Efficient Multi-View Clustering via Greedy Automatic View Selection and Diverse Feature Integration

Jyoti Mankar
Department of Computer Engineering, K. K. Wagh Institute of Engineering Education and Research Panchavati

Snehal Kamalapur
Department of Computer Engineering, K. K. Wagh Institute of Engineering Education and Research Panchavati

https://doi.org/10.5109/7388866

出版情報：Evergreen. 12 (3), pp.1802-1826, 2025-09. 九州大学グリーンテクノロジー研究教育センター
バージョン：

Efficient Multi-View Clustering via Greedy Automatic View Selection and Diverse Feature Integration

Jyoti Mankar[1,*], Snehal Kamalapur[1]

[1]Department of Computer Engineering, K. K. Wagh Institute of Engineering Education and Research Panchavati, Nashik-422003, Maharashtra, India

*Author to whom correspondence should be addressed:
E-mail: jrmankar@kkwagh.edu.in

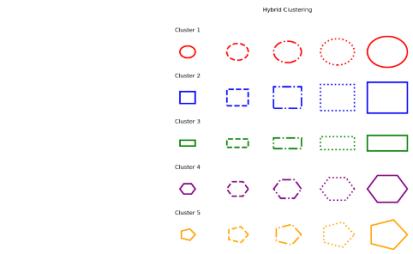
(Received May 30, 2025; Revised September 01, 2025; Accepted September 04, 2025)

Abstract: Multi-view clustering leverages complementary information from multiple feature representations, yet its success relies on selecting optimal feature combinations and clustering algorithms. We propose a Greedy Automatic View Selection (GAVS) algorithm to identify the most informative subset of feature views that maximize clustering performance. GAVS iteratively adds feature views based on their contribution to clustering quality, measured by normalized mutual information (NMI). We evaluate GAVS on Coil20, UCI Digits, Movies, and Caltech 7 datasets using Spectral, Agglomerative, and Affinity Propagation clustering with diverse features (GIST, LBP, HOG, CENTRIST). Results show optimal combinations vary across datasets, with GAVS achieving peak NMIs of 1.000 (Coil20), 0.9351 (UCI Digits), 0.6937 (Movies), and 0.9806 (Caltech 7). This adaptive strategy offers practical guidance for improving clustering accuracy in real-world applications.

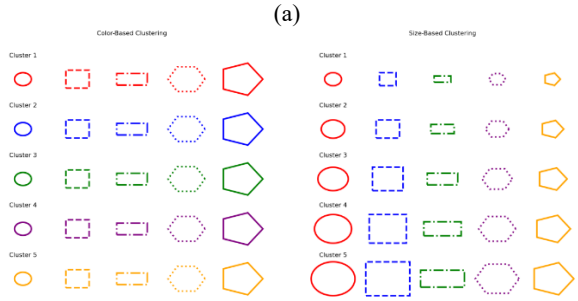
Keywords: Benchmark Datasets; Clustering Algorithms; feature extractions; Multiview clustering

1. Introduction

Clustering is a fundamental unsupervised machine learning technique widely used in various domains to group similar data points based on their inherent characteristics. Traditional clustering methods, such as K-means, hierarchical clustering, and Gaussian Mixture Models (GMM), operate under the assumption that data can be represented using a single feature space. However, in real-world scenarios, data is often inherently multi-faceted, containing information from multiple perspectives or "views." For example, in image analysis, an object can be described using different feature sets such as shape, texture, and color, each providing distinct but complementary information. Similarly, in text mining, documents can be represented by different features, including term frequency, topic distributions, and word embeddings[1)]. Multiview clustering (MVC) aims to integrate multiple feature representations into the clustering process, leading to improved clustering accuracy and robustness[2)]. The multi-view clustering problem as shown in Figure 1, involves grouping geometric shapes (circles, hexagons, squares, rectangles, and pentagons) based on different views or representations of the same shape. Each shape may appear in different styles, such as variations in color, size, orientation, or boundary styles, representing multiple views of the same underlying category. The goal of multi-view clustering is to leverage these diverse representations to improve clustering performance by integrating information from multiple perspectives. Instead of clustering based on a single feature space, this approach combines multiple feature sets to achieve more accurate and robust clustering results.

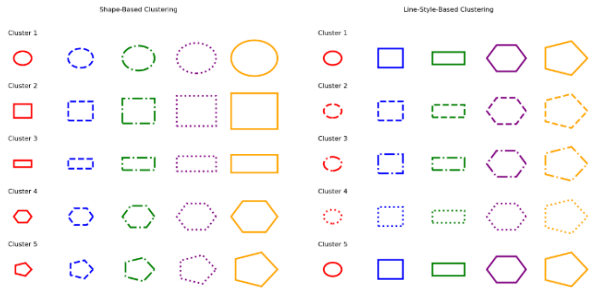


(a)



(b) (c)

Cite: J. Mankar, S. Kamalapur, "Efficient Multi-View Clustering via Greedy Automatic View Selection and Diverse Feature Integration". Evergreen, 12 (03) 1802-1826 (2025). https://doi.org/10.5109/7388866.



(d) (e)

