

13	27/10/25	Architecture of Pre-trained models	Eg. J
14	27/10/25	Pre-trained CNN model as a feature Extractor using Transfer learning	Eg. J
15	27/10/25	Yolo model to Detect the objects.	Eg. J

27/10/25

understanding the Architecture of pre-trained models.

Aim:

To Study and analyse the architecture of pre-trained CNN models such as VGG16, Resnet50, and InceptionV3.

Objectives:

- + To understand layer configuration of pre-trained CNN models.
- + To visualize input, hidden and output layers.
- + To interpret the feature extraction process of pre-trained models.

Observation:

~~→ pre-trained models are trained on large dataset like ImageNet.~~

~~+ layers closer to input extract general features (edges, texture) while deeper layers capture more abstract features (shape, objects).~~

~~+ Model summary provides the total parameters and layer structure.~~

for working with convolution  
output. about benefit of model

Verb ( )  
(faster): sequential ( )  
and padding ( )  
④ : conv2d (3, 64, kernel\_size=(3, 3), stride=(1, 1),  
padding=(1, 1)) mean be  
ReLU (inplace=True)  
(1) : relu (inplace=True)

(2) : conv2d (64, 64, kernel\_size=(3, 3), stride=(1, 1),  
padding=(1, 1)) mean be  
(3) : ReLU (inplace=True) mean be  
(4) : MaxPool2d (kernel\_size=2, stride=2, padding=0,  
dilation=1, ceil\_mode=False)  
activation with length of 4  
; later be used for exams  
;

(5) : MaxPool2d (kernel\_size=2, stride=2,  
padding=0, dilation=1, ceil\_mode=False)  
; support and test it equal

(AvgPool) : AdaptiveAvgPool (output\_size=(3, 3))  
; support and test it equal  
(6) : linear (in\_features=256, out\_features=100,  
bias=True) mean be  
(7) : ReLU (inplace=True) mean be  
(8) : Dropout (p=0.5, inplace=False)  
;

(9)

### Pseudocode:

- 1) Import library
- 2) Load a pre-trained model
- 3) Display model summary using  
model.summary()
- 4) Visualize architecture using  
plot-model ()
- 5) Print layers names and output shape.  
order

feature extractor layer  
sequential ( )  
④ conv2d  
;

(5) MaxPool2d

classification layer

sequential ( )

(6) : linear (in\_features=256, out\_features=100,  
bias=True)

(7) ReLU (inplace=True)

(8) Dropout (p=0.5, inplace=False)

Total trainable parameters : 134752544

### Result:

Successfully implemented the  
Architecture of pre-trained model.

