# Invasive Species Detection Through Image Classification

Robert Relyea Ali Batayneh

Spring 2019

## 1 Introduction

The main goal of this final project was to detect an invasive flower species in various images of forests and foliage. The specific invasive species to be detected was the Hydrangea, which is invasive to the forests of Brazil. In addition to detecting the invasive flower species, five other species of flowers will be detectable by combining the Hydrangea dataset with an additional flowers dataset. Deep Learning techniques will be used to train and test a model which will have seven output classes (Hydrangea, Chamomile, Tulip, Rose, Sunflower, Dandelion, and none.) The datasets are provided by past Kaggle challenges [2, 1]. The combined datasets will be split into training and testing subsets.

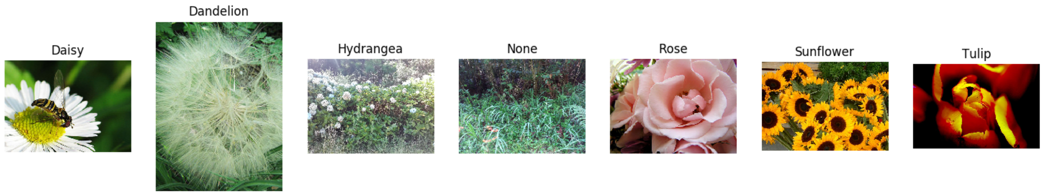


Figure 1: Example images of all flower classes.

## 2 Related Work

There have been several previous works for classifying the presence of different species in images. Toth et al. [8] presented several methods for classifying different plant species in images using an SVM with SIFT features, and Inception CNN model, and a combination of both. Klein et al. [5] presented a pipeline for data collection and classification for general biodiversity applications which utilized CNNs and other machine learning techniques. Both of these prior works demonstrate successful invasive species detection through deep learning based image classification.

## 3 Methodology

For this experiment, invasive and non-invasive species detection was posed as an image classification problem. Classes were defined as whether one of six species of flowers were present or not. Data augmentation techniques such as random crops and variations in color content were applied to the combined training dataset to improve validation results.

We will utilize the ResNet [3] CNN architecture for feature extraction and classification of training images. The ResNet architecture was able to achieve improved classification results with vastly deeper networks than models such as VGG16 [7] and AlexNet [6]. This improved performance was due to the introduction of residual features from previous convolutional layers which were added to the features of subsequent convolutional layers. This operation was designed to reduce the affect of vanishing gradients as the loss is backwards propagated throughout the extremely deep network. In addition to the ResNet architecture, the DenseNet[4] architecture was also investigated for this classification task. DenseNet builds upon the residual concepts posed by ResNet by introducing dense skip connections. These connections jump back to more convolutional layers than the residual features in ResNet. The skip connection features in DenseNet are also concatenated to the intermediate convolution layer features rather than summed like the ResNet implementation.

For this exercise, the network models will be trained using transfer learning as well as from scratch. A Pytorch implementation of both models will be trained on a workstation running Ubuntu 16.04 with an Nvidia Titan V GPU. The final fully-connected classification layers will be redefined to have seven class outputs to match the problem.

Training and testing dataset splits were produced from the combined Kaggle challenge datasets[2, 1] and were utilized for our model evaluation.

# 4 Experiments

The Hydrangea dataset utilized for the project was provided in a Kaggle challenge [2] which includes RGB images of foliage from the Brazilian forest. Each training image is labeled as invasive or not invasive. The dataset containing five other species of flowers was also provided by a Kaggle challenge [1] and contains 4242 images with no pre-existing train/test split. This data was split and combined with the training and testing datasets from the Hydrangea dataset. In total the resulting dataset represented 7 classes where a train/test split of 80/20 was created for each class. The resulting seven classes to be predicted are Daisy, Dandelion, Rose, Sunflower, Tulip, Hydrangea, and None. Details about the per-class training and testing image counts are provided in Table 1.