Fig. 1: Multiview clustering problem

Multi-view clustering is an advanced technique in unsupervised learning that aims to group data points by leveraging multiple feature representations or perspectives. In the given implementation, geometric shapes serve as a visual metaphor for multi-view clustering, where each subplot represents a different clustering approach: shape-based, color-based, line-style-based, size-based, and hybrid clustering. These perspectives reflect how different features contribute to grouping patterns, similar to real-world scenarios where diverse data sources (e.g., textual, visual, or numerical) provide complementary information for clustering. By simultaneously analyzing multiple clustering solutions within a single framework, multi-view clustering enhances robustness, mitigates biases from individual views, and improves overall cluster quality. This visualization highlights how different clustering criteria lead to varied structures, emphasizing the importance of considering multiple perspectives in unsupervised learning tasks, especially in applications like bioinformatics, social network analysis, and computer vision. Multiview clustering has gained significant attention in recent years due to its ability to leverage complementary information from different data perspectives. By integrating multiple views, multiview clustering can better capture the underlying data distribution, enhance clustering stability, and provide more interpretable results. Various approaches have been developed for multiview clustering, including co-training-based methods, subspace learning methods, and deep learning-based techniques[3]. Co-training methods iteratively update cluster assignments by training on separate views and enforcing consistency across them. Subspace learning methods attempt to find a shared latent space where the clustering structure is more evident. Deep learning-based methods use neural networks to learn view-specific representations while enforcing cross-view consistency[4]. Despite these advancements, multiview clustering still faces significant challenges that limit its practical application. Issues such as view heterogeneity, inconsistencies in feature alignment, computational scalability, and missing views pose obstacles to achieving robust and interpretable clustering outcomes[5]. This paper aims to systematically explore these challenges, review state-of-the-art solutions, and propose novel approaches to improve the efficiency, effectiveness, and applicability of multiview clustering in real-world scenarios[6]. While multiview clustering offers advantages over traditional single-view clustering, it also introduces several complexities that must be addressed to achieve optimal performance. These challenges primarily stem from variations in feature representation, inconsistencies between views, computational demands, missing data, and model interpretability. One of the fundamental challenges in multiview clustering is the heterogeneity of data representations across different views. In practical applications, different feature spaces may have varying distributions, scales, and dimensional structures, making it difficult to integrate them effectively[7]. Some views may contain redundant or noisy information, which can distort the clustering structure and lead to suboptimal results. Additionally, not all views contribute equally to clustering performance, and determining the relative importance of each view remains a complex problem. Addressing this challenge requires the development of view-weighting mechanisms or adaptive learning strategies that dynamically adjust the importance of different views during clustering[8]. Another major issue in multiview clustering is ensuring consistency and alignment between different views. While different views provide complementary information, they may not always be aligned due to variations in data collection methods, feature extraction techniques, or domain-specific differences. For example, in medical image analysis, different imaging modalities such as MRI, CT scans, and PET scans provide distinct perspectives on the same anatomical structure, but discrepancies in resolution, contrast, or orientation can create misalignment. Effective clustering requires robust alignment techniques that can map different feature spaces onto a common latent space while preserving structural relationships within the data[9]. Multiview clustering techniques often involve integrating multiple high-dimensional feature spaces, which significantly increases computational complexity. Many traditional clustering algorithms do not scale well in multiview settings, especially when dealing with large datasets in domains such as genomics, e-commerce, and social network analysis. The increased dimensionality and dependency on multiple feature spaces lead to greater memory requirements and longer processing times. This challenge necessitates the development of efficient dimensionality reduction, feature selection, and optimization techniques that can handle large-scale multiview clustering problems with reduced computational overhead[10]. In real-world scenarios, some views may be missing or incomplete due to limitations in data collection processes. For instance, in multimodal sentiment analysis, text, audio, and video data may not always be available simultaneously due to sensor failures, recording limitations, or privacy concerns[11]. In such cases,

Cite: J. Mankar, S. Kamalapur, "Efficient Multi-View Clustering via Greedy Automatic View Selection and Diverse Feature Integration". Evergreen, 12 (03) 1802-1826 (2025). https://doi.org/10.5109/7388866.

standard multiview clustering methods struggle to maintain performance when information from one or more views is unavailable. Handling missing views effectively requires the use of imputation techniques, generative models, or graph-based methods that can estimate missing data while maintaining clustering consistency[12]. While deep learning-based multiview clustering methods have shown promising results, they often suffer from poor interpretability. Many deep clustering models act as black boxes, making it difficult to understand how different views contribute to the final clustering decisions[13]. This lack of transparency limits their applicability in high-stakes domains such as healthcare, finance, and cybersecurity, where interpretability is critical for trust and decision-making. Additionally, generalizing multiview clustering models across different datasets and application domains remains a challenge[14]. Many models are optimized for specific data distributions and do not adapt well to new datasets with varying feature spaces. Developing explainable and generalizable multiview clustering frameworks is essential for broader real-world adoption[15]. The rapid advancements in machine learning, clustering techniques, classification models, and material property optimization have led to significant improvements across multiple scientific and engineering domains. This literature review synthesizes recent studies, covering spectral clustering, synthetic data balancing, material property enhancements, and classification techniques. Spectral clustering has gained traction as an effective approach for high-dimensional data clustering. Abhadiomhen et al. [12] conducted a comparative analysis of spectral clustering techniques, emphasizing their robustness in handling complex datasets. Liu et al. [16] introduced a multiview spectral clustering method leveraging a weighted tensor low-rank constraint, which demonstrated improved clustering accuracy and reduced computational costs. Xu et al. [17] proposed a cooperative manifold learning technique that integrates low-rank representation, enhancing clustering stability and precision. Additionally, Gao et al. [18] developed a low-rank correlation representation method, improving clustering outcomes in unsupervised learning scenarios. Wang et al. [19] implemented a hybrid approach combining spectral clustering with deep learning techniques, demonstrating increased efficiency and adaptability in high-dimensional spaces. Classification models play a crucial role in predictive analytics. Traditional methods such as decision trees and support vector machines (SVMs) have been widely used for various applications. However, imbalanced datasets pose a challenge, often leading to biased predictions. To address this issue, researchers have explored synthetic data balancing techniques. Wang et al. [20] evaluated the impact of synthetic minority oversampling techniques (SMOTE) and NearMiss strategies, demonstrating their effectiveness in improving model robustness. Further, Tran et al. [21] applied reinforcement learning to optimize data resampling methods, achieving improved model generalization. Recent studies also highlight the effectiveness of hybrid balancing techniques, where machine learning and statistical methods are combined for better performance [22]. Despite the significant advancements in multiview clustering, there remain several critical gaps that hinder its full potential and practical implementation in real-world applications. One of the main challenges is the heterogeneity of data representations across different views. While existing methods aim to integrate multiple views, variations in the data distribution, scale, and dimensional structure across views often create difficulties in achieving seamless integration[23]. There is a need for more robust techniques that can effectively handle such heterogeneity, ensuring consistent performance across diverse datasets.

