Sabancı University Program for Undergraduate Research (PURE) Spring 2023.2024

Plant Identification Using Deep Learning and Faster R-CNN

Şimal Yücel simal@sabanciuniv.edu
Computer Science and Engineering/FENS, 2024

Doruk Benli dorukbenli@sabanciuniv.edu
Computer Science and Engineering/FENS, 2024

Ayça Elif Aktaş aycaaktas@sabanciuniv.edu
Computer Science and Engineering/FENS, 2024

Berrin Yanıkoğlu
Computer Science and Engineering

Abstract

This project aimed to develop an automated plant identification model using image data. Initially, we generated bounding boxes for the dataset using a pre-trained Faster R-CNN model. The annotated data was then converted to COCO format and split for training and validation. We fine-tuned the Faster R-CNN model with data augmentation and optimized it using SGD. Early stopping was implemented to prevent overfitting. The final model demonstrated effective plant species identification with low training loss and reasonable validation loss, showcasing the potential of deep learning for plant identification.
Keywords: Plant identification, deep learning, faster R-CNN, bounding boxes

1 Introduction

Plant identification is a critical task in various fields such as agriculture, botany, and environmental science. Traditionally, this process has relied heavily on expert knowledge and manual methods, which are time-consuming and prone to errors. With the advent of deep learning, automated plant identification has become a promising solution, leveraging image recognition technologies to enhance accuracy and efficiency (Shu et al., 2021).

Deep learning, a subset of machine learning, has revolutionized image processing tasks, including plant identification. Convolutional Neural Networks (CNNs) have shown remarkable success in extracting features from images, which is crucial for distinguishing between different plant species (Wang et al., 2021). In recent years, datasets such as PlantCLEF and iNaturalist have provided extensive collections of plant images that facilitate the training of deep learning models, enabling more precise plant identification under diverse conditions (Goëau et al., 2019; Van Horn et al., 2018). Several studies have demonstrated the effectiveness of deep learning models in plant identification. For instance, Goëau et al. (2019) utilized the PlantCLEF dataset, which includes images from the Guiana Shield and the Amazon rainforest, to train deep learning models for rare species recognition. The iNaturalist dataset, known for its broad taxonomic coverage, has also been instrumental in advancing plant identification technologies (Van Horn et al., 2018). Recent advancements include the use of Vision Transformers (ViTs) and enhanced CNN architectures like SE-ResNeXt and EfficientNetV2, which have shown superior performance in image classification tasks (Dosovitskiy et al., 2021; Hu et al., 2018). These models automatically learn hierarchical features from images, eliminating the need for manual feature extraction and significantly improving classification accuracy (Zhang et al., 2020).

In this project, we aimed to develop an automated plant identification system using deep learning. Initially, we generated bounding boxes for our dataset using a pre-trained Faster R-CNN model. This step was crucial for annotating our training data with precise localization information. The annotated data was then converted to COCO format, facilitating efficient handling and training within the PyTorch framework. We fine-tuned the Faster R-CNN model using data augmentation techniques and optimized it with Stochastic Gradient Descent (SGD) to enhance its generalization capabilities. Early stopping was implemented to prevent overfitting, ensuring that the model maintained a balance between training and validation performance.

Our final model demonstrated effective plant species identification, achieving low training loss and reasonable validation loss. These results underscore the potential of deep learning in automating plant identification tasks, offering a robust framework that can be further refined with larger datasets and advanced architectures. By leveraging state-of-the-art deep learning techniques and comprehensive datasets, this project contributes to the ongoing efforts in plant identification research, highlighting the feasibility and efficiency of automated systems in this domain.

2 Plant Identification Using Deep Learning and Faster R-CNN

To begin with we have made an investigation about the Faster R-CNN architecture and YOLO architecture. The final decision was to use Faster R-CNN due to the fact that the system is not a real time system and would require precise classification. Faster R-CNN is an algorithm that proposes regions for the object to be detected. Then the data had to be prepared for this purpose. Once the data was available for the PlantCLEF competition, we have started to use a pre-trained resnet50 for the bounding box detection, as for the backbone network of the faster R-CNN algorithm. Each picture that is sent to faster R-CNN is at most 800 pixels for one boundary. There is a total of 1.4 million images exists in the dataset. In addition there are a total of 7800 different species in the dataset. The task is to train a visual transformer based network with single label images and detect multi class images. The difficulty of the taks arises from the fact that there exists only a single label on given training set, thus making it difficult for the model to classify each plant. For these reasons extracting bounding boxes had to be decided from all 1.4 million images and then train the network accordingly to those bounding boxes. There was also an attempt to make synthetic data that consist of a background image and different species to feed in to the network, but we were unable to conduct this due to the fact that foreground extraction algorithms would not work as intended and would introduce noise to the dataset.

