COMPUTER VISION: PLANT SEEDLINGS CLASSIFICATION

PGAIML – University Of Texas McCombs School Of Business

November 17, 2023

Presented by Greg Wenzel

CONTENTS / AGENDA

Executive Summary I

Objective:

The project aims to apply computer vision techniques for the classification of plant seedlings. The goal is to accurately identify and classify different types of plant seedlings, leveraging the capabilities of machine learning and image processing. This has significant applications in agriculture, enabling more precise and efficient crop management.

Methodology:

Data Preparation: The dataset is comprised of images from various plant seedlings. The data was preprocessed, including normalization and splitting into training, validation, and testing sets.

Exploratory Data Analysis (EDA): An EDA was conducted to understand the dataset's characteristics. This included analyzing the distribution of classes, examining image properties, and identifying any data imbalances or anomalies.

Model Development: Multiple machine learning models were developed and trained. The process involved evaluating different architectures and tuning parameters to optimize performance.

Model Evaluation: Models were rigorously tested and evaluated using metrics such as accuracy, confusion matrices, and classification reports.

Executive Summary II

Key Findings: The models demonstrated varying levels of accuracy in classifying plant seedlings. Specific challenges, such as class imbalances or similarities between seedling types, were encountered in a few instances and addressed during model development.

Actionable Insights: Machine learning models, particularly those based on computer vision, can effectively classify plant seedlings, aiding in agricultural decision-making. The performance of models was enhanced through techniques such as data augmentation, learning rate adjustments, and parameter tuning.

Recommendations: Further Data Collection: Enhance the dataset with more varied and numerous images to improve model robustness.

Model Optimization: Continue to refine the models by exploring advanced neural network architectures or transfer learning techniques.

Real-World Testing: Deploy the models in a controlled agricultural environment to assess their practical utility and make necessary adjustments.

Expand Scope: It may prove beneficial to consider extending the project to include additional plant species or different growth stages of seedlings.

Business Problem Overview

Context:

In modern agriculture, the accurate identification and classification of plant seedlings are crucial for efficient crop management. This involves distinguishing between different crop species and weeds at an early stage, which is essential for optimizing yield, reducing herbicide use, and managing resources effectively.

Challenges:

Manual Identification Limitations: Traditional methods rely heavily on manual labor, which is time-consuming and prone to errors.

Complexity in Diverse Conditions: Variability in lighting, background, and plant morphology across different environments adds complexity.

Resource Intensive: Manual classification requires extensive labor and expertise, which can be costly and inefficient.

Scalability: Scaling manual processes for large farms is challenging and often impractical.

Solution Approach

Strategy:

Implementing a machine learning-based computer vision system to automate and enhance the accuracy of plant seedling classification.

Key Components:

Data Collection & Preparation: Assemble a comprehensive dataset of seedling images, covering various species and environmental conditions. Preprocess the data for machine learning compatibility.

Exploratory Data Analysis (EDA): Analyze the dataset to understand its characteristics, identify patterns, and detect any anomalies or imbalances.

Model Development & Training: Utilize advanced machine learning models, especially Convolutional Neural Networks (CNNs), to learn from the dataset. Experiment with different architectures and parameters.

Model Evaluation & Optimization: Assess model performance using accuracy metrics, confusion matrices, and classification reports. Refine the models based on these evaluations.

Implementation & Testing: Deploy the model in a controlled environment for testing. Gather feedback and make necessary adjustments.

Solution Approach Benefits

Benefits:

- **Increased Efficiency**: Automating seedling classification speeds up the process and reduces manual labor.
- **Enhanced Accuracy**: AI models can achieve higher accuracy levels than manual methods.
- **Scalability**: The solution can be scaled to handle large-scale agricultural operations.
- **Resource Optimization**: More accurate classification leads to better crop management and resource utilization.

Conclusion:

The adoption of AI-driven computer vision for plant seedling classification represents a transformative approach in agriculture. It addresses the key challenges of manual methods, offering a scalable, efficient, and accurate solution for modern farming needs.

EDA Results I

Image Resizing: The original images were resized to a uniform dimension of 64x64 pixels to reduce computational costs while maintaining sufficient detail for model training.

Class Distribution: A count plot was generated to check for class imbalance within the target variable. This is crucial as an imbalanced dataset can bias the model toward more common classes.

Color Space Conversion: The images were converted from BGR (Blue, Green, Red) to RGB (Red, Green, Blue) color space using OpenCV's cvtColor function. This aligns with the standard color format used in most image processing tasks.

