Experiment No: 11

Aim: Using a pre-trained Imagenet network to predict images into one of the1000 Imagenet classes.

Theory:

Using a pre-trained ImageNet network for image classification involves leveraging deep learning models that have been trained on a large-scale dataset of 1.2 million images across 1000 distinct classes. This approach offers several advantages:

Transfer Learning: Pre-trained ImageNet models serve as a powerful foundation for transfer learning. They have already learned useful features from a wide range of images, which can be fine-tuned on smaller, specialized datasets for specific tasks, reducing the need for extensive labeled data.

High Accuracy: ImageNet models have demonstrated state-of-the-art performance in image classification tasks. Their ability to capture intricate visual patterns, textures, and objects makes them a valuable asset for a broad spectrum of applications, from object detection to facial recognition.

Efficiency: Leveraging pre-trained models saves computational resources and time, as training deep neural networks from scratch is computationally intensive. Fine-tuning a pre-trained model allows you to take advantage of the initial training's feature extraction capabilities.

Wide Applicability: Pre-trained ImageNet networks can be adapted to various domains, such as healthcare, autonomous driving, and natural language processing. Their hierarchical features make them versatile tools for understanding and interpreting complex visual data.

ImageNet is a large-scale, publicly available dataset of labeled images, and it has played a pivotal role in advancing the field of computer vision and deep learning. Here's a brief overview of the theory and significance of ImageNet:

Dataset Size: ImageNet contains millions of images categorized into thousands of object classes. In its most well-known version, ImageNet Large Scale Visual Recognition Challenge (ILSVRC), there are over a million images distributed across 1000 different classes.

Benchmark for Image Classification: ImageNet has been widely used as a benchmark for evaluating the performance of image classification algorithms and models. The challenge of classifying objects in such a large and diverse dataset has spurred the development of increasingly sophisticated deep learning models.

Deep Learning Breakthroughs: The ImageNet Large Scale Visual Recognition Challenge, initiated in 2010, played a pivotal role in driving the development of deep learning and Convolutional Neural Networks (CNNs). It led to the resurgence of neural networks and the development of models like AlexNet, VGG, GoogLeNet, and ResNet, which demonstrated remarkable accuracy in image classification.

Transfer Learning: The pre-trained models on ImageNet have been used for transfer learning. Researchers and practitioners fine-tune these models on smaller datasets for specific tasks, such as object detection, segmentation, and more. This transfer learning approach significantly reduces the need for large annotated datasets and computational resources.

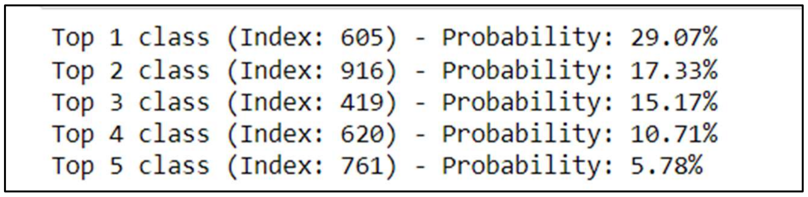
Generalization: ImageNet models have shown the ability to generalize features learned from a wide variety of images. This generalization makes them valuable tools for a wide range of computer vision and image understanding tasks.

Challenges and Datasets: ImageNet has inspired the creation of numerous other datasets and challenges, addressing various aspects of computer vision, including object detection, image segmentation, and scene recognition. These challenges have further accelerated research in the field.

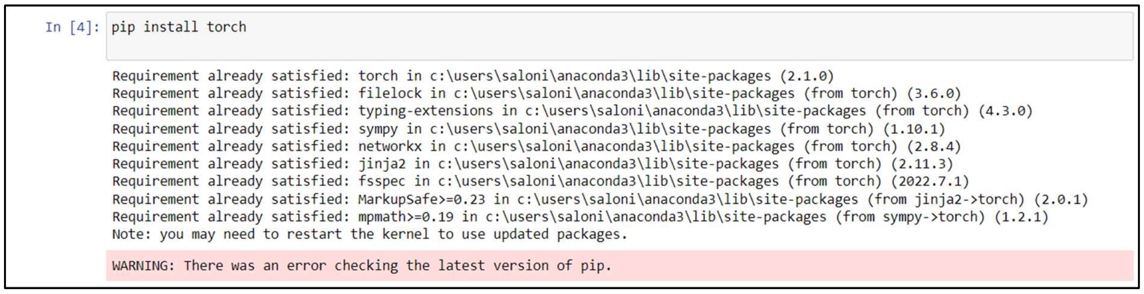
Real-World Applications: The principles and techniques developed through ImageNet and related datasets have found applications in real-world scenarios, from autonomous vehicles to medical image analysis, content moderation, and more.