In the era of high-dimensional data, objects are often represented through multiple heterogeneous feature views that capture different aspects of the same entity. Multi-view clustering has emerged as a promising technique to exploit this diversity and improve clustering accuracy. However, a major challenge lies in determining which combination of feature views contributes most effectively to the clustering outcome. The inclusion of redundant or irrelevant views can degrade clustering performance, increase computational complexity, and obscure meaningful patterns. Furthermore, the effectiveness of a clustering algorithm can vary significantly depending on the dataset characteristics and the selected feature views. Despite advancements in multi-view clustering, there is a lack of systematic methods for selecting the most informative subset of views that ensure optimal clustering performance across different datasets. This study addresses this critical gap by introducing a Greedy Automatic View Selection (GAVS) algorithm to iteratively identify and combine feature views that contribute most significantly to clustering quality.

2. Methodology

This study employs a robust multi-view clustering methodology, utilizing benchmark datasets to evaluate clustering performance across multiple feature representations. The process involves data preprocessing, dimensionality reduction, feature fusion, clustering, and evaluation using a variety of statistical and machine learning techniques. The selected datasets for this study include COIL-20, Caltech-7 (Dollar Bill, Faces, Garfield (Cartoon Cat), Motorbike, Snoopy (Cartoon Dog), Stop Sign, Windsor Chair), UCI Digit, and Movies dataset. These datasets provide diverse challenges, including variations in image features, text attributes, and categorical data, making them ideal for benchmarking multi-view clustering techniques.

Cite: J. Mankar, S. Kamalapur, "Efficient Multi-View Clustering via Greedy Automatic View Selection and Diverse Feature Integration". Evergreen, 12 (03) 1802-1826 (2025). https://doi.org/10.5109/7388866.

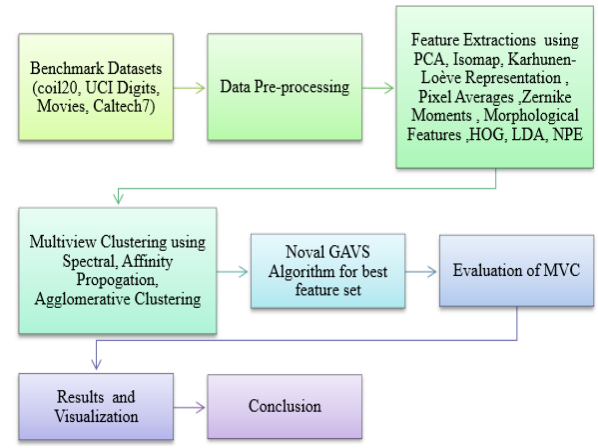
COIL-20 Dataset: The Columbia Object Image Library (COIL-20) consists of grayscale images of 20 different objects captured from various angles (0° to 360°) at 5° intervals, totaling 1,440 images. The dataset is commonly used in image clustering tasks where multiple views correspond to different angles of the same object[24].

Caltech-7 Dataset: This dataset is a subset of the larger Caltech-101 dataset, containing images from seven categories. Each image is represented using multiple feature descriptors such as Gabor filters, wavelet moments, and histogram-based representations, providing a strong test case for multi-view learning.

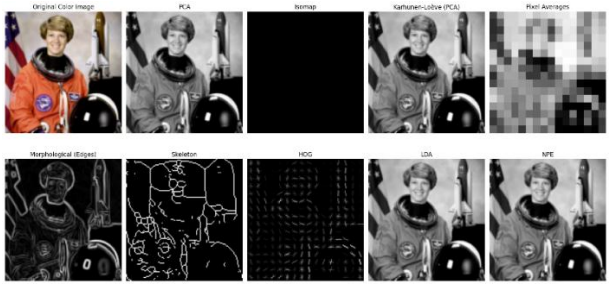
UCI Digit Dataset: This dataset comprises handwritten digits from 0 to 9, extracted from different sources such as postal mail and bank checks. It provides multiple feature representations, including pixel intensities, gradient-based features, and contour descriptors.

Movies Dataset: The Movies dataset consists of multiple views, including metadata (genre, director, cast), text reviews, and ratings. It is used to assess how well multi-view clustering techniques handle categorical and textual data alongside numerical features. The proposed methodology as shown in Figure 2a and 9 feature extractions are shown in Figure 2b, for multi-view clustering involves several stages, including data preprocessing, dimensionality reduction, feature fusion, clustering, and evaluation. The proposed methodology presents a novel and efficient approach to multi-view clustering by integrating a diverse set of nine feature extraction techniques, multiple clustering algorithms, and a unique Greedy Automatic View Selection (GAVS) strategy. This framework is designed to address the challenges of heterogeneity and redundancy in multi-view datasets by selecting the most informative feature subsets in a time-efficient manner. Traditional multi-view clustering methods often require manual selection or exhaustive evaluation of all possible view combinations, which becomes computationally intensive and impractical for high-dimensional data. In contrast, our GAVS algorithm adopts a greedy approach that incrementally identifies the most impactful views based on performance gains, thereby significantly reducing computational complexity without compromising clustering accuracy.

The novelty of this methodology lies in its comprehensive feature representation strategy. By employing nine distinct feature extraction methods—including both linear (PCA, LDA, Karhunen-Loève Transform) and nonlinear (Isomap, NPE) techniques, along with statistical and morphological descriptors (HOG, Zernike Moments, Pixel Averages, and morphological features)—the model captures complementary aspects of the data. This multi-perspective view enhances the robustness and discriminative power of the clustering process, allowing the algorithm to uncover deeper patterns within complex datasets. To evaluate clustering performance, we use a suite of metrics, including Adjusted Rand Index (ARI), Fowlkes-Mallows Index (FMI), Purity, Silhouette Score, and Normalized Mutual Information (NMI). Among these, NMI is selected as the primary evaluation criterion for the GAVS process because of its ability to measure the amount of shared information between predicted clusters and ground truth labels in a normalized and unbiased manner. Unlike ARI or FMI, which may be sensitive to the number of clusters or data imbalance, NMI provides a more stable and reliable measure across different datasets and clustering scenarios. Its bounded range [0, 1] and symmetry make it particularly suitable for view selection, ensuring consistent assessment of clustering quality at each step of the greedy selection process. Overall, this methodology offers a best-in-class balance between accuracy and efficiency. The GAVS mechanism eliminates the need to test all combinations of feature views, while the inclusion of varied feature descriptors ensures that important structural, statistical, and visual information is preserved. The combination of intelligent view selection, diverse feature fusion, and robust clustering evaluation makes this approach not only novel but also significantly more time-saving and scalable for real-world multi-view clustering applications.