Data Normalization: Pixel values were normalized by scaling them to a range between 0 and 1. This standardizes the input for the neural network, helping in faster convergence during training.

Data Splitting: The dataset was split into training, validation, and testing sets with ratios of 80%, 10%, and 10%, respectively, using train_test_split from scikit-learn. Stratification was used to maintain the same class distribution across the splits.

Encoding Target Classes: The target labels were encoded using LabelBinarizer to transform class names into one-hot encoded vectors, which is a suitable format for classification tasks with neural networks.

EDA Results II

Random Images Visualization: The function plot_images is used to display random images from each class, providing a visual understanding of the dataset's variety and complexity.

Data Shape Verification: The shapes of the train, validation, and test datasets are checked to confirm the consistency of data after splitting.

Encoding Verification: The shape of the encoded target variables for train, validation, and test sets is verified to ensure the encoding process was successful.

Plotting Random Images: By plotting random images from each class, the diversity within and across the classes can be visually assessed. This can reveal variations in plant size, color, and morphology, which are critical for informing the modeling strategy.

Checking Distribution of Target Variable: Visualizing the distribution of classes can help identify if there's a need for techniques to handle class imbalance, such as oversampling, undersampling, or weighted loss functions during model training.

Images Before and After Resizing: By comparing images before and after resizing, one can ensure that the resizing process maintains the integrity of visual features necessary for classification.

Image before resizing

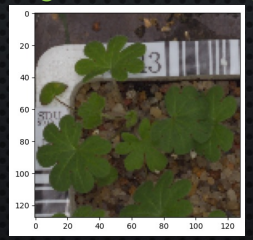
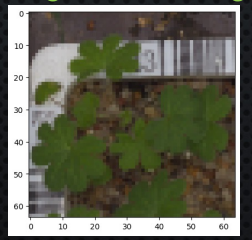
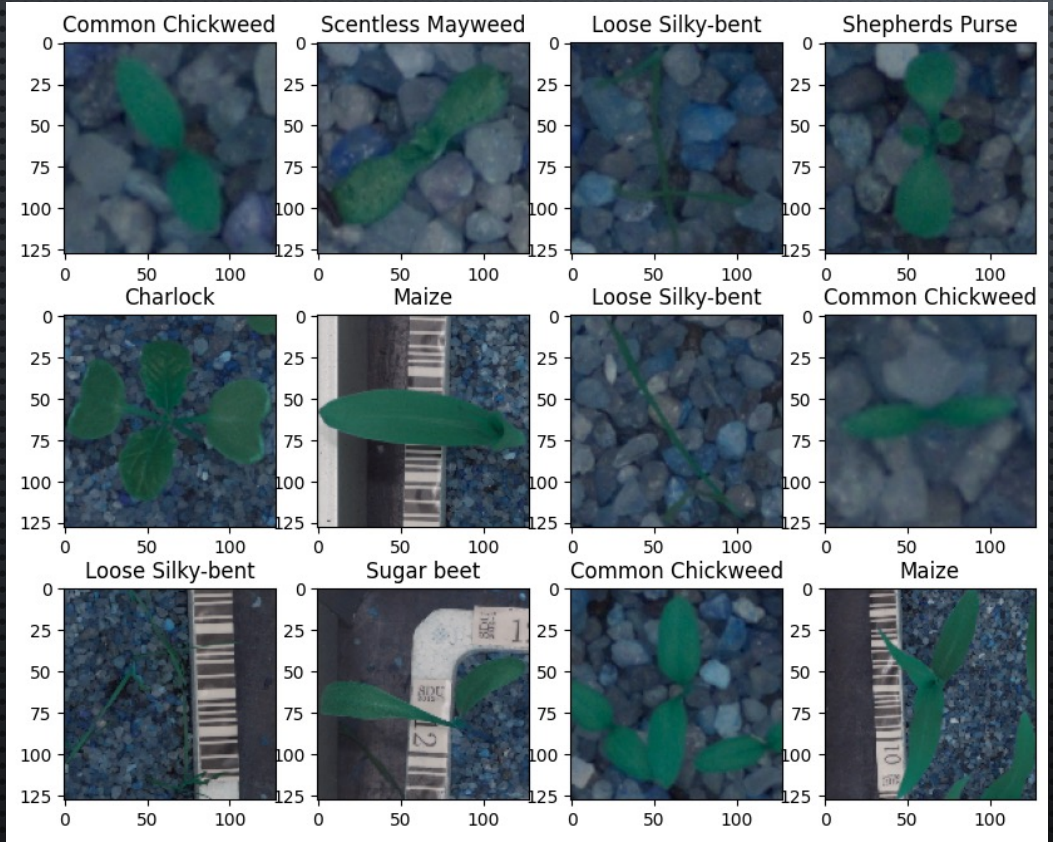