Implementation:

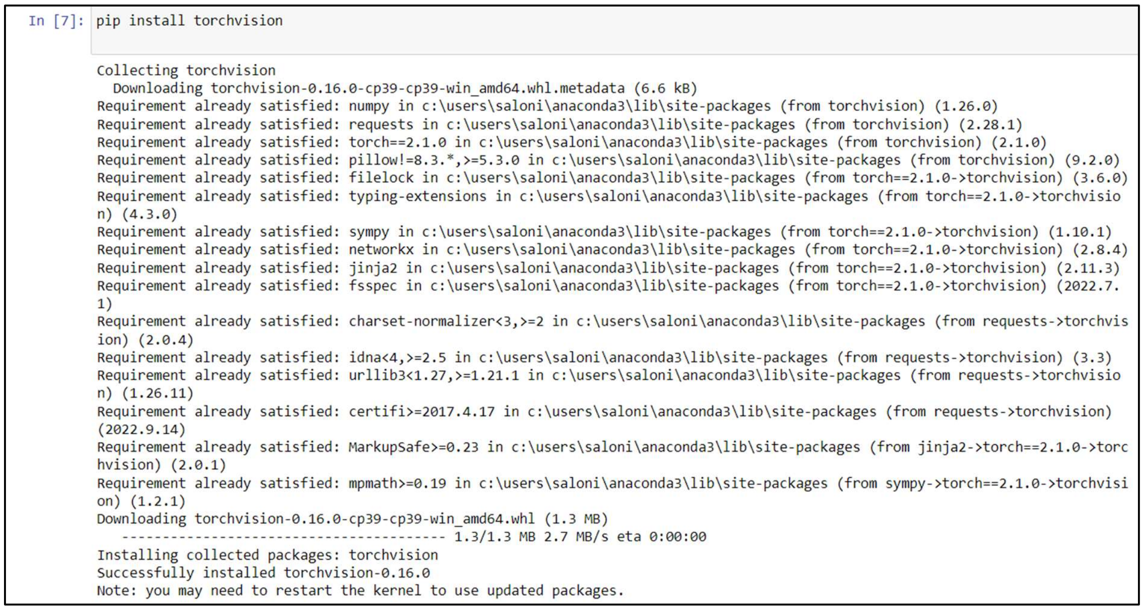
```
import torch
import torch.nn as nn
import torchvision.transforms as transforms
import torchvision.models as models
from PIL import Image

# Load the pre-trained ResNet model
model = models.resnet50(pretrained=True)

# Set the model to evaluation mode
model.eval()

# Define the transformation to preprocess input images to match the model's requirements
transform = 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])
])

# Load and preprocess the input image
image_path = 'C:\\Users\\SALONI\\Pictures\\cb4.png'
image = Image.open(image_path)

# Convert the image to RGB format if it's not already
if image.mode != 'RGB':
    image = image.convert('RGB')

image = transform(image)
image = image.unsqueeze(0)  # Add a batch dimension

# Perform inference to get class probabilities
with torch.no_grad():
    outputs = model(image)

# Get the class predictions by applying softmax to the model's output
probabilities = torch.nn.functional.softmax(outputs[0], dim=0)

# Get the top 5 class indices and their corresponding probabilities
top5_prob, top5_index = torch.topk(probabilities, 5)

# Print the top 5 predicted classes and their probabilities
for i in range(5):
    print(f'Top {i+1} class (Index: {top5_index[i].item()}) - Probability: {top5_prob[i].item()
* 100:.2f}%')
```



```
Top 1 class (Index: 605) - Probability: 29.07%
Top 2 class (Index: 916) - Probability: 17.33%
Top 3 class (Index: 419) - Probability: 15.17%
Top 4 class (Index: 620) - Probability: 10.71%
Top 5 class (Index: 761) - Probability: 5.78%
```



```
In [4]: pip install torch

Requirement already satisfied: torch in c:\users\saloni\anaconda3\lib\site-packages (2.1.0)
Requirement already satisfied: filelock in c:\users\saloni\anaconda3\lib\site-packages (from torch) (3.6.0)
Requirement already satisfied: typing-extensions in c:\users\saloni\anaconda3\lib\site-packages (from torch) (4.3.0)
Requirement already satisfied: sympy in c:\users\saloni\anaconda3\lib\site-packages (from torch) (1.10.1)
Requirement already satisfied: networkx in c:\users\saloni\anaconda3\lib\site-packages (from torch) (2.8.4)
Requirement already satisfied: jinja2 in c:\users\saloni\anaconda3\lib\site-packages (from torch) (2.11.3)
Requirement already satisfied: fsspec in c:\users\saloni\anaconda3\lib\site-packages (from torch) (2022.7.1)
Requirement already satisfied: MarkupSafe>=0.23 in c:\users\saloni\anaconda3\lib\site-packages (from jinja2->torch) (2.0.1)
Requirement already satisfied: mpmath>=0.19 in c:\users\saloni\anaconda3\lib\site-packages (from sympy->torch) (1.2.1)
Note: you may need to restart the kernel to use updated packages.

WARNING: There was an error checking the latest version of pip.
```



```
In [7]: pip install torchvision

Collecting torchvision
