FLOWER RECOGNITION AND SEGMENTATION: LEVERAGING CONVOLUTIONAL NEURAL NETWORKS FOR BOTANICAL CLASSIFICATION

Muhammad Abubakar

University of Nottingham
Email: *hcyma5@nottingham.ac.uk

Abstract-This paper explores the application of Convolutional Neural Networks (CNNs) for flower classification and semantic segmentation. We leverage the Oxford dataset and apply image pre-processing, data augmentation, and transfer learning due to limited data. Our classification model achieves a training accuracy of 86% and validation accuracy of 82%. The semantic segmentation model exhibits a global accuracy of about 93.5% and Mean IoU of 84.2%. The results highlight the potential of CNNs for effective flower classification and segmentation, opening avenues for applications in botany, computer vision, and precision agriculture.

***Index Terms*— Flower classification, Flower segmentation, Transfer learning**

I. INTRODUCTION

Despite the existence of approximately 369,000 recognized species of flowering plants globally [1][2], their identification often poses a challenge for the untrained eye. Conventional methods of identification typically involve seeking expert advice [1][3], referencing guidebooks, or conducting online research. With the proliferation of digital technology, a more modern and efficient solution for flower recognition lies in the classification of flower images captured by devices such as digital cameras or mobile phones.

Flower classification holds significant importance due to its wide-ranging applications in numerous sectors including computer vision, precision agriculture, botanical research, plant monitoring, ayurvedic medicine, and Content-Based Image Retrieval (CBIR).

This paper delves into the utilization of Convolutional Neural Networks (CNNs) for flower classification and segmentation. Known for their prowess in image processing and pattern recognition, CNNs offer an innovative approach to identify various types of flowers based on their unique physical attributes.

An integral part of this study involves the segmentation of flowers, a process that entails distinguishing the flower from its background. This technique not only enhances the precision of flower classification but also fosters a deeper understanding of flower structures. Such knowledge is potentially beneficial for fields like botany, ecology, and computer vision [4][5].

The classification component of this research focuses on seventeen distinct flower categories, using the Oxford dataset for analysis and incorporating segmentation techniques. Thus, the study underscores the potential of CNNs in revolutionizing the domain of flower recognition and classification.

II. METHODS

A. Classification

In this section, the proposed methodology is discussed. The data pre-processing, image augmentation method and the CNN model is discussed along with our proposed model network.

Import Dataset

This operation is responsible to read the dataset images from the local hard drive into datastore for image datatype in MATLAB platform (i.e., *ImageDatastore* object). *ImageDatastore* can manage a collection of image files with categorization where each class name is derived from the folder holding its corresponding images.

Pre-processing

In this section, we elaborate on the methodology employed for flower classification, including data preprocessing, image augmentation, and the Convolutional Neural Network (CNN) model. The import dataset operation is crucial in retrieving the dataset images from the local hard drive and transforming them into an ImageDatastore object on the MATLAB platform, which conveniently manages a collection of image files, each categorized based on the folder containing the respective images.

The preprocessing phase involves two primary steps. Initially, we resize all the images in the Oxford dataset to a uniform size of 256 x 256 pixels, a requirement for our model.

Subsequently, we partition the dataset into training, validation, and test sets, with a 70-10-20 split. This ensures a balanced representation of all flower classes in each set, thereby avoiding data imbalance.

To enhance the dataset and mitigate overfitting, we utilize data augmentation, specifically on the training data. This strategy not only increases data volume but also promotes the extraction of robust and generalizable patterns. We employ techniques such as rotation, translation, and reflection. An augmented image datastore supplies the training images, randomly altering the training data for each epoch, thereby reducing overfitting by increasing data diversity and preventing pixel memorization.

We further enhance the classification process through dataset shuffling. This random rearrangement of dataset images promotes an unbiased dataset distribution and mitigates model bias towards any particular class.[7]

Model architecture

The proposed model is a 28-layer CNN with five convolutional layers, each employing the 'relu' function and different filter configurations. The introduction of a dropout layer aids in preventing overfitting by periodically deactivating neurons during training, thereby fostering better model generalization.