Image after resizing

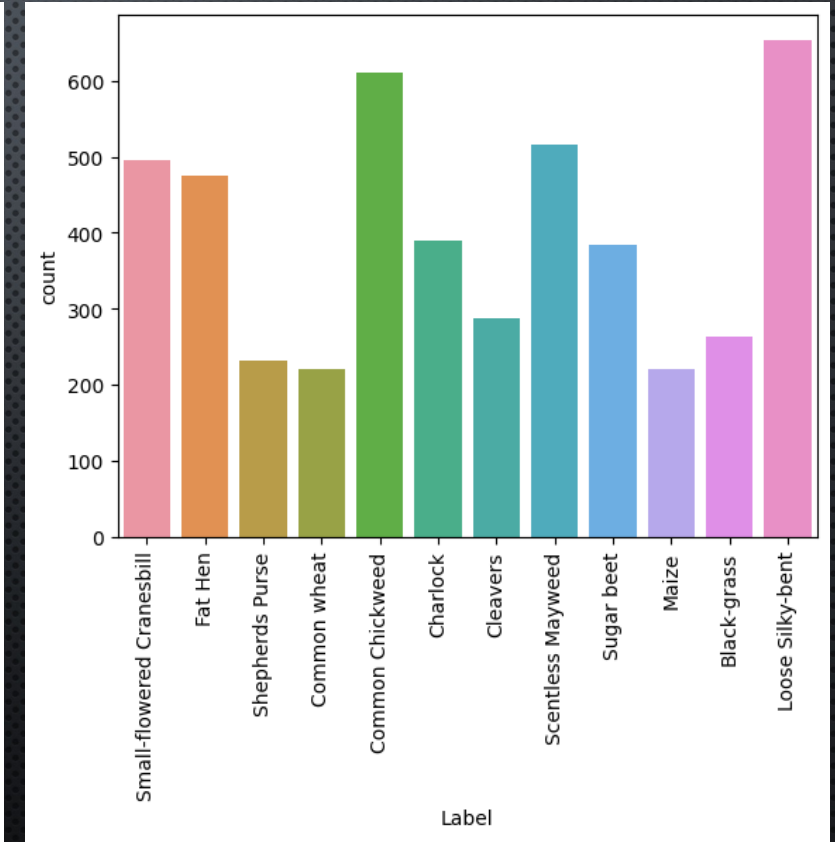


EDA Results III

Random Images from Each Seedling Class



Distribution of Target Variable



Data Pre-Processing Overview

1 **Color Space Conversion**: Images are converted from BGR to RGB. This is a crucial step because most image data in machine learning uses the RGB color space, and it is important for the consistency and accuracy of the model's performance.

2 **Image Resizing**: The images are resized from an original larger size to 64x64 pixels. Resizing images helps to reduce computational costs and allows the model to train faster while maintaining enough detail for the classification task.

3 **Data Splitting**: The dataset is split into training, validation, and testing sets with a distribution of 80%, 10%, and 10%, respectively. The train_test_split function is used, with stratification to ensure that each set represents the overall class distribution. This is important for training robust models and avoiding overfitting.

4 **Encoding Target Labels**: The LabelBinarizer is used to convert class labels into one-hot encoded vectors, a necessary preprocessing step for categorical target variables in classification problems.

5 **Data Normalization**: Pixel values are normalized by scaling them to the range [0, 1]. Normalization is a common practice that helps in the convergence of the model during training because it ensures that all features (pixel values, in this case) are on a similar scale.

Data Pre-Processing : Significance of Steps

Color Space Conversion: Ensures that the image data is compatible with most libraries and standards that expect RGB format.

Image Resizing: Creates uniformity in the input data and reduces the model's computational demands.

Data Splitting: Provides a basis for training models in a way that can be validated and tested for generalization to new, unseen data.

Encoding Target Labels: Transforms categorical labels into a format that can be effectively used by machine learning models for classification.

Data Normalization: Helps prevent issues related to the scale of data, like slow learning and convergence to suboptimal solutions.

A complete and proper data preprocessing schedule is fundamental for building effective machine learning models, particularly in the domain of image classification, where the input data can be highly varied and complex.

Performance Summary : Model 1

Model 1 Architecture : Sequential

3 Convolutional Layers
2 Max Pooling Layers
Flattening Layer
Dense Layer
Output Layer
Compilation : Adam optimizer

Summary:

The model has a total of 128,828 parameters, all of which are trainable.

