Name of faculty:

Mrs kirti

Submitted by:

1. Agam Mourya

12212217(73)

2.Prince Kumar

12221412(72)

Title: Deep Learning-Based Fruit Classification using Transfer Learning with MobileNetV2

Abstract

In recent years, deep learning has emerged as a powerful approach for image classification tasks, particularly in the domain of agriculture and food technology. This paper presents a comprehensive system for fruit classification using a Convolutional Neural Network (CNN) model with transfer learning from MobileNetV2. Leveraging a labeled dataset of fruit and vegetable images, the system preprocesses data, performs data augmentation, trains a CNN model, and evaluates its performance using standard metrics. The model achieves high accuracy on test data and is deployed with a user-friendly prediction function. This research demonstrates the efficacy of transfer learning in real-world classification tasks and proposes future directions for improvement.

Keywords: Deep Learning, Fruit Classification, MobileNetV2, Transfer Learning, CNN, Image Classification, Data Augmentation, Agricultural Automation

1. Introduction Agriculture and food industries are experiencing a transformation through AI-powered solutions. One critical task is the automated classification of fruits based on visual characteristics such as shape, color, and texture. Accurate classification helps in quality control, inventory management, and reducing human error. Deep learning models, particularly Convolutional Neural Networks (CNNs), have shown promising results in image classification tasks. This paper explores a transfer learning approach using MobileNetV2 for classifying fruits from images.

The growing demand for automated food sorting systems has created a need for accurate and efficient classification models. With a robust AI-based classification model, farmers, retailers, and food processing industries can improve sorting accuracy, reduce labor dependency, and enhance customer satisfaction. The proposed method also provides a foundation for integrating vision-based classification with robotic arms for real-time sorting and packaging.

2. Related Work Previous studies have explored fruit classification using classical machine learning algorithms such as Support Vector Machines (SVMs), k-Nearest Neighbors (k-NN), and decision trees. These models depend on handcrafted features extracted using techniques such as Histogram of Oriented Gradients (HOG), Scale-Invariant Feature Transform (SIFT), and color histograms. While effective for small datasets and simple tasks, these models struggle to generalize on large-scale datasets with diverse features.

The advent of deep learning brought significant improvements to image classification. CNNs, introduced by LeCun et al., became a cornerstone of modern computer vision. Models such as AlexNet, VGG16, ResNet, and Inception networks have achieved state-of-the-art performance in benchmarks like ImageNet. Transfer learning, where a pre-trained model is adapted to a new task, has become a practical approach to leverage large-scale learning for specific domains.

Recent applications of CNNs in agriculture include plant disease detection, fruit ripeness estimation, and defect identification. The use of pre-trained models like MobileNetV2 has gained traction due to their efficiency and performance on resource-constrained devices.

3. Dataset Description The dataset used in this study is the "Fruits Dataset Images" from Kaggle, which includes three subsets: training, testing, and validation. The dataset comprises high-quality images of 36 classes of fruits and vegetables. Each class contains sufficient samples for effective training, validation, and testing.

- Training set path: ../Fruits_Vegetables/train
- Testing set path: ../Fruits_Vegetables/test
- Validation set path: ../Fruits_Vegetables/validation

Each image is stored in a folder corresponding to its label. The dataset contains thousands of images across different lighting conditions, backgrounds, and angles. Such diversity aids in improving the model's ability to generalize.

Data preprocessing involved extracting file paths, creating labels based on directory names, shuffling the dataset, and saving labels for future reference. Sample images from each class were visualized using matplotlib to ensure data integrity.

4. Methodology

4.1 Data Preprocessing Images were collected using Python’s pathlib and processed into a DataFrame using pandas. Labels were extracted from directory names using string manipulation. The dataset was randomized using sample(frac=1) to ensure that batches during training had diverse samples.

4.2 Data Augmentation To overcome overfitting and increase generalization, extensive augmentation techniques were applied:

- Rotation: Randomly rotating images up to 30 degrees.
- Zoom: Random zoom in range of 15%.
- Width and Height Shift: Translational shifts up to 20%.
- Shear: Shearing transformations for geometric variation.
- Horizontal Flip: Mirroring images to simulate left-right variations.
- Fill Mode: Filling in pixels post transformation using 'nearest' strategy.

This is implemented using Image DataGenerator from TensorFlow, which also applies MobileNetV2-specific preprocessing.

4.3 Transfer Learning with MobileNetV2 MobileNetV2, developed by Google, is a lightweight CNN designed for mobile and embedded vision applications. Its architecture is based on depthwise separable convolutions, reducing the number of parameters and computational cost.

