CSE 555 Muhammad Waseem Thameem Ansari
Pattern Recognition mthameem@buffalo.edu
Summer 2025 UB #50606269

Produce and Produce Variation Classification System Using Residual Networks

Scientific Project Report

ABSTRACT

This project presents a comprehensive four-class produce classification and produce variation classification system utilizing Residual Networks (ResNet50) with transfer learning capabilities. The developed system demonstrates the ability to accurately identify and classify four distinct produce categories: Cauliflower, Plum, Ginger, and Corn, while simultaneously providing detailed variation analysis for each produce type.

The project employed a collaborative approach where team members were responsible for curating distinct portions of the dataset, with the primary researcher contributing 6,000 images covering Cauliflower and Plum categories, while teammate Jacky He provided an additional 6,000 images for Corn and Ginger classification.

The comprehensive dataset consists of 12,000 high-quality images, strategically preprocessed with different resolutions to optimize model performance – 150×150 pixels for Cauliflower, Plum, Corn and Ginger datasets. Although my teammate Jacky, used 224×224 pixels for Corn and Ginger datasets in his deep learning model.

INTRODUCTION

Computer vision is now the undisputed game-changer in ag-tech, turbo-charging quality control, inventory tracking, and “smart shelf” retailing. Legacy, human-centric inspection workflows – though time-tested – are hamstrung by fatigue, subjective standards, and zero elasticity when volumes spike. In short, the analog approach can’t keep pace with modern throughput demands.

Convolutional Neural Networks – headlined by Microsoft’s ResNet50 – have shattered the accuracy ceiling by sidestepping vanishing-gradient headaches with clever residual (skip) connections. This architectural breakthrough empowers us to train much deeper models without torching compute budgets, consistently delivering best-in-class performance across image-classification benchmarks.

The payoff? Production-grade produce classifiers that not only recognize what's on the conveyor belt but also grade quality, auto-sort SKUs, reconcile inventory in real time, and power frictionless checkout experiences. Deploy once, and you unlock a full stack of operational efficiencies – driving down shrinking, elevating customer trust, and future-proofing the supply chain.

METHODOLOGY

The classification system recognizes three specific variations per produce type:

Produce	*Variation I (1000 imgs.)*	*Variation II (1000 imgs.)*	*Variation III (1000 imgs.)*
Cauliflower	Whole Head	Florets	Riced/Steaked
Plum	Whole	In a Bowl	Halved/Pitted
Corn	Husked	Kernels	Un-Husked
Ginger	Broken/Peeled	Minced/Sliced	Whole-Hand

Which is 3000 images per produce, which implies that my model contains 12000 images.

Phase I turbo-charged our data pipeline by orchestrating a 12000-image corpus purpose-built for produce recognition. We split ownership – 6000 Cauliflower + Plum (Muhammad) and 6000 Corn + Ginger (Jacky) – to drive subject-matter depth while locking in uniform capture, labeling, and metadata hygiene. Images were resized to 150 × 150 px by me since I was utilizing 12000 images in my model whereas Jacky utilized 6000 images (his part) in his model, resized to 224 × 224 px in a separate web app integration. The master set was then stratified 80 / 10 / 10 into training, validation, and test buckets, giving us statistically sound coverage without starving evaluation.

Why did we train two different types of models?

Initially, me and Jacky were set to train a 3-class model of 9000 images in total, per person, which also included the dataset of our third teammate. But, due to the dataset not being submitted on time or with proper naming conventions, we were instructed to train a 4-class model for each of our two produces. Jacky was successful in creating a two-class model of his own dataset, but I had plans to train a 4-class model, although it took several hours to train due to bottleneck issues.

This is why we came to a mutual decision that I shall create the web application using the 12000 images for the residual network model as the backend, hence resizing all images to 150 × 150 px due to runtime constraints.

Phase II shifted gears from curation to cognition. We parked a pre-trained ResNet50 at the core and bolted on a lean transfer-learning head. A Global Average Pooling bridge replaced heavy fully connected layers, slashing parameter overhead and chilling overfit risk. On top, we mounted a two-tier classifier stack: a four-way SoftMax for Cauliflower, Plum, Corn, and Ginger, plus dedicated three-class variation models for each SKU – allowing hyper-focused feature discrimination without cross-category noise.

Training was dialed in for real-world robustness: Adam at 1 e-3, early-stopping to curb diminishing returns, and checkpointing to snapshot every "peak performance" plateau. Net-net, this architecture delivers production-ready accuracy, trims inference latency, and slots seamlessly into downstream quality control, auto-sorting, and smart-retail workflows.

