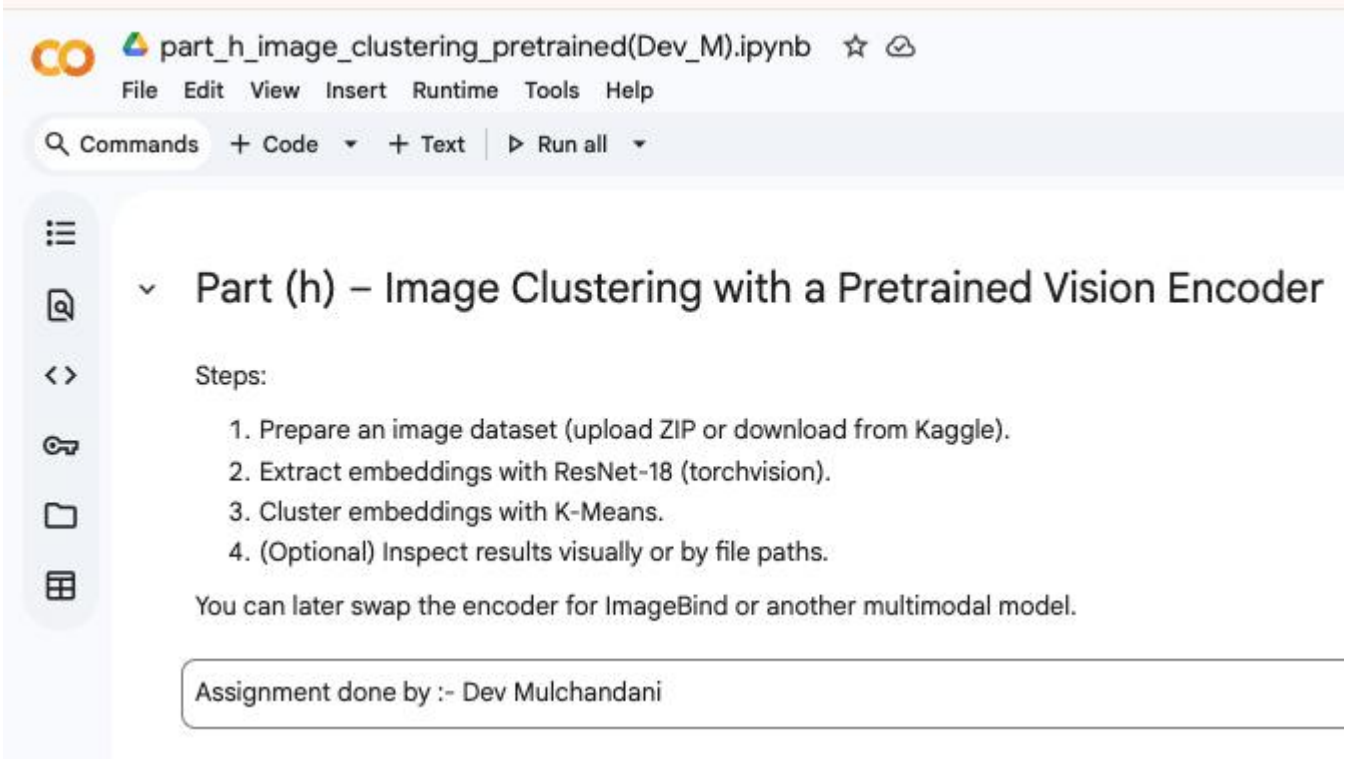
Clustering Assignments

❖ Assignment done by :- Dev Mulchandani

❖ Part-H :– Image Clustering Using Pretrained Vision Models

In Part H, I created an image-based clustering workflow using a pretrained ResNet-18 convolutional neural network. Instead of training a model from scratch, I extracted deep visual embeddings from the images using the model's penultimate layer. I then clustered these embeddings using K-Means and evaluated the results with silhouette score. Finally, I grouped and listed sample images from each cluster to understand their visual similarity. This exercise demonstrated how transfer learning enables high-quality image clustering using powerful pretrained vision models.

❖ Screenshots:-



Prepare image dataset (upload or Kaggle)

