

smart sorting.ipynb

File Edit View Insert Runtime Tools Help

Commands + Code + Text Run all Connecting

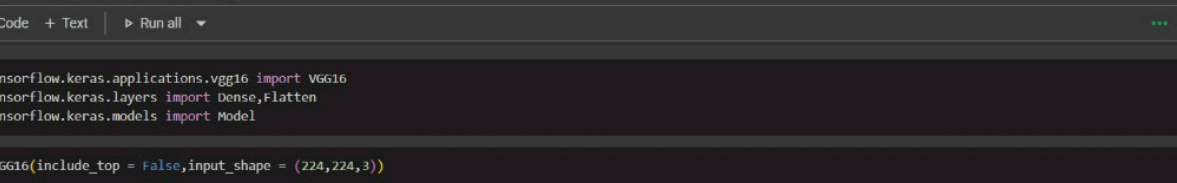
```
test_dir = os.path.join(dataset_dir, 'test')

# Define ImageDataGenerator for training with augmentation
train_datagen = ImageDataGenerator(
    rescale=1./255,
    rotation_range=20,
    width_shift_range=0.2,
    height_shift_range=0.2,
    shear_range=0.2,
    zoom_range=0.2,
    horizontal_flip=True,
    fill_mode='nearest'
)

# Define ImageDataGenerator for validation and testing (no augmentation)
val_test_datagen = ImageDataGenerator(rescale=1./255)

# Create generators
train_generator = train_datagen.flow_from_directory(
    train_dir,
    target_size=IMG_SIZE,
    batch_size=32,
    class_mode='categorical'
)

val_generator = val_test_datagen.flow_from_directory(
    val_dir,
    target_size=IMG_SIZE,
    batch_size=32,
    class_mode='categorical'
)
```



```
[ ] from tensorflow.keras.applications.vgg16 import VGG16
from tensorflow.keras.layers import Dense, Flatten
from tensorflow.keras.models import Model

[ ] vgg = VGG16(include_top = False, input_shape = (224,224,3))

Downloading data from https://storage.googleapis.com/tensorflow/keras-applications/vgg16/vgg16_weights_tf_dim_ordering_tf_kernels_notop.h5
58889256/58889256 0s 0us/step

for layer in vgg.layers:
    print(layer)

<InputLayer name=input_layer, built=True>
<Conv2D name=block1_conv1, built=True>
<Conv2D name=block1_conv2, built=True>
<MaxPooling2D name=block1_pool, built=True>
<Conv2D name=block2_conv1, built=True>
<Conv2D name=block2_conv2, built=True>
<MaxPooling2D name=block2_pool, built=True>
<Conv2D name=block3_conv1, built=True>
<Conv2D name=block3_conv2, built=True>
<Conv2D name=block3_conv3, built=True>
<MaxPooling2D name=block3_pool, built=True>
<Conv2D name=block4_conv1, built=True>
<Conv2D name=block4_conv2, built=True>
<Conv2D name=block4_conv3, built=True>
<MaxPooling2D name=block4_pool, built=True>
<Conv2D name=block5_conv1, built=True>
<Conv2D name=block5_conv2, built=True>
<Conv2D name=block5_conv3, built=True>
```

smart sorting.ipynb

File Edit View Insert Runtime Tools Help

Commands + Code + Text Run all

Connecting

Share Gemini M

import os
import shutil
from sklearn.model_selection import train_test_split
from tensorflow.keras.preprocessing.image import ImageDataGenerator, load_img, img_to_array
from tensorflow.keras.applications.vgg16 import VGG16, preprocess_input
from tensorflow.keras.layers import Dense, Flatten
from tensorflow.keras.models import Model
from tensorflow.keras.preprocessing import image
from tensorflow.keras.optimizers import Adam

mkdir ~/.kaggle

[] !ls

sample_data

[] from google.colab import files
files.upload()

Choose Files No file chosen Upload widget is only available when the cell has been executed in the current browser session. Please rerun this cell to enable.
Saving kaggle.json to kaggle.json
{'kaggle.json': b'{"username": "manthiniresmasri", "key": "c2c1f9ad92242f24e033af090e72ab94"}'}

[] !ls

kaggle.json sample_data

Variables Terminal

Resuming execution

smart sorting.ipynb

File Edit View Insert Runtime Tools Help

Commands + Code + Text ▶ Run all

Connecting

Share Gemini M

import os

os.listdir("/content")