(a) Methodology



(b) Feature Extractions using PCA, Isomap, Karhunen-Loève Representation ,Pixel Averages ,Zernike Moments , Morphological Features ,HOG, LDA, NPE

Fig. 2: (a) Methodology and (b) Feature extraction using 9 different methods

Cite: J. Mankar, S. Kamalapur, "Efficient Multi-View Clustering via Greedy Automatic View Selection and Diverse Feature Integration". Evergreen, 12 (03) 1802-1826 (2025). https://doi.org/10.5109/7388866.

2.1. Mathematical Model

Let,

$X=\{X_1, X_2,..., X_M\}$ be the dataset with N samples and M views.

Each view $X_m \epsilon R^{N\times d_m}$ represents a different feature extraction method.

The goal is to find an optimal clustering assignment C such that intra-cluster similarity is maximized and inter-cluster similarity is minimized.

Step 1: Feature Transformation for Multiview Data

Each view X_m undergoes a transformation:

$Z_m=f_m(X_m)$, $m\in\{1,2,...,M\}$

where f_m represents transformations such as PCA, ICA, Isomap, NMF, LDA, Random Projection, and HOG.

$Z=\{Z_1,Z_2,...,Z_M\}$, $Z_m \epsilon R^{N\times d_m}$

Step 2: Feature Fusion

To form a combined feature representation, we concatenate selected views:

$Z^*= [Z_{m1}, Z_{m2},...,Z_{mk}]$, $Z*\in RN\times d* Z^*\epsilon R^{N\times d^*}$

where $d^* = \sum_{i=1}^{k} d_{mi}$ is the total dimensionality of the combined views.

Step 3: Affinity Matrix Construction

For Spectral Clustering and Hierarchical Clustering, an affinity matrix A is computed[24]:

$$A_{ij} = \exp\left(-\frac{\left\|z_i^*-z_j^*\right\|^2}{\sigma^2}\right)$$

Where, σ is a scaling parameter.

Step 4: Clustering Formulation

2.1.1. Algorithm: Multiview Clustering Using Greedy Automatic View Selection and Spectral/Hierarchical/Affinity Propagation

Input:

Multiview dataset $X = \{X_1, X_2, ..., X_m\}$ with N samples and M views

Feature transformation methods: {PCA, ICA, Isomap, NMF, LDA, GRP, HOG,NPE}

Clustering method: {Spectral Clustering, Affinity Propagation, Hierarchical Clustering}

Number of clusters k (only for methods that require it)

Output:

Clustering assignment C

Evaluation scores: {ARI, NMI, FMI, Silhouette, Homogeneity, Completeness, V-Measure, Purity}

Step 1: Feature Extraction

For each view $X_m \in \mathbb{R}^{n\times dm}$:

a. Apply appropriate transformation:

PCA → Fourier Coefficients, Karhunen-Loève

ICA → Zernike Moments

Isomap → Profile Correlations

NMF → Pixel Averages

LDA → LDA Features

GRP → Morphological Features

HOG → HOG Features

NPE→ NPE Features

Store transformed features as $Z_m = f_m(X_m)$, forming $Z = \{Z_1, Z_2, ..., Z_m\}$

Step 2: Feature Fusion

Select a subset of views (e.g., $\{Z_1, Z_2, ..., Z_k\}$)

Concatenate the views:

$Z* = [Z_{m1}, Z_{m2}, ..., Z_{mk}] \in \mathbb{R}^{n\times d*}$

where $d* = \sum d_{mi}$ is the total fused feature dimension

Step 3: Similarity Matrix Construction

Compute pairwise similarities:

For Spectral/Hierarchical Clustering:

$A_{ij} = \exp(-\|Z_i* - Z_j*\|^2 / \sigma^2)$

For Affinity Propagation:

$S(i, k) = -\|Z_i* - Z_k*\|^2$

Step 4: Clustering

Apply the chosen clustering algorithm:

Cite: J. Mankar, S. Kamalapur, "Efficient Multi-View Clustering via Greedy Automatic View Selection and Diverse Feature Integration". Evergreen, 12 (03) 1802-1826 (2025). https://doi.org/10.5109/7388866.

Spectral Clustering:

a. Construct Laplacian matrix L = D − A

b. Compute eigenvectors of L corresponding to k smallest eigenvalues

c. Apply k-means on eigenvector space

Affinity Propagation:

a. Initialize responsibility R and availability A matrices

b. Iteratively update:

$R(i,k) = S(i,k) - \max_{k' \neq k} \{A(i,k') + S(i,k')\}$

$A(i,k) = \min(0, R(k,k) + \sum_{i' \neq i,k} \max(0, R(i',k)))$

c. Select exemplars and assign cluster labels

Hierarchical Clustering:

a. Compute distance matrix $D_{ij} = \|Z_i* - Z_j*\|$

b. Apply linkage method (average, complete, or single)

c. Cut dendrogram to obtain k clusters

Step 5: Evaluation

Compute clustering evaluation metrics:

Adjusted Rand Index (ARI)

Normalized Mutual Information (NMI)

Fowlkes-Mallows Index (FMI)

Silhouette Score

Homogeneity, Completeness, V-Measure

Purity Score

Step 6: Greedy Feature Combination Search (Detailed)

Initialization:

Start by evaluating each individual view's transformed features separately.

For each view, apply the clustering algorithm and calculate the evaluation metric ***NMI***

Identify the single view with the highest NMI score. This view becomes the first selected feature set.

Initialize the current best fused feature matrix as the features from this selected view.

Keep track of the selected views in a list (e.g., *selected_views = [best_single_view]*).

Iterative Addition of Views:

For the remaining views that are ***not yet selected***, do the following:

a. Temporarily concatenate the features of one candidate view to the currently selected fused features to form a new combined feature matrix.

b. Perform clustering on this new fused feature matrix.

c. Calculate the NMI (or chosen evaluation metric) for the clustering result.

d. Compare this NMI with the current best NMI score.

