, which is also the number of categories. e^{z_i} is the exponentiation of z_i . $\sum_{j=1}^{n} e^{z_j}$ is the sum of the exponential results of all input values z_j, used for normalization, so that the output values represent probabilities.

2.3 Model training

In the model training, the Adam optimizer and cross-entropy loss function were used. At the same time, to prevent overfitting, the project employed an early stopping method to avoid its occurrence. At the same time, model checkpoints were used to save the model, ensuring that the best model parameters were saved as best_model.keras for subsequent model evaluation.

2.4 Model evaluation and prediction

In the evaluation, the model's performance is measured through four indicators, namely (3), (4), (5), and (6).

$$\mathrm{Loss} = -\frac{1}{\mathrm{N}}\sum_{\mathrm{i}=1}^{\mathrm{N}}\sum_{\mathrm{c}=1}^{\mathrm{C}} \mathrm{y}_{\mathrm{i,c}}\log\left(\widehat{\mathrm{y}_{\mathrm{i,c}}}\right) \tag{3}$$

The difference between the values that the model predicts and the actual values is typically measured using loss. Here, N stands for the total number of samples, C for the number of categories, $y_{i,c}$ represents sample i's true label in category c, and $\widehat{y_{i,c}}$ represents sample i's expected probability in category c.

$$\mathrm{Accuracy} = \frac{\mathrm{TN} + \mathrm{TP}}{\mathrm{FP} + \mathrm{FN} + \mathrm{TP} + \mathrm{TN}} \tag{4}$$

The terms True Positives (TP), True Negatives (TN), False Positives (FP), and False Negatives (FN) are used in this context. The ratio of correctly predicted samples to total samples is known as accuracy.

$$\mathrm{Precision} = \frac{\mathrm{TP}}{\mathrm{FP} + \mathrm{TP}} \tag{5}$$

The percentage of real positive cases among the samples that the model predicts as positive is known as precision. True Positives (TP) and False Positives (FP) are the terms used in this context.

$$\mathrm{Recall} = \frac{\mathrm{TP}}{\mathrm{TP} + \mathrm{FN}} \tag{6}$$

Recall is the percentage of real positive samples that the model accurately detected. In this context, TP stands for True Positives, and FN stands for False Negatives.

However, the model still needs optimization. Although the accuracy has reached 99.56%, both precision and recall are very low. In this experiment, the number of images for each type of flower differs by about 100, which may lead to a lower accuracy rate. Initially, the resolution of the training image data was 64 × 64, and the lower resolution may lead to a mismatch between the model's precision and recall. Later, when training on images with a resolution of 300×200, these two metrics were still almost similar to those of 64×64. Therefore, the low Precision and Recall should not be attributed to the dataset, but rather to issues with the model itself. Thus, to improve these two metrics in the future, it is essential to further optimize the model. Meanwhile, the accuracy (Acc) during the model training period reached 1.0, which is likely a sign of overfitting. In the subsequent optimization, increasing the amount of training data, adjusting the learning rate, and paying attention to the adjustment of model complexity will be important.

3 RESULTS

The data comes from Kaggle, where the number of rose and sunflower images is relatively small, around 700-800 each, while the other two types have about 1,000 photos each, totaling over 3,000 photos. Their resolution is generally 300×200, and compressing these photos to a resolution of 64 × 64 can significantly reduce the training time required. 70% of the total dataset was allocated for training, while 30% was used as the validation set. During the training, a total of 40 sessions were conducted, with 50 photos used in each session. Efforts were made to ensure that the photos in the training set were fully utilized. Training for more than 50 sessions could lead to a decrease in accuracy, so 50 sessions is considered a reasonable.

The model training results are displayed in Table 1. The recall and precision rates are still comparatively poor, as may be shown. Consequently, the main goal of future research should be to optimize the model and modify its complexity in order to increase the value of each metric. This will help to ensure that the model operates effectively in reality and improves accuracy.

Table 2 shows the results of predictions based on a single photo. Examining a photo of a rose, as shown in Figure 4, it indicates the likelihood of it being a dandelion. It may be due to the resolution of the photos being compressed to 64 × 64 for training, which leads to insufficient feature extraction during training, resulting in inaccuracies during classification, and ultimately failing to accurately identify which category the image belongs to. The possibility of dandelions in Table 2 should theoretically be very low, but the identified results are close to 50%.