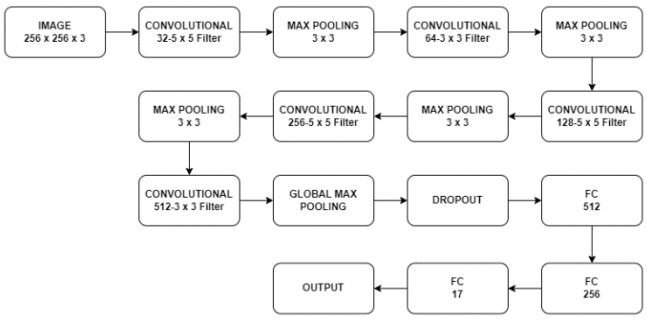

Fig.1. Proposed CNN architecture design

Adding a dropout layer helps prevent overfitting by randomly deactivating a portion of the neurons during each training step, promoting better generalization and robustness of the model.

Using the ReLU (Rectified Linear Unit) activation function for image classification provides better gradient propagation, computational efficiency, and reduced vanishing gradient issues, enhancing the model's ability to learn and extract meaningful feature.

Sequentially using five convolutional layers in a deep neural network allows for hierarchical feature extraction at multiple levels of abstraction, enabling the model to learn increasingly complex representations and capture intricate patterns in the data.

Using different filter sizes in convolutional layers allows the model to capture and learn features at multiple spatial scales, enabling it to extract both fine-grained and larger-scale patterns, enhancing the model's ability to understand complex structures in the data.

TABLE 1. Classification Training options

Training option	Value
Optimizer	Adam
Max Epoch	30
Mini batch size	32
Initial learning rate	1e-4
Shuffle	Every-epoch
Execution environment	gpu

The training options, as summarized in Table 1, utilize the Adam Optimizer for loss minimization, 30 epochs for model training, and a small learning rate for steady convergence towards the optimal solution. The shuffling of data at every epoch enhances model robustness and promotes efficient optimization. Finally, we leverage a GPU for computational processing to expedite model training and facilitate quicker experimentation and iteration.

B. Segmentation

Given the limited size of our dataset, we opted to employ data augmentation techniques, following a similar approach as in our classification model. At the very least, the model necessitates training and testing data. In light of the scarcity of data, we chose an 80-20 train-test split. Each time this split occurs, data is selected randomly to ensure diverse training and testing sets.

To mitigate the constraints of a smaller dataset, we utilized transfer learning. This approach leverages pre-existing models, which have been trained on related tasks, to enhance performance on the task at hand by transferring learned knowledge and representations.

ResNet-18 was selected for its notable performance, efficient architecture, the availability of pre-trained models, and extensive community support. The architecture of the model, depicted in Figure X, originally has an input layer of 224 by 224. Our objective was to adjust this model for an input size of 256 by 256. To accomplish this, modifications were made to the input layer to accept 256 by 256 images. The output layer was also adapted to differentiate between two classes: the flower and the background.

SegNet employs overlay techniques to visually represent the results of its image segmentation tasks by superimposing predicted segmentation masks onto the original images.

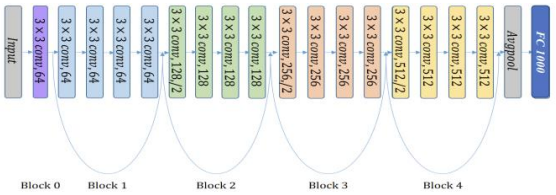

Fig.2. ResNet-18 architecture[7]

TABLE 2. Segmentation Training options

Training option	value
Training/Testing Model	Transfer Learning via ResNet-18
Optimizer	Adam
Max Epoch	30
Mini batch size	32
Initial learning rate	1e-4
Shuffle	Every-epoch
Execution environment	gpu

III. EVALUATION

A. Classification

Figure 3 presents a graphical representation of the training and validation accuracy for our model. The validation accuracy, a crucial metric, gauges the model's performance on data it has not encountered during training, facilitating hyperparameter tuning in the process. Our proposed model exhibits a promising training accuracy of **86%**. When applied to unseen data, it demonstrates an accuracy of **82%**. These figures suggest minimal overfitting as the training accuracy is not nearing 100%. However, a slight overfitting is observed towards the end of the training phase as the validation accuracy plateaus, indicating a halt in the model's learning process.