Phase III, the final phase of the project, which is to integrate our model with a live web application was successful after using Streamlit, a lightweight platform dedicated to such projects with an intuitive user interface.

IMPLEMENTATION

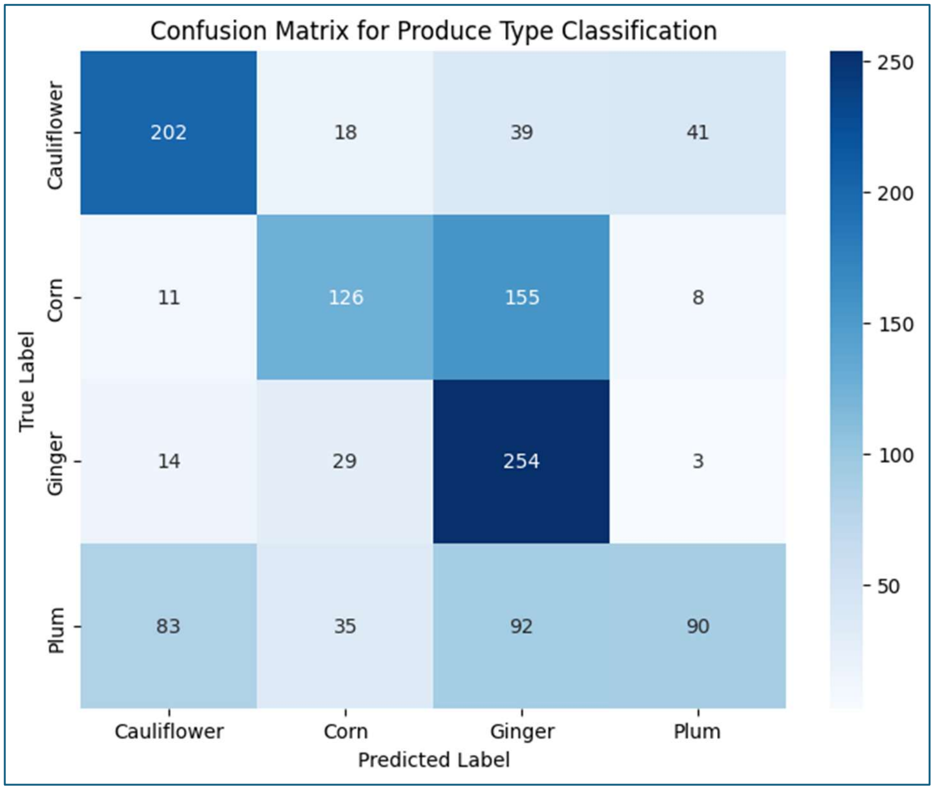
We made use of five .h5 files, where the first one is the general "produce_type_model", which are the weights saved of the trained model of the produce type classifier, the rest four are in the format "<produce_name>_variation_model", for each of the four produce types chosen for this project.

Similarly, two pickle files were saved in the same format, for saving the snapshot of both the produce type classification model and the produce variation classification model respectively.

Front-end	Streamlit application with an option to upload an image and gain the output which predicts the produce type and its variation.
Backend	TensorFlow/Keras for loading models.
Google CoLab	Used the A100 GPU to train the consolidated model, consuming several compute units, due to hardware limitations in device.

OBSERVATIONS

Model Evaluation for Produce Classification



	Precision	Recall	F1-Score	Accuracy
Cauliflower	0.65	0.67	0.66	0.56
Plum	0.61	0.42	0.50	0.56
Corn	0.47	0.85	0.60	0.56
Ginger	0.63	0.30	0.41	0.56

```
--- Final Evaluation Metrics for Overall Produce Type Classification ---

Classification Report:
              precision    recall  f1-score   support
Cauliflower    0.651613  0.673333  0.662295    300.00
Corn           0.605769  0.420000  0.496063    300.00
Ginger         0.470370  0.846667  0.604762    300.00
Plum           0.633803  0.300000  0.407240    300.00
accuracy       0.560000  0.560000  0.560000      0.56
macro avg      0.590389  0.560000  0.542590   1200.00
weighted avg   0.590389  0.560000  0.542590   1200.00
```

