Greenclassify: Deep Learning-Based Approach For Vegetable Image Classification

1. **Introduction**

   a. **Project overviews**

   Greenclassify is a deep learning-driven project focused on accurately classifying vegetable images. The project aims to achieve high accuracy in identifying various types of vegetables through meticulous dataset curation, model development using Convolutional Neural Networks (CNNs), and rigorous performance testing. By integrating the developed models into user-friendly applications, Greenclassify seeks to enhance agricultural processes and decision-making, offering scalable solutions for efficient vegetable classification in real-world scenarios.

   b. **Objectives**

   To develop a Deep Learning-Based model for accurately classifying vegetable images into 15 categories

2. **Project Initialization and Planning Phase**

   The "Project Initialization and Planning Phase" is where the project begins, setting clear goals, defining scope, and identifying stakeholders. This critical phase establishes project parameters, selects key team members, allocates resources, and outlines a feasible timeline. It also involves evaluating risks and planning to mitigate them. A successful initiation lays the groundwork for a well-structured and effectively managed machine learning project, ensuring clarity, alignment, and proactive strategies for potential obstacles.

   a. **Define Problem Statement**

   Problem Statement: Many grocery store owners and farmers face significant challenges with manually sorting and categorizing vegetables due to the varying sizes, shapes, and colors. This process is time-consuming, prone to errors, and labor-intensive, leading to operational inefficiencies, a pile-up of orders, and stress. Consequently, there is a pressing need for an automated solution to streamline vegetable sorting, reduce errors, and enhance efficiency for both grocery store operations and farming activities.

   Problem Statement Report:click here

b. Project Proposal (Proposed Solution)

1. Collect and preprocess the dataset.

2. Implement a CNN architecture suitable for image classification.

3. Train the model on the training set, validate it on the validation

set, and test its performance.

4. Try different models such as VGG16, ResNet50, Inception, Xception. Compare their accuracy.

5. Fine-tune hyperparameters to improve accuracy.

6. Deploy the final model using Flask for real-time classification

Project Proposal (Proposed Solution) Report: click here

c. Initial Project Planning

During the initial planning phase of the vegetable image classification project, objectives are defined, the scope is outlined, and stakeholders are identified. Timelines, resources, and a project strategy are set. The team deeply understands the vegetable image dataset, establishes analysis goals, and plans data preprocessing workflows. This phase ensures a systematic approach, laying the groundwork for successful model development and accurate classification results.

Initial Project Planning Report:click here

3. Data Collection and Preprocessing Phase

The Data Collection and Preprocessing Phase involves executing a plan to gather relevant vegetable image data from Kaggle and ensuring data quality. Preprocessing tasks include resizing, normalization and organizing the dataset for subsequent analysis and model development.

a. Data Collection Plan and Raw Data Sources Identified

The project aims to develop and train a deep learning-based model for accurately classifying various types of vegetables based on their images. Data will be sourced from publicly available datasets on Kaggle, specifically a vegetable image dataset containing 21,000 images from 15 classes, each with 1,400 images in 224x224 resolution and in .jpg format. The dataset will be split into three parts: 70% for training, 15% for testing, and 15% for validation. The vegetable image dataset, available as a 560 MB zip file, can be accessed publicly via the provided Kaggle link.

Raw Data Sources and Data Collection Report:click here

b. **Data Quality Report**

The Vegetable Image Dataset on Kaggle contains 21,000 high-resolution (224x224) JPG images of 15 different vegetable types, with 1,400 images per class. The dataset is balanced and publicly available, suitable for training and validating deep learning models. It is recommended to split the data into 70% for training, 15% for testing, and 15% for validation. Quality considerations include ensuring varied image conditions and accurate labeling for effective model training.

Data Quality Report:click here

c. **Data Preprocessing**

The images will be preprocessed by resizing, normalizing, augmenting, denoising, adjusting contrast, detecting edges, converting color space, cropping, batch normalizing, and whitening data. These steps will enhance data quality, promote model generalization, and improve convergence during neural network training, ensuring robust and efficient performance across various computer vision tasks.

Data Preprocessing Report:click here

4. Model Development Phase

The Model Development Phase for the "GreenClassify: Deep Learning-Based Approach for Vegetable Image Classification" project involves creating a predictive model for accurately classifying vegetables. This includes strategic feature selection and evaluating and selecting models such as CNN, Xception, Inception, ResNet50, and VGG16. The process starts with initiating training using appropriate code, followed by rigorous validation and performance assessment of these models to ensure reliable vegetable classification for various applications.

a. **Model Selection Report**

The Model Selection Report details the rationale behind choosing CNN, Xception, Inception, ResNet50, and VGG16 models for vegetable image classification. Each model is selected for its strengths in handling complex image patterns, high accuracy, robustness to variations, and overall performance in image classification tasks. CNN is chosen for its general effectiveness in image processing, Xception and Inception for their depth and efficiency, ResNet50 for handling deeper networks with residual connections, and VGG16 for its simplicity and high performance in benchmarks, ensuring a well-rounded and informed model selection aligned with project objectives.