Decision on Adding a View:

If adding the candidate view results in an improvement in NMI by at least a predefined threshold (for example, 0.01), then:

Update the selected views list to include this new view.

Update the current best fused feature matrix to include this view's features.

Update the current best NMI score to this new higher value.

Otherwise, discard the candidate view from consideration for this iteration.

Repeat Until No Improvement:

Repeat the iterative addition process by testing all remaining unselected views in the same way.

Continue adding views one-by-one, only if they improve the NMI score beyond the threshold.

Stop the iteration when no remaining views can improve the NMI score by the threshold.

Final Output:

The final selected subset of views and their fused features constitute the best performing feature combination.

Proceed with clustering and evaluation using this final fused feature set.

Step 7: Visualization

Use t-SNE on Z* to project to 2D

Plot clustered data points for qualitative analysis

End of Algorithm

2.1.2. Algorithm: Greedy Automatic View Selection (GAVS) for Multi-View Clustering

Input:

V = {v1, v2, ..., vm}: A set of *m* feature views extracted from the data (e.g., PCA, HOG, ICA features). Each view *vi* is a matrix of shape *n* × *di*, where *n* is the number of samples and *di* is the number of features in that view.

y: Ground-truth labels for the *n* samples (used only for evaluation).

ε > 0: A small positive threshold (e.g., 0.01) indicating the minimum improvement in clustering performance needed to add a new view.

C: Clustering algorithm (e.g., Spectral Clustering) with fixed parameters.

Output:

S ⊆ V: A selected subset of views that gives the best clustering results.

FS: The final combined feature matrix formed by concatenating the views in S.

M: Evaluation metrics (such as NMI, ARI, or Purity) calculated from clustering FS.

Step 1: Initialization

Start with an empty selected view set:

S ← ∅

Initialize the best clustering performance score:

*q ← 0**

For each view **vi** in **V**:

Perform clustering using algorithm **C** on **vi**

Evaluate clustering using NMI against ground-truth labels **y**

Identify the single view **vb** that gives the highest NMI score.

Update:

S ← {vb} (select the best performing view)

*q ← NMI(vb)** (store its clustering score)

FS ← vb (initial combined feature matrix)

Step 2: Greedy Iterative Selection

Set **R ← V \ S**, the set of remaining views not yet selected.

Repeat the following steps until no significant improvement is observed:

a. For each view **vc** in **R**:

Concatenate the current features **FS** with **vc** horizontally to form **F_candidate = [FS | vc]**

Perform clustering on **F_candidate** using algorithm **C**

Evaluate clustering with NMI against **y**, and store the score as **qc**

b. Find the view **v*** in **R** that gives the highest **qc** score

c. If **qc − q*** > ε (i.e., improvement is significant):

Update selected views: *S ← S ∪ {v}**

Update best score: *q ← qc**

```
Update combined features: FS ← [FS | v]*
Remove selected view from remaining views: R ← R \ {v}*
d. Else:
Stop the iteration as no significant improvement can be achieved
Step 3: Output
Return:
S: The final set of selected views
FS: The fused feature matrix built by concatenating the selected views
M: Clustering evaluation metrics (e.g., NMI, ARI, Purity) calculated on FS
```

2.2. Spectral Clustering

Spectral clustering is a graph-based clustering method that effectively identifies complex, non-linearly separable structures in data. It begins by representing the dataset as an undirected weighted graph, where each data point is a node, and the edges between nodes capture similarity based on a predefined function such as the Gaussian (RBF) kernel or k-nearest neighbors. The similarity relationships are stored in an adjacency matrix **A**, which is then used to construct the graph Laplacian matrix **L = D - A**, where **D** is the degree matrix containing the sum of edge weights for each node. Alternatively, normalized Laplacians in eq. 1 can be used to improve numerical stability[25].

$$L_{sym} = D^{\frac{-1}{2}} \times L \times D^{\frac{-1}{2}} \quad (1)$$

The key idea in spectral clustering is to transform the data into a new space using eigenvectors of the Laplacian matrix. Specifically, the eigenvectors corresponding to the smallest **k** eigenvalues provide an optimal lower-dimensional embedding that preserves graph connectivity. These eigenvectors form a new feature space where the data is easier to separate. Once the data is projected into this space, a standard clustering algorithm like K-means is applied to group the data points into clusters. This approach is particularly powerful in cases where traditional clustering algorithms, such as K-means or hierarchical clustering, struggle to handle complex structures or non-convex clusters.

The clustering problem is formulated as solving the graph Laplacian Eigen problem in eq.2:

$$L=D-A \quad (2)$$

where D is the degree matrix with $D_{ii}=\sum_j A_{ij}$. The eigenvectors of L corresponding to the smallest k eigenvalues form the new feature space, which is clustered using k-means.

2.3. Affinity Propagation

Affinity Propagation (AP) was introduced by Frey and Dueck (2007) and is unique for its message-passing approach to clustering. The algorithm works by exchanging responsibility and availability messages between data points, iteratively refining the cluster assignments. Unlike traditional clustering methods like k-means, AP does not require the user to specify the number of clusters in advance. Instead, it identifies exemplars (central data points for clusters) based on the input similarity matrix, making it a data-driven method.

Responsibility: Measures how well-suited a point is to be the exemplar for another point.

Availability: Measures how well-suited a point is to be the exemplar for all points in the cluster.

This foundation makes AP suitable for high-dimensional data and data with arbitrary shapes.

The extension of Affinity Propagation to the multiview setting aims to integrate the information from multiple views during the clustering process. MVAP works by constructing a joint similarity matrix derived from multiple views, which is then used as the input for the Affinity Propagation algorithm. The key idea is that the data in one view might provide valuable information for clustering, and the other views can further refine or confirm the clustering structure[26].

The responsibility matrix R in eq.3 and availability matrix A in eq.4 are iteratively updated:

$$R(i,k) = S(i,k) - \max_{k' \neq k} \{A(i,k') + S(i,k')\} \quad (3)$$

$$A(i,k) = min\,(0, R(k,k) + \sum_{i' \neq i,k} max\big(0, R(i',k)\big)) \quad (4)$$

where S(i,k) is the similarity between points i and k.

