Classifying Invasive Plant Species in Indiana Using Image Data

Owen Kleinmaier[1], **Connor Malone**[2]

[1]Indiana University, *owklein@iu.edu*

[2]Indiana University, *conmalon@iu.edu*

Abstract

Invasive plant species pose significant threats to biodiversity, agriculture, and local ecosystems in Indiana. Traditional identification methods are labor-intensive and require expertise. This project explores the use of transfer learning via MobileNetV2 for automating invasive species classification based on image data. We processed a dataset of 8,000 plant images, fine-tuned a pretrained MobileNetV2 model, and evaluated its performance on 25 invasive species. The model achieved a validation accuracy of 40%, demonstrating the feasibility of applying deep learning to ecological challenges. Future work will focus on improving accuracy by expanding the data set and exploring additional modalities.

Index Terms

Transfer Learning, Image Classification, MobileNetV2, Invasive Plant Species, Deep Learning.

I. Introduction

Invasive plant species, such as Japanese honeysuckle and garlic mustard, are among the most significant threats to Indiana's ecosystems. These plants displace native vegetation, alter habitats, and disrupt ecological balance. Traditional methods for identifying invasive species require fieldwork and expertise, making large-scale monitoring a daunting task. For example, the state has an early detection and rapid response system that scientists and citizens can submit findings to. This software helps map these species. The problem is that very few people in the state can correctly identify invasive plant species and even fewer know how to correctly prevent them from spreading and causing harm.

Machine learning and computer vision provide promising alternatives to automate species identification. Deep learning models, particularly those that use transfer learning, have shown great success in image classification tasks. This project aims to leverage these advances by training a model to classify invasive plant species in Indiana. Specifically, we employ MobileNetV2, a lightweight convolutional neural network (CNN), to identify 25 invasive species from a labeled dataset of user-submitted plant images in Indiana.

This work has the potential to aid conservation efforts by allowing a faster and more accurate identification of invasive plants. By automating this process, researchers, conservationists, and policy makers can allocate resources more effectively to mitigate the ecological impact of these species.

II. Methods

A. Dataset

The data set comprises 8,000 labeled images of 25 invasive plant species, sourced from iNaturalist. This is a website/mobile application that has millions of users around the world. It allows anybody to submit pictures of plants and animals to the software, identifies the species, and provides information about it. This app is very helpful, but we used it's user-submitted images to help train our own model! Using a simple data scraper, we filtered the images for verified Indiana-based users. This provided us with the images and their verified, labeled species.

B. Data Preprocessing

This was a very important step in setting up our classifier. It took a while to make sure the data was both formatted correctly for use in the MobileNetV2 model, but also so that the model could correctly classify it based on labels. Since the image data was given in URL form, we had to find a way to translate that into readable Tensor shapes for the model. We tried a couple different methods including locally downloading the images, but in the end decided to use a Python module called Pillow to read the URL's, request them and then convert them.

We then used Pandas to load the data and normalize it by encoding and mapping over the labels. The CSV file had 10 columns but only the image url and species name were used, so we had to filter the rest of the information (user, description, etc) out. Each image was resized to 224x224 pixels and normalized for use in MobileNetV2. The data was split into training sets (60%), validation sets (20%) and tests (20%). We limited the dataset to 8,000 images due to computational constraints but ensured a balanced representation of all species.

C. Model Architecture

We used MobileNetV2, a pre-trained model on ImageNet, for transfer learning. MobileNetV2 is well-suited for this task due to its lightweight architecture, which balances performance and computational efficiency. Originally, we had planned to use ResNet50, a heavier, more complex version of MobileNetV2. After many different struggles including lack of time and computational power, we decided to switch to a lighter version. MobileNetV2 is often used for mobile applications (hence the name), which works perfectly for our future plans of implementing the model into a user-accessible mobile application.

The pre-trained base was frozen, and a custom classification head was added, consisting of:

- A Global Average Pooling layer to reduce spatial dimensions.
- A Dropout layer (rate = 0.3) to mitigate overfitting.
- A Dense layer with softmax activation to output probabilities for 25 classes.

D. Training Setup

The model was trained for five epochs using the Adam optimizer with a learning rate of 0.001. A batch size of 64 was used, and categorical cross-entropy was the loss function. The training process was monitored using accuracy and loss metrics for both training and validation sets. Early stopping was originally enabled, but after a few errors we disabled it to allow full exploration of model performance over the defined epochs. The training was done on Kaggle using their built in GPU P100.

E. Evaluation Metrics

The model's performance was assessed using:

- Validation accuracy and loss.
- Individual testing on sample images.

III. Results

A. Training and Validation Performance

Figure 1 shows the training and validation accuracy over five epochs. The validation accuracy plateaued at around 40%, while training accuracy continued to improve, indicating potential overfitting. Figure 2 illustrates a steady decrease in both training and validation loss, suggesting effective optimization.

B. Testing on Sample Images

To get a better idea of how the model worked in a more practical way, we created a function that took in a sample URL image. Using the model it then classifies the image and checks in a list of invasive species whether the plant is invasive or not. This is super simple, but simulates exactly what would happen when we deploy the model to a mobile application for users. It performed decently, correctly classifying many different images of invasive species including Garlic Mustard and Japanese Honeysuckle, but fell short on less common species.

