

Cucumber Image Classification Report

1. Used Python Code

```
import os
import numpy as np
import torch
from torchvision import models, transforms
from PIL import Image
import json

# Load ImageNet class labels
with open('imagenet_class_index.json') as f:
    labels = json.load(f)

# Define models to compare
model_names = ['resnet18', 'resnet50', 'mobilenet_v2']
models_dict = {
    'resnet18': models.resnet18(pretrained=True),
    'resnet50': models.resnet50(pretrained=True),
    'mobilenet_v2': models.mobilenet_v2(pretrained=True)
}

# Transform for input image
transform = transforms.Compose([
    transforms.Resize((224, 224)),
    transforms.ToTensor(),
    transforms.Normalize(mean=[0.485, 0.456, 0.406],
                          std=[0.229, 0.224, 0.225])
])

# Folder with images
image_folder = "cucumber"
image_files = [f for f in os.listdir(image_folder) if f.endswith((''.jpg', '.jpeg', '.png'))]

results = {}

for name in model_names:
    model = models_dict[name]
    model.eval()
    correct = 0
```

```

misclassifications = {}

for file in image_files:
    img_path = os.path.join(image_folder, file)
    image = Image.open(img_path).convert("RGB")
    input_tensor = transform(image).unsqueeze(0)

    with torch.no_grad():
        output = model(input_tensor)
        probs = torch.nn.functional.softmax(output[0], dim=0)
        top_prob, top_idx = torch.max(probs, dim=0)

    label = labels[str(top_idx.item())][1]

    if 'cucumber' in label.lower():
        correct += 1
    else:
        misclassifications[label] = misclassifications.get(label, 0) + 1

total = len(image_files)
accuracy = correct / total * 100
results[name] = {'accuracy': accuracy, 'correct': correct, 'misclassified': misclassifications}

# Display results
for model_name, data in results.items():
    print(f"\nModel: {model_name}")
    print(f"Accuracy: {data['accuracy']:.2f}%")
    print(f"Correct: {data['correct']} out of {len(image_files)}")
    print("Misclassified classes:")
    for cls, count in data['misclassified'].items():
        print(f"{cls}: {count} images")

```

2. Classification Results

Comparison of classification accuracy using three models:

Model	Correctly Identified	Accuracy
ResNet-18	30/50	60.00%
ResNet-50	39/50	78.00%
MobileNet v2	32/50	64.00%

```
Image: ILSVRC2012_val_00042061_n07718472.jpg, Prediction: cucumber, Confidence: 0.6466
Image: ILSVRC2012_val_00042999_n07718472.jpg, Prediction: cucumber, Confidence: 0.6400
Image: ILSVRC2012_val_00043204_n07718472.jpg, Prediction: zucchini, Confidence: 0.9296
Image: ILSVRC2012_val_00045484_n07718472.jpg, Prediction: cucumber, Confidence: 0.5834
Image: ILSVRC2012_val_00045704_n07718472.jpg, Prediction: cucumber, Confidence: 0.8213
Image: ILSVRC2012_val_00046928_n07718472.jpg, Prediction: zucchini, Confidence: 0.5099
Image: ILSVRC2012_val_00047341_n07718472.jpg, Prediction: cucumber, Confidence: 0.5146
Image: ILSVRC2012_val_00047407_n07718472.jpg, Prediction: cucumber, Confidence: 0.5885
Image: ILSVRC2012_val_00047851_n07718472.jpg, Prediction: cucumber, Confidence: 0.5646

Total images: 50
Correctly identified (keyword match): 30
Accuracy: 60.00%
|
In [2]:
```

Name	Type	Size	Value
correct	int	1	30
f	TextIOWrapper	1	TextIOWrapper object of _io module
file	str	37	ILSVRC2012_val_00047851_n07718472.jpg
image	Image	(500, 375)	<Image @ 0x1E47B525DF0> Mode: RGB
image_files	list	50	['ILSVRC2012_val_00000463_n07718472.jpg', 'ILSVRC2012_val_00001392_n07 ...
image_folder	str	8	cucumber
img_path	str	46	cucumber\ILSVRC2012_val_00047851_n07718472.jpg
input_tensor	Tensor	(1, 3, 224, 224)	Tensor object of torch module
label	str	8	cucumber
labels	dict	1000	{'0':['n01440764', 'tench'], '1':['n01443537', 'goldfish'], '2':['n014 ...
output	Tensor	(1, 1000)	Tensor object of torch module
probs	Tensor	(1000,)	Tensor object of torch module
top_idx	Tensor	1	Tensor object of torch module
top_prob	Tensor	1	Tensor object of torch module
total	int	1	50

Misclassified Classes:

ResNet-18:

- zucchini: 10
- lemon: 3
- grocery_store: 2
- Granny_Smith: 2
- green_lizard: 1
- book_jacket: 1
- spaghetti_squash: 1

ResNet-50:

- zucchini: 5
- fig: 2
- grocery_store: 2
- packet: 1
- Granny_Smith: 1

MobileNet v2:

