

Flower Image Classification Using Deep Convolutional Neural Network

Author

Alipour, N, Tarkhaneh, O, Awrangjeb, M, Tian, H

Published

2021

Conference Title

2021 7th International Conference on Web Research (ICWR)

Version

Accepted Manuscript (AM)

DOI

<https://doi.org/10.1109/ICWR51868.2021.9443129>

Copyright Statement

© 2021 IEEE. Personal use of this material is permitted. Permission from IEEE must be obtained for all other uses, in any current or future media, including reprinting/republishing this material for advertising or promotional purposes, creating new collective works, for resale or redistribution to servers or lists, or reuse of any copyrighted component of this work in other works.

Downloaded from

<http://hdl.handle.net/10072/406566>

Griffith Research Online

<https://research-repository.griffith.edu.au>

Flower Image Classification Using Deep Convolutional Neural Network

*Neda Alipour

Department of Electrical Engineering
Shahed University
Tehran, Iran
Nedalipur@gmail.com

Mohammad Awrangjeb

Department of Information and Communication Technology
Griffith University
Brisbane, Australia
m.awrangjeb@griffith.edu.au

Omid Tarkhaneh

Department of Computer Science
University of Tabriz
Tabriz, Iran
Tarkhanehomid@gmail.com

Hongda Tian

Department of information Technology
University of Technology Sydney
Sydney, Australia
Hongda.Tian@uts.edu.au

Abstract— These days deep learning methods play a pivotal role in complicated tasks, such as extracting useful features, segmentation, and semantic classification of images. These methods had significant effects on flower types classification during recent years. In this paper, we are trying to classify 102 flower species using a robust deep learning method. To this end, we used the transfer learning approach employing DenseNet121 architecture to categorize various species of oxford-102 flowers dataset. In this regard, we have tried to fine-tune our model to achieve higher accuracy respect to other methods. We performed preprocessing by normalizing and resizing of our images and then fed them to our fine-tuned pretrained model. We divided our dataset to three sets of train, validation, and test. We could achieve the accuracy of 98.6% for 50 epochs which is better than other deep-learning based methods for the same dataset in the study.

Keywords—*Flower Classification; Deep Learning; Transfer Learning; DenseNet*

I. INTRODUCTION

Species information on the various ranges of flowers is important to protect and manage biodiversity. Also, flowers are considered as the most important part of the food chain and habitat for almost all insect pollinators. Therefore, owning an adequate recognition of flower species is essential for biodiversity protection. There are thousands of flowers grow in a wide variety in different countries of the world. Manual identification of all these flower species is a time-consuming and challenging task for even botanical experts.

Recently, with the development of computer vision technology, flower classification has significantly progressed in computer vision. Especially that different types of flowers own similar shapes, colors, and petals. Therefore, the development of a computer-aided method is an urgent step for fast and accurate flower categorization. Various techniques have been developed for image classification, which can be divided into two groups: traditional machine learning methods

and deep learning methods. In the first group of methods, raw images should be transformed into a suitable format in which machine can easily extract handcrafted features such as color, shape and texture [2]. In the second group, raw images can be fed to Convolutional Neural Networks (CNNs) directly without doing much preprocessing. This is one of the reasons these methods are efficient in recognition application [3]. In addition, CNNs are able to learn hierarchical features automatically with the goal of classification or segmentation of images. Due to the tremendous progress of CNNs in different areas, we utilize transfer learning of a CNN-based model to automatically differentiate various types of flowers. Transfer learning-based approaches advance the performance of our network on a small dataset. Our proposed model is based on fine-tuning of a pre-trained deep learning model, called DenseNet121 [4]. Recently, we have witnessed the outstanding performance of DenseNet on image classification. This network is more accurate and easier to train, since each layer is directly connected to every other layer [5]. Therefore, this model is an interesting case to deal with the problem of flower classification. Our pre-trained model is trained and tested on the 102-oxford flowers dataset [1] and achieves higher accuracy of 98.6% in comparison with other models. The framework we employ is Pytorch. Pytorch is primarily developed by Facebook's AI Research (FAIR) lab and is a deep learning research framework that provides optimum flexibility and speed.

The organization of our paper is as follows. Section 2 reviews some relevant works of flower classification. Then, in Section 3, we introduce the process of our classification model. In Section 4, we provide experimental results of our model. Finally, in Section 5, the conclusion will be drawn.

