1.Machine Learning Project

1.1 Introduction

Soybeans are one of the leading staple crops consumed around the world, with an annual global production of 240 million metric tons [1]. With the assistance of machine learning, this project aims to predict the quality of soybeans (based on the GB1352-2009 Standard of Soybean Classification), using the images of soybean seeds.

This paper will first formulate the problem and describe the dataset to be used. It will then go over various ML methods used. We have decided to use convolution neural networks, Support vector machines, and Random forest classifiers. For each of these methods, we also describe our motivation for choosing them, the feature extraction process, choice of loss function, and the test-train split. Lastly, we will compare the results we have obtained from the different ML methods and present the best one.

It should be noted that a trained ML model already exists that uses the same dataset on Kaggle for CNN [4]. However, no part of this project is **neither** copied, **nor** derived from the previous project. This work is hence completely independent from existing models using the same dataset.

1.2 Problem Formulation

The dataset, consisting of 5513 images of soybean seeds of different types, was obtained by taking pictures with industrial cameras and downloaded from Kaggle [2].

The problem to be solved is: Given a labeled dataset consisting of images of different types of soybean seeds, and when a new image is given as input, predict the quality (classify it into one of the given categories) of the input image.

The data is categorical in nature, and each image is assigned exactly one of these five categories: intact, spotted, immature, broken, and skin-damaged. Each datapoint is a RGB image of a soybean seed with a size 227 x 227 pixels. As each image has a size of 227 * 227 px, each image datapoint has 51529 features (so multidimensional), with the RGB intensity of each pixel representing a feature.

As the data is labeled, and the labels represent the category to which a seed belongs to, **supervised learning** methods will be used to train the models. The aim of the project is to use the trained models to predict the category of soybean seeds.

1.3 Methods

As previously stated, the total number of data points is 5513 and each of them belong to one of the 5 categories mentioned earlier. The dataset downloaded has the data in the correct format, and no additional data preprocessing is needed. In the beginning we selected the first 100 photos from each set (~9% in total) and separately stored them for testing purposes after selecting the best model.

We have trained three ML models on the dataset: Convolution Neural Network, Support Vector Machine, and Random Forest classifier. The motivation behind using these specific models, and their specific details are mentioned in the subsections below.

Since we are working with images, the features are the rgb values of each pixel of an image. As such, we have not used any specific feature extraction process but we have used PCA for the last two methods. CNNs and random forest classifiers can automatically extract features from images, and do not need any manual feature selection process, and PCA is an easy way to reduce the number of features for SVMs so that it is easy to train it.

1.3.1 Convolutional Neural Network (CNN)

The first model that was chosen for this problem is Convolutional Neural Network (CNN). The motivation behind using a CNN for this machine learning task was based on its suitability and excellence in image classification tasks. CNNs excel at capturing local features and learning patterns in images, making them ideal for identifying the visual differences that are essential for classifying the quality of soybean seeds based on input images. The model's ability to automatically extract relevant features from the data, coupled with their proven success in various image classification tasks, justifies its selection as the hypothesis space for this ML method.

The features are the RGB intensities of each pixel, and since a CNN is used, there is no need for a feature selection process unlike some classical machine learning algorithms. CNNs automatically learn and extract features from the input data during the training process. The only feature engineering that was done was changing the RGB intensities from a scale of [0, 255] to a scale of [0, 1] as it is easier to train CNNs with these smaller values.

Loss Function

The loss function used in this scenario is **Categorical Cross-Entropy**. The motivation behind choosing this loss function lies in its appropriateness for multi-class classification problems where each data point belongs to one and only one class, as is the case with classifying different types of soybean seeds. Categorical Cross-Entropy measures the dissimilarity between the predicted class probabilities and the ground truth labels. It encourages the model to assign a high probability to the correct class while penalizing incorrect class predictions. This loss function is particularly well-suited for the evaluation of the hypothesis because it not only provides a clear and interpretable measure of the model's performance but also helps in effectively training the model to make accurate and confident predictions among multiple classes, which is crucial for determining the quality of soybean seeds based on their classification.

$$\mathrm{CCE}(y, \hat{y}) = -\frac{1}{n_{\text{samples}}} \sum_{i=0}^{n_{\text{samples}}-1} y_i * \log \hat{y}_i$$

Fig 1: Categorical Cross Entropy Function

Data Split

The dataset was split into a training set, and a validation set via a single split. The training set comprised 80% of the total data, which equates to 4011 samples, while the validation set accounted for 20% of the data, totaling 1002 samples. We chose this specific split to strike a balance between having a sufficiently large training set to train the model effectively while also reserving a separate validation set for fine-tuning hyperparameters and monitoring the model's performance. This 80-20 split is a common practice in machine learning, as it provides a reasonable compromise between having enough data for training and having a reliable dataset for assessing the model's performance and preventing overfitting

1.3.2 Support Vector Machine (SVM)

For the second model, we chose Support Vector Machine (SVM) for its effectiveness in classification tasks and robustness. SVMs can handle both linear and non-linear data separation, making them suitable for classifying soybean seed quality based on images.