Table 1: Dataset class sample composition details.

| **Class** | **Total Images** | **Train Images** | **Test Images** |
|---|---|---|---|
| Daisy | 769 | 615 | 154 |
| Dandelion | 1055 | 844 | 211 |
| Hydrangea | 1448 | 1158 | 290 |
| None | 847 | 678 | 169 |
| Rose | 784 | 627 | 157 |
| Sunflower | 734 | 587 | 147 |
| Tulip | 984 | 787 | 197 |
| **Total** | **6621** | **5296** | **1325** |

The models used for experimentation include the ResNet18, ResNet152, and the DenseNet161 CNN architectures. Data augmentation in the form of random crops and horizontal flips were introduced for some of the training runs to evaluate any improvements in classification performance. Each model was trained for 60 epochs with a varying learning rate. The initial learning rate was set to 0.001 with a scheduled decay every 7 epochs by a factor of 0.1. Further training details can be seen in Tables 2, 3, and 4.

# 5 Results

The scope of the project involved taking the image datasets provided by Kaggle [2, 1] and classifying the images into seven classes (Hydrangea, Chamomile, Tulip, Rose, Sunflower, Dandelion, and None.) The Hydrangea Kaggle competition shows that participants were able to achieve above 99 percent accuracy in their classifier submissions which provides support that the classification problem of interest is achievable. Our best performing DenseNet161 model with data augmentation was able to achieve a validation accuracy of 89.32 percent. Using the same model and the same data

Table 2: Training details for experiments performed on the Hydrangea dataset. The best validation accuracy across all models was produced by the DenseNet161 architecture with data augmentation applied.

| Architecture | Training | Batch Size | Epochs | Data Augmentation | Best Val Accuracy |
|---|---|---|---|---|---|
| ResNet18 | Transfer Learning | 256 | 60 | Random Resize Crop, ColorJitter | 0.856209 |
| ResNet152 | Transfer Learning | 256 | 60 | Random Resize Crop, ColorJitter | 0.875817 |
| DenseNet161 | Transfer Learning | 256 | 60 | Random Resize Crop, ColorJitter | **0.893246** |

Table 3: Training details for experiments performed on the flowers dataset. The best validation accuracy across all models was produced by the DenseNet161 architecture with data augmentation applied.

| Architecture | Training | Batch Size | Epochs | Data Augmentation | Best Val Accuracy |
|---|---|---|---|---|---|
| ResNet18 | Transfer Learning | 256 | 60 | Random Resize Crop, ColorJitter | 0.871824 |
| ResNet152 | Transfer Learning | 256 | 60 | Random Resize Crop, ColorJitter | 0.89261 |
| DenseNet161 | Transfer Learning | 256 | 60 | Random Resize Crop, ColorJitter | **0.903002** |

Table 4: Training details for experiments performed on the combined dataset. The best validation accuracy across all models was produced by the DenseNet161 architecture with data augmentation applied.

| Architecture | Training | Batch Size | Epochs | Data Augmentation | Best Val Accuracy |
|---|---|---|---|---|---|
| ResNet18 | From Scratch | 256 | 60 | Random Resize Crop, ColorJitter | 0.876226 |
| ResNet18 | Transfer Learning | 256 | 60 | None | 0.870189 |
| ResNet18 | Transfer Learning | 256 | 60 | Random Resize Crop, ColorJitter | 0.879245 |
| ResNet152 | Transfer Learning | 256 | 60 | None | 0.886038 |
| ResNet152 | Transfer Learning | 256 | 60 | Random Resize Crop, ColorJitter | 0.892075 |
| DenseNet161 | Transfer Learning | 256 | 60 | None | 0.898868 |
| DenseNet161 | Transfer Learning | 256 | 60 | Random Resize Crop, ColorJitter | **0.908679** |