diffab13.ipynb

File Edit View Insert Runtime Tools Help

Q Commands + Code + Text ▶ Run cell

Connect ↗

ipip install torch torchvision matplotlib

```
Requirement already satisfied: torch in /usr/local/lib/python3.12/dist-packages (2.0.0rc1+cu26)
Requirement already satisfied: torchvision in /usr/local/lib/python3.12/dist-packages (0.17.0+cu26)
Requirement already satisfied: matplotlib in /usr/local/lib/python3.12/dist-packages (3.18.0)
Requirement already satisfied: filelock in /usr/local/lib/python3.12/dist-packages (from torch) (3.18.0)
Requirement already satisfied: typing-extensions>=4.1.0 in /usr/local/lib/python3.12/dist-packages (from torch) (4.1.0)
Requirement already satisfied: setuptools in /usr/local/lib/python3.12/dist-packages (from torch) (56.2.0)
Requirement already satisfied: sympy>1.13.3 in /usr/local/lib/python3.12/dist-packages (from torch) (1.13.3)
Requirement already satisfied: networkx in /usr/local/lib/python3.12/dist-packages (from torch) (3.0.0)
Requirement already satisfied: fsspec in /usr/local/lib/python3.12/dist-packages (from torch) (2023.3.0)
Requirement already satisfied: dask-cuda>0.7.0 in /usr/local/lib/python3.12/dist-packages (from torch) (0.7.0)
Requirement already satisfied: nvidia-cuda-runtime-cu12.6.77 in /usr/local/lib/python3.12/dist-packages (from torch) (12.6.77)
Requirement already satisfied: nvidia-cuda-cudatoolkit-cu12.6.77 in /usr/local/lib/python3.12/dist-packages (from torch) (12.6.77)
Requirement already satisfied: nvidia-cuda-cudatoolkit-cu12.6.77 in /usr/local/lib/python3.12/dist-packages (from torch) (12.6.77)
Requirement already satisfied: nvidia-cublas-cu12.6.4.1 in /usr/local/lib/python3.12/dist-packages (from torch) (12.6.4.1)
Requirement already satisfied: nvidia-cufft-cu12.6.4.1 in /usr/local/lib/python3.12/dist-packages (from torch) (12.6.4.1)
Requirement already satisfied: nvidia-curand-cu12.6.77 in /usr/local/lib/python3.12/dist-packages (from torch) (12.6.77)
Requirement already satisfied: nvidia-cusolver-cu12.6.7.0 in /usr/local/lib/python3.12/dist-packages (from torch) (12.6.7.0)
Requirement already satisfied: nvidia-cusparse-cu12.6.4.2 in /usr/local/lib/python3.12/dist-packages (from torch) (12.6.4.2)
Requirement already satisfied: nvidia-cusparselt-cu12.6.7.1 in /usr/local/lib/python3.12/dist-packages (from torch) (12.6.7.1)
Requirement already satisfied: nvidia-hccl-cu12.6.7.3 in /usr/local/lib/python3.12/dist-packages (from torch) (12.6.7.3)
Requirement already satisfied: nvidia-mtx-cu12.6.77 in /usr/local/lib/python3.12/dist-packages (from torch) (12.6.77)
Requirement already satisfied: nvidia-nvml-cu12.6.85 in /usr/local/lib/python3.12/dist-packages (from torch) (12.6.85)
Requirement already satisfied: nvidia-oflfile-cu12.6.1 in /usr/local/lib/python3.12/dist-packages (from torch) (12.6.1)
Requirement already satisfied: triton>0.8 in /usr/local/lib/python3.12/dist-packages (from torch) (0.8.0)
Requirement already satisfied: numpy in /usr/local/lib/python3.12/dist-packages (from torchvision) (1.20.2)
Requirement already satisfied: pillow>9.5.3 in /usr/local/lib/python3.12/dist-packages (from torchvision) (9.5.3)
Requirement already satisfied: contourpy>1.0.1 in /usr/local/lib/python3.12/dist-packages (from matplotlib) (1.0.1)
Requirement already satisfied:ycler>1.0 in /usr/local/lib/python3.12/dist-packages (from matplotlib) (0.12.3)
Requirement already satisfied: fonttools>4.22.0 in /usr/local/lib/python3.12/dist-packages (from matplotlib) (4.22.0)
Requirement already satisfied: kochiolymp>1.0 in /usr/local/lib/python3.12/dist-packages (from matplotlib) (1.0.0)
Requirement already satisfied: packaging>20.0 in /usr/local/lib/python3.12/dist-packages (from matplotlib) (20.0)
Requirement already satisfied: pygments>2.3 in /usr/local/lib/python3.12/dist-packages (from matplotlib) (2.3.5)
Requirement already satisfied: python-dateutil>2.21.3 in /usr/local/lib/python3.12/dist-packages (from matplotlib) (2.21.3)
Requirement already satisfied: six>1.5 in /usr/local/lib/python3.12/dist-packages (from python-dateutil>2.7>=2.7.0)
Requirement already satisfied: importlib>1.4.1 in /usr/local/lib/python3.12/dist-packages (from sympy>1.13.3>torch) (1.4.1)
Requirement already satisfied: MarkupSafe>2.0 in /usr/local/lib/python3.12/dist-packages (from jinja2>3.0)
Requirement already satisfied: MarkupSafe>2.0 in /usr/local/lib/python3.12/dist-packages (from torch) (2.0.0)
```