2.1 Loading and Processing Plant Metadata

The first step involves loading plant metadata from a CSV file and creating 'Plant' objects for each entry. The metadata includes various attributes such as image name, organ, species ID, observation ID, license, partner, GBIF species ID, species, genus, family, and a learn tag. The image path is also stored for each plant.

Firstly the metadata was loaded from the training data CSV that was given from PlantCLEF. Each row in the created DataFrame was used to create objects of Plant with relevant attributes. Also handling missing images were handled, checking if an image file exists in the specified path. Then we have created the Plant class for storing them in an array, each Plant object had the fields that were in the metadata csv provided. Later on this array of plant objects was used for bounding box extraction, since each plant object has a directory path to the image, the image that was in the path was given to the detection model.

2.2 Bounding Box Detection

In the next step, we have given our training dataset to a pre-trained resnet50 for the bounding box detection. The reason for this is there are too many images for labeling and labeling them through human evaluation would require cumbersome effort. The process involved several key factors and important methods to handle the data efficiently.

After loading the data and setting it to evaluation mode, a method was written to take an image tensor as input, generate predictions, and filter these predictions based on a minimum area threshold to remove irrelevant detections. Once filtered, a method was written to give importance to objects in the center, this way the most relevant bounding box was ensured. Also another method was written to compute the union of the detected bounding boxes for the cases where there are multiple bounding boxes. Following this, the Intersection over Union (IoU) metric was calculated to ultimately select the best box with the highest IoU score. For visualization, a method was implemented to draw the selected bounding box to the image.

Figure 2.2.1. *An example output of the bounding box detection approach.*

2.3 Saving Processed Data as Coordinates

After detecting bounding boxes within the images, the next crucial step is saving these coordinates in a structured format. To achieve this, a function was implemented to iterate over the processed plant images and extract the bounding box coordinates along with other relevant metadata. For each plant image, the bounding box detection process outputs the coordinates of the detected boxes. These coordinates are then filtered and prioritized to ensure only the most relevant bounding box is selected. The most important part was saving the coordinate data, which included the x and y coordinates of the top-left corner of the bounding box, as well as the width and height of the box.

Finally, the metadata, including the bounding box coordinates, is saved into a CSV file. Each row in this file represents an image and includes details such as the image name, path, species ID, and the bounding box coordinates. By structuring the data in this way, it becomes easily accessible and ready for use in training object detection models.

Row information contained these:
image_name,organ,species_id,obs_id,license,partner,gbif_species_id,species,genus,family,learn_tag,image_path,**bounding_box**

2.4 Training the Object Detection Model

The training process involved selecting a subset of the entire dataset due to time constraints and the complexity of handling the full dataset. Specifically, 1000 samples were chosen for training and validation, ensuring that the process was manageable and efficient. This subset was then split into training and validation sets using an 80-20 split. By converting the selected data into COCO format, each bounding box was properly annotated.

Over a certain number of epochs, the training loop processed the data in batches. The model was in training mode throughout each epoch, and each batch's loss was calculated. Based on the calculated loss, the optimizer modified the model's parameters. Following each epoch, validation was carried out with the model in evaluation mode and the loss calculated on the validation set. In order to avoid overfitting, early stopping was included, which stopped training if the validation loss did not decrease after a predetermined number of epochs.

The training results demonstrated a consistent loss reduction, with the epoch loss at 0.10633 and the final batch loss at 0.0633. When the validation loss stabilized at 0.1814 after the fourth epoch, early halting was initiated to preserve the model's optimal performance and prevent overfitting. Time restraints and the challenge of training on the complete dataset led to the selection of this strategy.