II. RELATED WORKS

Up to now, various algorithms have been proposed to counter the problem of flower classification. Several

traditional methods have been proposed which use handcrafted features to perform image classification. Guru et al. [6] proposed a textural feature-based method to categorize flower images. They analyzed the efficiency of their method on 35 types of flowers. They could get a maximum result of 79% with their algorithm. In [7], color and texture features are used on 18 classes of flowers. They employed gray-level co-occurrence matrix (GLCM) to calculate textural features and a Multilayer Perceptron (MLP) for classifying and also learning the features. In [8], a novel algorithm is proposed based on the histogram and Local Binary Pattern features (LBP) features for image classification. They could get a higher accuracy of 91% for an image size of 8×8 . Lodh and Patekh [9] proposed an approach to recognize flowers using combined colors and GIST features. In recent years, deep neural networks have attracted a lot of attention in image processing and computer vision areas to solve the problem of traditional machine learning methods. Most of these methods employ CNNs to classify images semantically. For instance, Xia et al. [10] proposed a method based on the transfer learning approaches. They used a pre-trained model of Inception v3 and could get the accuracy of 94% on the oxford 102-flower dataset. In [11], two pre-trained networks of AlexNet and LeNet were tested on flower datasets collected from Flickr, Google, and Kaggle. The modified network of LeNet could achieve a higher accuracy of 57.73% for image classification. Mete, and Ensari [12] investigated the performance of flower classification using four classifiers of Support Vector Machine (SVM), K-Nearest Neighbor (KNN), Random Forest (RF), and MLP. They realized the SVM and MLP have the highest accuracies. Generally, the transfer learning approach plays a key role in using the learned parameters from one model to another one in order to solve new and complex problems. So, we are able to retrain a predefined model to classify our arbitrary dataset. Hence, in this study the pre-trained model of DenseNet121 was employed to classify 102 types of flower images [1].

III. PROPOSED WORK

In this section, we aim to use transfer learning approach for the process of flower classification. Our construction process is divided into four fundamental components including,

- Image pre-processing
- Algorithm training
- Verification process
- Testing process

The schematic of our proposed model is shown in Fig.1.

A. Image Pre-processing

This stage is performed before feeding our images to the network, which can lead to better results from our method. As our images are different in size and width, we resized all of them. After adjusting and normalizing the images, our model can be trained on them using their label.

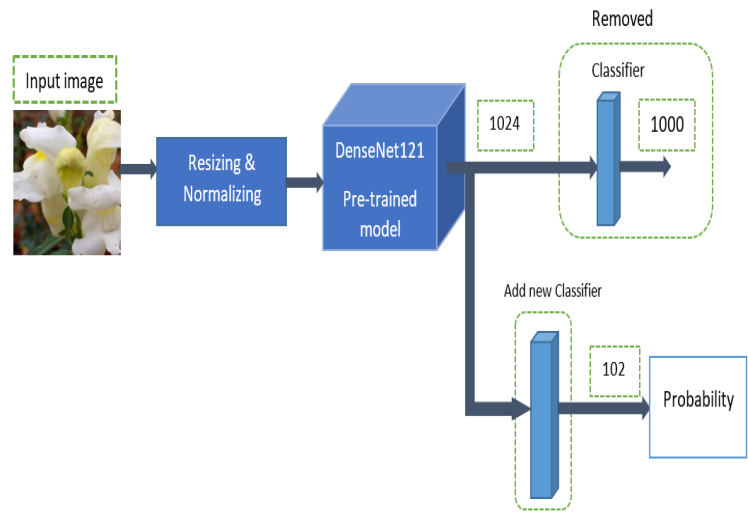


Fig.1: Block diagram of the proposed work

Then, we feed these images to our fine-tuned model.

B. DenseNet121 Model

In this stage, we focus on the transfer learning process by fine-tuning our pre-trained model. Transfer learning means we do not train the models from scratch because it is time-consuming and requires a considerable amount of data for training. Thus, we just transfer features using employing a pre-trained model for our task, which was trained before on the ImageNet dataset, which is a dataset consisting of millions of mages in 1000 classes. So, in the fine-tuning procedure, we replace the classification function to compute the probability of our 102 classes. Pytorch framework helped us to retrain DenseNet's final layer for our 102 classes using fine-tuning. We kept the parameters of the previous layers fixed and removed the classifier layer of our model, and then retrained it again.