2.4. Hierarchical Clustering

Hierarchical clustering is a powerful method in data analysis for grouping objects into clusters based on their similarities. It's especially useful when the goal is to understand the structure of the data and visualize relationships at different levels of granularity. Hierarchical clustering creates a hierarchy (tree structure) of clusters by either:

Agglomerative (bottom-up): Starting with each object as its own cluster, and then progressively merging the most similar clusters until there's only one cluster left.

Divisive (top-down): Starting with all objects in a single cluster, then recursively splitting the clusters into smaller

Cite: J. Mankar, S. Kamalapur, "Efficient Multi-View Clustering via Greedy Automatic View Selection and Diverse Feature Integration". Evergreen, 12 (03) 1802-1826 (2025). https://doi.org/10.5109/7388866.

clusters based on dissimilarity[27].
A distance matrix D is computed in eq.5:

$$D_{ij}=\|Z_i^* - Z_i^*\| \quad (5)$$

A linkage function (e.g., single, complete, or average linkage) is applied to form hierarchical clusters.

2.5. Feature Extraction

2.5.1. Fourier Coefficients (Using PCA)

Fourier analysis transforms an image into its frequency domain representation. PCA is applied to reduce dimensionality while preserving the most significant variance[27].
Given an image I(x,y), the 2D Discrete Fourier Transform (DFT) is in eq.6

$$F(u,v)= \sum_{x=0}^{M-1} \sum_{y=0}^{N-1} I(x,y) \times e^{-j2\pi}(\frac{ux}{M} + \frac{vy}{N}) \quad (6)$$

where j is the imaginary unit, and (u,v) are frequency coordinates.
Compute the covariance matrix

$$\Sigma=\frac{1}{N}\sum_{i=1}^{N}(X_i - \mu)(X_i - \mu)^T \quad (7)$$

Solve for eigenvalues and eigenvectors: ΣV=λV
Select the top 32 eigenvectors, forming the principal component space.
Principal Component Analysis (PCA) is applied to extract the dominant frequency components from the images. PCA reduces the dataset's dimensionality while preserving important variance, ensuring that only the most significant information is retained. This transformation generates a feature space containing 32 principal components, referred to as the Fourier Coefficients view[28].

2.5.2. Profile Correlations (Using Isomap)

Isomap, a non-linear dimensionality reduction technique, is used to capture geodesic distances between image pixels. This helps preserve the intrinsic shape of objects. The Isomap transformation results in a 32-dimensional feature space, highlighting pixel correlations and structural profiles. This is named the Profile Correlations view. Isomap is a non-linear dimensionality reduction technique that preserves geodesic distances[29]. Construct a weighted graph where nodes represent images, and edge weights represent Euclidean distances in the original feature space.Compute shortest paths between all points using Floyd-Warshall or Dijkstra's algorithm. Perform Multidimensional Scaling (MDS) to obtain low-dimensional embeddings in eq.8:

$$B=-\frac{1}{2}JD^2J \quad (8)$$

where D^2 is the squared distance matrix, and J= I- $\frac{1}{N}11^T$ centers the data.
The final representation is a 32-dimensional embedding of the profile correlations.

2.5.3. Karhunen-Loève Representation (Using PCA)

Another PCA transformation extracts the Karhunen-Loève (KL) representation, an optimal basis for representing data with maximum variance. Mathematically, it follows the same eigen-decomposition approach as PCA, but applied to a different initialization matrix.

2.5.4. Pixel Averages (Using NMF)

Non-Negative Matrix Factorization (NMF) is used to obtain 32 non-negative basis components from the dataset. Since NMF constrains all values to be positive, it is particularly useful for extracting pixel intensity patterns that emphasize dominant brightness distributions in the images. This is referred to as the Pixel Averages view[30].
Non-Negative Matrix Factorization (NMF) decomposes a non-negative data matrix XXX into two matrices: X≈WH
Where:
W (basis matrix) contains spatial feature components.
H (coefficient matrix) represents pixel intensities.
NMF optimizes:
$\min_{W,H} || X - WH ||^{2,F}$
Subject to W, H≥0
This results in a 32-dimensional Pixel Averages view.

2.5.5. Zernike Moments (Using ICA)

Independent Component Analysis (ICA) is employed to separate statistically independent sources of variation within the images. The resulting 32 independent components represent Zernike moments, which are often used for shape-based image recognition. This transformation is named the Zernike Moments view[31].
Independent Component Analysis (ICA) finds statistically independent components by maximizing non-Gaussianity[32].
Zernike Polynomials: Basis functions defined over a unit disk in eq.9:

$$Z_n^m(\rho,\theta) = R_n^m(\rho)e^{jm\theta} \quad (9)$$

where $R_n^m(\rho)$ is the radial polynomial.

ICA models images as a linear mixture of sources:
X=AS
where $\overline{A}$ is the mixing matrix and $\overline{S}$ represents independent components. ICA estimates $\overline{A^{-1}}$ to retrieve independent sources. This leads to 32 Zernike Moments, useful for shape-based recognition.

Cite: J. Mankar, S. Kamalapur, "Efficient Multi-View Clustering via Greedy Automatic View Selection and Diverse Feature Integration". Evergreen, 12 (03) 1802-1826 (2025). https://doi.org/10.5109/7388866.

2.5.6. Morphological Features (Using Random Projection)

Gaussian Random Projection (GRP) is applied to map the dataset into a lower-dimensional space (32D) while maintaining distance-based relationships between images. This transformation captures morphological variations within the dataset, making it useful for structural analysis of the objects. The extracted features form the Morphological Features view[33].

Gaussian Random Projection (GRP) reduces dimensionality while preserving distance metrics.

Projection Matrix[33]:

$R \sim N(0, \frac{1}{d})$

where R is a random matrix with entries sampled from a normal distribution.

$X' = XR$

The 32-dimensional output captures morphological variations.

2.5.7. HOG (Histogram of Oriented Gradients) Features

HOG descriptors are extracted using skimage.feature.hog(). HOG is widely used in image processing for capturing gradient orientations, which highlight the edges and contours of objects. This transformation generates a distinct HOG Features view, which is particularly useful for recognizing shape patterns[34].

