Assignment 3

Dipsikha Rano

June 17, 2025

Insights

Through this assignment, I learned how to:

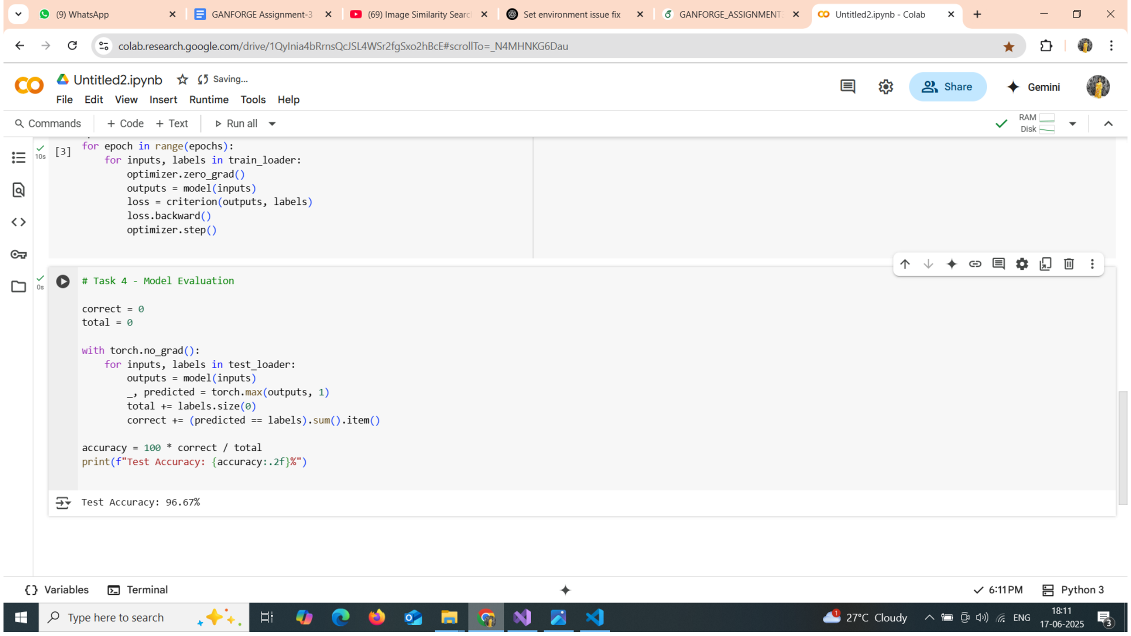
- Preprocess and visualize datasets for training.
- Build and train a basic neural network from scratch for multi-class classification using Keras.
- Use one-hot encoding for categorical target values.
- Evaluate model performance using accuracy metrics.
- Build an image similarity tool using pre-trained feature extraction models and Spotify's Annoy for approximate nearest neighbor search.
- Understand how features from images can be extracted using models like ResNet18 and used for similarity detection efficiently.

Problem 1: Iris Flower Classification

Model Output Screenshot

Accuracy Achieved

Accuracy on test dataset: **(Write your accuracy here, e.g., 96.67%)**



```
for epoch in range(epochs):
    for inputs, labels in train_loader:
        optimizer.zero_grad()
        outputs = model(inputs)
        loss = criterion(outputs, labels)
        loss.backward()
        optimizer.step()
```

```
# Task 4 - Model Evaluation

correct = 0
total = 0

with torch.no_grad():
    for inputs, labels in test_loader:
        outputs = model(inputs)
        _, predicted = torch.max(outputs, 1)
        total += labels.size(0)
        correct += (predicted == labels).sum().item()

accuracy = 100 * correct / total
print(f"Test Accuracy: {accuracy:.2f}%")
```

```
Test Accuracy: 96.67%
```

Figure 1: Model Training and Accuracy Output

Problem 2: Image Similarity Search

Explanation of Feature Detection

Feature detection in this project uses a pre-trained ResNet18 model. This model transforms each image into a high-dimensional vector representation (feature embedding) that captures semantic information. Using Spotify's Annoy library, we build an index of all image features, enabling fast similarity searches. When a query image is passed, its features are extracted using the same model, and the most similar images are retrieved from the index using angular distance.