# lab13\_pretrained\_model\_architecture.py

```
import torch
import torchvision.models as models

# Load a pretrained model (vgg16)
model = models.vgg16(pretrained=True)

# Print model architecture
print("***** VGG16 Pre-trained Model Architecture *****")
print(model)

# Print only feature extractor part
print("***** Feature Extractor Layers *****")
print(model.features)

# Print classifier part
print("***** Classifier Layers *****")
print(model.classifier)

# Print total trainable parameters
total_params = sum(p.numel() for p in model.parameters() if p.requires_grad)
print(f"Total Trainable Parameters: {total_params}")

/usr/local/lib/python3.12/dist-packages/torchvision/models/_utils.py:288: UserWarning: The parameter 'pretrained' is deprecated since 0.13 and may be removed in the future, please use 'weights' instead.
  warnings.warn()
/usr/local/lib/python3.12/dist-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=Weights.IMGNET1K_V1'. You can also use 'weights=Weights.IMGNET1K_DEFAULT' to get the most up-to-date weights.
  warnings.warn(msg)
Downloaded: "https://download.pytorch.org/models/vgg16-397923ef.pth" to /root/.cache/torch/hub/checkpoints/vgg16-397923ef.pth
288M [██████████] 528PV528M [00:03:00.88, 24984/S]
www VGG16 Pre-trained Model Architecture www
Vgg[  
  (features): Sequential(  
    (0): Conv2d(3, 64, kernel_size=(3, 3), stride=(1, 1), padding=(1, 1))  
    (1): ReLU(inplace=True)  
    (2): Conv2d(64, 64, kernel_size=(3, 3), stride=(1, 1), padding=(1, 1))  
    (3): ReLU(inplace=True)
```

```
dlab13.pyimb ★ 66
File Edit View Insert Runtime Tools Help
Commands + Code + Text ▾ RunAll ▾
Connect ↗ Share

# Print total trainable parameters
total_params = sum(p.numel() for p in model.parameters() if p.requires_grad)
print(f"Total Trainable Parameters: {total_params}")

# user/local/lib/python3.12/dist-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=VGG16_Weights.DEFAULT'. You can also use 'weights=VGG16_Weights.DEFAULT' to get the most up-to-date weights.
# warnings.warn(msg)
# downloading: https://download.pytorch.org/models/vgg16-39792ef.pth to /root/.cache/torch/hub/checkpoints/vgg16-39792ef.pth
#66| [██████████] 528M/528M [99%] 149MB/s
#66| VGG16 Pre-trained Model Architecture
#66
#(Features): sequential
#(0): Conv2d(3, 64, kernel_size=(3, 3), stride=(1, 1), padding=(1, 1))
#(1): ReLU(inplace=True)
#(2): Conv2d(64, 64, kernel_size=(3, 3), stride=(1, 1), padding=(1, 1))
#(3): ReLU(inplace=True)
#(4): MaxPool2d(kernel_size=2, stride=2, padding=0, dilation=1, ceil_mode=False)
#(5): Conv2d(64, 128, kernel_size=(3, 3), stride=(1, 1), padding=(1, 1))
#(6): ReLU(inplace=True)
#(7): Conv2d(128, 128, kernel_size=(3, 3), stride=(1, 1), padding=(1, 1))
#(8): ReLU(inplace=True)
#(9): MaxPool2d(kernel_size=2, stride=2, padding=0, dilation=1, ceil_mode=False)
#(10): Conv2d(128, 256, kernel_size=(3, 3), stride=(1, 1), padding=(1, 1))
#(11): ReLU(inplace=True)
#(12): Conv2d(256, 256, kernel_size=(3, 3), stride=(1, 1), padding=(1, 1))
#(13): ReLU(inplace=True)
#(14): Conv2d(256, 256, kernel_size=(3, 3), stride=(1, 1), padding=(1, 1))
#(15): MaxPool2d(kernel_size=2, stride=2, padding=0, dilation=1, ceil_mode=False)
#(16): Conv2d(256, 512, kernel_size=(3, 3), stride=(1, 1), padding=(1, 1))
#(17): ReLU(inplace=True)
#(18): Conv2d(512, 512, kernel_size=(3, 3), stride=(1, 1), padding=(1, 1))
#(19): ReLU(inplace=True)
#(20): Conv2d(512, 512, kernel_size=(3, 3), stride=(1, 1), padding=(1, 1))
#(21): ReLU(inplace=True)
#(22): MaxPool2d(kernel_size=2, stride=2, padding=0, dilation=1, ceil_mode=False)
#(23): Conv2d(512, 512, kernel_size=(3, 3), stride=(1, 1), padding=(1, 1))
#(24): ReLU(inplace=True)
#(25): Conv2d(512, 512, kernel_size=(3, 3), stride=(1, 1), padding=(1, 1))
#(26): ReLU(inplace=True)
#(27): Conv2d(512, 512, kernel_size=(3, 3), stride=(1, 1), padding=(1, 1))
#(28): ReLU(inplace=True)
#(29): MaxPool2d(kernel_size=2, stride=2, padding=0, dilation=1, ceil_mode=False)
#(30): AdaptiveAvgPool2d(output_size=(7, 7))
#(Classification): Sequential
#(0): Linear(in_features=19688, out_features=4096, bias=True)
#(1): ReLU(inplace=True)
#(2): Dropout(p=0.5, inplace=False)
#(3): Linear(in_features=4096, out_features=4096, bias=True)
#(4): ReLU(inplace=True)
#(5): Dropout(p=0.5, inplace=False)
#(6): Linear(in_features=4096, out_features=1000, bias=True)

# Feature Extractor Layers
sequential
#(0): Conv2d(3, 64, kernel_size=(3, 3), stride=(1, 1), padding=(1, 1))
#(1): ReLU(inplace=True)
#(2): Conv2d(64, 64, kernel_size=(3, 3), stride=(1, 1), padding=(1, 1))
#(3): ReLU(inplace=True)
#(4): MaxPool2d(kernel_size=2, stride=2, padding=0, dilation=1, ceil_mode=False)
```

