ex:-13 Understanding the architecture of Pre-trained model

Aim:- To explore and understand the architecture, layer and feature extraction capability of pre-trained deep learning model on standard dataset like. imagenet, using tensorflow/keras.

objectives:-

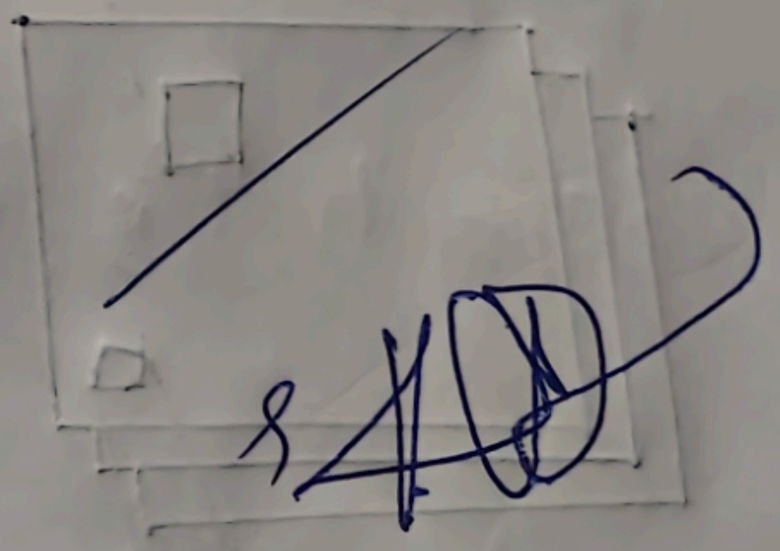
1. To load a pre-trained model with weights trained on Imagenet.
2. To analyze the layer structure and number of parameters in the model.
3. To perform feature extraction using the pre-trained model
4. To visualize intermediate feature maps for understanding how the model processes images
5. To predict class of sample images using the pre-trained model.

Pseudocode

1. Import necessary libraries
2. Load a pre trained model with ImageNet weights
3. print model summary to understand the architecture
4. Load a sample image and preprocess it for the chosen model
5. use the model to predict the class of the image
6. Decode predictions to get human-readable labels.
7. optionally create a new model to output intermediate layers for visualization.
8. visualize feature maps from convolutional layers to understand learned features
9. Analyze results and summarize observations

Result:-

Successfully loaded and understood the architecture of VGG16.



output:-

1. model summary
   - vgg 16 has 16 layers mostly convolution + pooling layers followed by fully connected layers
   - total parameters:- ~138 million (trainable + non trainable)

2. predictions:-
   for the sample elephant image, the model predicts African elephant or indian elephant depending ontop probabilities.

3. Feature maps.
   - Early convolutional layers capture edges, textures and simple patterns.
   - deeper layers capture complex structures and object parts.

Compression on MNIST dataset | Welcome to Colab - Colab | (57) Oh Priya Priya Full Video S

colab.research.google.com/#scrollTo=yqzUuG69RSGY

Gmail | Maps | News | Translate | Disney+ Hotstar -... | YouTube | All Bookmarks

Welcome to Colab | Cannot save changes

File Edit View Insert Runtime Tools Help

Share

Commands | + Code | + Text | Run all | Copy to Drive | Reconnect T4

Table of contents

- Welcome to Colab!
  - Getting started
  - Data science
  - Machine learning
  - More resources
    - Featured examples
- + Section

```
[ ]
import tensorflow as tf
from tensorflow.keras.applications import VGG16
from tensorflow.keras.applications.vgg16 import preprocess_input, decode_predictions
from tensorflow.keras.preprocessing.image import img_to_array
import numpy as np
import matplotlib.pyplot as plt

# Load CIFAR-10 dataset
(x_train, y_train), (_, _) = tf.keras.datasets.cifar10.load_data()

# Pick one sample image
img = x_train[0]  # shape (32,32,3)
plt.imshow(img)
plt.axis('off')
plt.title("Sample CIFAR-10 Image")
plt.show()

# Resize image to VGG16 input size (224x224) and make it writable
img_resized = tf.image.resize(img, (224, 224)).numpy().copy()
x = np.expand_dims(img_resized, axis=0)
x = preprocess_input(x)  # preprocess for VGG16

# Load pre-trained VGG16
model = VGG16(weights='imagenet', include_top=True)
model.summary()

# Predict image class
preds = model.predict(x)
print("Top 3 Predictions:")
for pred in decode_predictions(preds, top=3)[0]:
    print(f"{pred[1]}: {pred[2]*100:.2f}%")
```

