Experiment 10 - Transfer Learning - Pre Trained Model VGG16

Problem Statement:

To implement transfer learning using the pre-trained model (VGG16) on image dataset.

GitHub & Google Colab Link:

GitHub Link: https://github.com/piyush-gambhir/ncu-lab-manual-and-end-semester-projects/blob/main/NCU-CSL312%20-%20DL%20-%20Lab%20Manual/Experiment%2010/Experiment%2010.ipynb

Google Colab Link:

CO Open in Colab

Installing Dependencies:

In [ ]:

```
! pip install tabulate numpy pandas matplotlib seaborn
```

```
Collecting tabulate
  Downloading tabulate-0.9.0-py3-none-any.whl.metadata (34 kB)
Requirement already satisfied: numpy in c:\users\mainp\appdata\local\programs\python\python311\lib\site-packages
(1.26.4)
Collecting pandas
  Downloading pandas-2.2.2-cp311-cp311-win_amd64.whl.metadata (19 kB)
Requirement already satisfied: matplotlib in c:\users\mainp\appdata\local\programs\python\python311\lib\site-pac
kages (3.8.4)
Collecting seaborn
  Downloading seaborn-0.13.2-py3-none-any.whl.metadata (5.4 kB)
Requirement already satisfied: python-dateutil>=2.8.2 in c:\users\mainp\appdata\local\programs\python\python311\
lib\site-packages (from pandas) (2.9.0.post0)
Collecting pytz>=2020.1 (from pandas)
  Downloading pytz-2024.1-py2.py3-none-any.whl.metadata (22 kB)
Collecting tzdata>=2022.7 (from pandas)
  Downloading tzdata-2024.1-py2.py3-none-any.whl.metadata (1.4 kB)
Requirement already satisfied: contourpy>=1.0.1 in c:\users\mainp\appdata\local\programs\python\python311\lib\si
te-packages (from matplotlib) (1.2.1)
Requirement already satisfied: cycler>=0.10 in c:\users\mainp\appdata\local\programs\python\python311\lib\site-p
ackages (from matplotlib) (0.12.1)
Requirement already satisfied: fonttools>=4.22.0 in c:\users\mainp\appdata\local\programs\python\python311\lib\s
ite-packages (from matplotlib) (4.51.0)
Requirement already satisfied: kiwisolver>=1.3.1 in c:\users\mainp\appdata\local\programs\python\python311\lib\s
ite-packages (from matplotlib) (1.4.5)
Requirement already satisfied: packaging>=20.0 in c:\users\mainp\appdata\local\programs\python\python311\lib\sit
e-packages (from matplotlib) (24.0)
Requirement already satisfied: pillow>=8 in c:\users\mainp\appdata\local\programs\python\python311\lib\site-pack
ages (from matplotlib) (10.3.0)
Requirement already satisfied: pyparsing>=2.3.1 in c:\users\mainp\appdata\local\programs\python\python311\lib\si
te-packages (from matplotlib) (3.1.2)
Requirement already satisfied: six>=1.5 in c:\users\mainp\appdata\local\programs\python\python311\lib\site-packa
ges (from python-dateutil>=2.8.2->pandas) (1.16.0)
Downloading tabulate-0.9.0-py3-none-any.whl (35 kB)
Downloading pandas-2.2.2-cp311-cp311-win_amd64.whl (11.6 MB)
   ---------------------------------------- 0.0/11.6 MB ? eta -:--:--
   - -------------------------------------- 0.5/11.6 MB 14.9 MB/s eta 0:00:01
   ---- ----------------------------------- 1.2/11.6 MB 14.8 MB/s eta 0:00:01
   ---- ----------------------------------- 1.4/11.6 MB 11.0 MB/s eta 0:00:01
   ------ --------------------------------- 1.8/11.6 MB 10.2 MB/s eta 0:00:01
   --------- ------------------------------ 2.8/11.6 MB 12.9 MB/s eta 0:00:01
   ------------ --------------------------- 3.5/11.6 MB 13.9 MB/s eta 0:00:01
   ----------------- ---------------------- 5.0/11.6 MB 15.9 MB/s eta 0:00:01
   --------------------- ------------------ 6.3/11.6 MB 17.5 MB/s eta 0:00:01
   -------------------------- ------------- 7.7/11.6 MB 18.9 MB/s eta 0:00:01
   ------------------------------- -------- 9.2/11.6 MB 20.2 MB/s eta 0:00:01
   ----------------------------------- ---- 10.4/11.6 MB 21.1 MB/s eta 0:00:01
   ---------------------------------------  11.6/11.6 MB 26.2 MB/s eta 0:00:01
   ---------------------------------------- 11.6/11.6 MB 24.2 MB/s eta 0:00:00
Downloading seaborn-0.13.2-py3-none-any.whl (294 kB)
   ---------------------------------------- 0.0/294.9 kB ? eta -:--:--
   --------------------------------------- 294.9/294.9 kB 17.8 MB/s eta 0:00:00
Downloading pytz-2024.1-py2.py3-none-any.whl (505 kB)
   ---------------------------------------- 0.0/505.5 kB ? eta -:--:--
   --------------------------------------- 505.5/505.5 kB 16.0 MB/s eta 0:00:00
Downloading tzdata-2024.1-py2.py3-none-any.whl (345 kB)
   ---------------------------------------- 0.0/345.4 kB ? eta -:--:--
   --------------------------------------- 345.4/345.4 kB 20.9 MB/s eta 0:00:00
Installing collected packages: pytz, tzdata, tabulate, pandas, seaborn
Successfully installed pandas-2.2.2 pytz-2024.1 seaborn-0.13.2 tabulate-0.9.0 tzdata-2024.1
```