Model Selection Report:click here

b. **Initial Model Training Code, Model Validation and Evaluation Report**

The Initial Model Training Code employs selected algorithms, including CNN, Xception, Inception, ResNet50, and VGG16, on the vegetable image dataset, setting the foundation for effective image classification. The subsequent Model Validation and Evaluation Report rigorously assesses model performance using metrics such as accuracy and precision to ensure reliability and effectiveness in accurately classifying various vegetable types. This comprehensive approach ensures that the models are robust and capable of performing well in real-world scenarios.

Initial Model Training Code, Model Validation and Evaluation Report:click here

5. **Model Optimization and Tuning Phase**

The Model Optimization and Tuning Phase involves refining neural network models for peak performance. It includes optimized model code, fine-tuning hyperparameters, comparing performance metrics, and justifying the final model selection for enhanced predictive accuracy and efficiency.

a. **Tuning Documentation**

For the GreenClassify project, deep learning models CNN, Xception, Inception, ResNet50, and VGG16 were deployed as-is, while ResNet50 underwent hyperparameter tuning. The tuning process for ResNet50 involved experimenting with learning rates and units. The best configuration achieved a learning rate of 0.0001 and number of units in dense layer as 32.

Model Optimization and Tuning Report:click here

b. **Final Model Selection Justification**

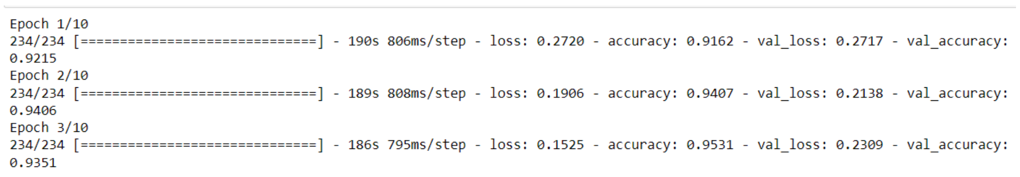
The Final Model Selection Justification articulates the rationale for choosing Xception as the ultimate model. Its exceptional accuracy, ability to handle complexity, and successful hyperparameter tuning align with project objectives, ensuring accurate prediction of vegetable classes.

Model Optimization and Tuning Report:click here

6. **Results**

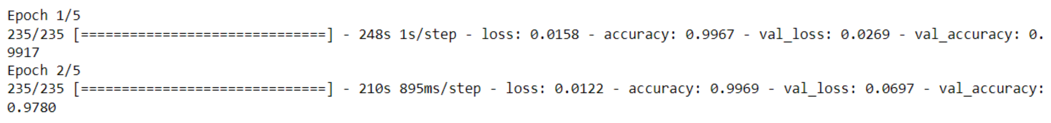
   a. **Output Screenshots**

1. **CNN:**



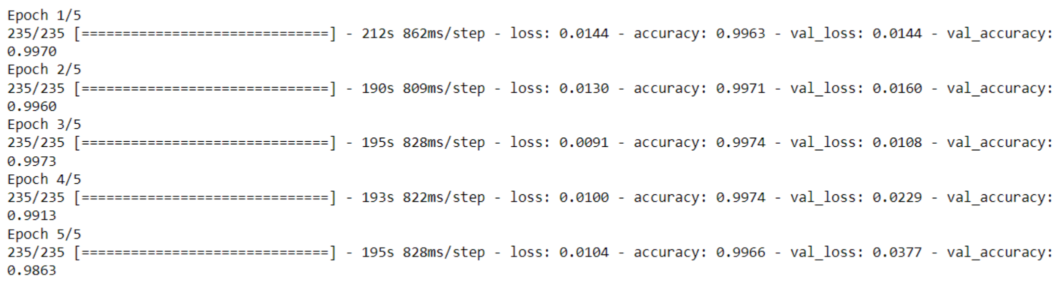
```
Epoch 1/10
234/234 [==============================] - 190s 806ms/step - loss: 0.2720 - accuracy: 0.9162 - val_loss: 0.2717 - val_accuracy:
0.9215
Epoch 2/10
234/234 [==============================] - 189s 808ms/step - loss: 0.1906 - accuracy: 0.9407 - val_loss: 0.2138 - val_accuracy:
0.9406
Epoch 3/10
234/234 [==============================] - 186s 795ms/step - loss: 0.1525 - accuracy: 0.9531 - val_loss: 0.2309 - val_accuracy:
0.9351
```