[

['.config',

'fresh_tomato',

'fresh_banana',

'fresh_orange',

'fresh_bitter_gourd',

'stale_apple',

'stale_banana',

'fresh_capsicum',

'stale_tomato',

'fresh_apple',

'stale_orange',

'stale_bitter_gourd',

'fresh-and-stale-images-of-fruits-and-vegetables.zip',

'kaggle.json',

'stale_capsicum',

'imageLabels.txt',

'sample_data']

]

[]

import os

import shutil

import numpy as np

from sklearn.model_selection import train_test_split

Set the path to the dataset

dataset_dir = '/content'

Valid image extensions

valid_extensions = ('.jpg', '.jpeg', '.png', '.bmp')

Variables Terminal

Resuming execution

smart sorting.ipynb

File Edit View Insert Runtime Tools Help

Commands + Code + Text Run all

Connecting


Share Gemini M

```
[ ] # Specify the correct path to your image folder
    folder_path = '/content/output_dataset/train/stale_tomato' # + Added leading slash

    # list all image files
    image_files = [f for f in os.listdir(folder_path) if f.endswith((''.jpg', '.png', '.jpeg'))]

    # Select a random image from the list
    selected_image = random.choice(image_files)

    # Display the randomly selected image
    image_path = os.path.join(folder_path, selected_image)
    display(Image(filename=image_path))
```



Variables Terminal

Resuming execution

smart sorting.ipynb

File Edit View Insert Runtime Tools Help

Commands + Code + Text Run all

Connecting

block2_conv2 (Conv2D)	(None, 112, 112, 128)	147,584
block2_pool (MaxPooling2D)	(None, 56, 56, 128)	0
block3_conv1 (Conv2D)	(None, 56, 56, 256)	295,168
block3_conv2 (Conv2D)	(None, 56, 56, 256)	590,880
block3_conv3 (Conv2D)	(None, 56, 56, 256)	590,880
block3_pool (MaxPooling2D)	(None, 28, 28, 256)	0
block4_conv1 (Conv2D)	(None, 28, 28, 512)	1,180,160
block4_conv2 (Conv2D)	(None, 28, 28, 512)	2,359,808
block4_conv3 (Conv2D)	(None, 28, 28, 512)	2,359,808
block4_pool (MaxPooling2D)	(None, 14, 14, 512)	0
block5_conv1 (Conv2D)	(None, 14, 14, 512)	2,359,808
block5_conv2 (Conv2D)	(None, 14, 14, 512)	2,359,808
block5_conv3 (Conv2D)	(None, 14, 14, 512)	2,359,808
block5_pool (MaxPooling2D)	(None, 7, 7, 512)	0
flatten (Flatten)	(None, 25088)	0
dense (Dense)	(None, 1)	393,068

Total params: 15,015,756 (57.28 MB)

Trainable params: 393,068 (1.15 MB)

Non-trainable params: 14,714,688 (56.13 MB)

smart sorting.ipynb

File Edit View Insert Runtime Tools Help

Commands + Code + Text Run all Connecting

<Conv2D name=block5_conv3, built=True>
<MaxPooling2D name=block5_pool, built=True>

[] len(vgg.layers)

[] 19

[] for layer in vgg.layers:
 layer.trainable = False

[] x = Flatten()(vgg.output)

[] output = Dense(train_generator.num_classes, activation='softmax')(x)

[] vgg16 = Model(vgg.input,output)

[] vgg16.summary()

Model: "functional"

Layer (type)	Output Shape	Param #
input_layer (InputLayer)	(None, 224, 224, 3)	0
block1_conv1 (Conv2D)	(None, 224, 224, 64)	1,792
block1_conv2 (Conv2D)	(None, 224, 224, 64)	36,828
block1_pool (MaxPooling2D)	(None, 112, 112, 64)	0

