ENEL 645 Final Project: Image Classification of Agricultural Crops

Dillon Pullano
Dept. of Electrical and Computer Engineering
dillon.pullano@ucalgary.ca

Jonathan Lui
Dept. of Electrical and Computer Engineering
jonathan.lui1@ucalgary.ca

Yaad Sra
Dept. of Electrical and Computer Engineering
yaadkaran.sra@ucalgary.ca

Andrew Lee
Dept. of Electrical and Computer Engineering
andrew.lee4@ucalgary.ca

Sami Abdelhalem
Dept. of Electrical and Computer Engineering
sami.abdelhalem@ucalgary.ca

***Abstract*—This project explores the application of deep learning techniques for the classification of crops using a publicly available dataset. Given the growing demand for automated agricultural systems, the accurate classification of crop types from images can significantly enhance crop management and yield prediction. This report details our approach using convolutional neural networks (CNNs) to classify images into multiple crop categories, discusses the model's performance, and highlights the potential implications for agricultural practices and research.**

I. Introduction

The classification of crops from images represents a vital step towards realizing the potential of precision agriculture, enabling targeted interventions, and optimizing resource use. With the advent of deep learning, specifically convolutional neural networks (CNNs), there has been significant progress in image classification tasks. This project leverages a dataset from Kagglethat consists of 30 different types of agricultural crop images that were used to train our CNN models. Our project aimed to accurately classify the images into their respective crop categories, thereby contributing to the body of knowledge in automated agricultural systems. This report outlines our methodology, from data preprocessing and model selection to training and evaluation. It then discusses the broader implications of our findings for the agricultural sector.

II. Related Work

THe application of deep learning in agriculture has been a subject of increasing interest, with studies demonstrating its potential in areas such as disease detection, yield prediction, and crop classification. Convolutional Neural Networks (CNNs), in particular, have shown great promise due to their ability to extract and learn features from images. This project builds upon existing research by applying CNNs to the task of classifying crops from images, a key component in the automation of agricultural monitoring systems. Several notable studies have laid the groundwork for this application, showcasing the effectiveness of deep learning models in achieving high accuracy rates in image-based classification tasks within the agricultural domain.

III. Materials and Methods

A. Kaggle Dataset

The primary dataset for this project was obtained from Kaggle, under the title ”Agricultural Crops Image Classification”. This dataset encompasses a broad spectrum of agricultural crop images, meticulously categorized to represent a diverse range of crop types commonly found across various agricultural practices globally.

Each image in the dataset is provided in a high-resolution format, ensuring that critical visual features necessary for accurate classification are preserved. The images encapsulate various growth stages of the crops, from seedlings to mature plants, and include different parts of the plant such as leaves, stems, and fruits, providing a rich dataset for training the CNN model.

Each image is contained in a folder that is labeled with its corresponding crop type. The dataset contains a diverse set of 30 crop categories, including, but not limited to, wheat, maize, and rice, encompassing a wide range of visual characteristics. When creating training, validation, and testing datasets for all of our models, the splits were done for each crop type individually and then were combined into a single split.

B. Data Preprocessing

Data preprocessing is a critical step in preparing the raw images for input into the CNN models. Two separate preprocessing pipelines were followed to run our models.

The first approach was to use the original size of the dataset with a (80%, 10%, 10% split) for the training, validation, and testing datasets, respectively. This resulted in the number of training samples being 652, validation samples being 79, and testing samples being 98. The reason that there are more testing samples is that when a split could not be applied evenly, the remainder was added to the testing set.

The second approach initially split the dataset with a (60%, 20%, 20% split) for the training, validation, and testing datasets, respectively. After the split was done, each image in the training dataset was augmented with the following torchvision transformations:

- Horizontal flip
- Vertical flip
- Random rotation between 0 and 10 degrees
- Random color jitter between 0 and 0.1
- Gaussian blur (kernel size of 3)
- Random sharpness (factor of 1.5)
- Perspective shift (distortion of 0.3)

This resulted in the final number of training samples being 3880, validation samples being 167, and testing samples being 177.