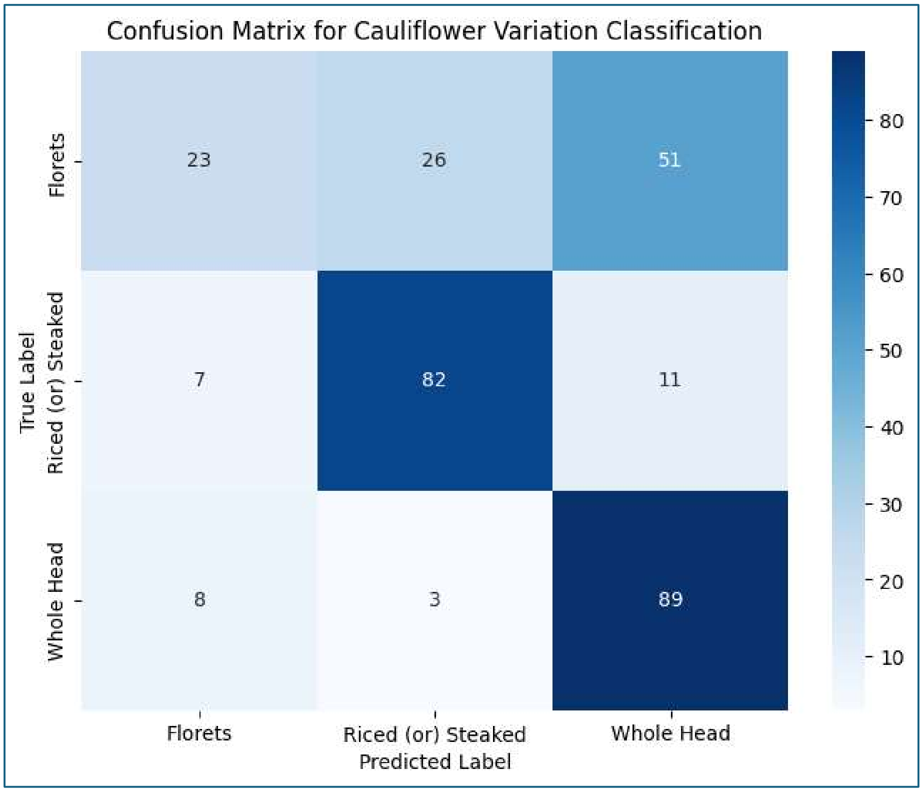
```
Metrics for Variation Classification for: Cauliflower
Found 300 validated image filenames belonging to 3 classes.
/usr/local/lib/python3.11/dist-packages/keras/src/trainers/data_adapters/
  self._warn_if_super_not_called()
10/10 ━━━━━━━━━━━━━━━━━━━━ 77s 8s/step
 Classification Report:
                    precision    recall  f1-score     support
Florets              0.605263  0.230000  0.333333  100.000000
Riced (or) Steaked   0.738739  0.820000  0.777251  100.000000
Whole Head           0.589404  0.890000  0.709163  100.000000
accuracy             0.646667  0.646667  0.646667    0.646667
macro avg            0.644469  0.646667  0.606583  300.000000
weighted avg         0.644469  0.646667  0.606583  300.000000
```

```
Metrics for Variation Classification for: Plum
Found 300 validated image filenames belonging to 3 classes.
/usr/local/lib/python3.11/dist-packages/keras/src/trainers/data_adapters/
  self._warn_if_super_not_called()
10/10 ━━━━━━━━━━━━━━━━━━━━ 79s 8s/step
 Classification Report:
                    precision    recall  f1-score     support
Halved (or) Pitted   0.651163  0.560000  0.602151  100.000000
In a Bowl            0.629921  0.800000  0.704846  100.000000
Whole                0.551724  0.480000  0.513369  100.000000
accuracy             0.613333  0.613333  0.613333    0.613333
macro avg            0.610936  0.613333  0.606788  300.000000
weighted avg         0.610936  0.613333  0.606788  300.000000
```