smart sorting.ipynb

File Edit View Insert Runtime Tools Help

Commands + Code + Text Run all

Connecting

```
optimizer=opt,
loss='categorical_crossentropy',
metrics=['accuracy']
)

# Train the model - assumes train_generator and val_generator are defined
history = vgg16.fit(
    train_generator,
    validation_data=val_generator,
    epochs=15,
    steps_per_epoch=len(train_generator), # or 20 if you prefer
    validation_steps=len(val_generator),
    callbacks=[early_stopping]
)
```

```
... /usr/local/lib/python3.11/dist-packages/keras/src/trainers/data_adapters/py_dataset_adapter.py:121: UserWarning: Your `PyDataset` class should call `super().__init__(**kwargs)` in its
self._warn_if_super_not_called()
Epoch 1/15
45/45 1276s 28s/step - accuracy: 0.1886 - loss: 2.3673 - val_accuracy: 0.6187 - val_loss: 1.4568
Epoch 2/15
45/45 1242s 28s/step - accuracy: 0.5425 - loss: 1.5490 - val_accuracy: 0.6729 - val_loss: 1.0714
Epoch 3/15
45/45 1294s 28s/step - accuracy: 0.6384 - loss: 1.2319 - val_accuracy: 0.8042 - val_loss: 0.8272
Epoch 4/15
45/45 1203s 27s/step - accuracy: 0.7316 - loss: 1.0068 - val_accuracy: 0.8542 - val_loss: 0.6954
Epoch 5/15
45/45 1246s 27s/step - accuracy: 0.7524 - loss: 0.9066 - val_accuracy: 0.8250 - val_loss: 0.6432
Epoch 6/15
45/45 1263s 27s/step - accuracy: 0.7739 - loss: 0.8386 - val_accuracy: 0.8687 - val_loss: 0.5580
Epoch 7/15
45/45 1211s 27s/step - accuracy: 0.8097 - loss: 0.7167 - val_accuracy: 0.8792 - val_loss: 0.4984
Epoch 8/15
```


smart sorting.ipynb

File Edit View Insert Runtime Tools Help

Commands + Code + Text ▶ Run all

Connecting

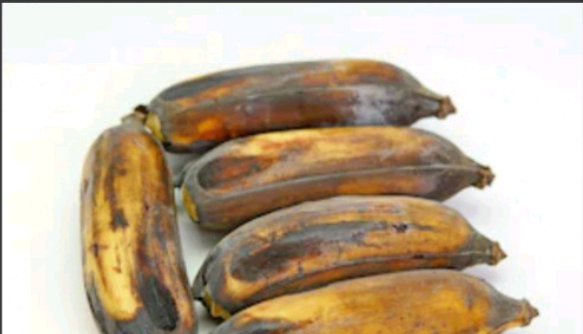
Share Gemini M

```
# Specify the correct path to your image folder
folder_path = '/content/output_dataset/train/stale_banana' # ← Added leading slash

# List all image files
image_files = [f for f in os.listdir(folder_path) if f.endswith(('.jpg', '.png', '.jpeg'))]

# Select a random image from the list
selected_image = random.choice(image_files)

# Display the randomly selected image
image_path = os.path.join(folder_path, selected_image)
display(Image(filename=image_path))
```



Variables Terminal

Resuming execution

smart sorting.ipynb

File Edit View Insert Runtime Tools Help

Commands + Code + Text Run all

Connecting

```
classes = []
for cls in os.listdir(dataset_dir):
    cls_path = os.path.join(dataset_dir, cls)
    if os.path.isdir(cls_path):
        files = os.listdir(cls_path)
        if any(file.lower().endswith(valid_extensions) for file in files):
            classes.append(cls)

# Create the directories for train, val, and test sets
output_dir = 'output dataset'
os.makedirs(output_dir, exist_ok=True)
os.makedirs(os.path.join(output_dir, 'train'), exist_ok=True)
os.makedirs(os.path.join(output_dir, 'val'), exist_ok=True)
os.makedirs(os.path.join(output_dir, 'test'), exist_ok=True)

# Split and copy images
for cls in classes:
    os.makedirs(os.path.join(output_dir, 'train', cls), exist_ok=True)
    os.makedirs(os.path.join(output_dir, 'val', cls), exist_ok=True)
    os.makedirs(os.path.join(output_dir, 'test', cls), exist_ok=True)

    class_dir = os.path.join(dataset_dir, cls)
    images = [img for img in os.listdir(class_dir) if img.lower().endswith(valid_extensions)][1:200]

    print(cls, len(images))

    train_and_val_images, test_images = train_test_split(images, test_size=0.2, random_state=42)
    train_images, val_images = train_test_split(train_and_val_images, test_size=0.25, random_state=42)

    for img in train_images:
        shutil.copy(os.path.join(class_dir, img), os.path.join(output_dir, 'train', cls, img))
```