The pre-trained model we employed for our method is DeseNet121. DenseNet121 is a convolutional neural network with 24 layers where each layer is connected to all other layers that are deeper in the network. In other words, each block is an iterative concatenation of previous layers. The output of each layer is calculated as in (1)

$$x_l = H_l([x_{l-1}, x_{l-2}, \dots, x_0]) \quad (1)$$

Where x_l is the output of the l^{th} layer, H_l is a non-linear transformation applied to the output of the previous layers of $[x_{l-1}, x_{l-2}, \dots, x_0]$. DenseNet has fewer parameters than an equivalent traditional CNN, as there is no need to learn redundant feature maps. On the other hand, DenseNet layers are very narrow (e.g., 12 filters), and they just add a small set of new feature-maps. Also, each layer of DenseNet has direct access to the gradients from the loss function and the original input image; therefore, there are not any problems related to the flow of information and gradients during training procedure. The structure of DenseNet model is shown in Fig.2.

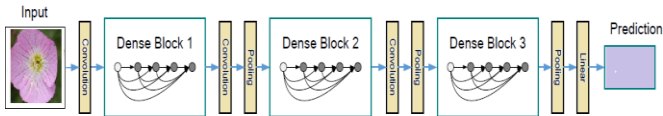


Fig.2: Architecture of proposed model [4]

IV. EXPERIMENTAL RESULTS

In our experiment, the pre-trained model of DenseNet121 of Pytorch, Oxford 102-flower dataset [1] and the hardware of 1xTesla K80 GPU of Google Colab were employed for our study.

A. DataSet

In this paper, in order to train and test the fine-tuned model, our dataset was collected from Oxford 102-flower [1], which is used in many types of researches. This dataset includes flowers that are common in the United Kingdom and on the other hand consists of 40 to 258 images in each category. All flower images are from the real-world with the format of JPEG. The images of the dataset have large scale, pose and light variations. This is a competitive dataset that is used in many research studies.

B. Implementation

Some samples of our dataset for training and testing of our algorithm are shown in Fig.3. We use the transfer learning technique of a pre-trained model and fine-tune it for flower image classification. In this way, we have a faster and cost-effective train without requiring high computational power and time-consuming process. The proposed algorithm was implemented on 1xTesla K80 GPU using the Pytorch framework. At first, we rescaled all of our 8,228 images into constant size of 224×224 to be suitable for training of our model and then normalize them. Also, we could label images in dataset using a Jason file. Next, we split them into three sets of train (6,583), validation (823), and test datasets (822) with batch-size of 64. In addition, our optimization algorithm for computing loss function was Stochastic Gradient Descent (SGD) with a learning rate of 0.001 and our activation function in each layer was ReLU. Under these settings, we should keep weights and parameters of previous layers fixed and only remove the last classifier layer and initialize a new one. Then, we can start retraining the new last layer by feeding the flower dataset. The result of this process is the changing of output nodes from 1000 to 102.



Fig.3: Samples of flowers species from the Oxford 102-Flower [1]

We measure the performance of our model classification by the Cross-Entropy Loss function to adjust weights and calculate the difference between the output and the label of our image dataset for 50 epochs and batch size of 64.

V. RESULTS

Our experiment was conducted using the Google Colab environment. The accuracy of our model during training and testing was evaluated for 50 epochs with batch size of 64. Fig.4 and Fig.5 illustrate the results of the accuracy and loss of our algorithm on Oxford-102 dataset during training phase [1]. The blue graph shows the training set and the red graph shows the validation set.