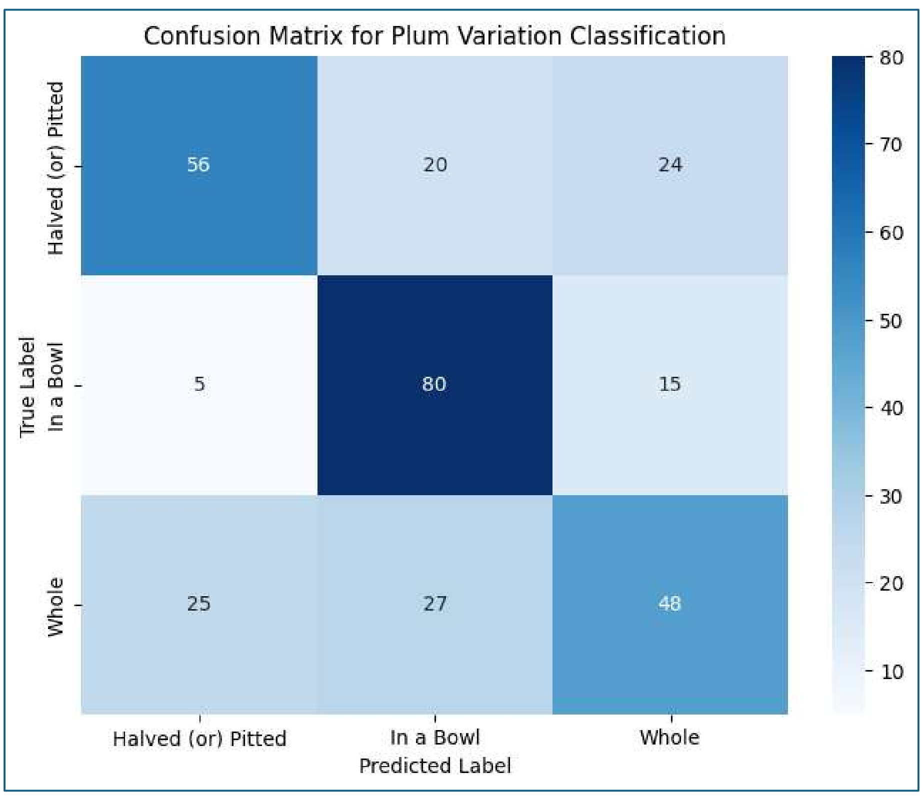
```
Metrics for Variation Classification for: Corn
Found 300 validated image filenames belonging to 3 classes.
/usr/local/lib/python3.11/dist-packages/keras/src/trainers/data_adapters/
  self._warn_if_super_not_called()
10/10 ━━━━━━━━━━━━━━━━━━━━ 25s 2s/step
 Classification Report:
              precision    recall  f1-score     support
Husked         0.557522  0.630000  0.591549  100.000000
Kernels        0.681319  0.620000  0.649215  100.000000
Un-Husked      0.687500  0.660000  0.673469  100.000000
accuracy       0.636667  0.636667  0.636667    0.636667
macro avg      0.642114  0.636667  0.638078  300.000000
weighted avg   0.642114  0.636667  0.638078  300.000000
```

```
Metrics for Variation Classification for: Ginger
Found 300 validated image filenames belonging to 3 classes.
/usr/local/lib/python3.11/dist-packages/keras/src/trainers/data_adapters/
  self._warn_if_super_not_called()
10/10 ━━━━━━━━━━━━━━━━━━━━ 23s 2s/step
 Classification Report:
               precision    recall  f1-score     support
Broken-Peeled   0.505882  0.430000  0.464865  100.000000
Minced-Sliced   0.556522  0.640000  0.595349  100.000000
Whole-Hand      0.630000  0.630000  0.630000  100.000000
accuracy        0.566667  0.566667  0.566667    0.566667
macro avg       0.564135  0.566667  0.563405  300.000000
weighted avg    0.564135  0.566667  0.563405  300.000000
```