The architecture is designed to perform image classification into one of 12 classes.

The use of dropout is intended to mitigate overfitting, which is common in deep learning models, especially when dealing with image data.

Layer (type)	Output Shape	Param #
conv2d (Conv2D)	(None, 64, 64, 128)	3584
max_pooling2d (MaxPooling2D)	(None, 32, 32, 128)	0
conv2d_1 (Conv2D)	(None, 32, 32, 64)	73792
max_pooling2d_1 (MaxPooling2D)	(None, 16, 16, 64)	0
conv2d_2 (Conv2D)	(None, 16, 16, 32)	18464
max_pooling2d_2 (MaxPooling2D)	(None, 8, 8, 32)	0
flatten (Flatten)	(None, 2048)	0
dense (Dense)	(None, 16)	32784
dropout (Dropout)	(None, 16)	0
dense_1 (Dense)	(None, 12)	204

Total params: 128828 (503.23 KB)
Trainable params: 128828 (503.23 KB)
Non-trainable params: 0 (0.00 Byte)

In Depth Summary I : Model 1

First Convolutional Layer:

It consists of 128 filters of size 3x3.
The 'relu' activation function is used.
Padding is set to 'same' to maintain the output size equal to the input size.
The input shape is set to 64x64x3, which corresponds to the image dimensions and color channels.

First Max Pooling Layer:

A 2x2 pooling window is used.
Padding is set to 'same'.

Second and Third Convolutional Layers:

Two additional sets of convolutional layers followed by max pooling layers are included.
The second convolutional layer has 64 filters, and the third has 32 filters, both with a size of 3x3 and 'relu' activation.

Flattening Layer:

The output of the last max pooling layer is flattened to create a single long feature vector.

Dense Layer:

A fully connected layer with 16 neurons and 'relu' activation follows the flattening layer.
A dropout layer with a rate of 0.3 is included after the dense layer to reduce overfitting.

In Depth Summary II : Model 1

Output Layer:
The output layer has 12 neurons, one for each class, with a 'softmax' activation function. This is typical for multi-class classification problems.

Compilation:
The Adam optimizer is used for training the model.
The loss function is 'categorical_crossentropy', suitable for multi-class classification tasks.
The metric for evaluation is 'accuracy'.

Summary:
The model has a total of 128,828 parameters, all of which are trainable.
The architecture is designed to perform image classification into one of 12 classes.
The use of dropout is intended to mitigate overfitting, which is common in deep learning models, especially when dealing with image data.

This CNN is structured to progressively reduce the spatial dimensions while increasing the depth of feature maps, a common approach in CNNs for image classification. The model is expected to capture complex features at various levels of abstraction, with the dense layers at the end serving to make predictions based on these features.

Validation Data loss : 1.0049 - accuracy: 0.6799
Test Data loss: 1.0431 - accuracy: 0.6526

Performance Summary : Model 2

Model 2 Architecture : Sequential

2 Convolutional Layers
2 Max Pooling Layers
Batch Normalization Layer
Flattening Layer
Dense Layer
Dropout Layer
Output Layer w/softmax activation
Compilation : Adam optimizer
Loss Function : Categorical Crossentropy

Summary:
The model has a total of 151,676 parameters.

ImageDataGenerator with 20 degrees rotation & nearest fill with 30 epochs & batch size of 64 are used in this model. A learning rate adjustment callback is employed if no improvement is seen to improve model training.

Layer (type)	Output Shape	Param #
conv2d (Conv2D)	(None, 64, 64, 64)	1792
max_pooling2d (MaxPooling2D)	(None, 32, 32, 64)	0
conv2d_1 (Conv2D)	(None, 32, 32, 32)	18464
max_pooling2d_1 (MaxPooling2D)	(None, 16, 16, 32)	0
batch_normalization (Batch Normalization)	(None, 16, 16, 32)	128
flatten (Flatten)	(None, 8192)	0
dense (Dense)	(None, 16)	131088
dropout (Dropout)	(None, 16)	0
dense_1 (Dense)	(None, 12)	204

Total params: 151676 (592.48 KB)
Trainable params: 151612 (592.23 KB)
Non-trainable params: 64 (256.00 Byte)

In Depth Summary I : Model 2

Convolutional Layers: The model includes two convolutional layers. The first has 64 filters, and the second has 32 filters, both with a kernel size of 3x3 and 'relu' activation function. Padding is set to 'same' to retain the dimensionality post-convolution.

Max Pooling Layers: Each convolutional layer is followed by a max pooling layer with a 2x2 window, also with 'same' padding to reduce the spatial dimensions of the feature maps.