For both of these pipelines, a final transformation was applied to all final images that match the recommendations when applying transfer learning from the Resnet18 and Resnet50 models that are trained from the ImageNet dataset. This final transformation first resized all images to a uniform size of 256x256 pixels using the bilinear interpolation method. It then performed a center crop to resize all images to a uniform size of 224x224 pixels. Images were then converted to a tensor and were finally normalized in the RGB bands with mean = [0.485, 0.456, 0.406] and standard deviation = [0.229, 0.224, 0.225], as recommended in the Pytorch documentation.

C. Model Architecture

The Convolutional Neural Network (CNN) models developed in this project are based on the architecture of preexisting, proven models for image classification tasks, through leveraging the concept of transfer learning. Specifically, the base of our models utilizes the ResNet-18 and ResNet-50 architectures that are trained using the ImageNet dataset. These models are renowned for their ability to learn from a large number of parameters without overfitting, thanks to residual connections.

For all models that were tested, the respective ResNet weights were first loaded in and frozen. We then applied either a single or sequential final layer to output a probability distribution over the 30 crop categories that were present in our dataset. The sequential layer was used to apply a dropout to prevent overfitting.

A categorical cross-entropy loss function was applied for all tests, suitable for multi-class classification problems. A stochastic gradient descent optimizer with a momentum of 0.9 was also applied for all tests, while the learning rate was adjusted to either 0.001 or 0.0001. Finally, our tests also varied the batch size to 32 or 64. These parameters were modified in an attempt to increase our validation and testing accuracies and to prevent overfitting to our training datasets.

D. Training Process

We decided to train and test our models locally on our computers since this allowed us to run multiple jobs simultaneously while discussing approaches through online meetings. Initially, we created a base Python file together to be used as a template. This file loaded in the dataset, split and preprocessed the data through 2 pipelines into the original and augmented datasets, created 2 separate models to run both datasets, trained both models using ResNet-18 using a single output layer, a learning rate of 0.0001, and a batch size of 64. This file then tested the models, calculated evaluation metrics, and output a confusion matrix to visualize classification results for both datasets.

Early stopping was implemented using the validation loss. If the current epoch had a validation loss that was higher than the lowest current loss 3 times, the code would automatically stop the training loop. The epoch with the lowest validation loss was saved so that it could be reloaded in for the testing phase.

The training process was carried out over 50 epochs, with early stopping implemented to prevent overfitting if the validation loss did not improve for 10 consecutive epochs. Model performance was evaluated based on accuracy and loss metrics for both the training and validation sets, providing insights into the model's learning progression and generalization ability.

E. Evaluation Metrics

To assess the model's performance, we employed accuracy, precision, recall, and F1-score metrics. These metrics were calculated on the test set, which the model had not seen during the training or validation phases, ensuring an unbiased evaluation of its classification capabilities. We also presented a confusion matrix for a visual representation.

IV. Results and Discussion

A. Model Performance

Our Convolutional Neural Network (CNN) model, trained on the Kaggle dataset for crop image classification, demonstrated promising results. The best model results achieved a testing accuracy of 79.59% on the training dataset that had no image augmentations and used an 80%, 10%, 10% split. The model used the ResNet-50 architecture (with dropout applied) that was set up with a learning rate of 0.001 and a batch size of 64. The high testing accuracy indicates a strong ability to generalize from the training data to unseen images.

1) Accuracy and Loss Metrics: This model ran for 46 epochs with a steady decrease in training and validation loss over epochs and an increase in accuracies with a final training accuracy of 89.26% and a final validation accuracy of 69.62%. Such trends are indicative of effective learning without a significant amount of overfitting.

2) Precision, Recall, and F1-Score: Beyond accuracy, the model's precision, recall, and F1-score metrics offer a deeper insight into its classification performance across different crop types. Table I below presents these metrics for each crop category, highlighting areas where the model excels and where improvements are needed.