- zucchini: 10
- grocery_store: 2
- pomegranate: 1
- packet: 1
- bell_pepper: 1
- butternut_squash: 1
- Granny_Smith: 1
- guacamole: 1

```
In [2]: runfile('D:/AI/Labwork_4/untitled0.py', wdir='D:/AI/Labwork_4')
Reloaded modules: torch.ops, torch.classes
C:\Users\Avater\anaconda3\Lib\site-packages\torchvision\models\_utils.py:223: UserWarning: Arguments other than a weight
enum or 'None' for 'weights' are deprecated since 0.13 and may be removed in the future. The current behavior is equivalent
to passing 'weights=ResNet50_Weights.IMAGENET1K_V1'. You can also use 'weights=ResNet50_Weights.DEFAULT' to get the most
up-to-date weights.
  warnings.warn(msg)
Downloading: "https://download.pytorch.org/models/resnet50-0676ba61.pth" to C:
\Users\Avater\.cache\torch\hub\checkpoints\resnet50-0676ba61.pth
100%|██████████| 97.8M/97.8M [00:31<00:00, 3.24MB/s]
C:\Users\Avater\anaconda3\Lib\site-packages\torchvision\models\_utils.py:223: UserWarning: Arguments other than a weight
enum or 'None' for 'weights' are deprecated since 0.13 and may be removed in the future. The current behavior is equivalent
to passing 'weights=MobileNet_V2_Weights.IMAGENET1K_V1'. You can also use 'weights=MobileNet_V2_Weights.DEFAULT' to get the
most up-to-date weights.
  warnings.warn(msg)
Downloading: "https://download.pytorch.org/models/mobilenet_v2-b0353104.pth" to C:
\Users\Avater\.cache\torch\hub\checkpoints\mobilenet_v2-b0353104.pth
100%|██████████| 13.6M/13.6M [00:04<00:00, 3.31MB/s]

Model: resnet18
Accuracy: 60.00%
Correct: 30 out of 50
Misclassified classes:
grocery_store: 2 images
green_lizard: 1 images
zucchini: 10 images
lemon: 3 images
book_jacket: 1 images
spaghetti_squash: 1 images
Granny_Smith: 2 images

Model: resnet50
Accuracy: 78.00%
```

```
zucchini: 10 images  
lemon: 3 images  
book_jacket: 1 images  
spaghetti_squash: 1 images  
Granny_Smith: 2 images
```

```
Model: resnet50  
Accuracy: 78.00%  
Correct: 39 out of 50  
Misclassified classes:  
grocery_store: 2 images  
packet: 1 images  
zucchini: 5 images  
Granny_Smith: 1 images  
fig: 2 images
```

```
Model: mobilenet_v2  
Accuracy: 64.00%  
Correct: 32 out of 50  
Misclassified classes:  
grocery_store: 2 images  
zucchini: 10 images  
pomegranate: 1 images  
packet: 1 images  
bell_pepper: 1 images  
butternut_squash: 1 images  
Granny_Smith: 1 images  
guacamole: 1 images
```

```
In [3]:
```

Name	Type	Size	Value
accuracy	float	1	64.0
cls	str	9	guacamole
correct	int	1	32
count	int	1	1
data	dict	3	{'accuracy':64.0, 'correct':32, 'misclassified':{'grocery_store':2, 'z ...
f	TextIOWrapper	1	TextIOWrapper object of _io module
file	str	37	ILSVRC2012_val_00047851_n07718472.jpg
image	Image	(500, 375)	<Image @ 0x1E47C6D31D0> Mode: RGB
image_files	list	50	['ILSVRC2012_val_00000463_n07718472.jpg', 'ILSVRC2012_val_00001392_n07 ...
image_folder	str	8	cucumber
img_path	str	46	cucumber\ILSVRC2012_val_00047851_n07718472.jpg
input_tensor	Tensor	(1, 3, 224, 224)	Tensor object of torch module
label	str	8	cucumber
labels	dict	1000	{'0':['n01440764', 'tench'], '1':['n01443537', 'goldfish'], '2':['n014 ...
misclassifications	dict	8	{'grocery_store':2, 'zucchini':10, 'pomegranate':1, 'packet':1, 'bell_ ...
model_name	str	12	mobilenet_v2
model_names	list	3	['resnet18', 'resnet50', 'mobilenet_v2']

3. Explanation of Results

The models used (ResNet-18, ResNet-50, and MobileNet v2) are all pre-trained on the ImageNet dataset. Each of them was used to classify 50 cucumber images.

- ResNet-50 performed the best with an accuracy of 78%.
- MobileNet v2 followed with 64%.
- ResNet-18 had 60% accuracy.

Misclassified Image Analysis:

Common incorrect predictions included:

- Zucchini (most frequent)
- Lemon, Fig, Granny Smith, and Grocery Store

These misclassifications are understandable given the visual similarity of these vegetables to cucumbers. Additionally, model performance may vary due to:

- Lighting and background noise
- Variation in cucumber size or shape
- Image quality

While these predictions were not labeled "cucumber," they may still be contextually relevant and show how closely related classes challenge model differentiation.

4. Conclusions

- Using multiple models provided a clearer understanding of performance variation. ResNet-50 showed the highest accuracy at 78%.
- Misclassifications were consistent across models and often involved visually similar green fruits or vegetables.
- While keyword matching provided a practical evaluation method, it is not perfect. Fine-tuning on domain-specific cucumber images would likely improve performance.
- Pre-trained models like MobileNet are useful for lighter applications, though they may sacrifice some accuracy.

This experiment demonstrates how different models interpret visual input differently and how model selection can impact classification accuracy in specific domains.