Batch Normalization: Included after the second set of convolution and pooling layers to stabilize learning by normalizing the activations.

Flatten Layer: The feature maps are flattened into a single vector to connect to the dense layers.

Dense Layers: A dense layer with 16 neurons is used for intermediate processing, followed by a dropout layer with a rate of 0.3 to reduce overfitting.

Output Layer: The final dense layer has 12 neurons with a 'softmax' activation function, suitable for multi-class classification of 12 classes.

Optimizer: The Adam optimizer is employed for training.

Loss Function: Categorical crossentropy is used, appropriate for multi-class classification tasks.

Model Size: The model has a total of 151,676 parameters, all of which are trainable.

In Depth Summary II : Model 2

Initial Performance: The model starts with a training accuracy of 25.22% and a validation accuracy of 16.36%.

Learning Progress: Over the epochs, the model shows consistent improvement in both training and validation accuracy.

Final Performance: By the end of the training, the model achieves a training accuracy of 75.26% and a validation accuracy of 77.80%.

Test Data Performance Metrics
Accuracy on Test Data: The model achieves an accuracy of 76.63% on the test data.

Classification Report:

- The model performs well on certain classes, as evidenced by high precision and recall for classes like 1, 3, and 10.
- There are classes where the model does not perform as well, indicated by lower precision and recall, such as class 0 and class 4.
- The macro and weighted averages of precision, recall, and F1-score are around 76%, aligning with the test accuracy.

Summary and Analysis
Model 2 is well-structured and has been trained to recognize various plant seedlings with a good degree of accuracy. The use of data augmentation and learning rate reduction strategies during training has helped improve the model's ability to generalize. The performance metrics indicate that the model may have some bias towards certain classes, which could be addressed with further training, data augmentation, or class weighting. The model's strengths are evident in its ability to classify several classes with high accuracy, and it demonstrates a balanced approach to precision and recall across most classes.

Validation Data	loss : 0.7327 - accuracy: 0.7780
Test Data	loss: 0.7680 - accuracy: 0.7663

Performance Summary : Model 3

Model 1 Architecture : Sequential

2 Convolutional Layers
2 Max Pooling Layers
Batch Normalization
Flattening Layer
Dense Layer
Dropout Layer
Output Layer
Compilation : Adam optimizer
Lost Function : Categorical Crossentropy

Summary:
This model represents a fine-tuned version of earlier models with a total of 269,036 parameters featuring enhancements in the data augmentation process, network architecture, and training regimen. Additional models were tested and tuned during this process and Model 3 is the chosen version used for comparison.

Layer (type)	Output Shape	Param #
conv2d (Conv2D)	(None, 64, 64, 64)	1792
max_pooling2d (MaxPooling2D)	(None, 32, 32, 64)	0
conv2d_1 (Conv2D)	(None, 32, 32, 32)	18464
max_pooling2d_1 (MaxPooling2D)	(None, 11, 11, 32)	0
batch_normalization (Batch Normalization)	(None, 11, 11, 32)	128
flatten (Flatten)	(None, 3872)	0
dense (Dense)	(None, 64)	247872
dropout (Dropout)	(None, 64)	0
dense_1 (Dense)	(None, 12)	780

Total params: 269036 (1.03 MB)
Trainable params: 268972 (1.03 MB)
Non-trainable params: 64 (256.00 Byte)

In Depth Summary I : Model 3

Data Augmentation: Enhanced data augmentation techniques including horizontal and vertical flips, random rotations, shifts, shearing, zooming, and fill mode settings to improve the model's ability to generalize and prevent overfitting.

Convolutional Layers: The first convolutional layer has 64 filters of size 3x3, followed by a max pooling layer with a 2x2 window. The second convolutional layer has 32 filters of size 3x3, followed by a max pooling layer with a 3x3 window, indicating a deeper feature extraction compared to the original model.

Batch Normalization: Applied after the convolutional layers to normalize the activations and speed up training.

Flattening Layer: Flattens the feature maps into a one-dimensional vector to prepare for dense connections.

Dense Layer: Increased to 64 neurons (from the previous 16) with 'relu' activation, suggesting a more complex model capable of capturing more intricate relationships in the data.

Dropout Layer: Maintained at a rate of 0.3 to combat overfitting.

Output Layer: Remains with 12 neurons using 'softmax' activation, appropriate for the 12-class classification problem.

Optimizer: Uses the Adam optimizer.

Loss Function: Categorical crossentropy, which is standard for multi-class classification.

