17-CATEGORY SEGMENTATION & CLASSIFICATION OF FLOWER IMAGES AND FEATURE EXPLANATION

Raunak Agrawal

Class of 2026, Columbia University

ABSTRACT

This project aims to solve the problem of how humans are rather inadequate at identifying what species of flowers they might come across anywhere, unless of course they are flower enthusiasts. The model pipeline solves this problem through not just mere classification of the flowers, but rather steps to improve classification accuracy, along with an insight into the classifier's decision-making through the use of feature explanation. The model pipeline consists of a segmentation model, the classification model, and a feature explanation model, which will be expanded upon more later. After experimenting with overall results, it is evident to see that segmenting the flower images provided a better accuracy in classification.

1. INTRODUCTION

As mentioned above, this project aims to solve the problem of recognizing flower species for non-experts. Overall, the work of classifying flower images presents significant challenges due to the high variability between flower species as well as similarities, not to mention images containing multiple types of flowers in the background or alongside each other. Compounding this variability alongside with differences in lighting, pose, and scale, it is evident that robust approaches are needed to accurate results. Primarily, segmentation, which isolates the primary object of interest (flowers) from the background, and classification, which assigns the label to the isolated flower image.

The project builds upon the works of Nilsback and Zisserman's work in flower classification[1] and segmentation[2]. Their pioneering work, alongside their accumulation of the Oxford-17 category flower dataset, demonstrated the efficacy of combining features such as color, texture, and shape to develop robust classifiers for visually similar categories[1]. Their work on segmentation, which used probabilistic models and graph-based optimization to delineate foregrounds, inspired my desire to use segmentation to understand whether it improves classification accuracy.

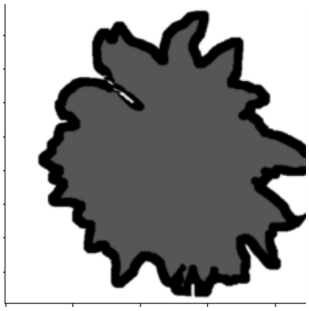
We will now look towards the next few things as part of the paper: Data Curation, Methodology in Model Pipeline, and Results.

2. DATA CURATION

The dataset I used is the Oxford-17 Category Flowers Dataset, curated by Nilsback and Zisserman for their work[1][2]. It consists of 17 different types of flowers, of which 80 images exist for each class. This totals at 1360 flower images to be used in training the classification model. The dataset containing these original images was curated into datasets using PyTorch Datasets, including pre-processing the images to resize them and normalize them.

`flower_train, flower_test, flower_val`

Another dataset created was the segmented flowers dataset. There were 849 images in the Oxford-17 Category dataset that contained annotated ground-truth masks, courtesy of Nilsback and Zisserman. This dataset contained the 849 ground-truth mask files along with the corresponding original-flower files from the dataset, as this is what is required to train the U-net segmentation model. Here, the transformations were applied to both the inputs and the target labels as the U-net model requires both images, unlike the classification model that requires the input images and the class labels.



`seg_train, seg_test, seg_val`

The last dataset was curated post-segmentation model processing, and this dataset contains all of the same images that is contained in the flower_ datasets, except the images are now all the flowers isolated from the background, where the background now became a uniform black. This dataset also contained 1360 images, with the same splits as the flower_ datasets.

`iso_flower_train, iso_flower_test, iso_flower_val`

See the code pdf file attached for the data curation work.

3. METHODOLOGY

3.1. Model Pipeline

3.1.1. Segmentation

The segmentation model employs the U-Net architecture, created by Ronneberger et al.[3] It employs a ResNet50 encoder pre-trained on ImageNet for robust feature extraction, and is then fine-tuned on the segmentation dataset I curated. The U-Net architecture is effective in object delineation tasks due to its symmetric design, which combines high-level semantic features with low-level spatial details. U-net was utilized by employing the segmentations_models.pytorch library by Pavel Iakubovskii.[4] By fine-tuning the model on annotated segmentation masks provided for the 13 flower categories in Oxford-17 Category Flower Dataset, the model received a pixel accuracy of 99% in the segmentation task, displaying its robustness to create masks for flower images. At Fig1 is a figure of an original image, along with its predicted mask.