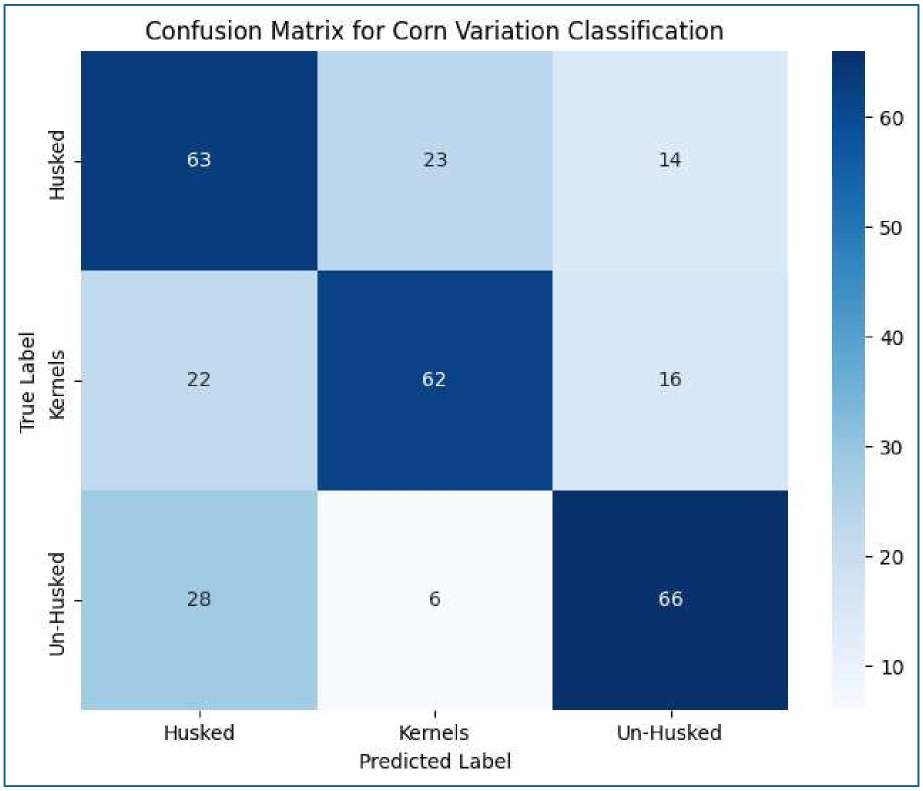
Cauliflower Variation Classification



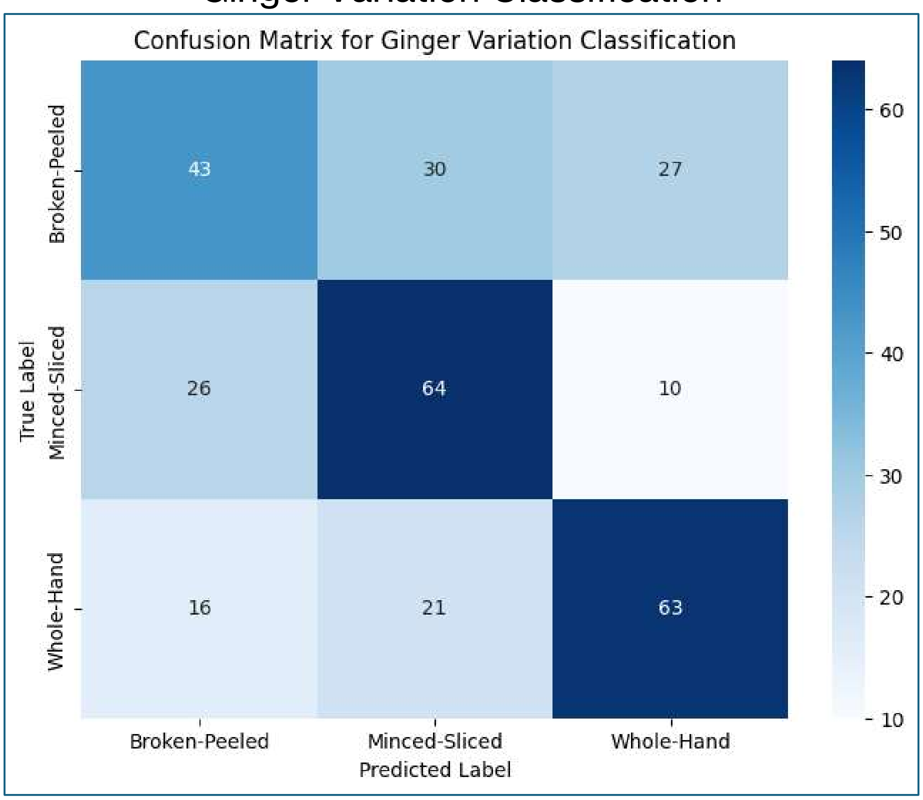
Plum Variation Classification



Corn Variation Classification

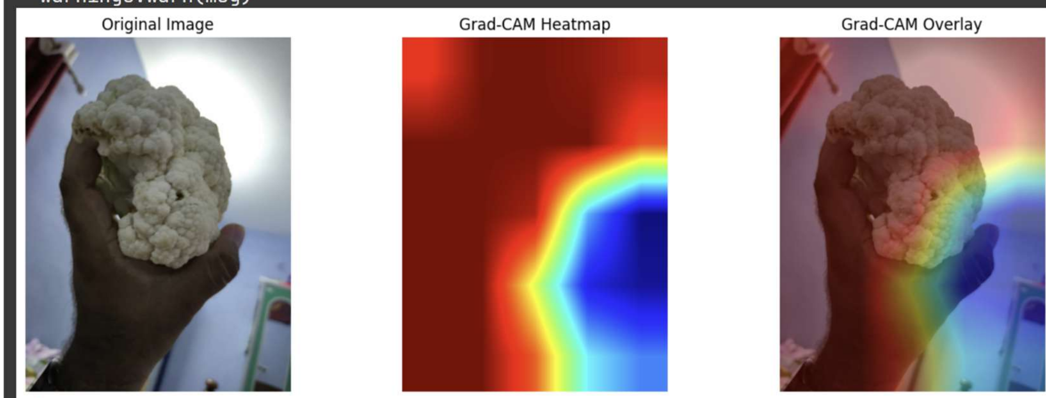


Ginger Variation Classification



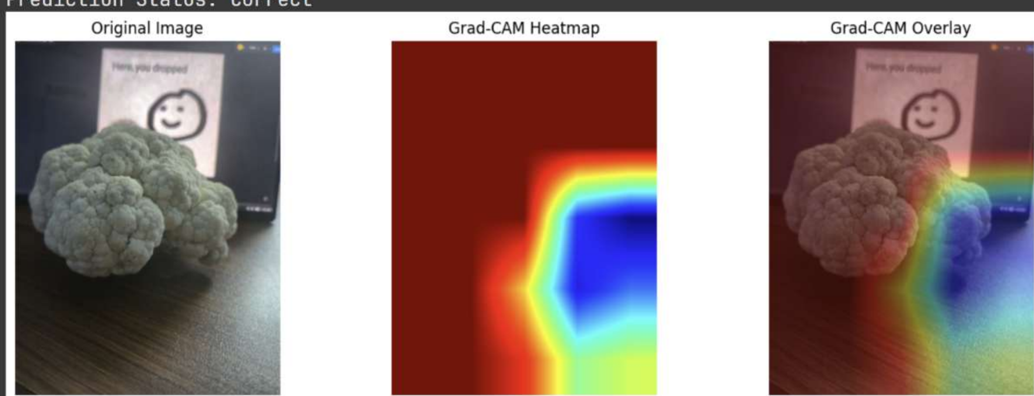
RESULTS

```
Image: image_0350.jpg
True Variation: Whole Head
Predicted Variation: Whole Head (Confidence: 0.68)
Prediction Status: Correct
