Project 4: Report
Team Name: Team Visionaries

Names: Abdullah Yassine
Jackson Sutherland
Suman Upreti

VisionQuest: Revolutionizing Supermarket Checkouts with AI-Powered Image Classification

1. Problem Space

Traditional check-out systems in supermarkets use barcode-scanning methods, and thus stickers are attached to fruits and vegetables. Handling these stickers is time-consuming for the cashier and customers. Further, it also contributes to microplastic pollution. Fruits and Vegetable Image classifiers would be helpful for checkout processes without barcodes, simplifying the checkout processes and reducing microplastic pollution.

The solution we provide here is different from the solutions out there in that we use a VGG-16 model which consists of 13 convolutional layers with three other layers acting as classifiers. Traditionally, CNNs were used to solve multi-class image problems. However, our model uses VGG-16 which has a total of 16 layers that helps analyze a complex data set which a CNN model might find difficult to do. We also normalize the data before we start using it which helps make it uniform among each other and this part significantly helps train our model at a higher accuracy than an accuracy of a model with normalization.

We also use a pre-trained VGG-16 model which helped greatly with our model accuracy because instead of building a model from scratch which might have took a lot of time and possibly not achieve the end-results that we were seeking for, we used a pre-trained model which helped us get the accuracy that we were looking for and at the same time train faster than a model which was built from the base up.

2. Approach

Our approach uses a deep convolutional neural network-driven supervised machine learning pipeline. It has evolved in response to early feedback and testing:

Model Selection and Variation: While our initial plan focused strongly on transfer learning in relation to well-understood, top-performing architectures such as VGG16 and InceptionV3 these models remain foundational. However, feedback encouraged us to explore additional architectures, including custom CNNs and other pre-trained networks. Exploring these numerous architectures helps mitigate overfitting and identifies the best configuration for subtle variations in produce categories.

Transfer Learning and Fine-Tuning: The models are initialized with pre-trained weights, such as VGG16, which are proficient in extracting features at large scales. Freezing most of the layers of these models and training only a few layers on top allows the models to adapt to produce-specific images. For fine-tuning, selected layers are gradually unfrozen. This controlled adaptation helps capture the minute variations in texture and color that distinguish one apple variety from another, for example, Gala from Fuji.

Simplifying Data Management: Early feedback highlighted the complexity of merging multiple datasets. Our initial plan was to integrate several Kaggle sources into one large, diverse dataset. To remain focused and avoid data-integration challenges, we streamlined efforts around a single, well-curated dataset, enabling us to concentrate on refining the model and extending augmentation techniques rather than on resolving data harmonization issues.

Regularization and Normalization: we introduced dropout layers into our models, encouraging robust feature learning and reducing reliance on any single set of features. We also used ImageNet normalization across the dataset so that the model can expect the same type of data each time it took an image as an input to its model.

Cross-Validation and Metrics: Train-Validation-Test split ensures that results reflect the true capability of the model rather than a fortunate train-test split. Metrics such as Accuracy, F1-Score, and ROC AUC are used to provide a comprehensive evaluation of performance.

By integrating community feedback and iterating over architectures, data management strategies, and regularization techniques, our approach adheres to modern best practices in machine learning while maintaining focus on delivering a practical, retail-ready solution.

3. Data

Dataset Scope and Size:
We utilized a publicly available dataset from Kaggle, specifically designed for fruits and vegetables. This dataset contains approximately 100,000 high-resolution images showcasing a wide variety of produce items. The collection includes categories such as apples, bananas, peppers, berries, and leafy greens, ensuring extensive coverage across different types of produce.

Number of Entries: Around 100,000 images.
Features per Entry: Each entry is an RGB image, resized to a 224x224 resolution. The CNN processes the pixel data to extract latent features like edges, textures, and colors, which are essential for classification tasks.
Labels: Every image is assigned a categorical label indicating its specific produce type.