File Edit View Insert Runtime Tools Help

Q Commands + Code + Text + Run All Connect ↻ Share

```
11
❶ # print total trainable parameters
total_params = sum(p.numel() for p in model.parameters() if p.requires_grad)
print(f'Total Trainable Parameters: {total_params}')


(Sequential):
(0): Linear(in_features=2888, out_features=496, bias=True)
(1): ReLU(inplace=True)
(2): Dropout(p=0.5, inplace=False)
(3): Linear(in_features=496, out_features=400, bias=True)
(4): ReLU(inplace=True)
(5): Dropout(p=0.5, inplace=False)
(6): Linear(in_features=400, out_features=1000, bias=True)
}

---- Feature Extractor Layers ----
sequential:
(0): Conv2d(3, 64, kernel_size=(3, 3), stride=(1, 1), padding=(1, 1))
(1): ReLU(inplace=True)
(2): Conv2d(64, 64, kernel_size=(3, 3), stride=(1, 1), padding=(1, 1))
(3): ReLU(inplace=True)
(4): MaxPool2d(kernel_size=2, stride=2, padding=0, dilation=1, ceil_mode=False)
(5): Conv2d(64, 128, kernel_size=(3, 3), stride=(1, 1), padding=(1, 1))
(6): ReLU(inplace=True)
(7): Conv2d(128, 128, kernel_size=(3, 3), stride=(1, 1), padding=(1, 1))
(8): ReLU(inplace=True)
(9): MaxPool2d(kernel_size=2, stride=2, padding=0, dilation=1, ceil_mode=False)
(10): Conv2d(128, 256, kernel_size=(3, 3), stride=(1, 1), padding=(1, 1))
(11): ReLU(inplace=True)
(12): Conv2d(256, 256, kernel_size=(3, 3), stride=(1, 1), padding=(1, 1))
(13): ReLU(inplace=True)
(14): Conv2d(256, 256, kernel_size=(3, 3), stride=(1, 1), padding=(1, 1))
(15): ReLU(inplace=True)
(16): MaxPool2d(kernel_size=2, stride=2, padding=0, dilation=1, ceil_mode=False)
(17): Conv2d(256, 512, kernel_size=(3, 3), stride=(1, 1), padding=(1, 1))
(18): ReLU(inplace=True)
(19): Conv2d(512, 512, kernel_size=(3, 3), stride=(1, 1), padding=(1, 1))
(20): ReLU(inplace=True)
(21): Conv2d(512, 512, kernel_size=(3, 3), stride=(1, 1), padding=(1, 1))
(22): ReLU(inplace=True)
(23): MaxPool2d(kernel_size=2, stride=2, padding=0, dilation=1, ceil_mode=False)
(24): Conv2d(512, 512, kernel_size=(3, 3), stride=(1, 1), padding=(1, 1))
(25): ReLU(inplace=True)
(26): Conv2d(512, 512, kernel_size=(3, 3), stride=(1, 1), padding=(1, 1))
(27): ReLU(inplace=True)
(28): Conv2d(512, 512, kernel_size=(3, 3), stride=(1, 1), padding=(1, 1))
(29): ReLU(inplace=True)
(30): MaxPool2d(kernel_size=2, stride=2, padding=0, dilation=1, ceil_mode=False)

---- Classifier Layers ----
sequential:
(0): Linear(in_features=2888, out_features=496, bias=True)
(1): ReLU(inplace=True)
(2): Dropout(p=0.5, inplace=False)
(3): Linear(in_features=496, out_features=400, bias=True)
(4): ReLU(inplace=True)
(5): Dropout(p=0.5, inplace=False)
(6): Linear(in_features=400, out_features=1000, bias=True)

Total Trainable Parameters: 13835754
```