Lab 13: Understanding the Architecture of a Pre-trained Model

Aim:

To study and analyze the structure of a pre-trained CNN model such as VGG16 or ResNet50

Description :

Pre trained models are trained on large datasets and can be reused for feature extraction or fine-tuning.

they contain multiple convolutional and fully connected layers designed for hierarchical feature learning

Procedure :

1.) Import a pre-trained model (e.g. keras.application)

2.) Display layers names, types, and output shapes

3.) Analyse the number of parameters and filter sizes.

4.) Visualize intermediate feature maps for a sample image

Pseudocode:

Load VGG16

Print model summary

select an input image

Extract outputs from selected layers

visualise feature maps?

Observation:

Early layers capture edges and colors, while deeper layers detect shapes and objects. The network shows hierarchical representation learning.

```
print("Feature Extractor:\n", feature_extractor)
print("\nClassifier Head:\n", classifier_head)

# ----------------------------------
# 🖼️ Test with an Image
# ----------------------------------
# Example image (download or use your own)
# Replace "example.jpg" with your local image path
# You can skip this block if you just want architecture visualization

try:
    img_path = "example.jpg"  # put your image path here
    img = Image.open(img_path).convert("RGB")
    plt.imshow(img)
    plt.title("Input Image")
    plt.axis("off")
    plt.show()

    # Preprocess for ResNet-50
    preprocess = transforms.Compose([
        transforms.Resize(256),
        transforms.CenterCrop(224),
        transforms.ToTensor(),
        transforms.Normalize(mean=[0.485, 0.456, 0.406],
                             std=[0.229, 0.224, 0.225]),
    ])

    input_tensor = preprocess(img).unsqueeze(0)  # shape: [1, 3, 224, 224]
```

```
    print("\n🏷️ Top 5 predicted class indices:", top5.indices)
    print("📊 Corresponding probabilities:", top5.values)
except FileNotFoundError:
    print("\n⚠️ No sample image found — skipping inference.")
    print("Place an image named 'example.jpg' in this folder to test the model.")

# ----------------------------------
# 🧱 Summary of architecture structure
# ----------------------------------
print("""
🧱 Summary of ResNet-50 architecture:

Input (3x224x224)
    ↓
Conv1 + BatchNorm + ReLU
    ↓
MaxPool
    ↓
Residual Blocks (layer1–layer4)
    ↓
Global Average Pooling
    ↓
Fully Connected (1000 classes)
    ↓
Softmax → Predicted Class
""")
```

Lab_13.ipynb Saving failed since 11:29 PM

File Edit View Insert Runtime Tools Help

Commands + Code + Text Run all

Share

RAM Disk

[1] 19s

```
    ↓
Fully Connected (1000 classes)
    ↓
Softmax → Predicted Class
""")
```

```
🔍 Model Architecture:

ResNet(
  (conv1): Conv2d(3, 64, kernel_size=(7, 7), stride=(2, 2), padding=(3, 3), bias=False)
  (bn1): BatchNorm2d(64, eps=1e-05, momentum=0.1, affine=True, track_running_stats=True)
  (relu): ReLU(inplace=True)
  (maxpool): MaxPool2d(kernel_size=3, stride=2, padding=1, dilation=1, ceil_mode=False)
  (layer1): Sequential(
    (0): Bottleneck(
      (conv1): Conv2d(64, 64, kernel_size=(1, 1), stride=(1, 1), bias=False)
      (bn1): BatchNorm2d(64, eps=1e-05, momentum=0.1, affine=True, track_running_stats=True)
      (conv2): Conv2d(64, 64, kernel_size=(3, 3), stride=(1, 1), padding=(1, 1), bias=False)
      (bn2): BatchNorm2d(64, eps=1e-05, momentum=0.1, affine=True, track_running_stats=True)
      (conv3): Conv2d(64, 256, kernel_size=(1, 1), stride=(1, 1), bias=False)
      (bn3): BatchNorm2d(256, eps=1e-05, momentum=0.1, affine=True, track_running_stats=True)
      (relu): ReLU(inplace=True)
      (downsample): Sequential(
        (0): Conv2d(64, 256, kernel_size=(1, 1), stride=(1, 1), bias=False)
        (1): BatchNorm2d(256, eps=1e-05, momentum=0.1, affine=True, track_running_stats=True)
      )
    )
    (1): Bottleneck(
      (conv1): Conv2d(256, 64, kernel_size=(1, 1), stride=(1, 1), bias=False)
      (bn1): BatchNorm2d(64, eps=1e-05, momentum=0.1, affine=True, track_running_stats=True)
```

Automatic saving failed. This file was updated remotely or in another tab. Show diff

11:30 PM T4 (Python 3)

Lab_13.ipynb ☆ Saving failed since 11:29 PM

File Edit View Insert Runtime Tools Help

Share

mands + Code + Text ▷ Run all

RAM
Disk

```
Classifier Head:
 Linear(in_features=2048, out_features=1000, bias=True)

⚠️ No sample image found — skipping inference.
Place an image named 'example.jpg' in this folder to test the model.

📘 Summary of ResNet-50 architecture:

Input (3x224x224)
   ↓
Conv1 + BatchNorm + ReLU
   ↓
MaxPool
   ↓
Residual Blocks (layer1–layer4)
   ↓
Global Average Pooling
   ↓
Fully Connected (1000 classes)
   ↓
Softmax → Predicted Class
```