The code in particular first breaks down the images to flat readable RGB values which can then be used by the model to conduct its tests. PCA is also used to reduce the number of features down to 50.

Loss Function

We have used hinge loss as it is the most common choice of loss function in SVMs. Hinge loss maximizes the margin between classes, promoting effective class separation. It penalizes misclassification, promoting effective separation by positioning the decision boundary to balance accurate classification and maximizing class separation.

Data Split

We used an 80-20 split for training and validation, ensuring fairness in model comparison. This approach allowed for an equal opportunity for the models and provided sufficient data for training and evaluation.

Result

Unfortunately, this method didn't produce any meaningful results. The process took too much time and even after extended periods of time failed to produce any results. Thus, we will consider the project a failure, but we believe it is still worth mentioning.

```
# Train the SVM model on the PCA-transformed training data
svm_model.fit(X_train_pca, y_train)

# Evaluate the svm_model on the PCA-transformed testing data
y_pred_pca = svm_model.predict(X_test_pca)
accuracy_pca = accuracy_score(y_test, y_pred_pca)
report_pca = classification_report(y_test, y_pred_pca)

# Print accuracy and classification report
print(f'Accuracy with PCA: {accuracy_pca}')
print(f'Classification Report with PCA:\n{report_pca}')
```

1679m 28.8s Python

Runtime as of the moment the program was terminated.

1.3.3 Random Forest Classifier

For our final method to test we have chosen the Random Forest Classifier as a key machine learning model. Random Forests are useful for their versatility and robust performance across a spectrum of classification tasks. Particularly adept at handling high-dimensional data, such as images, they offer an ensemble approach that combines the strengths of multiple decision trees.

Random Forests alleviate the need for explicit feature engineering. Raw RGB pixel intensities served as the input features. To ensure model convergence and consistency, the RGB intensities underwent scaling, transitioning from the original range of [0, 255] to a more suitable scale of [0, 1].

Loss Function

Differing from traditional loss functions used in neural networks, Random Forests rely on impurity measures, like Gini impurity or entropy, to guide decision tree splits. Model evaluation hinges on performance metrics, with accuracy as a primary indicator.

Data Split

As with the first 2 models, an 80-20 split was chosen to ensure a similar split, and an accurate comparison between the results.

1.4 Comparison and Conclusion

In this machine learning project, we aimed to predict the quality of soybeans based on image data. We explored three distinct approaches: Convolutional Neural Networks (CNN), Support Vector Machines (SVM), and Random Forest Classifier. A dataset split of 80-20 was used to ensure the fair comparisons.

CNNs were chosen for their image classification prowess and the Categorical Cross-Entropy loss function was used. The CNN yielded promising results, showing an accuracy of 0.85, and loss of 0.45 on the validation set.

SVMs were selected for their effectiveness in classification tasks. The hinge loss was utilized to maximize class separation. Unfortunately, SVM failed to produce any meaningful results, with the process taking absurd amounts of time and thus producing no results.

Finally, Random Forests were deployed for their versatility in handling high-dimensional data. RGB pixel intensities were used without feature engineering and loss functions were replaced by impurity measures. The accuracy on validation set first increased exponentially but slowed later on and seems to have peaked as the accuracy of 400 and 500 trees are equal at 0.72.

Based on these validation errors, CNN showed the most promise in its classification. It had the highest accuracy within the shortest time. SVM failed to produce any meaningful results and is deemed a failure. Random Forests were promising at first, but tapered off with bigger tree sizes and the accuracy stagnated. To further test the CNN model, a test set containing some of the soybean images from the same dataset, that were not used for testing or validation, were selected. The results from the test set were also positive, with the test error and test accuracy at 0.41 and 83% which are comparable to the validation errors and accuracies.

To enhance the machine learning method, future directions may involve collecting additional training data, experimenting with different features, models, or loss functions, and optimizing the CNN architecture for better generalization. These findings highlight the potential for further optimization and the need to address the test set's performance for more robust classification.

To conclude, the project was aimed at classifying soybean seeds in five categories based on their images. Three models were chosen, and out of them, CNN showed the best performance, and the error on the test set was also comparable to that of the validation set. Further improvements may involve fine-tuning hyperparameters and exploring more advanced techniques. The limitations of each method should be considered, with room for refinement to achieve a more satisfactory solution.

1.5 References

[1] Staple Foods

https://en.wikipedia.org/wiki/Staple_food

[2] Soybean Seeds dataset

https://www.kaggle.com/datasets/warcoder/soyabean-seeds

[3] Categorical Cross-Entropy Function

https://ml-explained.com/blog/metrics-explained

[4] Other ML model trained on the same dataset

https://www.kaggle.com/code/philopateergeorgei/99-tensorflow

Appendix

Link to the codes

Convolutional Neural Network

https://github.com/rathee-harsh/soybean-seeds-classifier/blob/master/model.ipynb

Support Vector Machine (Unsuccessful)

https://github.com/Irmath/soybean-seeds-classifier1/blob/main/model2.ipynb

Random Forest Classifier

https://github.com/Irmath/soybean-seeds-classifier1/blob/main/model3.ipynb