Output Image

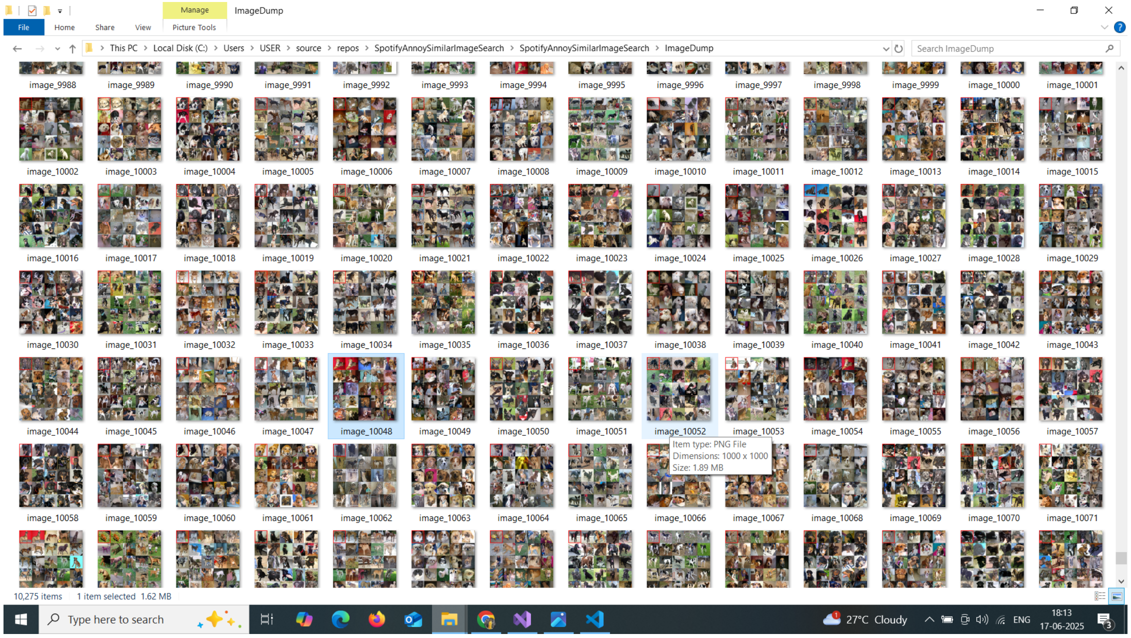
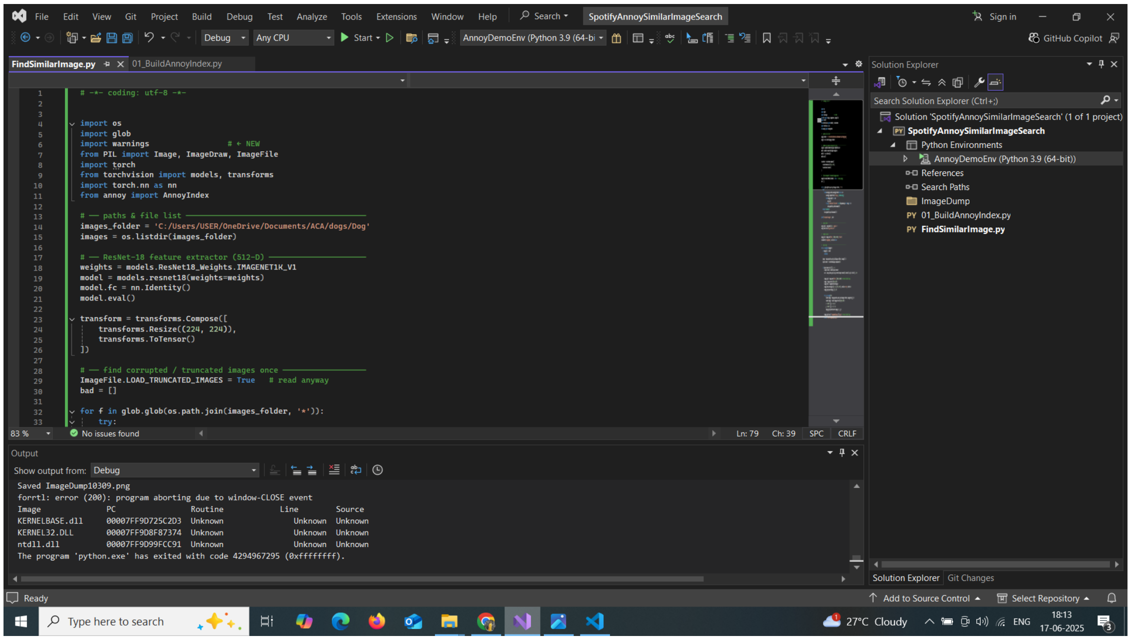

