Strawberry Farm Classification From UAV Data and Machine Learning

Shirish Shekhar Jha 20257

Abstract

The classification of strawberry farms is an important task in the agricultural industry for efficient management and production. In this study, a Convolutional Neural Network (CNN) model and machine learning techniques are employed to classify strawberry farms based on their visual characteristics. The dataset used in this study consists of images of different strawberry farms at different levels of growth captured using an UAV. The images were extracted from an .ecw file and were made in the form of patches of size 512×512. The CNN model is trained on the numpy dataset to classify the strawberry farms into different categories based on their features such as size, shape, and color. The performance of the model is evaluated using different evaluation metrics, and the results show that the CNN model outperforms other machine learning models in terms of accuracy. Also, the feature extraction technique improved accuracy when integrated with the machine learning model. The proposed CNN-based classification model can be used for accurate and efficient classification of strawberry farms.

Keywords— Classification, CNN model, Feature Extraction

I. INTRODUCTION AND BACKGROUND

The problem statement handed over was to classify the strawberry farms based on their growth level. There were three classes of the strawberry farm classification, class 0 where the strawberry farm has just been made ready for cultivation, class 1 comprised of the strawberry farms where the cultivation has just started, and the class 2 comprises of the strawberry farms where the strawberries are ready for harvesting. The task was to classify the strawberry farms into these growth levels by using features like shape, color, size, etc.

Former works have been done using the CNN models to classify agricultural crops and satellite images. The works which gave us an inspiration to apply were "*Precision agriculture classification using convolutional neural networks for paddy growth level".*

II. METHODOLOGY

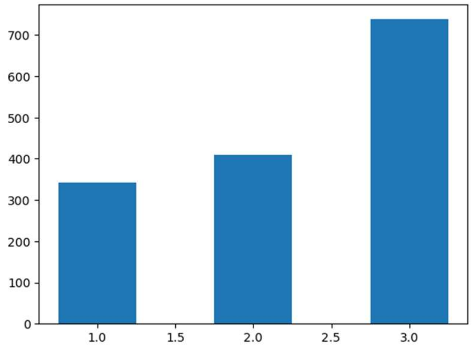
A. Step1

It involved creating dataset from the UAV images.

The UAV data file was first converted into .tiff file format from which later patches of size 512×512 were extracted. These patches included images of forest area, buildings, cars, roads, farms and other miscellaneous objects. Filtering and segregation of strawberry farms images based on their growth level was done manually so as to avoid any digital erros. This resulted in creation of final dataset where we had approx. 9000 images of miscellaneous objects and 1149 cumulative images of the strawberry farms, strawberry farms images were only used to perform the task. Later all the images were stored as an array with entries as images along with their labels. Before saving images, they were resized to a size of 228×*228×3.* Here 228 are length and width of images and 3 is the number of channels since the image was in RGB format.

B. Step 2

This step involved understanding the images and class distributions. It was understood that there was a high-class imbalance as evident by this graph.

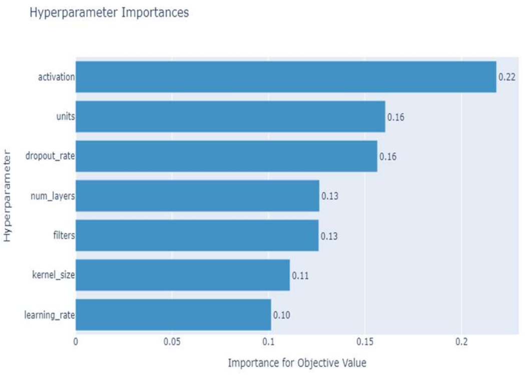


To resolve this error one can, provide extra weightage to the class with lower sample and also one can use data augmentation to increase the number of images of a particular class. For our first CNN model we applied class weightage based on the number of samples.

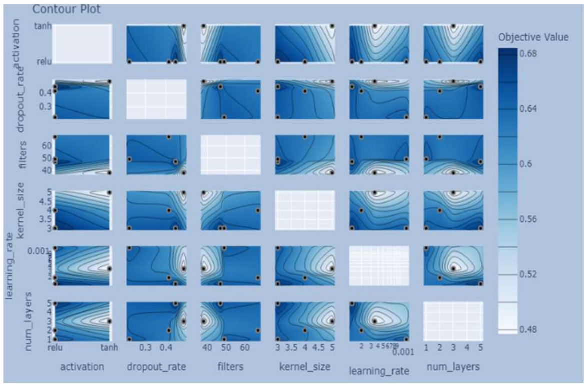
C. Step 3

Model building came as the third step in achieving the goals of the project. A Convolutional Neural Network model was first deployed to understand its working. It was later found that it required hyperparameter tuning which was done using the Optuna library. Optuna is a Python library for hyperparameter optimization that uses the Bayesian optimization technique. Bayesian optimization is a sequential model-based optimization technique that builds a probabilistic objective function model and uses it to guide the search for the best set of hyperparameters. Bayesian optimization aims to find the set of hyperparameters that maximizes the expected improvement over the current best solution. After hyperparameter tuning a CNN model was build based on the hyperparameters to classify the images and it resulted in increased accuracy. More optimizations could be done based on better computing resources and that act as a scope of improvement in the study.

