Dimensionality's Blessing: Clustering Images by Underlying Distribution

I. Implementation Details

A. Process Flow

- The process begins by loading images from a specified directory, using the **ResNet Architecture** to extract features from each image.
- The extracted features are stored in a feature matrix, which serves as the input for various clustering algorithms.
- Multiple clustering methods are applied to the feature matrix, including:
  - KMeans Clustering
  - Gaussian Mixture Model (GMM) Clustering
  - Spectral Clustering
  - **Distribution-based Clustering** using affinity matrices.
- For each clustering method, the labels are generated, and the clustering results are saved in respective directories.
- The clustering outcomes are then evaluated using various metrics such as:
  - Silhouette Score
  - Calinski-Harabasz Score
  - Davies-Bouldin Score
- Finally, the evaluation results are compiled into a **Pandas DataFrame** for easier comparison and visualization of clustering performance across different methods.
- The implementation aims to generate and display clusters for a sample dataset and can generate results for other datasets

B. Understanding Distribution clustering implementation

1) **Feature Extraction**: The algorithm uses a pre-trained AlexNet model, implemented in the `TestModel` class, to extract deep features from images. The model reduces the feature maps to a one-dimensional vector using adaptive pooling and flattening, capturing important characteristics for each image.
2) **Distance Calculation**: Once features are extracted, pairwise squared Euclidean distances are computed between the image feature vectors. This step is performed in the `cluster` function, which forms the foundation for clustering by determining how "close" images are in the feature space.
3) **Affinity Matrix Computation**: The algorithm calculates a second-order distance based on the pairwise distances to create an affinity matrix. The affinity matrix represents similarities between the distributions of the images, as implemented in the `compute_affinity_matrix` function.
4) **Clustering with Distribution Clustering Algorithm**: The algorithm iteratively groups images based on second-order distances using the `distribution_clustering` function. Images that have low second-order distances form candidate clusters, and only groups that meet the minimum size are assigned valid clusters.
5) **Outlier Handling**: Images that do not fit well into existing clusters are handled as outliers and placed into separate clusters, ensuring that the main clusters remain pure. This is part of the robustness provided by the `distribution_clustering` function.

II. Datasets

A. Testing Dataset

The unit test dataset used for evaluation can be found at the following link: Unit Test Dataset. It contains 100 images of colesium and the clustered images are shown in the implementation in `Main.ipynb`

B. Additional Evaluation

For evaluation and comparison following datasets are used and in stored in Other Dataset folder in the repository:

- **MNIST** - A samller sample of the digit recoginition dataset used of testing of classification.
- **CalTech 101** - Some folders at random chosen from the Caltech 101 dataset for evaluation containing various object/ animals images.

III. Results

Scoring result on the given test dataset and the MNIST dataset for all the clustering algorithms. More results can be generated by changing the folder name in the Main.ipynb file in the github repository.

method	Silhouette Score	Calinski-Harabasz Index	Davies-Bouldin Index
kmeans	0.024251	4.933275	2.036408
gmm	0.044698	5.367283	2.150070
spectral	0.037343	4.938626	2.277607
distribution	0.028362	3.192529	2.287611

method	Silhouette Score	Calinski-Harabasz Index	Davies-Bouldin Index
kmeans	0.092072	26.127189	2.183616
gmm	0.097487	27.322520	2.262916
spectral	0.070050	26.046151	2.235247
distribution	0.189591	17.558190	1.549544

Fig. 1. Table 1 contains result on Sample Dataset and Table 2 contains result on MNIST Datatset

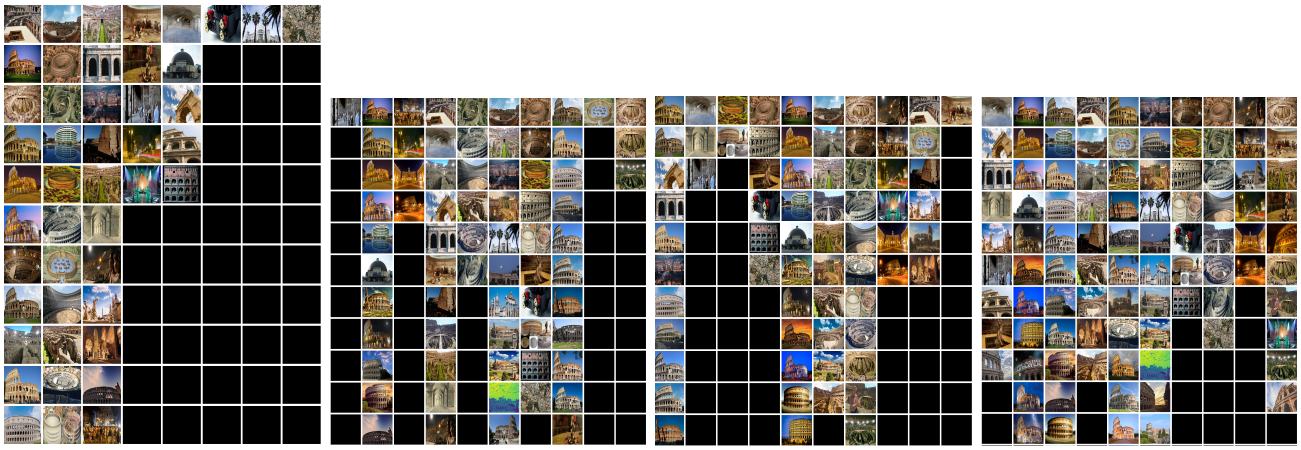
A. Analysis

- The Silhouette Score, Calinski-Harabasz Score, and Davies-Bouldin Score are popular evaluation metrics used to assess the quality of clustering algorithms. The Silhouette Score measures how similar an object is to its own cluster compared to other clusters, with values ranging from -1 to 1. A higher Silhouette Score indicates well-separated clusters, with 1 being ideal, 0 indicating overlapping clusters, and negative values suggesting incorrect clustering. The Calinski-Harabasz Score evaluates cluster dispersion and compactness, where a higher score means more distinct and well-separated clusters. There is no upper bound, but higher values are better. The Davies-Bouldin Score measures the average similarity ratio of each cluster with its most similar cluster, with lower values indicating better clustering. The ideal score is close to 0, as it reflects that clusters are compact and well-separated. We can see the distribution clustering method doing a decent performance in comparision with the other methods.

B. Challenges

- The code available in the paper was not working properly and the files suggested in the README were also missing, thus I was needed to rewrite the whole code understanding the paper and the available code.
- All the mentioned dataset were not publicly available and there were some ambiguity in the dataset used as well like some of the Caltech image dataset. Thus some of the datasets were taken randomly for checking apart from given one.
- There were no true labels present and thus purity score couldn't be calculated as done in the paper and thus some alternatives were used to compare the clustering performances.
- Results thus not exactly matching with the paper, yet the algorithm aims to satisfy the conditions mentioned in the paper and giving a decent performance for the clustering aspect.

C. Some Sample Clusters



- Distribution Clustering Kmeans GMM Spectral