[9] ✓ 12s

```python
# @title Prepare image dataset (upload or Kaggle)
import os, zipfile
from pathlib import Path

try:
    from google.colab import files  # type: ignore
    IN_COLAB = True
except Exception:
    IN_COLAB = False

data_dir = Path("images")
data_dir.mkdir(exist_ok=True)

print("Option 1: upload a ZIP that contains image files.")
print("Option 2: use Kaggle to download an image dataset, then unzip into 'images/'.")
mode = input("Type 'upload' or 'kaggle' (or 'skip' if already prepared): ").strip().lower()

if mode == "upload":
    if not IN_COLAB:
        raise RuntimeError("Upload only works in Colab.")
    uploaded = files.upload()
    for fn in uploaded:
        if fn.lower().endswith(".zip"):
            with zipfile.ZipFile(fn, "r") as zf:
                zf.extractall(data_dir)
elif mode == "kaggle":
    if IN_COLAB:
        print("Please upload kaggle.json from your Kaggle account.")
        uploaded = files.upload()
        kaggle_path = Path("~/.kaggle").expanduser()
        kaggle_path.mkdir(parents=True, exist_ok=True)
        for fn in uploaded:
            Path(fn).replace(kaggle_path / "kaggle.json")
        os.chmod(kaggle_path / "kaggle.json", 0o600)

    !pip -q install kaggle
    DATASET_SLUG = input("Enter Kaggle dataset slug (e.g. 'zalando-research/fashionmnist'): ").strip()
    !kaggle datasets download -d $DATASET_SLUG -p images
    for zpath in data_dir.glob("*.zip"):
        with zipfile.ZipFile(zpath, "r") as zf:
            zf.extractall(data_dir)
else:
    print("Skipping dataset download; make sure 'images/' contains image files.")
```

```
Option 1: upload a ZIP that contains image files.
Option 2: use Kaggle to download an image dataset, then unzip into 'images/'.
Type 'upload' or 'kaggle' (or 'skip' if already prepared): upload
Choose files sample_100_images.zip
sample_100_images.zip(application/zip) - 336042 bytes, last modified: 02/12/2025 - 100% done
Saving sample_100_images.zip to sample_100_images.zip
```

Extract embeddings with ResNet-18

[10] ✓ 13s

```
# @title Extract embeddings with ResNet-18
!pip -q install torch torchvision

import torch
from torchvision import models, transforms
from PIL import Image
import numpy as np
from pathlib import Path

device = "cuda" if torch.cuda.is_available() else "cpu"
print("Using device:", device)

model = models.resnet18(weights=models.ResNet18_Weights.DEFAULT)
model.fc = torch.nn.Identity()
model = model.to(device)
model.eval()

preprocess = 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],
    ),
])
```

```
image_paths = [p for p in Path("images").rglob("*") if p.suffix.lower() in [".jpg", ".jpeg", ".png"]]
print("Found", len(image_paths), "images.")

embeddings_list = []
valid_paths = []
for p in image_paths:
    try:
        img = Image.open(p).convert("RGB")
    except Exception:
        continue
    x = preprocess(img).unsqueeze(0).to(device)
    with torch.no_grad():
        emb = model(x).cpu().numpy()
    embeddings_list.append(emb[0])
    valid_paths.append(p)

if not embeddings_list:
    raise ValueError("No valid images found in 'images/'.")

embeddings = np.vstack(embeddings_list)
print("Embeddings shape:", embeddings.shape)
```

```
Using device: cpu
Found 105 images.
Embeddings shape: (105, 512)
```

Cluster image embeddings

[11] ✓ 0s

```python
# @title Cluster image embeddings
from sklearn.cluster import KMeans
from sklearn.metrics import silhouette_score

k = 5
kmeans = KMeans(n_clusters=k, random_state=42, n_init=10)
labels = kmeans.fit_predict(embeddings)

score = silhouette_score(embeddings, labels)
print(f"Silhouette score: {score:.3f}")

from collections import defaultdict
clusters = defaultdict(list)
for path, lab in zip(valid_paths, labels):
    clusters[lab].append(str(path))

for lab, files_in_cluster in clusters.items():
    print("=" * 70)
    print("Cluster", lab)
    for f in files_in_cluster[:15]:
        print("-", f)
```

```
Silhouette score: 0.156
======================================================================
Cluster 4
- images/car_31.jpg
- images/car_98.jpg
- images/car_96.jpg
- images/car_52.jpg
- images/car_13.jpg
- images/car_38.jpg
- images/car_22.jpg
- images/car_47.jpg
- images/car_28.jpg
- images/car_76.jpg
- images/car_42.jpg
- images/car_26.jpg
- images/car_62.jpg
- images/car_11.jpg
- images/car_24.jpg
======================================================================
Cluster 1
- images/car_2.jpg
- images/car_65.jpg
- images/car_41.jpg
- images/car_53.jpg
- images/car_86.jpg
- images/car_78.jpg
- images/car_23.jpg
- images/car_51.jpg
- images/car_16.jpg
- images/car_63.jpg
- images/car_82.jpg
- images/car_48.jpg
- images/car_9.jpg
- images/car_89.jpg
- images/car_67.jpg
======================================================================
```

```
==================================================
Cluster 2
- images/sample_1.jpg
- images/sample_4.jpg
- images/sample_3.jpg
- images/sample_2.jpg
- images/sample_0.jpg
==================================================
Cluster 3
- images/car_7.jpg
- images/car_79.jpg
- images/car_66.jpg
- images/car_34.jpg
- images/car_77.jpg
- images/car_40.jpg
- images/car_95.jpg
- images/car_87.jpg
- images/car_80.jpg
- images/car_10.jpg
- images/car_39.jpg
- images/car_97.jpg
- images/car_15.jpg
- images/car_60.jpg
- images/car_45.jpg
==================================================
Cluster 0
- images/car_30.jpg
- images/car_0.jpg
- images/car_6.jpg
- images/car_73.jpg
- images/car_75.jpg
- images/car_50.jpg
- images/car_64.jpg
- images/car_32.jpg
- images/car_3.jpg
- images/car_58.jpg
- images/car_94.jpg
- images/car_44.jpg
- images/car_81.jpg
- images/car_72.jpg
- images/car_17.jpg
```