Total Parameters: 269,036 trainable parameters, an increase that indicates a more complex model with the potential for better performance but also an increased risk of overfitting.

In Depth Summary II : Model 3

Training Regimen and Performance Metrics

Learning Rate Reduction: Implements ReduceLROnPlateau to halve the learning rate when the validation accuracy does not improve for three epochs, with a minimum learning rate set to 0.00001. This adaptive approach helps to fine-tune the model's weights, especially during the later stages of training.

Epochs and Batch Size: The model is trained for 30 epochs with a batch size of 64.

Training Performance: The training process demonstrates progressive improvement, with initial accuracy starting at 27.78% and improving over time.

Validation Performance: Validation accuracy improves from 14.25% to a peak of 82.71%, indicating good generalization.

Model Evaluation on Test Data

Test Accuracy: Model 3 achieves an accuracy of 80% on the test data, which is a robust performance metric.

Classification Report:

- Model 3 shows high precision and recall for several classes, indicating effective learning.
- The precision for classes 1, 2, 3, 4, 5, 7, 8, 10, and 11 is particularly high, suggesting that the model is confident and accurate in its predictions for these classes.
- The recall is also high for most classes, meaning the model is able to identify most positive instances correctly.
- The macro average and weighted average scores are both solid, showing that the model's performance is consistently strong across different classes.

In Depth Summary III : Model 3

Visual Inspection and Predictions

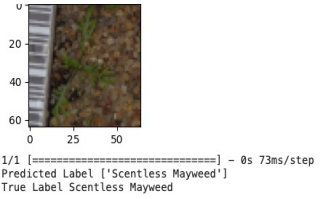
The model's predictions are visually inspected by comparing the predicted labels with the true labels for individual test images. Visual confirmation of the code, labels and random images indicate that the model's predictions align well with the true labels, suggesting that the model has effectively learned to recognize and classify the plant seedling images.

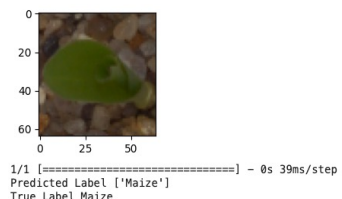
Summary and Analysis

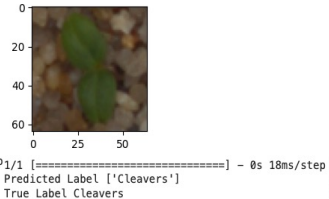
Model 3 represents a significant improvement over the initial iterations. The enhancements in data augmentation and model complexity have led to a substantial increase in accuracy and a balanced precision-recall trade-off across classes. The use of adaptive learning rate adjustments through ReduceLROnPlateau has likely contributed to this improved performance by fine-tuning the model's weights effectively during training.

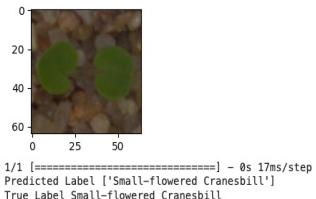
The test results, supported by the detailed classification report and visual confirmations, suggest that Model 3 is a robust classifier for this plant seedling classification task, with strong predictive capabilities that could likely be deployed in a practical setting with confidence.

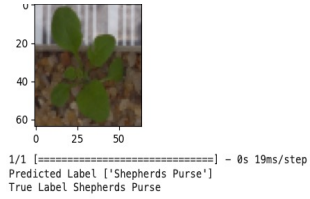
Validation Data loss : 0.5449 - accuracy: 0.8294 Test Data loss: 0.6779 - accuracy: 0.7979











Conclusion I

Model Evolution: Starting with a basic CNN architecture, the models evolved in complexity and capability. The final iteration, Model 3, incorporated comprehensive data augmentation techniques and architectural refinements that significantly enhanced its ability to generalize from the training data to unseen images.

Performance Metrics: The performance of Model 3 on the training and validation datasets showed a consistent improvement over the training epochs, achieving a final validation accuracy of 82.71%. The test accuracy reached 80%, which is a strong indicator of the model's robustness.

Class-specific Performance: The classification report revealed that Model 3 performed well across all seedling categories, with particularly high precision and recall for several classes. This suggests that the model is not only accurate on average but also maintains reliable performance across diverse seedling types.