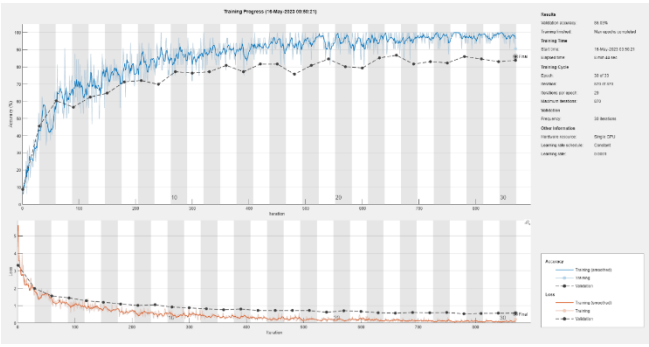

Fig.3. Training and Validation information

The confusion matrix shown in figure 4 is another important tool, offers a comprehensive overview of the predicted and actual classes across the dataset. It enables us to assess the model's performance via metrics such as true positives, true negatives, false positives, and false negatives. Furthermore, it provides a detailed breakdown of the model's performance for each class, enhancing our understanding of its strengths and weaknesses.[6]

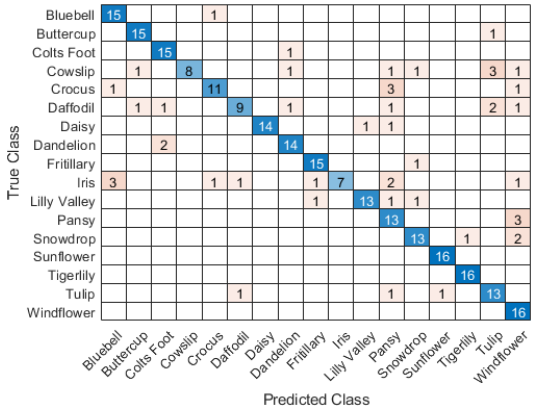

Fig.4. Multi class confusion matrix

TABLE 3. Precision, Recall and F1 for classes

Class	Precision	Recall	F1
Bluebell	0.78947	0.9375	0.85714
Buttercup	0.88235	0.9375	0.90909
Colts Foot	0.83333	0.9375	0.88235
Cowslip	1	0.5	0.66667
Crocus	0.84615	0.6875	0.75862
Daffodil	0.81818	0.5625	0.66667
Daisy	1	0.875	0.93333
Dandelion	0.82353	0.875	0.84848
Fritillary	0.88235	0.9375	0.90909
Iris	1	0.4375	0.6087
Lilly Valley	0.92857	0.8125	0.86667
Pansy	0.56522	0.8125	0.66667
Snowdrop	0.8125	0.8125	0.8125
Sunflower	0.94118	1	0.9697
Tigerlily	0.94118	1	0.9697
Tulip	0.68421	0.8125	0.74286
Windflower	0.64	1	0.78049

In assessing classification tasks, recall and precision are essential metrics as they shed light on distinct facets of a model's performance. High recall signifies the model's competence in recognizing positive instances, whereas high precision implies the reliability of the model's positive predictions. These metrics facilitate striking a balance between different types of errors. The F1 score, a measure

that amalgamates precision and recall, provides a comprehensive evaluation of a model's performance in binary classification tasks. It is computed as the harmonic mean of precision and recall, thus equally weighing both metrics. The calculation formula for the F1 score is:

F1 score = 2 * (precision * recall) / (precision + recall)

The F1 score spans from 0 to 1, with higher values signifying superior performance. It delivers a balanced evaluation, taking into account both the model's ability to identify positive instances (recall) and the correctness of these positive predictions (precision).

Results

Results showcased in Figure 5 illustrate the model's ability to classify flowers. The model demonstrates satisfactory accuracy and confidence in its predictions. However, there exists potential for further enhancement in accuracy, achievable through providing the model with an expanded.