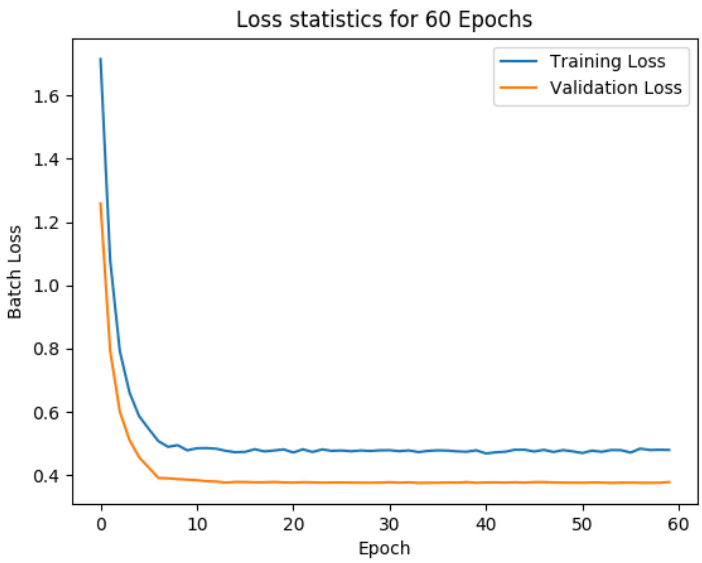


Figure 2: Training and validation loss of DenseNet161 with data augmentation and transfer learning during training for 60 epochs on the combined dataset.

augmentation scheme for the flowers dataset produced the best accuracies for those experiments as well. The flowers dataset experiment results can be seen in Table 3.

Several performance results were recorded from the conducted experiments which may be viewed

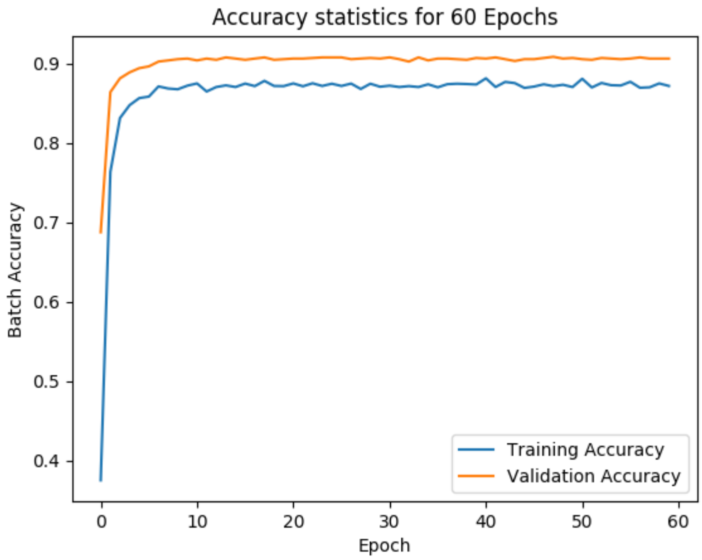


Figure 3: Training and validation accuracy of DenseNet161 with data augmentation and transfer learning during training for 60 epochs on the combined dataset.

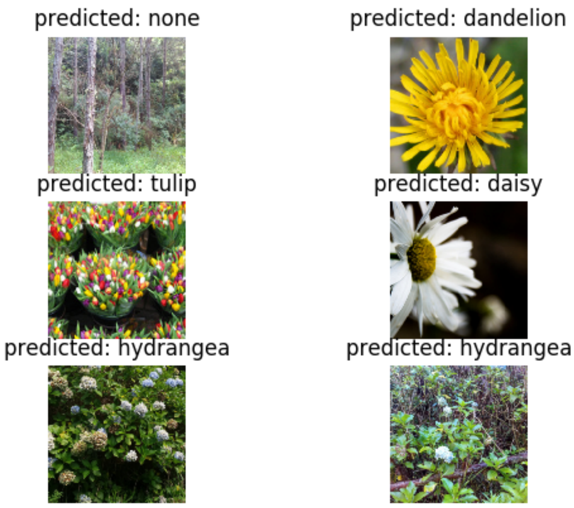


Figure 4: Classification inference examples for multiple classes from the combined dataset.

in Table 4. The overall results for the combined Hydrangea and Flowers data set shows that the best validation accuracy achieved was 90.49 percent using the DenseNet161 architecture with transfer learning and data augmentation applied.