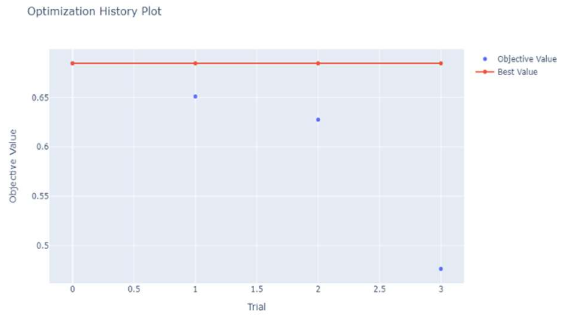
Optuna helped understand the importance of parameters and saved the best params for which we get the best results. The below plot shows the importance of each parameter.



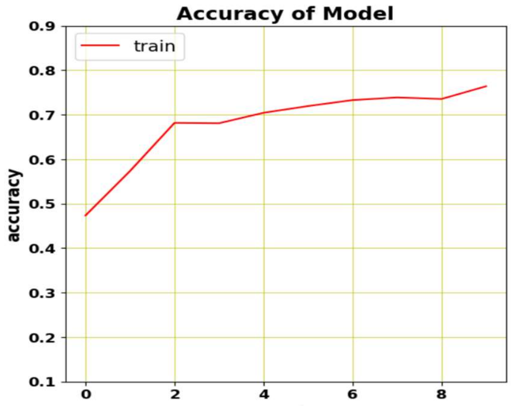
Below is a contour plot describing relationships.



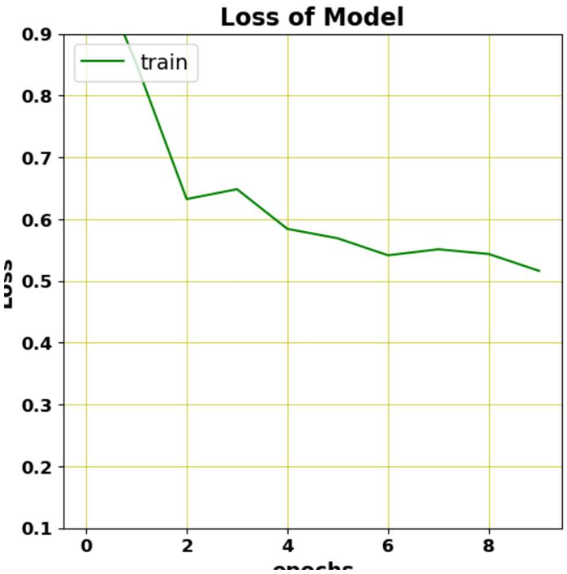
The history of optimization is as follows



CNN model convergence is evident from the following training and loss graph.



Loss of model is

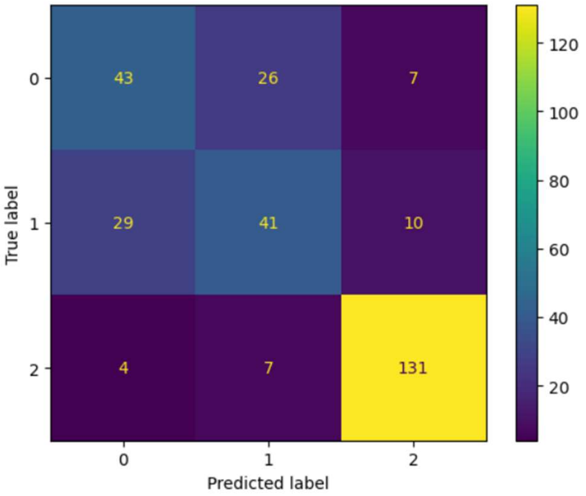


D. Step 4

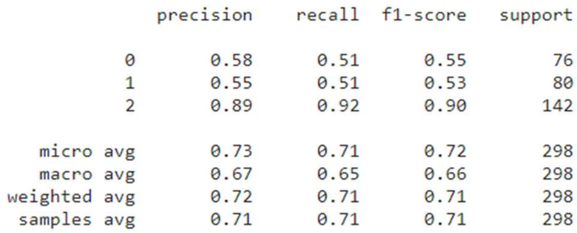
The last step of the study involved getting results and extracting features from the images that could be fed into machine learning models. Several models were implemented that included Random Forest Classifier, K Neighbors Classifier and an ensemble of several weak classifiers that included Linear SVC, Logistic Regression, KNN, Decision Tree Classifier. Ensembles of weak classifiers were made using the majority voting technique. Machine Learning based classification increased the performance of the CNN models significantly.