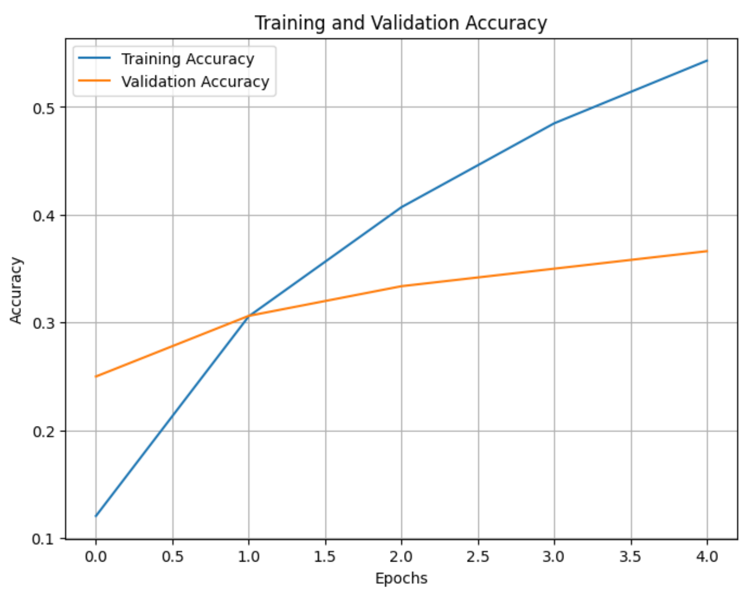

Fig. 1. Training and Validation Accuracy over 5 epochs. Validation accuracy reached 40% while training accuracy continued to improve.

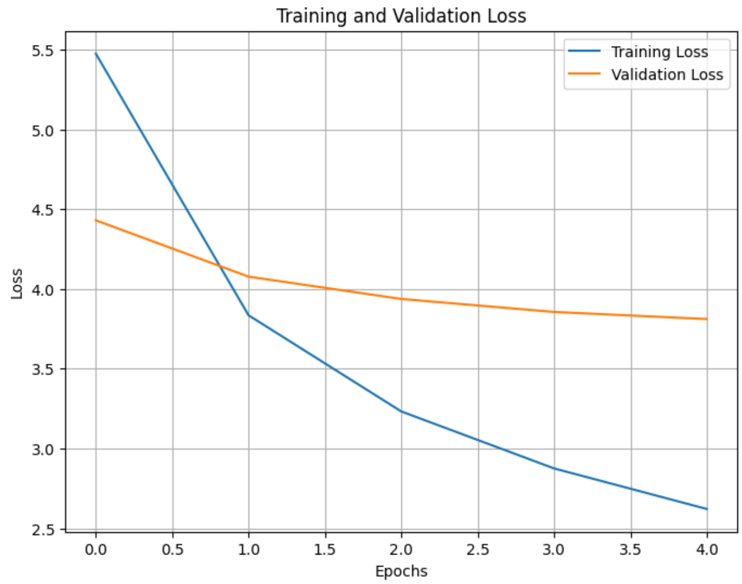

Fig. 2. Training and Validation Loss over 5 epochs. Both metrics decreased steadily, showing effective model training.

IV. Discussion

A. Strengths and Weaknesses

The model successfully classified many invasive species, demonstrating the potential of transfer learning for ecological applications. However, the limited size of the data set and the similarity between some species hindered performance. We also were lacking powerful computational resources which hindered the model's performance due to restrictions on time and GPU's.

Key strengths include:

- Efficient training using MobileNetV2's lightweight architecture.
- Promising results with relatively few epochs.

Limitations include:

- Misclassifications for visually similar species.
- The need for a larger and more diverse dataset to improve generalization.

B. Future Work

Future directions include:

- Collecting additional labeled images, especially for underrepresented species.
- Exploring ensemble models to improve classification accuracy.
- Incorporating metadata, such as geographic location or seasonal information, to enhance predictions.
- Fine-tuning hyperparameters to balance training efficiency and accuracy.

V. Author Contributions

Owen Kleinmaier and Connor Malone contributed equally to the project. Owen focused on data gathering, preprocessing and training, while Connor reasearched models, created the model and helped evaluate it. Generative AI tools were used to assist with idea generating, finding resources, and document formatting.

VI. References

1) TensorFlow Documentation: https://www.tensorflow.org.
2) MobileNetV2 Paper: Sandler et al., ”MobileNetV2: Inverted Residuals and Linear Bottlenecks.”
3) Indiana DNR Invasive Species: https://www.in.gov/dnr/rules-and-regulations/invasive-species/terrestrial-invasive-species-plants/.
4) Keras API Reference: https://keras.io/api/.
5) Transfer Learning Survey: Pan et al., ”A Survey on Transfer Learning.”
6) He et al., ”Deep Residual Learning for Image Recognition,” in Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 2016.
7) Howard et al., ”Searching for MobileNetV3,” in Proceedings of the IEEE International Conference on Computer Vision (ICCV), 2019.