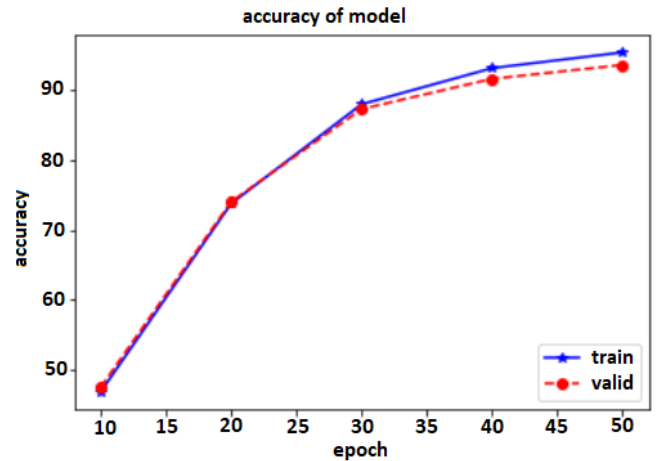


Fig.4: The accuracy of the proposed method for Oxford-102 dataset

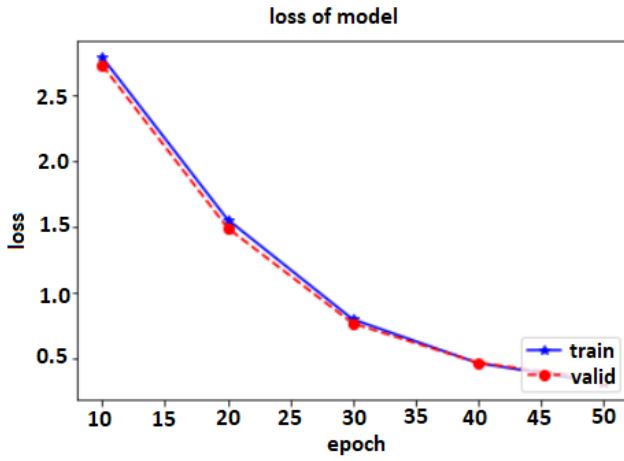


Fig.5: The loss of proposed method for Oxford-102 dataset

We retrained our network on the dataset in 50 epochs. Fig.6 shows the test result of our algorithm.

Fig.6 (a) shows a sample test image for testing of our method. Among 8,228 images for the training and testing of our model, 6,583 images are used for the training, 823 images for the validation and 822 images for the testing. Fig.6 (b) depicts the classes with the highest probability among the top 5 classes for the test image. As shown in this figure, our proposed algorithm can successfully recognize the correct class of the test image. In our experiments, our model could show excellent accuracy with growing number of epochs across the competitive flower dataset. It should be noted that, due to utilizing a pre-trained model of DenseNet121, our method could achieve superior results with much fewer parameters and less computations. Table I shows the parameters and results for our proposed method.

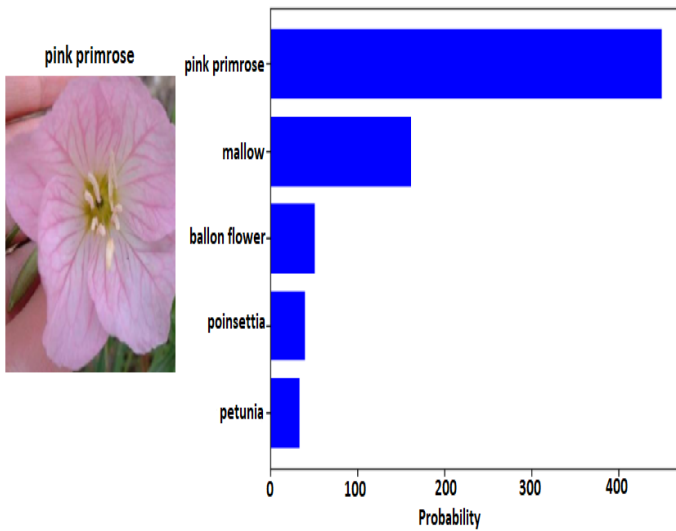


Fig. 6: Test result of our method

TABLE I. PARAMETERS AND RESULTS OF OUR PROPOSED METHOD FOR DATASET OF OXFORD-102

| Method | DenseNet121 |
|-----------|-------------|
| Epoch | 50 |
| Data size | 224 |
| Accuracy | 98.6% |
| Precision | 98.3% |
| Recall | 98% |

In order to evaluate the performance of our method, three metrics of Accuracy, Precision, and Recall are considered. They are based on True Positive (TP), True Negative (TN), False Positive (FP) and False Negative (FN), as can be seen in (2), (3), and (4)

$$Accuracy = \frac{TP+TN}{TP+TN+FP+FN} \quad (2)$$

$$Precision = \frac{TP}{TP+FP} \quad (3)$$

$$Recall = \frac{TP}{TP+FN} \quad (4)$$

Fig.7 shows the efficiency of our method for flower classification in comparison with the results of other algorithms on the same dataset.