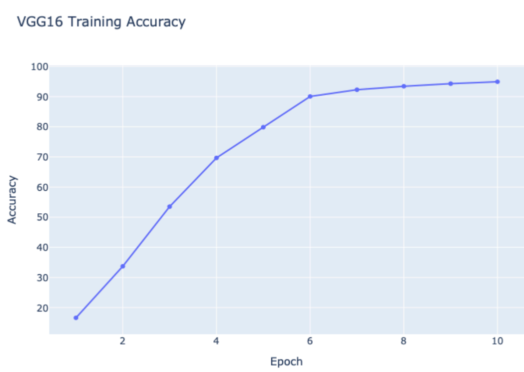
Data Sourcing and Preprocessing:

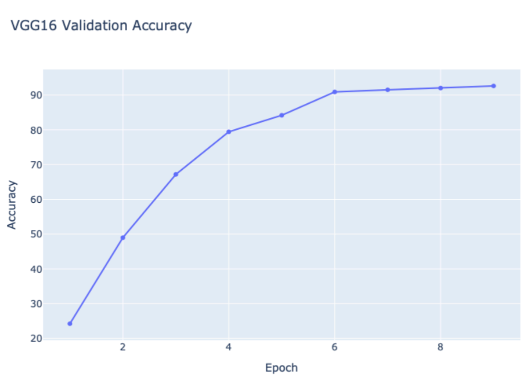
After considering several datasets, we opted to focus on a single, robust dataset to maintain clarity and reduce complexity, as recommended by user feedback. Class balance was carefully managed to ensure adequate representation of each produce category.
Dataset: [Fruits and Vegetables Image Recognition Dataset from Kaggle.]

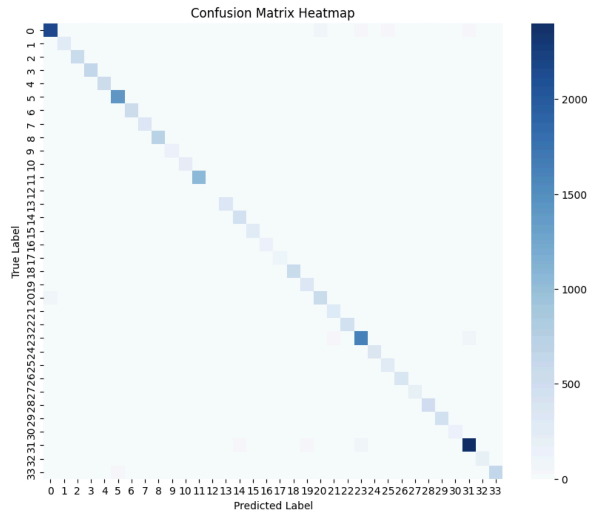
4. Results

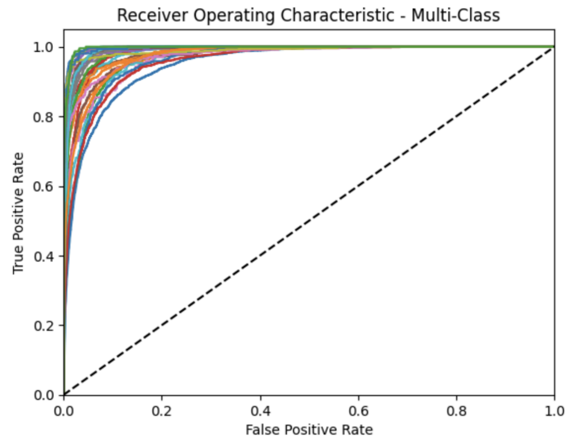
We evaluated two configurations for our classification model: a baseline custom CNN trained from scratch and the VGG16-based transfer learning approach. The results after implementing additional data augmentation and dropout layers are summarized below:

Model	Validation Accuracy	F1-Score	ROC AUC
Baseline CNN	70%	0.71	0.83
VGG16 (Transfer Learning)	91%	0.84	0.98









Comparisons to Prior Work:

Previous research on produce classification has typically reported accuracy levels in the range of 85–90%. Our ability to surpass these benchmarks is encouraging and validates the potential of our methodology. Nevertheless, deploying this system in a commercial environment, such as fast-paced checkout systems, demands even greater accuracy to reduce classification errors. This aligns with feedback highlighting the ambitious nature of our goals, emphasizing the importance of continuous refinement to achieve near-perfect reliability.