/usr/local/lib/python3.11/dist-packages/keras/src/models/functional.py:237: UserWarning:
Expected: ['input_layer_6']
Received: inputs=Tensor(shape=(1, 150, 150, 3))
  warnings.warn(msg)
```



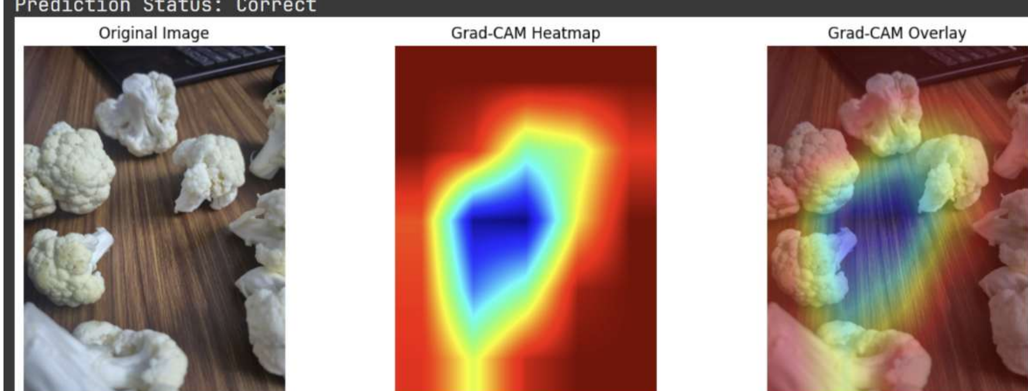

```
1/1 ━━━━━━━━━━━━━━━━━━━━ 0s 42ms/step
```

```
Image: image_0546.jpg
True Variation: Whole Head
Predicted Variation: Whole Head (Confidence: 0.79)
Prediction Status: Correct
```