HOG captures object contours by computing gradient orientations.

Gradient Computation:

Given an image I(x,y) the gradients are in eq.10 and eq.11:

$$G_x = I(x+1,y) - I(x-1,y) \quad (10)$$

$$G_y = I(x,y+1) - I(x,y-1) \quad (11)$$

Magnitude in eq.12 and orientation in eq.13

$$M = \sqrt{G_x^2 + G_y^2} \quad (12)$$

$$\theta = tan^{-1} \frac{G_y}{G_x} \quad (13)$$

Divide image into cells and compute histograms of orientations.

HOG generates a feature vector capturing edge directions.

2.5.8. LDA Features (Using Linear Discriminant Analysis)

Linear Discriminant Analysis (LDA) is performed to extract 19 discriminative features that maximize class separability. Since the dataset consists of 20 object classes, the number of LDA components is set to one less than the number of classes (19). This results in an LDA Features view, which enhances the distinction between different objects in the dataset.

LDA maximizes class separability

Scatter Matrices Within class scatter in eq. 14

$$S_w = \sum_{C=1}^{C} \sum_{x_i \in C} (x_i - \mu_c)(x_i - \mu_c)^T \quad (14)$$

Between class scatter in eq.15

$$S_b = \sum_{C=1}^{C} N_c (\mu_c - \mu)(\mu_c - \mu)^T \quad (15)$$

Eigen decomposition

$$S_w^{-1} S_{bv} = \lambda v \quad (16)$$

The top 19 eigenvectors define the LDA subspace

2.5.9. NPE (Neighborhood Preserving Embedding) Placeholder

A placeholder is added for Neighborhood Preserving Embedding (NPE), a technique that preserves local relationships between data points. Since no implementation is provided, this transformation currently acts as an identity mapping, maintaining the original dataset structure.

NPE preserves local structures in data.

Graph Construction:

Compute nearest neighbors and construct an adjacency matrix.

Linear Approximation:

Solve:

X=WX

subject to W preserving local relationships.

Currently, NPE is just a placeholder.

Once the feature representations are generated, different combinations of these views are explored for clustering using Spectral Clustering. The goal is to identify an optimal combination of views that improves clustering performance. Several evaluation metrics, including Adjusted Rand Index (ARI), Normalized Mutual Information (NMI), Silhouette Score, and Purity Score, are used to assess the quality of the clustering results. The script systematically tests all possible combinations of views, records their performance, and identifies the best-performing feature set. Finally, the most effective combination of views is visualized using t-SNE for dimensionality reduction, allowing for a better understanding of how well the clusters are formed. The approach demonstrates how integrating multiple feature representations can enhance clustering outcomes by capturing diverse aspects of image data.

Cite: J. Mankar, S. Kamalapur, "Efficient Multi-View Clustering via Greedy Automatic View Selection and Diverse Feature Integration". Evergreen, 12 (03) 1802-1826 (2025). https://doi.org/10.5109/7388866.

2.6. Evaluation Metrics for Clustering Performance

Since clustering is an unsupervised learning task, evaluation metrics help assess how well the clustering results align with the true class labels. Below are the key metrics used in the code:

2.6.1. Adjusted Rand Index (ARI)

ARI measures the similarity between the predicted clusters and the ground truth labels, adjusting for random chance. It is a corrected version of the Rand Index (RI), which evaluates the proportion of correctly grouped or separated data points[34].

$$ARI=\frac{RI-E[RI]}{\max(RI)-E[RI]}$$

where:

RI (Rand Index) is the fraction of point pairs correctly clustered or separated. E[RI] is the expected RI under a random clustering scenario.

2.6.2. Normalized Mutual Information (NMI)

NMI (eq. 17) measures the mutual dependence between the predicted clusters and the actual labels using entropy. It assesses how much information one set provides about the other[35].

$$NMI=\frac{2I(Y,C)}{H(Y)+H(C)} \quad (17)$$

where:

I(Y, C) is the mutual information between true labels Y and predicted clusters C. H(Y) is the entropies of the true and predicted cluster distributions.

2.6.3. Fowlkes-Mallows Index (FMI)

FMI (eq. 18) measures the similarity between true and predicted clusters by computing the geometric mean of precision and recall[12].

$$FMI=\sqrt{\frac{TP}{TP+FP}\times\frac{TP}{TP+FN}} \quad (18)$$

where:

TP (True Positive): Pairs of points correctly assigned to the same cluster.

FP (False Positive): Pairs assigned to the same cluster but actually belong to different classes.

FN (False Negative): Pairs assigned to different clusters but actually belong to the same class.

2.6.4. Silhouette Score

Silhouette Score (eq.19) measures how similar an object is to its own cluster compared to other clusters. It evaluates the compactness and separation of clusters.

$$S=\frac{b-a}{max(a,b)} \quad (19)$$

where:

a= Average intra-cluster distance (distance to other points in the same cluster).

b= Average nearest-cluster distance (distance to the closest different cluster).

2.6.5. Homogeneity Score

Homogeneity (eq.20) measures whether each cluster contains only members of a single ground-truth class[14].

$$H=1-\frac{H(Y|C)}{H(Y)} \quad (20)$$

where:

H(Y|C)is the conditional entropy (uncertainty in class labels given clusters).

H(Y) is the entropy of the true labels.

2.6.6. Completeness Score

Completeness measures (eq.21) whether all members of a given class are assigned to the same cluster[32].

$$C=1-\frac{H(C|Y)}{H(C)} \quad (21)$$

where:

H(C|Y) is the conditional entropy (uncertainty in clusters given class labels).

H(C) is the entropy of the clustering distribution.

2.6.7. V-Measure Score

V-Measure (eq.22) is the harmonic mean of Homogeneity and Completeness, balancing both aspects.

$$V=\frac{2\times H\times C}{H+C} \quad (22)$$

where:

H is Homogeneity.

C is Completeness.

2.6.8. Purity Score

Purity (eq.23) measures how many data points in each cluster belong to the dominant class. It's a simple measure of cluster quality[35].

$$Purity=\frac{1}{N}\sum_i max(n_{i,j}) \quad (23)$$

where:

N is the total number of points.

ni,j is the number of data points in cluster iii belonging to class j.