The table shows that many of the crop categories achieved an accuracy, precision, and recall score of 1. This is representative of our model performing well. There are some classes

TABLE I
PERFORMANCE METRICS BY CROP CATEGORY

Crop Type	Precision	Recall	F1-Score	Support
almond	1.00	1.00	1.00	3
banana	1.00	0.67	0.80	3
cardamom	1.00	0.75	0.86	4
Cherry	1.00	0.67	0.80	3
chilli	1.00	1.00	1.00	3
clove	1.00	1.00	1.00	3
Coffee-plant	1.00	1.00	1.00	3
cotton	0.60	1.00	0.75	3
Cucumber	1.00	0.75	0.86	4
Fox_nut(Makhana)	1.00	1.00	1.00	4
gram	1.00	0.67	0.80	3
jowar	1.00	0.67	0.80	3
jute	1.00	1.00	1.00	3
Lemon	1.00	0.67	0.80	3
maize	1.00	1.00	1.00	4
mustard-oil	1.00	0.33	0.50	3
Olive-tree	1.00	1.00	1.00	3
papaya	1.00	0.75	0.86	4
Pearl_millet(bajra)	0.75	1.00	0.86	3
pineapple	1.00	1.00	1.00	3
rice	0.75	1.00	0.86	3
soyabean	1.00	0.75	0.86	4
sugarcane	0.75	1.00	0.86	3
sunflower	1.00	0.75	0.86	4
Tea	1.00	1.00	1.00	3
Tobacco-plant	0.75	1.00	0.86	3
tomato	1.00	1.00	1.00	3
vigna-radiata(Mung)	1.00	0.33	0.50	3
wheat	1.00	1.00	1.00	3
			accuracy	0.86
			macro avg	0.86
			weighted avg	0.86

however, such as mustard oil and vigna-radiata(Mung) where the model has a lower F1-score due to low precision or recall, indicating potential areas for improvement.

B. Analysis of Misclassifications

In addition to the scores presented above, the confusion matrix for the best performing model can be seen below:

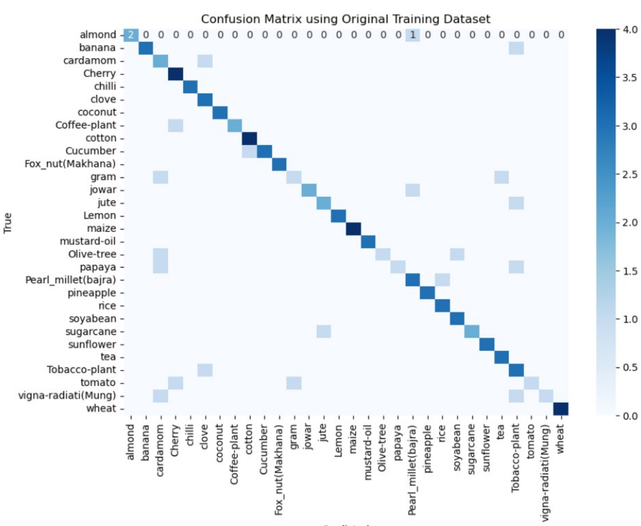

Fig. 1. Confusion Matrix

It can be seen from the confusion matrix that for the most part, crops were classified correctly. However, there are instances where several crops (Gram, Olive Tree, Papaya, and vigna-radiati (Mung) were mislabelled as cardamom. We believe the model would perform better if more unique images were added.

C. Discussion

The results obtained from this project underscore the potential of CNNs in the field of precision agriculture, specifically in automating the task of crop classification. The high accuracy and satisfactory precision, recall, and F1-scores suggest that such models can significantly contribute to agricultural practices, aiding in crop monitoring and management.

However, the analysis also points to challenges in distinguishing between crops with similar physical characteristics. This limitation highlights the importance of further research into more sophisticated neural network architectures or the integration of additional types of data beyond visual imagery, such as spectral data, to enhance classification accuracy.

Additionally, we found that our tests that used the augmented training dataset generally performed worse than the original set. Adjusting the batch size did not seem to change our accuracies by a significant amount. A faster learning rate of 0.001 always performed better than using 0.0001.

D. Implications for Future Work

The findings from this study open several avenues for future research. Firstly, experimenting with more complex CNN architectures or hybrid models that combine CNNs with other types of neural networks might yield improvements in classification performance. Secondly, incorporating data augmentation techniques that introduce a greater variety of transformations could help the model learn more robust features if applied differently. We believe it could be that we added too many augmentations that resulted in our lower accuracy results. Lastly, expanding the dataset to include more diverse images, especially of underrepresented crop types, would likely improve the model's generalization capabilities.

This study demonstrates the feasibility and effectiveness of using convolutional neural networks for the classification of agricultural crop images. While the results are promising, the discussed limitations and potential improvements highlight the ongoing need for research in this area to fully leverage the power of deep learning in agriculture.