Table 1 Results of various indicators for model training

Metric	Epoch	Loss	Accuracy	Precision	Recall
Value	40	0.1748	0.9657	0.2636	0.2727

Table 2 Results of Predictions for a Single Photo

Metric	The possibility of dandelions
Value	0.451324

Figure 4: Roses used for prediction (Mamaev, 2021).

4 CONCLUSIONS

This experiment used a convolutional neural network (CNN) model, employing ReLu as the activation function during training. It also incorporates early stopping and model checkpoints to save the trained model. Through the recognition of a single photo, it was found that although the accuracy reached 0.96, the precision was only 0.27. Therefore, there are some potential issues during the model training process. For example, issues such as excessive false positives, over-prediction of positive cases, and data imbalance need to be addressed. To achieve high accuracy in identifying flower species, it is necessary to further optimize the model, such as adjusting the learning rate to minimize the risk of overfitting.By adjusting the model, it may be possible to effectively identify the features of images and accurately classify them, thereby reducing the interference of the natural environment on classification recognition in real-world applications.

REFERENCES

Abadi, M., Barham, P., Chen, J., (2016). TensorFlow: A system for large-scale machine learning. In *12th USENIX Symposium on Operating Systems Design and Implementation (OSDI 16)* (pp. 265 – 283).

Agarap, A. F. (2018). Deep learning using rectified linear units (relu). *arXiv preprint arXiv:1803.08375*.

Banerjee, K., Gupta, R. R., Vyas, K., et al. (2020). Exploring alternatives to softmax function. *arXiv preprint arXiv:2011.11538*.

Demir, F., Abdullah, D. A., & Sengur, A. (2020). A new deep CNN model for environmental sound classification. *IEEE Access*, 8, 66529 – 66537. https://doi.org/10.1109/ACCESS.2020.2984903

Haloi, M., & Shekhar, S. (2021). Datum: A system for tfrecord dataset management. *GitHub*. Retrieved from https://github.com/openAGI/datum

Lee, J., Wang, Y., & Cho, S. (2022). Angular margin-mining softmax loss for face recognition. *IEEE Access*, 10, 43071 – 43080. https://doi.org/10.1109/ACCESS.2022.3168310

Liu, F., Zhang, Z., & Zhou, R. (2022). Automatic modulation recognition based on CNN and GRU. *Tsinghua Science and Technology*, 27(2), 422 – 431. https://doi.org/10.26599/TST.2020.9010057

Saba, T., Rehman, A., Jamail, N. S. M., Marie-Sainte, S. L., Raza, M., & Sharif, M. (2021). Categorizing the students' activities for automated exam proctoring using proposed Deep L2-GraftNet CNN network and ASO based feature selection approach. *IEEE Access*, 9, 47639 – 47656. https://doi.org/10.1109/ACCESS.2021.3068223

Wang, S. Y., Wang, O., Zhang, R., et al. (2020). CNN-generated images are surprisingly easy to spot... for now. In Proceedings of the *IEEE/CVF Conference on Computer Vision and Pattern Recognition* (pp. 8695 – 8704).

Yan, Y., Zhang, X., & Wu, B. (2019). Simplified early stopping criterion for belief-propagation polar code decoder based on frozen bits. *IEEE Access*, 7, 134691–134696. https://doi.org/10.1109/ACCESS.2019.2940135

Yan, Z., Guo, S., Xiao, G., & Zhang, H. (2020). On combining CNN with non-local self-similarity based image denoising methods. *IEEE Access*, 8, 14789–

14797. https://doi.org/10.1109/ACCESS.2019.2962809

Zhao, G., Zhang, Z., Guan, H., Tang, P., & Wang, J. (2018, August). Rethinking ReLU to train better CNNs. In *2018 24th International conference on pattern recognition (ICPR)* (pp. 603-608). IEEE.

Zheng, G., Han, G., & Soomro, N. Q. (2020). An inception module CNN classifiers fusion method on pulmonary nodule diagnosis by signs. *Tsinghua Science and Technology*, 25(3), 368–383. https://doi.org/10.26599/TST.2019.9010010

Mamaev, A. (2021). Flowers Recognition [Data set]. *Kaggle*. https://www.kaggle.com/datasets/alxmamaev/flowers-recognition