## 6 Discussion

The overall results from the project concluded an accuracy of 90.87 percent for the combined Flowers and Hydrangea data set. This demonstrates the ability for a model to be trained on a pool of multiple datasets while maintaining high classification accuracies.

The Hydrangea dataset results from this project are sub par to those of the Kaggle competition where the best validation accuracy was upwards of 99 percent. The reduced performance compared to the Kaggle entrants may be due to a reduced size of training dataset. The Kaggle competition provides the test dataset samples without class labels. In order to assess our combined dataset results offline, we split the Kaggle-provided training dataset into train and validation sets. The reduced training sample pool may have resulted in a less generalizable model. Results for additional models can also be seen in Table 2.

The reduced Hydrangea dataset accuracy may also be due to the specific architecture along with fine tuning that allows the model to train with very high validation accuracy. The experiments conducted in the project produced some interesting observations, the models with Data augmentation generally performed better than those without Data Augmentation included. One more interesting observation is that as deeper architectures are used, the accuracy of the model increases. This falls in line with the improved classification accuracies observed with deeper residual and dense networks.

Many entrants to the Kaggle flower dataset competition attained validation accuracies in the high 80s to low 90s. This is in line with our best DenseNet model which attained a 90.3 percent validation accuracy when trained for 60 epochs with data augmentation. Results for additional models can also be seen in Table 3.

Additional findings include an improved classification accuracy when training a ResNet18 model with pre-trained CNN weights and data augmentation compared to a ResNet18 model trained from scratch. Table 4 shows the resulting validation accuracy for the pre-trained model was 87.9 percent while the network trained from scratch achieved an accuracy of 87.6 percent. While the percentages are not very different, the ease of fine-tuning a pre-trained model makes it a more compelling option than training a model fully from scratch.

# 7 Teamwork

| **Section** | **Robert's Contribution** | **Ali's Contribution** |
|---|---|---|
| Introduction | 80% | 20% |
| Related Work | 20% | 80% |
| Methodology | 50% | 50% |
| Experiments | 40% | 60% |
| Results | 50% | 50% |
| Discussion | 50% | 50% |
| ResNet | 0% | 100% |
| DenseNet | 100% | 0% |
| Datasets | 60% | 40% |

# References


[1] Flowers recognition. https://www.kaggle.com/alxmamaev/flowers-recognition. Accessed: 2019-03-26.

[2] Invasive species monitoring. https://www.kaggle.com/c/invasive-species-monitoring/data. Accessed: 2019-03-20.

[3] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 770–778, 2016.

[4] Gao Huang, Zhuang Liu, and Kilian Q. Weinberger. Densely connected convolutional networks. *CoRR*, abs/1608.06993, 2016.

[5] David J Klein, Matthew W McKown, and Bernie R Tershy. Deep learning for large scale biodiversity monitoring. In *Bloomberg Data for Good Exchange Conference*, 2015.

[6] Alex Krizhevsky, Ilya Sutskever, and Geoffrey E Hinton. Imagenet classification with deep convolutional neural networks. In *Advances in neural information processing systems*, pages 1097–1105, 2012.

[7] Karen Simonyan and Andrew Zisserman. Very deep convolutional networks for large-scale image recognition. *arXiv preprint arXiv:1409.1556*, 2014.

[8] Bálint Pál Tóth, Márton József Tóth, Dávid Papp, and Gábor Szücs. Deep learning and svm classification for plant recognition in content-based large scale image retrieval. In *CLEF (Working Notes)*, pages 569–578, 2016.