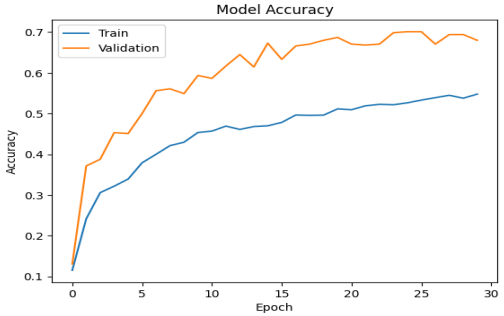
Learning Rate Adaptation: The application of ReduceLROnPlateau allowed for an adaptive learning rate, which improved model convergence and ultimately led to better performance on the validation and test sets.

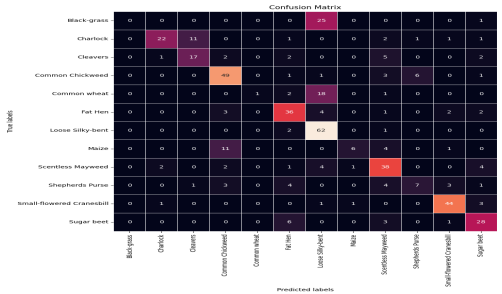
Insights and Recommendations: Collecting more data, especially for underrepresented classes, could further enhance the model's accuracy. Additionally, exploring more advanced neural network architectures or transfer learning could offer gains in both performance and efficiency.

Practical Implications: The successful classification of plant seedlings has direct applications in agriculture, such as automating the process of identifying and sorting seedlings, assisting in early weed detection, and contributing to advanced agriculture practices that can lead to better crop management and increased yields.

Conclusion II : Visual Summary

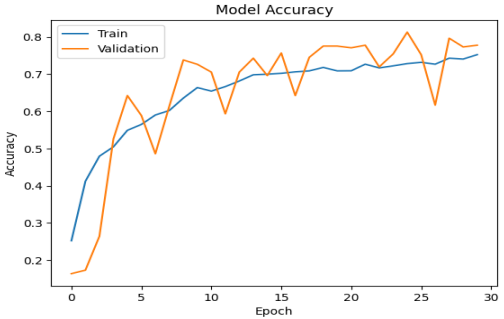
Model 1

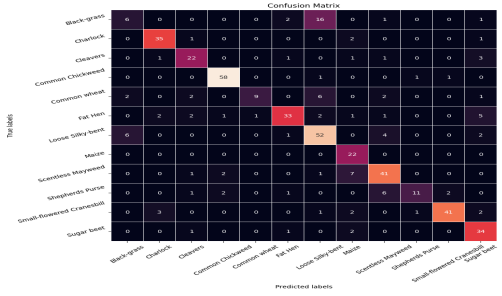




	precision	recall	f1-score	support
0	0.00	0.00	0.00	26
1	0.85	0.56	0.68	39
2	0.59	0.59	0.59	29
3	0.70	0.80	0.75	61
4	1.00	0.05	0.09	22
5	0.65	0.75	0.70	48
6	0.54	0.95	0.69	65
7	0.75	0.27	0.40	22
8	0.61	0.73	0.67	52
9	0.50	0.30	0.38	23
10	0.85	0.88	0.86	50
11	0.65	0.74	0.69	38
accuracy			0.65	475
macro avg	0.64	0.55	0.54	475
weighted avg	0.65	0.65	0.61	475

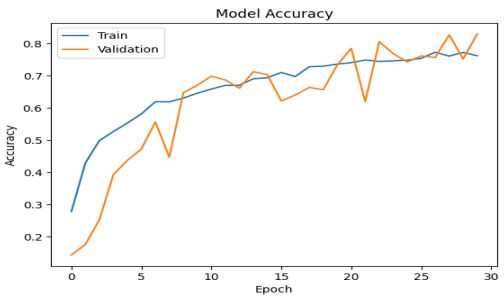
Model 2

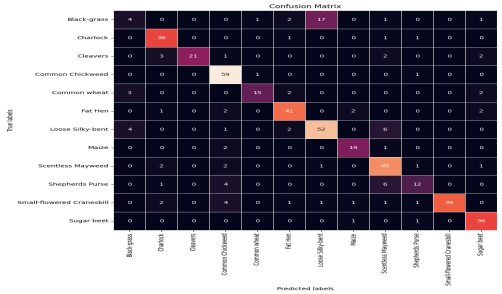




	precision	recall	f1-score	support
0	0.43	0.23	0.30	26
1	0.85	0.90	0.88	39
2	0.73	0.76	0.75	29
3	0.92	0.95	0.94	61
4	0.90	0.41	0.56	22
5	0.85	0.69	0.76	48
6	0.66	0.80	0.72	65
7	0.59	1.00	0.75	22
8	0.73	0.79	0.76	52
9	0.85	0.48	0.61	23
10	0.93	0.82	0.87	50
11	0.69	0.89	0.78	38
accuracy			0.77	475
macro avg	0.76	0.73	0.72	475
weighted avg	0.78	0.77	0.76	475

