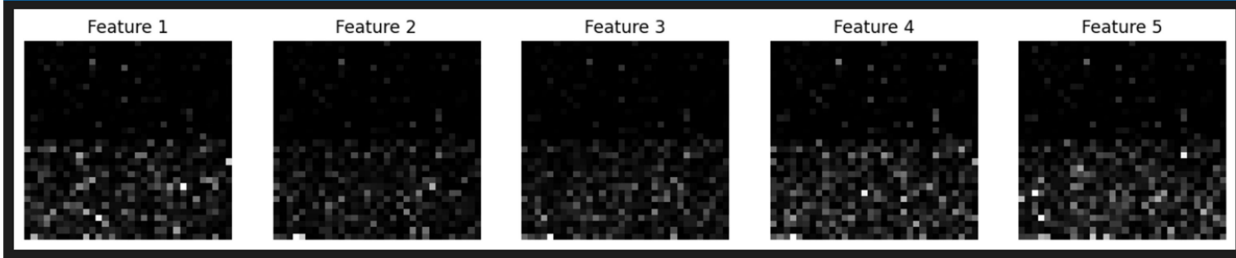
```python
import matplotlib.pyplot as plt
import numpy as np
from tensorflow.keras.preprocessing.image import ImageDataGenerator

# Create ImageDataGenerator instance
datagen = ImageDataGenerator()

# Load training images with class labels
train_generator = datagen.flow_from_directory(
    train_dir,
    target_size=(224, 224),
    batch_size=32,
    class_mode='categorical'
)

# Plot some sample images from the dataset
def plot_images_from_generator(generator):
    images, labels = next(generator)  # Get one batch of images
    plt.figure(figsize=(12, 8))
    for i in range(5):  # Display 5 images
        plt.subplot(1, 5, i + 1)
        plt.imshow(images[i].astype("uint8"))
        plt.axis('off')
        plt.title(f"Class: {np.argmax(labels[i])}")
    plt.show()

# Display sample images
plot_images_from_generator(train_generator)
```

```
Found 4428 images belonging to 5 classes.
```


```python
# Display the first feature vector
print("First 10 extracted feature values for the first sample in the training set:")
print(densenet_train_features[0][:10])  # Display the first 10 values

print("\nShape of feature vector:", densenet_train_features[0].shape)

# Display multiple feature vectors
print("\nFirst 5 feature vectors with first 5 values:")
for i in range(5):
    print(f"Sample {i + 1}:", densenet_train_features[i][:5])

```

```
First 10 extracted feature values for the first sample in the training set:
[5.9016613e-05 5.5387835e-03 3.8996928e-03 2.4631540e-03 6.4646542e-02
 1.0872296e+00 1.3914286e-03 2.1080947e-03 1.3035554e-01 3.9736979e-04]

Shape of feature vector: (1024,)

First 5 feature vectors with first 5 values:
Sample 1: [5.9016613e-05 5.5387835e-03 3.8996928e-03 2.4631540e-03 6.4646542e-02]
Sample 2: [0.00015548 0.00486723 0.00626041 0.00155096 0.05498223]
Sample 3: [0.0001007  0.00571491 0.00519302 0.00214454 0.07052059]
Sample 4: [0.00022117 0.00697958 0.00369843 0.00204438 0.05035614]
Sample 5: [0.0001594  0.00452116 0.0031273  0.00156499 0.05542429]
```

```python
def display_extracted_features(features, num_features=5):
    plt.figure(figsize=(15, 5))

    # Display the first few feature maps
    for i in range(num_features):
        feature_map = features[i]

        # Normalize feature map for better visualization
        feature_map_min = feature_map.min()
        feature_map_max = feature_map.max()

        normalized_feature_map = (feature_map - feature_map_min) / (feature_map_max - feature_map_min)

        plt.subplot(1, num_features, i + 1)
        plt.imshow(normalized_feature_map.reshape((int(normalized_feature_map.shape[0] ** 0.5), -1)), cmap='gray')
        plt.axis('off')
        plt.title(f"Feature {i + 1}")
    plt.show()

# Display sample extracted features
display_extracted_features(densenet_train_features)

```



Seed Classification System

This system can classify seeds into the following categories:

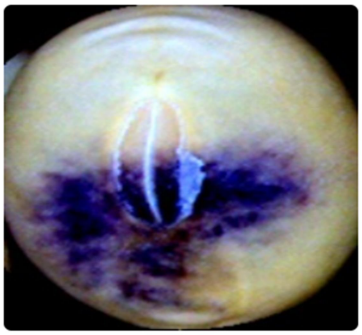
- **Broken** - Seeds with physical breakage
- **Immature** - Seeds that have not fully developed
- **Intact** - Completely healthy seeds
- **Skin-Damaged** - Seeds with damaged outer skin
- **Spotted** - Seeds with spots or discoloration

Upload Seed Image

Choose File | 1033 - Copy.jpg

Classify Seed

Image Preview:

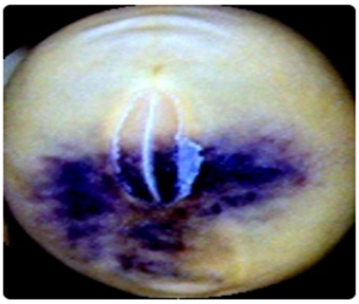


Upload Seed Image

Choose File | 1033 - Copy.jpg

Classify Seed

Image Preview:



Classification Result

The seed is classified as: **Spotted**

Confidence: 100.00%

Probability Distribution

Spotted	100.00%
Broken	0.00%
Immature	0.00%
Skin-Damaged	0.00%
Intact	0.00%