  Downloading torchvision-0.16.0-cp39-cp39-win_amd64.whl.metadata (6.6 kB)
Requirement already satisfied: numpy in c:\users\saloni\anaconda3\lib\site-packages (from torchvision) (1.26.0)
Requirement already satisfied: requests in c:\users\saloni\anaconda3\lib\site-packages (from torchvision) (2.28.1)
Requirement already satisfied: torch==2.1.0 in c:\users\saloni\anaconda3\lib\site-packages (from torchvision) (2.1.0)
Requirement already satisfied: pillow!=8.3.*,>=5.3.0 in c:\users\saloni\anaconda3\lib\site-packages (from torchvision) (9.2.0)
Requirement already satisfied: filelock in c:\users\saloni\anaconda3\lib\site-packages (from torch==2.1.0->torchvision) (3.6.0)
Requirement already satisfied: typing-extensions in c:\users\saloni\anaconda3\lib\site-packages (from torch==2.1.0->torchvisio
n) (4.3.0)
Requirement already satisfied: sympy in c:\users\saloni\anaconda3\lib\site-packages (from torch==2.1.0->torchvision) (1.10.1)
Requirement already satisfied: networkx in c:\users\saloni\anaconda3\lib\site-packages (from torch==2.1.0->torchvision) (2.8.4)
Requirement already satisfied: jinja2 in c:\users\saloni\anaconda3\lib\site-packages (from torch==2.1.0->torchvision) (2.11.3)
Requirement already satisfied: fsspec in c:\users\saloni\anaconda3\lib\site-packages (from torch==2.1.0->torchvision) (2022.7.
1)
Requirement already satisfied: charset-normalizer<3,>=2 in c:\users\saloni\anaconda3\lib\site-packages (from requests->torchvis
ion) (2.0.4)
Requirement already satisfied: idna<4,>=2.5 in c:\users\saloni\anaconda3\lib\site-packages (from requests->torchvision) (3.3)
Requirement already satisfied: urllib3<1.27,>=1.21.1 in c:\users\saloni\anaconda3\lib\site-packages (from requests->torchvisio
n) (1.26.11)
Requirement already satisfied: certifi>=2017.4.17 in c:\users\saloni\anaconda3\lib\site-packages (from requests->torchvision)
(2022.9.14)
Requirement already satisfied: MarkupSafe>=0.23 in c:\users\saloni\anaconda3\lib\site-packages (from jinja2->torch==2.1.0->torc
hvision) (2.0.1)
Requirement already satisfied: mpmath>=0.19 in c:\users\saloni\anaconda3\lib\site-packages (from sympy->torch==2.1.0->torchvisi
on) (1.2.1)
Downloading torchvision-0.16.0-cp39-cp39-win_amd64.whl (1.3 MB)
   ---------------------------------------- 1.3/1.3 MB 2.7 MB/s eta 0:00:00
Installing collected packages: torchvision
Successfully installed torchvision-0.16.0
Note: you may need to restart the kernel to use updated packages.
```

Conclusion:

Using a pre-trained ImageNet network provides a powerful and efficient way to classify images into one of the 1000 ImageNet classes. These models have learned from a vast dataset, making them capable of generalizing well to a wide range of visual recognition tasks. They can be fine-tuned for specific applications or used directly for image classification.