Code

In [ ]:

```python
import cv2
from keras.applications import vgg16
from keras.preprocessing import image
from keras.applications.vgg16 import preprocess_input, decode_predictions
import numpy as np
from os import listdir
from os.path import isfile, join
```

In [ ]:

```python
# Define the path to your images
IMAGE_PATH = "images/"

# Load the VGG16 model
vgg_model = vgg16.VGG16(weights='imagenet')
```

In [ ]:

```python
def load_and_preprocess_image(img_path):
    target_size = (224, 224)  # VGG16 uses 224x224 images
    img = image.load_img(img_path, target_size=target_size)
    x = image.img_to_array(img)
    x = np.expand_dims(x, axis=0)
    x = preprocess_input(x)
    return x

def get_predictions(model, x):
    preds = model.predict(x)
    return decode_predictions(preds, top=3)[0]

def draw_test(name, predictions, input_im):
    BLACK = [0, 0, 0]
    # Calculate needed expansion to fit text
    extra_width = max(len(pred[1]) for pred in predictions) * 20
    expanded_image = cv2.copyMakeBorder(input_im, 0, 0, 0, input_im.shape[1] + extra_width, cv2.BORDER_CONSTANT
    img_width = input_im.shape[1]
    cv2.putText(expanded_image, str(name), (img_width + 10, 30), cv2.FONT_HERSHEY_COMPLEX_SMALL, 1, (0, 0, 255)
    y_offset = 60
    for i, prediction in enumerate(predictions):
        string = f"{prediction[1]}: {prediction[2]:.2f}"
        cv2.putText(expanded_image, string, (img_width + 10, y_offset + (i * 30)), cv2.FONT_HERSHEY_COMPLEX_SMA
    cv2.imshow(name, expanded_image)

def process_images():
    file_names = [f for f in listdir(IMAGE_PATH) if isfile(join(IMAGE_PATH, f))]

    for file in file_names:
        img_path = join(IMAGE_PATH, file)
        x = load_and_preprocess_image(img_path)

        # Load image using opencv for display
        img_display = cv2.imread(img_path)
        img_display = cv2.resize(img_display, None, fx=0.5, fy=0.5, interpolation=cv2.INTER_CUBIC)

        # Get predictions from VGG16 model
        predictions_vgg = get_predictions(vgg_model, x)

        # Display results
        draw_test(f"VGG16 Predictions - {file}", predictions_vgg, img_display)
        cv2.waitKey(0)  # Wait for key press to continue

    cv2.destroyAllWindows()
```

In [ ]:

```python
# Main function to execute the process
if __name__ == '__main__':
    process_images()
```

```
1/1 ━━━━━━━━━━━━━━━━━━━━ 1s 903ms/step
```

Output Explanation

Example Output Interpretation: When you run the script, for each image, it displays:

- Name of the image file.
- Top 3 predictions where each line shows:
  - The predicted category.
  - The model's confidence in that prediction expressed as a percentage.

For instance, if the output for an image is:

```
VGG16 Predictions - cat.jpg
Persian cat: 0.45
Tabby cat: 0.30
Siamese cat: 0.10
```

This means:

- The model is 45% confident that the image is of a Persian cat.
- The second most likely category, according to the model, is a tabby cat, with 30% confidence.
- The third guess is a Siamese cat, with 10% confidence.

Loading [MathJax]/jax/output/CommonHTML/fonts/TeX/fontdata.js