III. Results and Discussion

The first deployment of CNN model gave an accuracy of 25.5% on the test set. This promised a scope of improvement and after using optuna the new model architecture gave an accuracy of 77% on the test set. Feature Extraction first with the random forest model gave an accuracy of 70.8%, its confusion matrix is



The classifiation report is as follows:

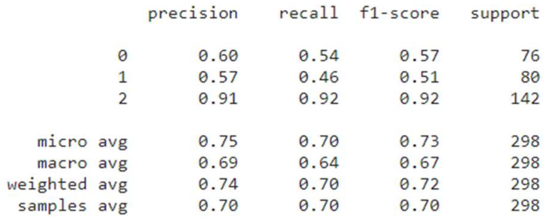


```
              precision    recall  f1-score   support

           0       0.58      0.51      0.55        76
           1       0.55      0.51      0.53        80
           2       0.89      0.92      0.90       142

   micro avg       0.73      0.71      0.72       298
   macro avg       0.67      0.65      0.66       298
weighted avg       0.72      0.71      0.71       298
 samples avg       0.71      0.71      0.71       298
```

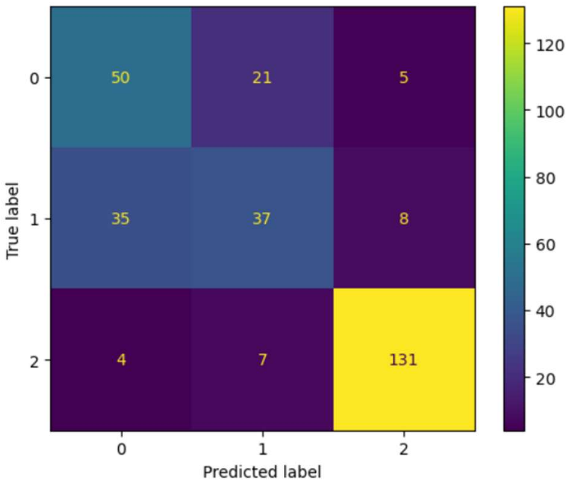
Feature extraction and classification with the KNN model gave an accuracy of 70.13% its classification report and confusion matrix are discussed below.



```
              precision    recall  f1-score   support

           0       0.60      0.54      0.57        76
           1       0.57      0.46      0.51        80
           2       0.91      0.92      0.92       142

   micro avg       0.75      0.70      0.73       298
   macro avg       0.69      0.64      0.67       298
weighted avg       0.74      0.70      0.72       298
 samples avg       0.70      0.70      0.70       298
```



The last model that made classification using the features extraction was an ensemble of weak classifiers and it gave an accuracy 71.14%. The performance increased from 25.5 % to a maximum of 77% and hence there is scope for improvement. As discussed, earlier with more computing resources one can much efficiently optimize the hyper parameters and train the model for a greater number of epochs that promises the scope of improvement.

REFERENCES

[1] Precision agriculture classification using convolutional neural networks for paddy growth level J. Clerk Maxwell, A Treatise on Electricity and Magnetism, 3rd ed., vol. 2. Oxford: Clarendon, 1892, pp.68–73.

[2] @misc{chollet2015keras,title={Keras},author={Chollet, Fran\c{c}ois and others}, year={2015}, howpublished={\url{https://keras.io}},

[3] Martín Abadi, Ashish Agarwal, Paul Barham, Eugene Brevdo,Zhifeng Chen, Craig Citro, Greg S. Corrado, Andy Davis,Jeffrey Dean, Matthieu Devin, Sanjay Ghemawat, Ian Goodfellow, Andrew Harp, Geoffrey Irving, Michael Isard, Rafal Jozefowicz, Yangqing Jia, Lukasz Kaiser, Manjunath Kudlur, Josh Levenberg, Dan Mané, Mike Schuster, Rajat Monga, Sherry Moore, Derek Murray, Chris Olah, Jonathon Shlens, Benoit Steiner, Ilya Sutskever, Kunal Talwar, Paul Tucker, Vincent Vanhoucke, Vijay Vasudevan, Fernanda Viégas, Oriol Vinyals, Pete Warden, Martin Wattenberg, Martin Wicke, Yuan Yu, and Xiaoqiang Zheng. TensorFlow: Large-scale machine learning on heterogeneous systems, 2015. Software available from tensorflow.org.

[4] @software{reback2020pandas,autho={The pandas development team,} title={pandas-dev/**pandas**: Pandas}

[5] Bradski, G., 2000. The OpenCV Library. Dr. Dobb&#x27;s Journal of Software Tools.

[6] arXiv:1907.10902 [cs.LG].

[7] Harris, C.R., Millman, K.J., van der Walt, S.J. et al. Array programming with NumPy. Nature 585, 357–362 (2020). DOI: 10.1038/s41586-020-2649-2.