Fig.5. Classification model's prediction

B. Segmentation

TABLE 4. Segmentation evaluation table

Evaluation technique	Value
GlobalAccuracy	0.93521
MeanAccuracy	0.95818
MeanIoU	0.8416
WeightedIoU	0.88556
MeanBFScore	0.39939

The performance metrics indicate that our model performs reasonably well in semantic segmentation tasks. With a global accuracy of around 93.5%, the model accurately classifies approximately 93.5% of all pixels. The mean accuracy, which measures performance specific to each class, is approximately 95.8%, indicating that, on average, the model accurately classifies 95.8% of pixels for each class. The Mean Intersection over Union (IoU), a measure of overlap between predicted and actual segments, stands at around 84.2%. This suggests an average overlap of 84.2% for each class, which is commendable. The Weighted IoU, which takes class imbalance into account, is approximately 88.6%, indicating a satisfactory overlap between predicted and actual segments. However, the Mean Boundary F1 Score, at about 0.4, suggests that the model could improve in its ability to accurately predict object boundaries. [8]

Results

As depicted in Figure 6, the model successfully segments flowers, despite some instances of over-segmentation in the first image. This could potentially be mitigated by providing more data.

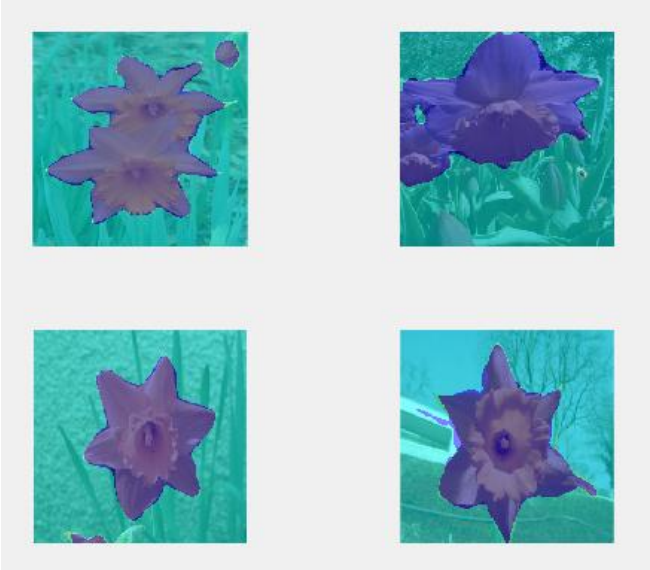

Fig.6. Segmentation model's prediction

IV. CONCLUSION

In conclusion, while both models perform admirably under the given constraints, there is room for improvement, particularly in terms of data quantity. Despite these challenges, the achieved accuracy levels by both models support their practical use in relevant tasks.

V. REFERENCES

[1] Madirakshi Das, R. Manmatha, and Edward M. Rireman, “Indexing Flower” using domain knowledge, University of Massachusetts,Intelligent Information Retrieval-1999.

[2] Yuanyuan liu,Fan Tang,Dengven Zhou,Yiping Meng,Weiming Dong “Flower Classification via Convolutional Neural Network”, IEEE International Conference on Functional Structural plant growth modeling simulation, visualization and applications 2016.

[3] Seeland M, Rzanny M, Alaqraa N, Wäldchen J, Mäder P, “Plant species classification using flower images—A comparative study of local feature representations”. doi:10.1371/journal.pone.0170629

[4] Hazem Hiary, Heba Saadeh, Maha Saadeh, Mohammad Yaqub,”Flower classification using deep convolutional neural networks” IET, April 2018.

[5] S.K. Datta “Present Status of Research on Floriculture in India” LS International Journal of Life Sciences · August 2019 DOI: 10.5958/2319-1198.2019.00006. - Volume 8, Number 2, May-August, 2019, pp. 71-93.

[6] Q. Abu Al-Haija, C.D. McCurry, S. Zein-Sabatto, “Intelligent SelfReliant Cyber-Attacks Detection and Classification System for IoT Communication Using Deep Convolutional Neural Network”, 12th International Network Conference 2020 (INC2020), Springer Lecture Notes in Networks and Systems, Sep. 2020.[6]

[7] Q. A. Al-Haija, M. A. Smadi and S. Zein-Sabatto, "Multi-Class Weather Classification Using ResNet-18 CNN for Autonomous IoT and CPS Applications," 2020 International Conference on Computational Science and Computational Intelligence (CSCI), Las Vegas, NV, USA, 2020, pp. 1586-1591, doi: 10.1109/CSCI51800.2020.00293.

[8] T. Luo, T. Cai, M. Zhang, S. Chen, D. He, and L.Wang, “Defective Convolutional Networks”, arXiv:1911.08432v2 [cs.CV] 6 Apr 2020.