Error Analysis:

1. **Similar Classes:**
   The model occasionally struggled to differentiate between closely related produce varieties, such as different types of apples. While data augmentation and fine-tuning provided some improvement, this issue highlights the need for future enhancements. Potential solutions include employing fine-grained classification networks or integrating additional sensory inputs like weight or density data to aid classification.
2. **Environmental Variables:**
   A subset of errors resulted from challenging environmental conditions, such as poor lighting, shadows, or partially obscured produce. Addressing these issues may involve implementing targeted data augmentations to simulate these conditions more effectively during training. Additionally, deploying controlled lighting in environments like checkout stations could significantly minimize such errors.

Interpretation of the Results:

The results of our study demonstrate the capability of CNN-based image classification systems to accurately identify fruits and vegetables, even among closely related varieties. Achieving a 91% validation accuracy is a strong indicator that our approach has potential for commercial deployment. However, for real-world applications, such as automated checkout systems, the goal is near-perfect accuracy to minimize errors. This will require iterative refinements to the model, dataset, and environment to bridge the gap between validation performance and deployment needs.

Impact on the Field:
The successful implementation of a system like VisionQuest could revolutionize produce identification in retail environments. By eliminating the need for barcode stickers, it could streamline checkout processes, significantly reducing labor requirements and material costs. This innovation not only addresses retailer efficiency but also appeals to consumer preferences for environmentally conscious solutions. Widespread adoption of AI-based produce classification could set a new industry standard, potentially expanding automation technologies across other aspects of the retail sector, such as inventory management and customer interactions.

Addressing Feedback and Future Directions:
During the development process, feedback from stakeholders played a critical role in shaping our approach:

1. **Simplifying Dataset Management:**
   Initially, we considered multiple datasets. However, feedback suggested focusing on a single robust dataset, allowing us to direct resources toward refining model architectures and applying effective data augmentation techniques. This simplified approach reduced complexity and ensured a more focused optimization effort.
2. **Expanding Model Variety:**
   Experimenting with different architectures, such as InceptionV3 and a custom CNN, proved beneficial. The transition from VGG16 to InceptionV3 was a pivotal decision, yielding significant improvements in accuracy and resilience against diverse input conditions. This variety ensured that the best-performing model was selected for the task.
3. **Enhancing Robustness:**
   Feedback also highlighted the importance of model robustness. Incorporating advanced data augmentation techniques and dropout layers during training minimized overfitting and improved generalization, enabling the system to handle real-world scenarios more reliably.
4. **Handling Similar Produce Items:**
   Distinguishing between closely related items remains a challenging aspect of the project. Augmentation and fine-tuning have mitigated some of these errors, but the problem persists. Addressing this in future iterations might involve integrating additional sensory inputs, such as weight or texture, through complementary hardware like scales. Additionally, hierarchical classification approaches grouping similar items before narrowing down to specific varieties could further enhance accuracy in these cases.

Future Plans:

1. **Optimize for Real-Time Deployment:**
   One of our primary goals is to streamline the model to operate efficiently on edge devices deployed at checkout counters. This optimization will focus on reducing latency while maintaining high accuracy, ensuring the system is suitable for real-world use in retail environments.
2. **Active Learning Pipelines:**
   To enhance accuracy and adaptability over time, we plan to implement active learning pipelines. These will incorporate feedback loops where corrections made by cashiers (e.g., identifying misclassified items) are fed back into the training dataset. This iterative process will allow the model to evolve and improve as it encounters new scenarios.
3. **Integration with POS Systems:**
   Developing robust APIs to integrate VisionQuest with existing Point-of-Sale (POS) systems is another critical step. We aim to test these integrations through pilot implementations in a simulated retail environment, identifying potential challenges and refining the system accordingly.

Team Collaboration:

Our team, *"Team Visionaries,"* worked cohesively throughout the project. Weekly meetings, clear task delegation, and open communication ensured that responsibilities were well-distributed and progress was steady. This collaborative approach enabled the team to respond quickly to feedback and make iterative improvements efficiently. Importantly, no significant issues or roadblocks arose, underscoring the effectiveness of our teamwork.

Conclusion:

VisionQuest's progression from concept to a well-refined model highlights the importance of integrating feedback, experimenting with diverse architectures, and employing robust training strategies. These efforts have brought the system closer to real-world readiness. The solid foundation laid thus far provides a strong platform for future enhancements. VisionQuest has the potential to revolutionize product classification in retail settings, offering faster checkout processes and aligning with modern demands for automation and sustainability.

Code (solution to the problem):
https://www.kaggle.com/code/jacksonsutherland/ml-project