smart sorting.ipynb

File Edit View Insert Runtime Tools Help

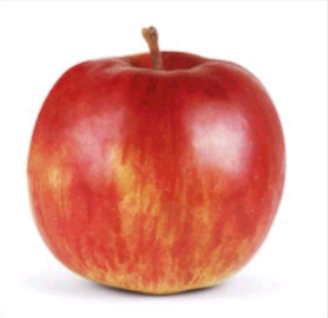
Commands + Code + Text Run all

Connecting

```
# List all image files
image_files = [f for f in os.listdir(folder_path) if f.endswith(('.jpg', '.png', '.jpeg'))]

# Select a random image from the list
selected_image = random.choice(image_files)

# Display the randomly selected image
image_path = os.path.join(folder_path, selected_image)
display(Image(filename=image_path))
```



```
[ ] # Specify the correct path to your image folder
folder_path = 'C:/Users/your_username/Desktop/your_image_folder'
```

smart sorting.ipynb

File Edit View Insert Runtime Tools Help

Commands + Code + Text Run all Connecting

```
# Specify the correct path to your image folder
folder_path = '/content/output_dataset/train/fresh orange' # + Added leading slash

# List all image files
image_files = [f for f in os.listdir(folder_path) if f.endswith((''.jpg', '.png', '.jpeg'))]

# Select a random image from the list
selected_image = random.choice(image_files)

# Display the randomly selected image
image_path = os.path.join(folder_path, selected_image)
display(Image(filename=image_path))
```



colab.research.google.com/drive/1min_srbvKS649D7iINRCR5R4yMPPH0

smart sorting.ipynb

File Edit View Insert Runtime Tools Help

Commands + Code + Text Run all

Connecting

1

kaggle datasets list -s vegetables

ref	title	size	lastUpdated	downloadCount	voteCount
kritikseth/fruit-and-vegetable-image-recognition	Fruits and Vegetables Image Recognition Dataset	2136757290	2022-02-12 06:15:55.417000	48726	444
yashdogra/fnv2024	Fruits and Vegetables	3415	2025-02-14 13:29:21.607000	825	67
everydaycodings/produce-prices-dataset	Fruits and Vegetables Prices Dataset	237295	2023-12-11 13:40:33.147000	5076	61
moltean/fruits	Fruits-360 dataset	3725667254	2025-06-07 16:15:42.817000	191875	3208
ramkrijal/agriculture-vegetables-fruits-time-series-prices	Time Series Price Vegetables and Fruits	1401948	2023-03-05 16:54:21.760000	5171	56
yapwh1208/supermarket-sales-data	Supermarket Sales Data	8881137	2023-09-22 01:06:21.493000	19621	175
raghavrpotdar/fresh-and-stale-images-of-fruits-and-vegetables	Fresh and Stale Images of Fruits and Vegetables	1513322759	2021-05-17 16:20:32.193000	7093	65
anshikakashyap12/fruits-and-vegetables-prices-in-usa	Fruits And Vegetables Prices In USA	3072	2023-10-23 17:19:11.877000	1765	34
muhridinmuxiddinov/fruits-and-vegetables-dataset	Fruits and Vegetables dataset	911969808	2022-12-21 01:55:19.330000	3502	28
sergeynesteruk/packed-fruits-and-vegetables-recognition-benchmark	Packed Fruits and Vegetables Recognition Benchmark	31310524657	2024-08-19 13:17:44.510000	672	18
misraahmed/vegetable-image-dataset	Vegetable Image Dataset	560031432	2021-12-24 13:24:40.333000	33051	218
jorgebailon/fruits-vegetables	Fruits & Vegetables	586784620	2020-08-18 00:52:11.743000	1688	21
abhisheksubhashswami/fruits-and-vegetables	Fruits and Vegetables	571642037	2023-09-19 07:16:28.243000	893	16
ysthehurricane/vegetables-cultivation-data-exclusive	Vegetables Cultivation Data Exclusive	3042	2021-08-20 14:12:31.330000	564	45
shadikfaysal/fruit-and-vegetables-ssm	Fruit and Vegetables SSM	3571450159	2021-11-17 05:59:55.930000	634	15
nehaprabhavalkar/indian-food-101	Indian Food 101	6951	2020-09-30 06:23:43.063000	34310	607
rudraprasadbhuyan/vegetables-dataset	Vegetables Dataset	7872	2024-09-05 13:54:01.560000	303	6
swoyam2609/fresh-and-stale-classification	Fresh and Rotten Classification	3053594823	2023-06-07 07:01:15.067000	7797	65
utkarshsaxenadn/vegetable-classifier-acc-99.87	Vegetable Classifier ResNet50V2 Acc : 99.87%	93689165	2022-10-21 01:29:02.183000	1151	45
gpreda/daily-consumption-of-fruit-vegetables-in-europe	Daily consumption of fruit & vegetables in Europe	327286	2021-12-12 11:00:48.493000	1037	36