3.1.2. Classification

Following the segmentation, the original dataset was processed through the model to create the isolated_flowers dataset, and from there followed the classification task. The isolated flowers were used to train a ResNet50 model by Kaiming He et al, which is known for its depth and efficient feature extraction[5]. The model was loaded through PyTorch, and fine-tuning involved freezing the initial layers to retain the initial weights, while letting the model's adjusted

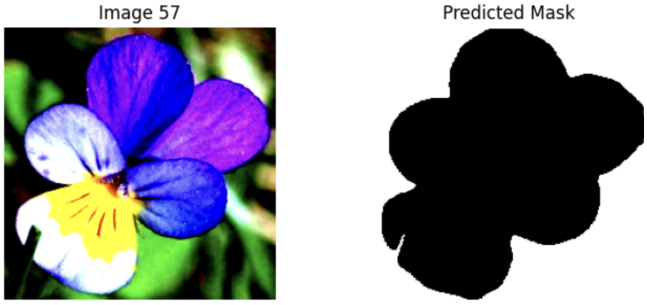


Fig. 1. Real image vs. Predicted Mask

fully connected layer learn flower-specific features. The loss function used was Cross Entropy Loss, which is useful for a multi-class classification problem. In classification, I compared a classification model trained on the isolated flower segments and the original dataset to compare the success of segmentation. As will be expanded upon in results, the model trained on the isolated flowers dataset outperformed the model trained on the original dataset in both testing tasks by a marginal percentage. Overall, the model trained on the isolated flowers dataset received an accuracy of 96% and 89% when tested on the isolated flowers test set and the original flowers test set. See Figure 2 for classification outputs.

3.1.3. GradCAM

GradCAM (Gradient-weighted Class Activation Mapping) is a powerful tool for enhancing the interpretability of the classification model by visualizing the regions of an image that contribute most to a model's decision-making process through a heat map. It was created by Selvaraju et al,[6] and I used Gildenblat's implementation in my project[7]. In this project, GradCAM was integrated into the pipeline to validate the classification model's predictions and to highlight key features influencing its output. This aligns with the broader goal of making the model more interpretable, particularly in applications like flower classification, where visual nuances are evident. The GradCAM uses the gradient of the target class flowing into the final convolutional layer to produce a course localization map. This map indicates which regions of the input image have the most significant impact on the model's predictions. Specifically, GradCAM:

1. Computes the gradients of the output score for the predicted class with respect to the feature maps of the last convolutional layer.