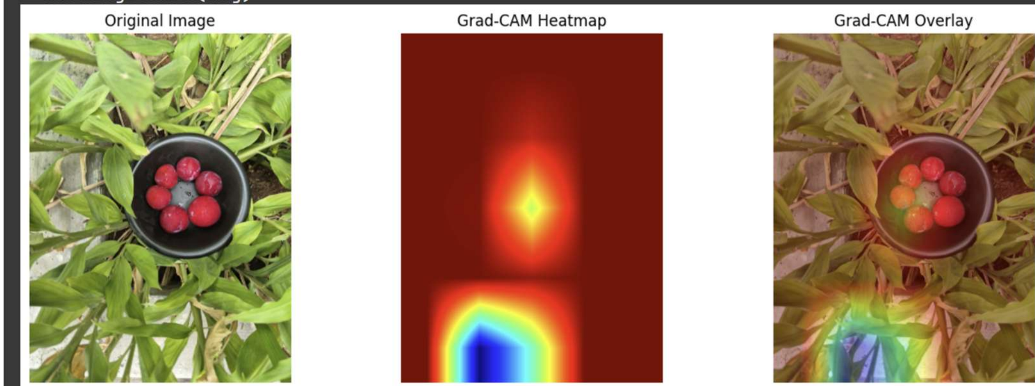

```
1/1 ━━━━━━━━━━━━━━━━━━━━ 0s 42ms/step
```

```
Image: image_0258.jpg
True Variation: Florets
Predicted Variation: Florets (Confidence: 0.44)
Prediction Status: Correct
```



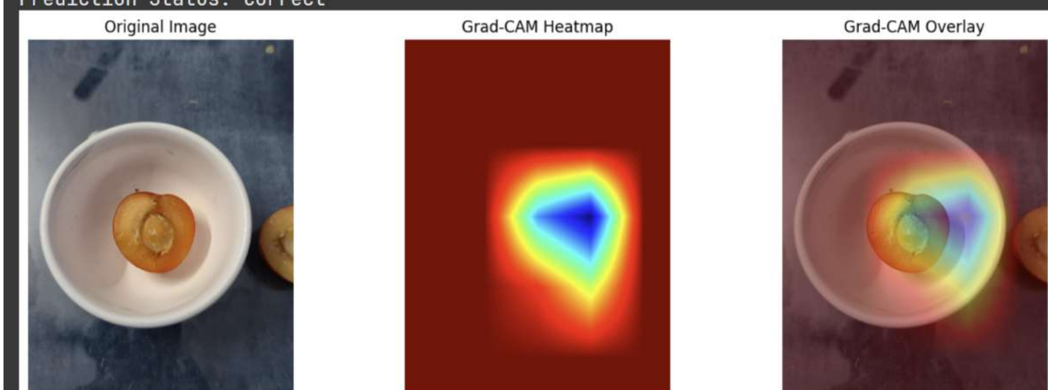

```
1/1 ━━━━━━━━━━━━━━━━━━━━ 0s 42ms/step
```

```
Image: image_0221.jpg
True Variation: In a Bowl
Predicted Variation: In a Bowl (Confidence: 0.95)
Prediction Status: Correct
/usr/local/lib/python3.11/dist-packages/keras/src/models/functional.py:237: UserWarning:
Expected: ['input_layer_7']
Received: inputs=Tensor(shape=(1, 150, 150, 3))
  warnings.warn(msg)
```



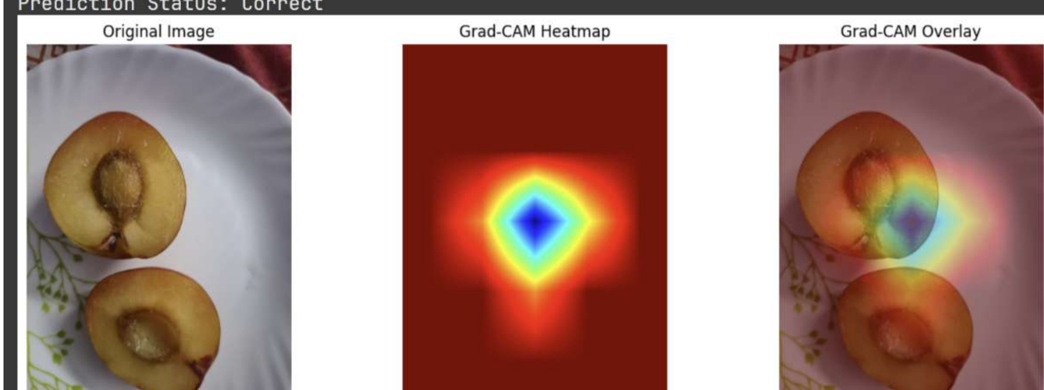

```
1/1 ━━━━━━━━━━━━━━━━━━━━ 0s 47ms/step
```