2

kaggle datasets download -d raghavrpotdar/fresh-and-stale-images-of-fruits-and-vegetables

Dataset URL: <https://www.kaggle.com/datasets/raghavrpotdar/fresh-and-stale-images-of-fruits-and-vegetables>

License(s): CC0-1.0


Downloading fresh-and-stale-images-of-fruits-and-vegetables.zip to /content

99% 1.39G/1.41G [00:13<00:00, 122MB/s]

Variables

Terminal


Resuming execution

 smart sorting.ipynb ☆ ☁

FileEditViewInsertRuntimeToolsHelp

🔍 Commands+ Code + Text ▶ Run all ▼

... Connecting ▾ ^



```
from keras.callbacks import EarlyStopping
from keras.optimizers import Adam

# Define optimizer
opt = Adam(learning_rate=0.0001)

# Early stopping to prevent overfitting
early_stopping = EarlyStopping(
    monitor='val_accuracy',
    patience=3,
    restore_best_weights=True
)

# Compile the model - assuming 'vgg16' has a softmax output layer for multi-class
vgg16.compile(
    optimizer=opt,
    loss='categorical_crossentropy',
    metrics=['accuracy']
)

# Train the model - assumes train_generator and val_generator are defined
history = vgg16.fit(
    train_generator,
    validation_data=val_generator,
    epochs=15,
    steps_per_epoch=len(train_generator), # or 20 if you prefer
    validation_steps=len(val_generator),
    callbacks=[early_stopping]
)
```

⌵ VariablesTerminal

Resuming execution

smart sorting.ipynb

File Edit View Insert Runtime Tools Help

Share Gemini M

Commands + Code + Text Run all Connecting

[]

!kaggle datasets download -d raghavrpotdar/fresh-and-stale-images-of-fruits-and-vegetables

Dataset URL: <https://www.kaggle.com/datasets/raghavrpotdar/fresh-and-stale-images-of-fruits-and-vegetables>

License(s): CC0 1.0

Downloading fresh-and-stale-images-of-fruits-and-vegetables.zip to /content

99% 1.39G/1.41G [00:13<00:00, 122MB/s]

100% 1.41G/1.41G [00:13<00:00, 110MB/s]

!unzip fresh-and-stale-images-of-fruits-and-vegetables.zip -d /content/

inflating: /content/stale_banana/rotated_by_15_Screen_Shot_2018-06-12_at_9.20.20_PM.png

inflating: /content/stale_banana/rotated_by_15_Screen_Shot_2018-06-12_at_9.20.31_PM.png

inflating: /content/stale_banana/rotated_by_15_Screen_Shot_2018-06-12_at_9.20.47_PM.png

inflating: /content/stale_banana/rotated_by_15_Screen_Shot_2018-06-12_at_9.20.52_PM.png

inflating: /content/stale_banana/rotated_by_15_Screen_Shot_2018-06-12_at_9.20.56_PM.png

inflating: /content/stale_banana/rotated_by_15_Screen_Shot_2018-06-12_at_9.21.00_PM.png

inflating: /content/stale_banana/rotated_by_15_Screen_Shot_2018-06-12_at_9.21.10_PM.png

inflating: /content/stale_banana/rotated_by_15_Screen_Shot_2018-06-12_at_9.21.14_PM.png

inflating: /content/stale_banana/rotated_by_15_Screen_Shot_2018-06-12_at_9.21.20_PM.png

inflating: /content/stale_banana/rotated_by_15_Screen_Shot_2018-06-12_at_9.21.25_PM.png

inflating: /content/stale_banana/rotated_by_15_Screen_Shot_2018-06-12_at_9.22.13_PM.png

inflating: /content/stale_banana/rotated_by_15_Screen_Shot_2018-06-12_at_9.22.25_PM.png

inflating: /content/stale_banana/rotated_by_15_Screen_Shot_2018-06-12_at_9.22.32_PM.png

inflating: /content/stale_banana/rotated_by_15_Screen_Shot_2018-06-12_at_9.22.37_PM.png