2. **VGG16**



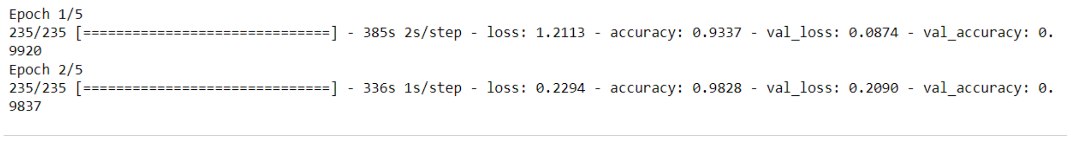
```
Epoch 1/5
235/235 [==============================] - 248s 1s/step - loss: 0.0158 - accuracy: 0.9967 - val_loss: 0.0269 - val_accuracy: 0.
9917
Epoch 2/5
235/235 [==============================] - 210s 895ms/step - loss: 0.0122 - accuracy: 0.9969 - val_loss: 0.0697 - val_accuracy:
0.9780
```

3. **RESNET50**



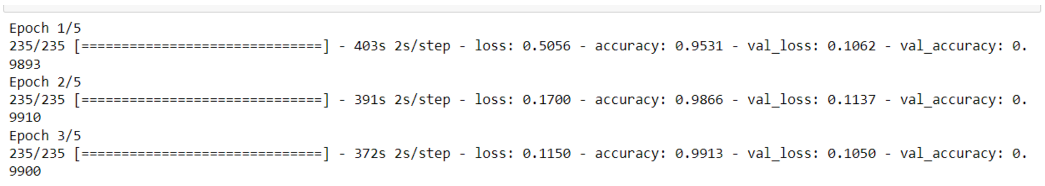
```
Epoch 1/5
235/235 [==============================] - 212s 862ms/step - loss: 0.0144 - accuracy: 0.9963 - val_loss: 0.0144 - val_accuracy:
0.9970
Epoch 2/5
235/235 [==============================] - 190s 809ms/step - loss: 0.0130 - accuracy: 0.9971 - val_loss: 0.0160 - val_accuracy:
0.9960
Epoch 3/5
235/235 [==============================] - 195s 828ms/step - loss: 0.0091 - accuracy: 0.9974 - val_loss: 0.0108 - val_accuracy:
0.9973
Epoch 4/5
235/235 [==============================] - 193s 822ms/step - loss: 0.0100 - accuracy: 0.9974 - val_loss: 0.0229 - val_accuracy:
0.9913
Epoch 5/5
235/235 [==============================] - 195s 828ms/step - loss: 0.0104 - accuracy: 0.9966 - val_loss: 0.0377 - val_accuracy:
0.9863
```

4. **INCEPTION**



```
Epoch 1/5
235/235 [==============================] - 385s 2s/step - loss: 1.2113 - accuracy: 0.9337 - val_loss: 0.0874 - val_accuracy: 0.
9920
Epoch 2/5
235/235 [==============================] - 336s 1s/step - loss: 0.2294 - accuracy: 0.9828 - val_loss: 0.2090 - val_accuracy: 0.
9837
```

5. **XCEPTION**



```
Epoch 1/5
235/235 [==============================] - 403s 2s/step - loss: 0.5056 - accuracy: 0.9531 - val_loss: 0.1062 - val_accuracy: 0.
9893
Epoch 2/5
235/235 [==============================] - 391s 2s/step - loss: 0.1700 - accuracy: 0.9866 - val_loss: 0.1137 - val_accuracy: 0.
9910
Epoch 3/5
235/235 [==============================] - 372s 2s/step - loss: 0.1150 - accuracy: 0.9913 - val_loss: 0.1050 - val_accuracy: 0.
9900
```

For source file: click here

7. **Advantages & Disadvantages**

Advantages:

1. Efficiency: Automates vegetable sorting, significantly reducing manual labor and associated errors.

2. Accuracy: Deep learning models provide high accuracy in classifying vegetables.

3. Scalability: Can be scaled to handle large volumes of data.

4. Versatility: Multiple models allow for comparison and selection of the best performing one.

5. Enhanced Operations: Improves operational efficiency for grocery store owners and farmers.

Disadvantages

1. Complexity: Requires expertise in deep learning and model tuning.

2. Resource-Intensive: High computational power and time are needed for training and validation.

3. Data Dependency: Performance is heavily reliant on the quality and diversity of the dataset.

4. Maintenance: Regular updates and maintenance of the model are necessary to ensure ongoing accuracy.

5. Cost: Initial setup and ongoing operational costs can be high.

8. **Conclusion**

In the GreenClassify project, after deploying and evaluating various deep learning models, Xception was chosen as the final model for vegetable image classification due to its superior performance and efficiency. While models like CNN, Inception, ResNet50, and VGG16 showed promising results, Xception's advanced architecture provided the highest accuracy and robustness across diverse conditions. This choice ensures the model's reliability in real-world applications, enhancing the operational workflows for grocery store owners and farmers by automating and optimizing the vegetable sorting process.

9. **Future Scope**

Future enhancements can include expanding the dataset to incorporate more vegetable types and conditions, integrating real-time classification systems, and employing transfer learning for improved accuracy. Additionally, developing a user-friendly interface and exploring edge computing solutions could further optimize the deployment and usability of the model in practical scenarios.

10. Appendix

a. Source Code

For source code, kindly refer to the link: click here

b. GitHub & Project Demo Link

Github link: github

Project demo link: video