V. CONCLUSION

This study embarked on the exploration of applying Convolutional Neural Networks (CNNs) to the classification of agricultural crop images, leveraging a comprehensive dataset from Kaggle. The project's primary aim was to assess the viability and efficiency of different CNNs in distinguishing between various crop types.

A. Main Findings

Our model demonstrated notable success, achieving an overall accuracy of 79.59% on the test dataset, which underscores the potential of deep learning techniques in agricultural applications. The precision, recall, and F1-score metrics further affirmed the model's robustness in classifying multiple crop types, despite the inherent challenges posed by the variability in image quality and crop appearance.

B. Implications for Precision Agriculture

The implications of this research extend beyond the technical achievements. By providing a reliable method for crop classification, this study contributes to the broader goals of precision agriculture: optimizing resource use, improving crop management strategies, and ultimately enhancing yield outcomes. The ability to accurately classify crops using images can facilitate real-time monitoring and decision-making, offering a scalable solution to meet the growing global food demand.

C. Limitations and Future Work

Despite the promising results, this study encountered limitations, particularly in the model's ability to distinguish between crops with similar visual characteristics. This challenge highlights the necessity for further model refinement and the exploration of more complex architectures or hybrid models.

Future research could also explore the integration of additional data modalities, such as spectral imaging or environmental data, to enrich the model's learning context. Expanding the dataset to include various crop types and growth stages could further enhance the model's generalization capabilities.

D. Concluding Remarks

In conclusion, the successful application of CNNs for agricultural crop image classification represents a significant step forward in the utilization of deep learning within the domain of precision agriculture. While challenges remain, the path forward is marked by numerous opportunities for innovation and improvement. Continued research in this field is not only poised to revolutionize agricultural practices but also to contribute to the sustainability of global food systems.

REFERENCES

[1] Y. LeCun, Y. Bengio, and G. Hinton, ”Deep learning,” *Nature*, vol. 521, no. 7553, pp. 436-444, May 2015.

[2] K. He, X. Zhang, S. Ren, and J. Sun, ”Deep residual learning for image recognition,” in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, Las Vegas, NV, USA, pp. 770-778, June 2016.

[3] L. Lu, Y. Li, and S. O. Oyedotun, ”Agricultural Crop Type Classification Using Deep Learning: A Review,” *Computers and Electronics in Agriculture*, vol. 178, pp. 105742, November 2020.

[4] T. N. Sainath, O. Vinyals, A. Senior, and H. Sak, ”Convolutional, long short-term memory, fully connected deep neural networks,” in *2015 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, Brisbane, QLD, Australia, pp. 4580-4584, April 2015.

[5] M. Kampffmeyer, A. B. Salberg, and R. Jenssen, ”Semantic segmentation of small objects and modeling of uncertainty in urban remote sensing images using deep convolutional neural networks,” in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition Workshops*, Las Vegas, NV, USA, pp. 1-9, June 2016.

[6] J. Deng, W. Dong, R. Socher, L.-J. Li, K. Li, and L. Fei-Fei, ”ImageNet: A large-scale hierarchical image database,” in *2009 IEEE Conference on Computer Vision and Pattern Recognition*, Miami, FL, USA, pp. 248-255, June 2009.

[7] A. Krizhevsky, I. Sutskever, and G. E. Hinton, ”ImageNet classification with deep convolutional neural networks,” in *Advances in neural information processing systems*, Lake Tahoe, NV, USA, pp. 1097-1105, December 2012.

[8] M.Azam,”Agricultural Crops Image Classification,” Kaggle, 2021. [Online]. Available: https://www.kaggle.com/datasets/mdwaquarazam/agricultural-crops-image-classification. [Accessed: 06-April-2024].

[9] PyTorch, ”torchvision.models.resnet18,” 2021. [Online]. Available: https://pytorch.org/vision/main/models/generated/torchvision.models.resnet18.html. [Accessed: 06-April-2024].

[10] PyTorch, ”torchvision.models.resnet50,” 2021. [Online]. Available: https://pytorch.org/vision/main/models/generated/torchvision.models.resnet50.html. [Accessed: 06-April-2024].