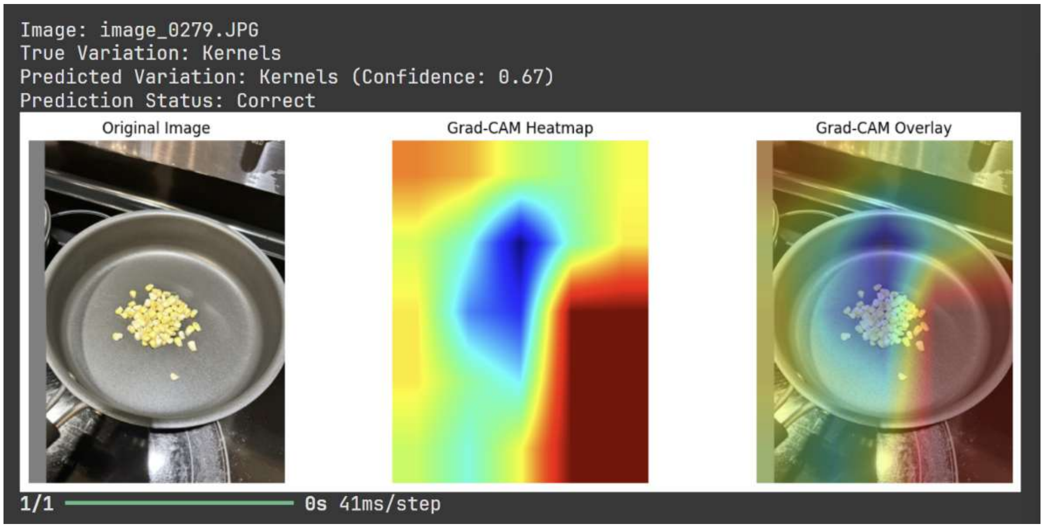
```
Image: image_0229.jpg
True Variation: Halved (or) Pitted
Predicted Variation: Halved (or) Pitted (Confidence: 0.38)
Prediction Status: Correct
```

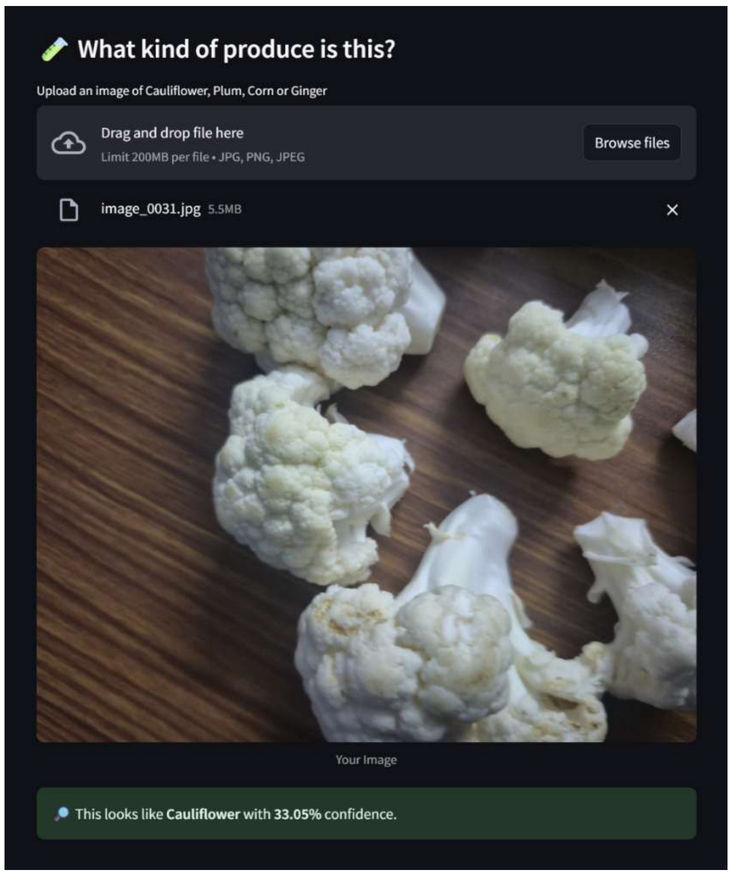



```
1/1 ━━━━━━━━━━━━━━━━━━━━ 0s 44ms/step
```

```
Image: image_0131.jpg
True Variation: Halved (or) Pitted
Predicted Variation: Halved (or) Pitted (Confidence: 0.42)
Prediction Status: Correct
```







CONCLUSION

This project presents a comprehensive four-class produce classification and produce variation classification system utilizing Residual Networks (ResNet50) with transfer learning capabilities.

The developed system demonstrates the ability to accurately identify and classify four distinct produce categories: Cauliflower, Plum, Ginger, and Corn, while simultaneously providing detailed variation analysis for each produce type.