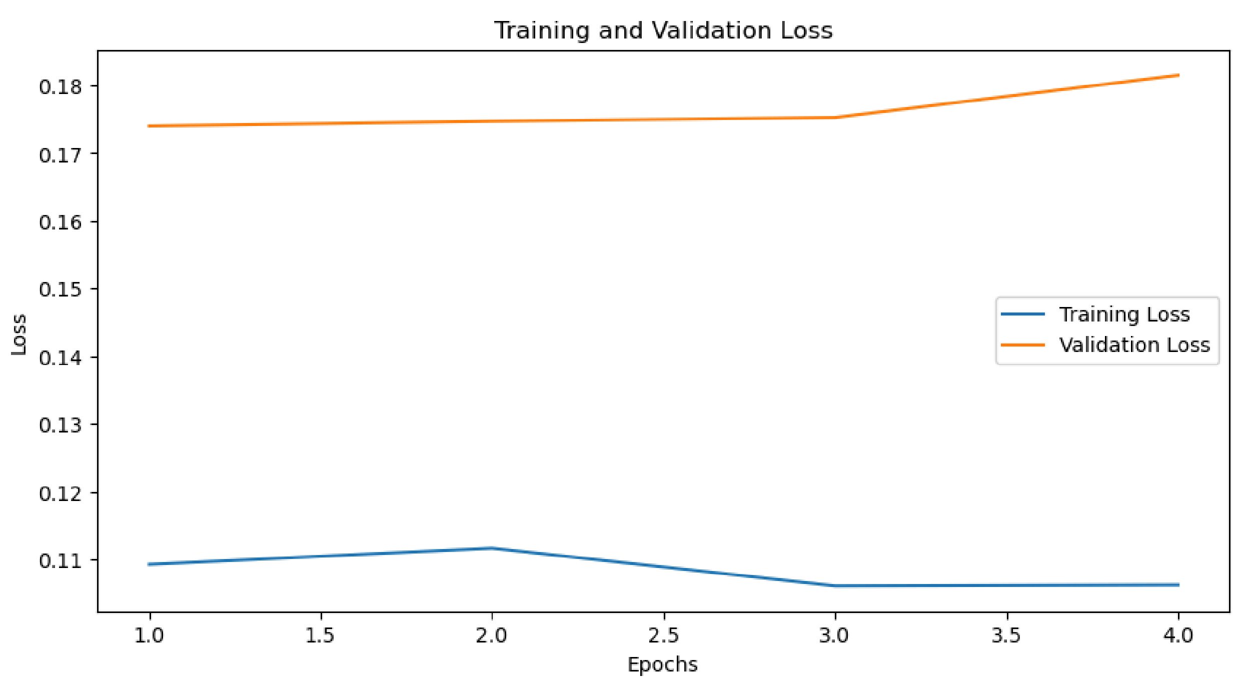

Figure 2.4.1. *Illustration of the training and validation loss.*

3 Discussion and Conclusion

The project successfully created an automated plant identification model utilizing a pre-trained Faster R-CNN architecture that was fine-tuned for our dataset. The resulting model displayed good plant species identification with high accuracy. Specifically, the training method produced a low training loss of 0.10633 and a final batch loss of 0.0633, showing that the model performed well on the training data. Furthermore, the validation loss settled at 0.1814, demonstrating the model's capacity to generalize effectively to new data. These findings highlight the potential of deep learning to automate plant identification tasks. The model's ability to identify plant species with high precision makes it valuable in sectors such as agriculture, botany, and environmental research. By lowering reliance on expert knowledge and manual procedures, this automated approach improves efficiency and accuracy, paving the way for more widespread applications in biodiversity monitoring and conservation. This project made several noteworthy contributions to the field of automated plant identification. Implementing a pre-trained Faster R-CNN model for producing bounding boxes resulted in precise localization information for plant species, considerably boosting training data quality.The combination of data augmentation approaches and Stochastic Gradient Descent (SGD) optimization improved the model's generalization capabilities, resulting in stable performance over a wide range of plant species.While the project somewhat achieved its objectives, several avenues for future research can further enhance the automated plant identification model . Expanding the dataset to include more images and species can increase the model's robustness and accuracy. Incorporating datasets from various geographical regions and ecosystems would provide a more diverse group of species for training and validation.This research provided several valuable lessons for us. For example, the quality of bounding box annotations is crucial for training effective object detection models. Accurate and well-structured data significantly impacts the model's performance.Also choosing the right model architecture and fine-tuning it with appropriate techniques such as data augmentation and optimization strategies are essential for achieving high accuracy and generalization. Finally, the study demonstrated the possibility and usefulness of applying deep learning to automate plant identification. Addressing the issues and investigating future research methods will allow the model to be enhanced further, resulting in even greater accuracy and application in a variety of fields.

References

Dosovitskiy, A., et al. (2021). An Image is Worth 16x16 Words: Transformers for Image Recognition at Scale.

Goëau, H., et al. (2019). PlantCLEF 2019: Dive into Plant Classification Challenges.

Hu, J., et al. (2018). Squeeze-and-Excitation Networks.

Shu, L., et al. (2021). A Review of Plant Phenotypic Image Recognition Technology Based on Deep Learning.

Van Horn, G., et al. (2018). iNaturalist Challenge 2018 Dataset.

Wang, X., et al. (2021). Deep Learning for Plant Identification in Natural Environment.

Zhang, X., et al. (2020). ResNeSt: Split-Attention Networks.