Variables | Terminal | 02:16

Compression on MNIST dataset | Welcome to Colab - Colab | (57) Oh Priya Priya Full Video S

colab.research.google.com/#scrollTo=yqzUuG69RSGY

Gmail | Maps | News | Translate | Disney+ Hotstar -... | YouTube | All Bookmarks

Welcome to Colab | Cannot save changes

File Edit View Insert Runtime Tools Help

Share

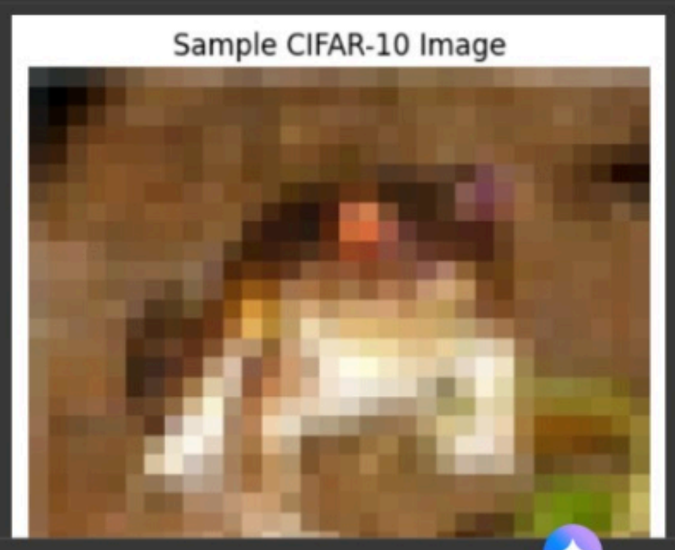
Commands | + Code | + Text | Run all | Copy to Drive | Reconnect T4

Table of contents

- Welcome to Colab!
  - Getting started
  - Data science
  - Machine learning
  - More resources
    - Featured examples
- + Section

```
[ ]
# Visualize feature maps from first conv layer
layer_outputs = [layer.output for layer in model.layers if 'conv' in layer.name]
feature_model = tf.keras.models.Model(inputs=model.input, outputs=layer_outputs)
feature_maps = feature_model.predict(x)

first_layer_features = feature_maps[0]
plt.figure(figsize=(12,6))
for i in range(6):
    plt.subplot(1,6,i+1)
    plt.imshow(first_layer_features[0, :, :, i], cmap='viridis')
    plt.axis('off')
plt.suptitle('First Conv Layer Feature Maps', fontsize=16)
plt.show()
```



Variables | Terminal | 02:16

Compression on MNIST dataset | Welcome to Colab - Colab | (57) Oh Priya Priya Full Video S

colab.research.google.com/#scrollTo=yqzUuG69RSGY

Gmail Maps News Translate Disney+ Hotstar -... YouTube All Bookmarks

Welcome to Colab | Cannot save changes

File Edit View Insert Runtime Tools Help

Share

Commands + Code + Text Run all Copy to Drive Reconnect T4

Table of contents

- Welcome to Colab!
  - Getting started
  - Data science
  - Machine learning
  - More resources
    - Featured examples
- + Section

Model: "vgg16"

Layer (type)	Output Shape	Param #
input_layer_3 (InputLayer)	(None, 224, 224, 3)	0
block1_conv1 (Conv2D)	(None, 224, 224, 64)	1,792
block1_conv2 (Conv2D)	(None, 224, 224, 64)	36,928
block1_pool (MaxPooling2D)	(None, 112, 112, 64)	0
block2_conv1 (Conv2D)	(None, 112, 112, 128)	73,856
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)	590,080
block3_conv3 (Conv2D)	(None, 56, 56, 256)	590,080
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

Variables Terminal 02:16

Compression on MNIST dataset | Welcome to Colab - Colab | (57) Oh Priya Priya Full Video S

colab.research.google.com/#scrollTo=yqzUuG69RSGY

Gmail Maps News Translate Disney+ Hotstar -... YouTube All Bookmarks

Welcome to Colab Cannot save changes

File Edit View Insert Runtime Tools Help

Share

Commands + Code + Text Run all Copy to Drive Reconnect T4

Table of contents

- Welcome to Colab!
  - Getting started
  - Data science
  - Machine learning
  - More resources
    - Featured examples
- + Section

block5_conv3 (Conv2D)	(None, 14, 14, 512)	2,359,808
block5_pool (MaxPooling2D)	(None, 7, 7, 512)	0
flatten (Flatten)	(None, 25088)	0
fc1 (Dense)	(None, 4096)	102,764,544
fc2 (Dense)	(None, 4096)	16,781,312
predictions (Dense)	(None, 1000)	4,097,000

Total params: 138,357,544 (527.79 MB)
Trainable params: 138,357,544 (527.79 MB)
Non-trainable params: 0 (0.00 B)
1/1 ━━━━━━━━━━━━━━━━━━━━ **3s** 3s/step
Top 3 Predictions:
Downloading data from https://storage.googleapis.com/download.tensorflow.org/data/imagenet_class_index.json
35363/35363 ━━━━━━━━━━━━━━━━━━━━ **0s** 1us/step
rock_python: 5.68%
consomme: 5.27%
Brabancon_griffon: 3.62%
1/1 ━━━━━━━━━━━━━━━━━━━━ **1s** 620ms/step