Model 3





	precision	recall	f1-score	support
0	0.36	0.15	0.22	26
1	0.80	0.92	0.86	39
2	1.00	0.72	0.84	29
3	0.79	0.97	0.87	61
4	0.88	0.68	0.77	22
5	0.84	0.85	0.85	48
6	0.73	0.80	0.76	65
7	0.83	0.86	0.84	22
8	0.71	0.87	0.78	52
9	0.71	0.52	0.60	23
10	1.00	0.78	0.88	50
11	0.82	0.95	0.88	38
accuracy			0.80	475
macro avg	0.79	0.76	0.76	475
weighted avg	0.79	0.80	0.79	475

Appendix

Data Background

Background & Contents:

- The data set comprises images of approximately 960 unique plants belonging to 12 species at several growth stages.

List of Species

Black-grass	Common Wheat	Maize	Small-flowered Cranesbill	Loose Silky-bent	Shepherds Purse
Charlock	Fat Hen	Scentless Mayweed	Sugar beet	Cleavers	Common Chickweed

Data Summary:

- The images are 128x128 pixels in color (RGB).
- A subset of the data contains about 10% of the images from each class.
- The data is pre-processed through several steps to facilitate effective machine learning modeling:
  - Images are resized from 128x128 to 64x64 pixels.
  - The color channels are normalized by dividing by 255, rescaling pixel values to the range [0,1].
  - Data augmentation techniques such as random rotations, width and height shifts, shearing, zooming, and flipping are applied to introduce robustness to the model by simulating various angles and conditions.
- The pre-processing and augmentation strategies are crucial for improving the model's ability to generalize from the training data to new, unseen images, which may vary in many of the ways captured by the augmentations (e.g., orientation, size, and lighting conditions). These steps are standard practices in preparing image data for deep learning models, particularly in the context of computer vision tasks where invariance to such transformations is desired.

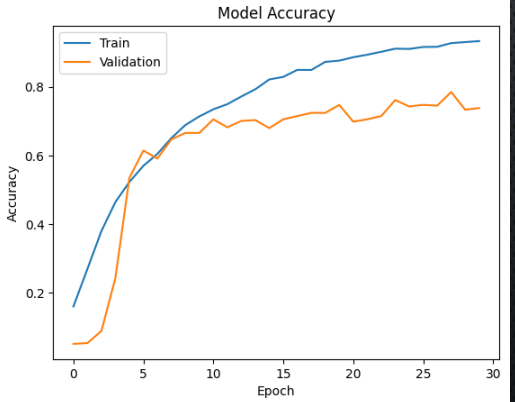
Additional Model Summaries

Model2_Tuned

Test Data = loss: 0.9769 - accuracy: 0.7326

Layer (type)	Output Shape	Param #
conv2d (Conv2D)	(None, 64, 64, 64)	1792
batch_normalization (BatchNormalization)	(None, 64, 64, 64)	256
max_pooling2d (MaxPooling2D)	(None, 32, 32, 64)	0
conv2d_1 (Conv2D)	(None, 32, 32, 32)	18464
batch_normalization_1 (BatchNormalization)	(None, 32, 32, 32)	128
max_pooling2d_1 (MaxPooling2D)	(None, 16, 16, 32)	0
flatten (Flatten)	(None, 8192)	0
dense (Dense)	(None, 64)	524352
dropout (Dropout)	(None, 64)	0
dense_1 (Dense)	(None, 12)	780

Total params: 545772 (2.08 MB)
Trainable params: 545580 (2.08 MB)
Non-trainable params: 192 (768.00 Byte)



Model2_Tuned2

Test Data = loss: 0.5709 - accuracy: 0.8400

Layer (type)	Output Shape	Param #
conv2d (Conv2D)	(None, 64, 64, 128)	1664
max_pooling2d (MaxPooling2D)	(None, 22, 22, 128)	0
conv2d_1 (Conv2D)	(None, 22, 22, 64)	131136
max_pooling2d_1 (MaxPooling2D)	(None, 8, 8, 64)	0
batch_normalization (BatchNormalization)	(None, 8, 8, 64)	256
flatten (Flatten)	(None, 4096)	0
dense (Dense)	(None, 32)	131104
dropout (Dropout)	(None, 32)	0
dense_1 (Dense)	(None, 12)	396

Total params: 264556 (1.01 MB)
Trainable params: 264428 (1.01 MB)
Non-trainable params: 128 (512.00 Byte)