The model is loaded without the top classification layers, and its weights are frozen to retain pre-trained features from ImageNet. These features capture general image features such as edges, textures, and shapes.

4.4 Model Architecture The architecture consists of:

- Input layer: Takes in 224x224 RGB images.
- Base: MobileNetV2 with include top=False and global average pooling.
- Dense layers: Two fully connected layers with 128 units each and ReLU activation.
- Output layer: Softmax activation with 36 output nodes for each class.

The model is compiled using Adam optimizer with categorical crossentropy as the loss function and accuracy as the evaluation metric.

4.5 Training Training is performed with early stopping to prevent overfitting. The model trains for 5 epochs with batch size 32. Training and validation accuracy and loss are plotted after training.

5. **Results**

5.1 Accuracy and Loss The model achieves over 90% accuracy on both training and validation sets. Loss curves indicate consistent learning with minimal overfitting.

5.2 Test Set Evaluation Predictions are generated on the test set. Results are evaluated using sklearn's accuracy_score. The final accuracy on the test set is 94.78%, indicating strong performance on unseen data.

5.3 Confusion Matrix A normalized confusion matrix is plotted using seaborn, showing accurate predictions across most classes. Minor confusion exists among visually similar classes like apple vs. tomato and lime vs. green apple.

5.4 Visual Inspection of Predictions Nine randomly selected test images are displayed with actual and predicted labels. Visual inspection confirms the model's robustness in identifying different fruits.

6. Deployment The model includes a user-friendly prediction interface:

- Accepts an image path
- Preprocesses image: resizing, normalizing
- Predicts class and outputs confidence score

The model is saved in Keras format for reuse and deployment in embedded systems or cloud services. It can be integrated with a robotic sorting arm for real-time fruit categorization and packaging.

7. **Discussion**

7.1 Model Performance The model performs exceptionally well, demonstrating the power of transfer learning for domain-specific tasks. Data augmentation played a critical role in improving robustness.

7.2 Comparative Analysis Compared with other architectures:

- MobileNetV2: 94.78% accuracy, fast, lightweight

- VGG16: 92.1% accuracy, slower, more parameters
- ResNet50: 93.5% accuracy, robust but heavier

MobileNetV2 strikes a balance between accuracy and computational efficiency, making it ideal for real-world deployment.

7.3 Limitations

- Misclassification in visually similar classes
- Limited performance in poor lighting conditions
- Dataset-specific bias

7.4 Practical Applications

- Automated fruit sorting in warehouses
- Quality control systems in packaging industries
- Mobile applications for fruit recognition in markets

7.5 Integration with Robotic Systems This classification model can serve as the perception layer in a robotic fruit sorting system. Combined with a robotic arm and conveyor mechanism, it can enable autonomous sorting and packaging in agricultural production lines.

8. Future Work Future improvements include:

- Fine-tuning MobileNetV2 to enhance accuracy
- Expanding dataset with more fruit types and real-world scenarios
- Introducing segmentation for multi-object images
- Building a mobile or web-based interface for end users
- Integration with edge AI devices like NVIDIA Jetson Nano
- Real-time inference optimizations using TensorFlow Lite

9. Conclusion This research demonstrates a complete pipeline for fruit classification using deep learning and transfer learning. With high accuracy, efficient architecture, and deployment capability, the system is well-suited for smart agriculture applications. By leveraging MobileNetV2, we achieve performance suitable for real-time and embedded systems.

The paper offers a reproducible and scalable approach for image classification tasks in food and agriculture. It sets the groundwork for future AI-driven automation in harvesting, sorting, and supply chain management.

10. References

- Howard, A. G., et al. (2017). MobileNets: Efficient Convolutional Neural Networks for Mobile Vision Applications.
- Simonyan, K., & Zisserman, A. (2014). Very Deep Convolutional Networks for Large-Scale Image Recognition.
- Krizhevsky, A., Sutskever, I., & Hinton, G. E. (2012). ImageNet Classification with Deep Convolutional Neural Networks.
- Chollet, F. (2017). Xception: Deep Learning with Depthwise Separable Convolutions.

- TensorFlow Documentation: https://www.tensorflow.org/
- Kaggle: Fruits Dataset Images - https://www.kaggle.com/datasets/shreyapmaher/fruits-dataset-images
- Keras Applications Documentation
- Scikit-learn Documentation
- LeCun, Y., Bengio, Y., & Hinton, G. (2015). Deep learning. Nature.