Variables Terminal 02:16

Compression on MNIST dataset | Welcome to Colab - Colab | (57) Oh Priya Priya Full Video S

colab.research.google.com/#scrollTo=yqzUuG69RSGY

Gmail Maps News Translate Disney+ Hotstar -... YouTube All Bookmarks

Welcome to Colab Cannot save changes

File Edit View Insert Runtime Tools Help

Share

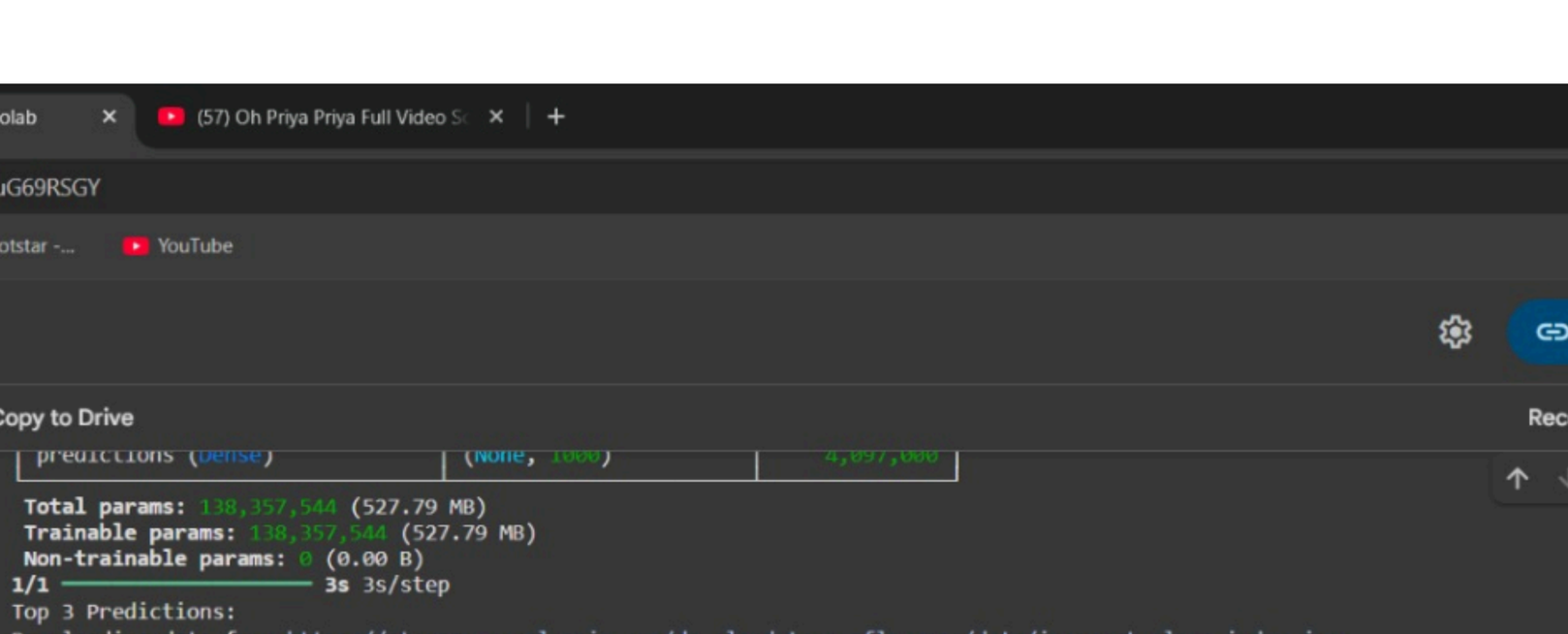
Commands + Code + Text Run all Copy to Drive Reconnect T4

Table of contents

- Welcome to Colab!
- Getting started
- Data science
- Machine learning
- More resources
- Featured examples
- + Section

predictions (Dense) (None, 1000) 4,097,000

```
Total params: 138,357,544 (527.79 MB)
Trainable params: 138,357,544 (527.79 MB)
Non-trainable params: 0 (0.00 B)
1/1 ━━━━━━━━━━━━━━━━━━━━ 3s 3s/step
Top 3 Predictions:
Downloading data from https://storage.googleapis.com/download.tensorflow.org/data/imagenet_class_index.json
35363/35363 ━━━━━━━━━━━━━━━━━━━━ 0s 1us/step
rock_python: 5.68%
consomme: 5.27%
Brabancon_griffon: 3.62%
1/1 ━━━━━━━━━━━━━━━━━━━━ 1s 620ms/step
```



Variables Terminal

02:16