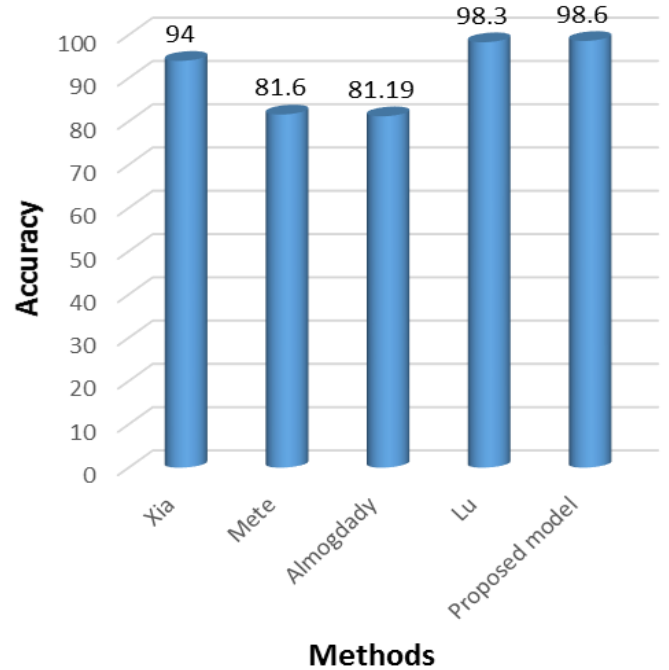


Fig.7: The performance of the proposed method compared to other algorithms on Oxford-102, Xia and Xu used Inception v3 [10], Mete and Ensari used SVM [12], Almgodady et al. used ANN [13], Lu et al. used neural architecture transfer [14]

The results of Fig.7 reveal that our method has a good efficiency in classification of flower images in comparison with other approaches. Our proposed fine-tuned method is robust as it reaches higher accuracy concerning the other employed methods.

CONCLUSION

In this paper, based on the transfer learning approach and fine-tuning of a pre-trained model of DenseNet121, we trained a flower classifier on the Oxford-102 flower dataset [1]. The results of classification showed that the proposed method could achieve higher accuracy than other methods. As future work, we are planning to develop a more accurate and effective deep network for flower image classification. Besides, we can extend the proposed method to be useful in other classification applications.

REFERENCES

- [1] M.E.Nilsback, and A. Zisserman, "Automated flower classification over a large number of classes", IEEE, 2008 [Sixth Indian Conference on Computer Vision, Graphics & Image Processing]
- [2] R.Shaparia, N. Patel, and Z.Shah, "Flower classification using texture and color features", 2: p. 113-118, 2017
- [3] K.Bae, J.Park,J.Lee,Y.Lee, and C.Lim, "Flower Classification with Modified Multimodal Convolutional Neural Networks", p. 113455, 2020.
- [4] G.Huang, Z.Liu, and L.Van Der Maaten, "Densely connected convolutional networks", IEEE, 2017 [Proceedings of the IEEE conference on computer vision and pattern recognition, 2017].
- [5] S.Jégou, M.Drozdzal, D.Vazquez, A.Romero, and Y.Bengio, "The one hundred layers tiramisu: Fully convolutional densenets for semantic segmentation", IEEE, 2017 [Proceedings of the IEEE conference on computer vision and pattern recognition workshops, 2017]
- [6] D.Guru, Y.Kumar, and S.Manjunath, "Textural features in flower classification", 54(3-4): p. 1030-1036, 2011.
- [7] H.Mohd-Ekhsan, J.Hamid, R.Ramle, and M.H.Ismail, "Classification of Flower Images Based on Colour and Texture Features Using Neural Network", 2010 International Conference on Intelligent Network and Computing (ICINC 2010), 2014.
- [8] M.Islam, M.Yousuf, and M.M. Billah, "Automatic plant detection using HOG and LBP features with SVM", 33(1): p. 26-38, 2019.
- [9] A.Lodh and R. Parekh., "Flower recognition system based on color and GIST features", IEEE, 2017 [Devices for Integrated Circuit (DevIC)]
- [10] X.Xia, C. Xu, and B. Nan., "Inception-v3 for flower classification", IEEE, 2017 [2nd International Conference on Image, Vision and Computing (ICIVC)].
- [11] K.Mitrović and D. Milošević., "Flower Classification with Convolutional Neural Networks", IEEE, 2019 [23rd International Conference on System Theory, Control and Computing (ICSTCC)].
- [12] BR.Mete and T. Ensari., "Flower Classification with Deep CNN and Machine Learning Algorithms",IEEE, 2019 [3rd International Symposium on Multidisciplinary Studies and Innovative Technologies (ISMSIT)].
- [13] H.Almogdady, S.Manaseer, and H.Hiary, "A Flower Recognition System Based On Image Processing And Neural Networks", 7(11), 2018
- [14] Z.Lu,G.Sreekumar, E.Goodman, W.Banzhaf, K.Deb, V.Boddeti, "Neural Architecture Transfer", arXiv preprint arXiv:2005.05859, 2020.