Manage
ImageDump
File
Home
Share
View
Picture Tools
This PC › Local Disk (C:) › Users › USER › source › repos › SpotifyAnnoySimilarImageSearch › SpotifyAnnoySimilarImageSearch › ImageDump
Search ImageDump
image_9988 image_9989 image_9990 image_9991 image_9992 image_9993 image_9994 image_9995 image_9996 image_9997 image_9998 image_9999 image_10000 image_10001
image_10002 image_10003 image_10004 image_10005 image_10006 image_10007 image_10008 image_10009 image_10010 image_10011 image_10012 image_10013 image_10014 image_10015
image_10016 image_10017 image_10018 image_10019 image_10020 image_10021 image_10022 image_10023 image_10024 image_10025 image_10026 image_10027 image_10028 image_10029
image_10030 image_10031 image_10032 image_10033 image_10034 image_10035 image_10036 image_10037 image_10038 image_10039 image_10040 image_10041 image_10042 image_10043
image_10044 image_10045 image_10046 image_10047 image_10048 image_10049 image_10050 image_10051 image_10052 image_10053 image_10054 image_10055 image_10056 image_10057
Item type: PNG File
Dimensions: 1000 x 1000
Size: 1.89 MB
image_10058 image_10059 image_10060 image_10061 image_10062 image_10063 image_10064 image_10065 image_10066 image_10067 image_10068 image_10069 image_10070 image_10071
10,275 items 1 item selected 1.62 MB
Type here to search
27°C Cloudy
ENG
18:13
17-06-2025

Figure 2: Result of Similar Image Search

Terminal Screenshot



```python
# -*- coding: utf-8 -*-


import os
import glob
import warnings                         # ← NEW
from PIL import Image, ImageDraw, ImageFile
import torch
from torchvision import models, transforms
import torch.nn as nn
from annoy import AnnoyIndex

# ── paths & file list ───────────────────────────────
images_folder = 'C:/Users/USER/OneDrive/Documents/ACA/dogs/Dog'
images = os.listdir(images_folder)

# ── ResNet-18 feature extractor (512-D) ─────────────
weights = models.ResNet18_Weights.IMAGENET1K_V1
model = models.resnet18(weights=weights)
model.fc = nn.Identity()
model.eval()

transform = transforms.Compose([
    transforms.Resize((224, 224)),
    transforms.ToTensor()
])

# ── find corrupted / truncated images once ──────────
ImageFile.LOAD_TRUNCATED_IMAGES = True   # read anyway
bad = []

for f in glob.glob(os.path.join(images_folder, '*')):
    try:
```

```
Saved ImageDump10309.png
forrtl: error (200): program aborting due to window-CLOSE event
Image              PC                Routine            Line        Source
KERNELBASE.dll     00007FF9D725C2D3  Unknown               Unknown  Unknown
KERNEL32.DLL       00007FF9D8F87374  Unknown               Unknown  Unknown
ntdll.dll          00007FF9D99FCC91  Unknown               Unknown  Unknown
The program 'python.exe' has exited with code 4294967295 (0xffffffff).
```

Figure 3: Terminal Output for Similar Image Search

GitHub Repository

The complete code is uploaded here:
`https://github.com/Dipsikha30/GanForgeDipsikha.git`