inflating: /content/stale_banana/rotated_by_15_Screen_Shot_2018-06-12_at_9.22.57_PM.png

inflating: /content/stale_banana/rotated_by_15_Screen_Shot_2018-06-12_at_9.23.10_PM.png

inflating: /content/stale_banana/rotated_by_15_Screen_Shot_2018-06-12_at_9.23.15_PM.png

inflating: /content/stale_banana/rotated_by_15_Screen_Shot_2018-06-12_at_9.23.24_PM.png

inflating: /content/stale_banana/rotated_by_15_Screen_Shot_2018-06-12_at_9.25.00_PM.png

inflating: /content/stale_banana/rotated_by_15_Screen_Shot_2018-06-12_at_9.25.04_PM.png

inflating: /content/stale_banana/rotated_by_15_Screen_Shot_2018-06-12_at_9.25.09_PM.png

inflating: /content/stale_banana/rotated_by_15_Screen_Shot_2018-06-12_at_9.25.13_PM.png

Variables Terminal Resuming execution

smart sorting.ipynb

File Edit View Insert Runtime Tools Help

Commands + Code + Text Run all

Connecting

Share Gemini M

```
train_images, val_images = train_test_split(train_and_val_images, test_size=0.2, random_state=42)

for img in train_images:
    shutil.copy(os.path.join(class_dir, img), os.path.join(output_dir, 'train', cls, img))
for img in val_images:
    shutil.copy(os.path.join(class_dir, img), os.path.join(output_dir, 'val', cls, img))
for img in test_images:
    shutil.copy(os.path.join(class_dir, img), os.path.join(output_dir, 'test', cls, img))

print("Dataset successfully split into training, validation, and test sets.")
```

fresh_tomato 200
fresh_banana 200
fresh_orange 200
fresh_bitter_gourd 200
stale_apple 200
stale_banana 200
fresh_capsicum 200
stale_tomato 200
fresh_apple 200
stale_orange 200
stale_bitter_gourd 200
stale_capsicum 200
Dataset successfully split into training, validation, and test sets.

```
[ ] # Set image size and directories
IMG_SIZE = (224, 224)
dataset_dir = '/content/output_dataset'
train_dir = os.path.join(dataset_dir, 'train')
val_dir = os.path.join(dataset_dir, 'val')
test_dir = os.path.join(dataset_dir, 'test')
```

Variables Terminal

Resuming execution

smart sorting.ipynb

File Edit View Insert Runtime Tools Help

Commands + Code + Text Run all

Connecting

Share Gemini M

```
[ ] # Print class indices
print(train_generator.class_indices)
print(val_generator.class_indices)
print(test_generator.class_indices)
```

```
Found 1440 images belonging to 12 classes.
Found 480 images belonging to 12 classes.
Found 480 images belonging to 12 classes.
{'fresh_apple': 0, 'fresh_banana': 1, 'fresh_bitter_gourd': 2, 'fresh_capsicum': 3, 'fresh_orange': 4, 'fresh_tomato': 5, 'stale_apple': 6, 'stale_banana': 7, 'stale_bitter_gourd': 8,
{'fresh_apple': 0, 'fresh_banana': 1, 'fresh_bitter_gourd': 2, 'fresh_capsicum': 3, 'fresh_orange': 4, 'fresh_tomato': 5, 'stale_apple': 6, 'stale_banana': 7, 'stale_bitter_gourd': 8,
{'fresh_apple': 0, 'fresh_banana': 1, 'fresh_bitter_gourd': 2, 'fresh_capsicum': 3, 'fresh_orange': 4, 'fresh_tomato': 5, 'stale_apple': 6, 'stale_banana': 7, 'stale_bitter_gourd': 8,
```

Double-click (or enter) to edit

```
lls /content/output_dataset/train
```

fresh_apple	fresh_capsicum	stale_apple	stale_capsicum
fresh_banana	fresh_orange	stale_banana	stale_orange
fresh_bitter_gourd	fresh_tomato	stale_bitter_gourd	stale_tomato

```
[ ] import os
import random
from IPython.display import Image, display
# Specify the correct path to your image folder
folder_path = '/content/output_dataset/train/fresh_apple' # + Added leading slash

# List all image files
image_files = [f for f in os.listdir(folder_path) if f.endswith((''.jpg', '.png', '.jpeg'))]
```

Variables Terminal

Resuming execution