2. Aggregates these gradients across spatial locations to assign importance weights to each feature map channel.

3. Produces a heatmap by applying these weights to the feature maps and performing a weighted sum.

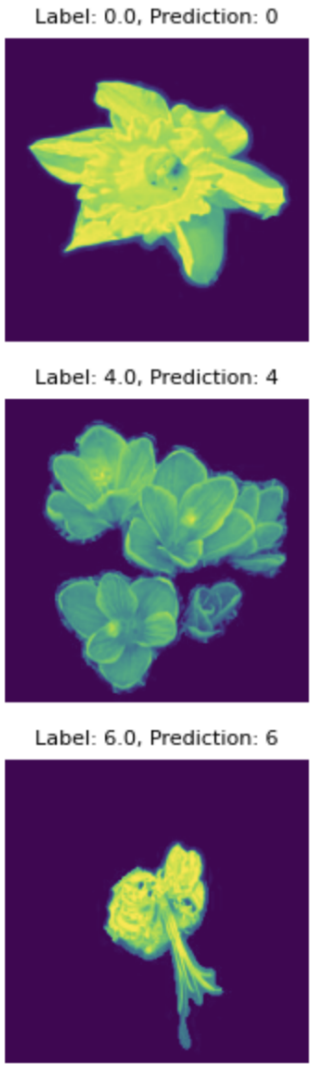


Fig. 2. Classification Outputs

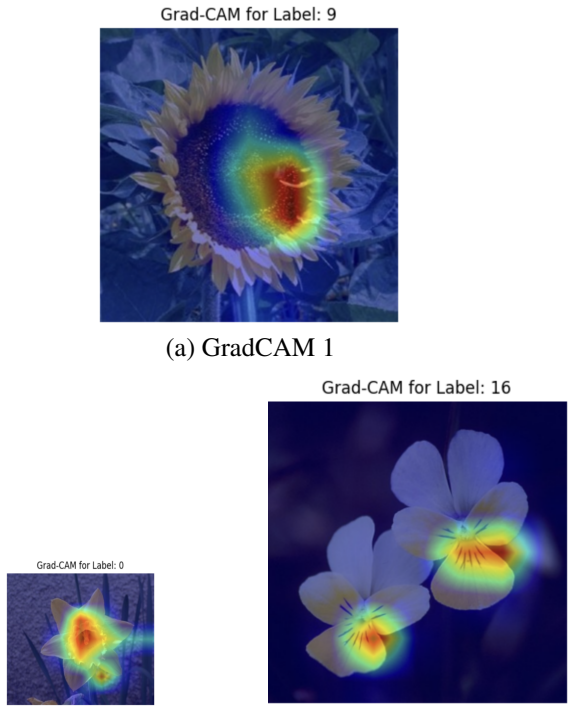
GradCAM was great here to understand why the classification model made the decisions that it did, and to verify that the classifier was making reasonable decisions. See Figure 3 for GradCAM visualizations.

4. RESULTS

In this section, we will discuss the results of the 3 different stages in the models pipeline.

4.1. Segmentation

The segmentation model was tested on the pixel-accuracy metric, achieving a 99% accuracy, effectively isolating flowers from diverse backgrounds. This confirms the utility of segmentation in preprocessing for flower datasets like the Oxford Flower Dataset.



(a) GradCAM 1

(b) GradCAM 2

(c) GradCAM 3 4

Fig. 3. GradCAM Visualizations

4.2. Classification

The ResNet50 model achieved accuracies of 96% and 89% when trained on the isolated flower dataset and tested on the isolated flowers test set and original flowers test set respectively. When trained on the original dataset and tested on the same test sets, the model received lower accuracies of 82% and 79%. The segmentation of the flowers improved the accuracy by a difference of +7% for the original flowers test set, and +17% for the segmented flowers test set, suggesting overall better performance but also that the segmented-trained classification model is very suited towards the segmented flowers and has lower performance by -7% for original flower images.

4.3. GradCAM

There were no metrics to specifically test GradCAM, however the visualization images help highlight the utility of using GradCAM to understand the features that outlined the classifier's decision making process.

5. DISCUSSION

The combined segmentation-classification pipeline underscores the importance of preprocessing in achieving state-of-the-art results in flower classification. The use of U-Net for segmentation aligns with traditional approaches that emphasize the utility of pixel-level annotations in complex visual tasks. ResNet50's hierarchical feature learning was instrumental in distinguishing visually similar categories, confirming findings from prior work.

The integration of Grad-CAM provided a means to validate model predictions, offering a bridge between model output and human interpretability.

6. CONCLUSION

Overall, this project highlights the usage of computer vision techniques such as segmentation to improve deep-learning tasks. Using such techniques with models can provide great results in challenging classification tasks such as flower classification. By leveraging the foundational works[1][2], and the usage of many different techniques and implementations[5][3], I was able to achieve relatively great results. Future work could explore expanding the dataset to include more challenging categories and experimenting with multimodal approaches that combine visual and textual descriptions of flowers for more robust classification.

Again, for any code implementation, please see the Jupyter notebook pdf file attached with my work.

7. REFERENCES

[1] Maria-Elena Nilsback and Andrew Zisserman, "A visual vocabulary for flower classification," in *IEEE Conference on Computer Vision and Pattern Recognition*, 2006, vol. 2, pp. 1447–1454.

[2] Maria-Elena Nilsback and Andrew Zisserman, "Delving into the whorl of flower segmentation," in *British Machine Vision Conference*, 2007, vol. 1, pp. 570–579.

[3] Olaf Ronneberger, Philipp Fischer, and Thomas Brox, "U-net: Convolutional networks for biomedical image segmentation," 2015.

[4] Pavel Iakubovskii, "Segmentation models pytorch," https://github.com/qubvel/segmentation$_m odels.pytorch$, 2019.

[5] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun, "Deep residual learning for image recognition," 2015.

[6] Ramprasaath R. Selvaraju, Michael Cogswell, Abhishek Das, Ramakrishna Vedantam, Devi Parikh, and Dhruv Batra, "Grad-cam: Visual explanations from deep networks via gradient-based localization," in *2017 IEEE International Conference on Computer Vision (ICCV)*, 2017, pp. 618–626.

[7] Jacob Gildenblat and contributors, "Pytorch library for cam methods," https://github.com/jacobgil/pytorch-grad-cam, 2021.