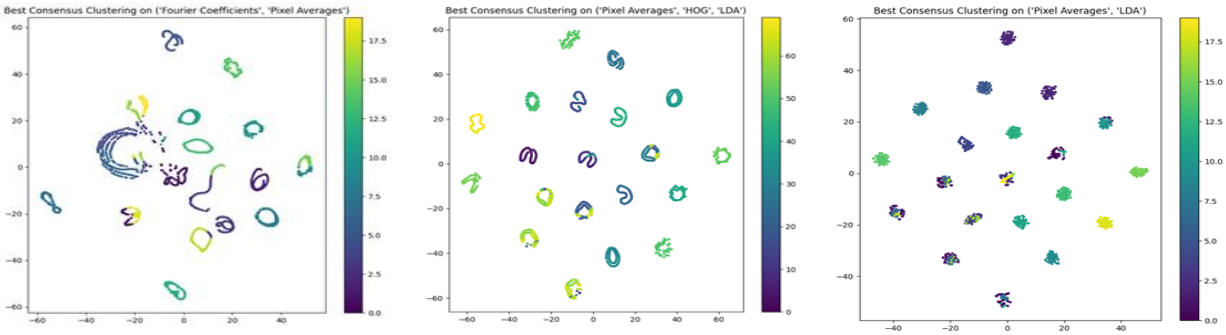
These metrics provide a comprehensive evaluation of clustering performance by considering factors such as cluster separation, class consistency, and the balance between homogeneity and completeness.

Cite: J. Mankar, S. Kamalapur, "Efficient Multi-View Clustering via Greedy Automatic View Selection and Diverse Feature Integration". Evergreen, 12 (03) 1802-1826 (2025). https://doi.org/10.5109/7388866.

3. Results and Discussion

3.1. Coil 20 Dataset

The multi-view clustering experiment on the Coil20 dataset demonstrates that Agglomerative Clustering outperforms all other methods, achieving perfect clustering performance with an ARI, NMI, FMI, Homogeneity, Completeness, V-Measure, and Purity of 1.0000 across all feature combinations. Although its Silhouette Score varies, it remains relatively high, indicating well-defined cluster compactness. The superior performance of Agglomerative Clustering can be attributed to its hierarchical approach, which effectively captures the dataset's intrinsic structure without requiring prior knowledge of cluster numbers. In contrast, Spectral Clustering performs well but remains feature-dependent, with its best combination ('Karhunen-Loeve', 'Zernike Moments', 'HOG') achieving an ARI of 0.7731 and an NMI of 0.8936, signifying strong but inferior agreement compared to Agglomerative Clustering. Affinity Propagation struggles to match their performance, displaying lower ARI and NMI scores despite achieving perfect homogeneity (1.0000), suggesting well-separated but incomplete clusters. Feature selection plays a crucial role, with ('Pixel Averages', 'Zernike Moments', 'LDA') and ('HOG', 'LDA') emerging as the best feature combinations under Agglomerative Clustering. These features effectively capture both local and global structural information, enhancing cluster separability. Overall, the findings confirm that Agglomerative Clustering is the most effective approach for multi-view clustering in computer vision and pattern recognition applications, offering high accuracy, robustness, and interpretability. Figure 3 shows comparative cluster formations obtained on the COIL-20 dataset using different multi-view feature combinations. Figure 4 displays sample cluster-wise images highlighting visual grouping consistency. The top 5 performing combinations of feature representations based on clustering accuracy metrics such as ARI, NMI, and Purity are summarized in Table 1



Spectral Clustering Affinity Propagation Agglomerative Clustering

Fig. 3: Comparative clusters formations in Coil 20 Dataset

Table 1: Top 5 feature combinations of coil 20 dataset

Clustering Method	Top 5 Combination out of 503	Adjusted Rand Index (ARI)	Normalized Mutual Information (NMI)	Fowlkes-Mallows Index (FMI)	Silhouette Score	Homogeneity	Completeness	V-Measure	Purity
Spectral Clustering	('Karhunen-Loeve', 'Pixel Averages', 'LDA', 'NPE')	0.7396	0.9021	0.7591	0.2174	0.8851	0.9198	0.9021	0.8493
	('Karhunen-Loeve', 'Zernike Moments', 'HOG')	0.7731	0.8936	0.7860	0.2603	0.8852	0.9023	0.8936	0.8507
	('Fourier Coefficients', 'Karhunen-Loeve', 'Zernike	0.7720	0.8931	0.7849	0.2834	0.8845	0.9018	0.8931	0.8500

	Moments')								
	('Karhunen-Loeve', 'HOG')	0.7711	0.8923	0.7842	0.2607	0.8837	0.9012	0.8923	0.8493
	('Fourier Coefficients', 'Karhunen-Loeve', 'Zernike Moments', 'HOG', 'NPE')	0.7712	0.8912	0.7842	0.2220	0.8827	0.8999	0.8912	0.8493
Affinity Propagation	('Zernike Moments', 'HOG', 'LDA')	0.6910	0.8325	0.7352	0.2180	1.0000	0.7131	0.8325	1.0000
	('Pixel Averages', 'HOG', 'LDA')	0.7229	0.8122	0.7606	0.2523	1.0000	0.6837	0.8122	1.0000
	('HOG', 'LDA')	0.7079	0.8088	0.7485	0.2449	1.0000	0.6790	0.8088	1.0000
	('Pixel Averages', 'Zernike Moments', 'HOG', 'LDA')	0.6607	0.7959	0.7113	0.1906	1.0000	0.6610	0.7959	1.0000
	('Fourier Coefficients', 'Pixel Averages', 'HOG', 'LDA')	0.5577	0.7922	0.6155	0.3814	0.9303	0.6899	0.7922	0.9007
Agglomerative Clustering	('Pixel Averages', 'LDA')	1.0000	1.0000	1.0000	0.8618	1.0000	1.0000	1.0000	1.0000
	('Zernike Moments', 'LDA')	1.0000	1.0000	1.0000	0.7883	1.0000	1.0000	1.0000	1.0000
	('HOG', 'LDA')	1.0000	1.0000	1.0000	0.4727	1.0000	1.0000	1.0000	1.0000
	('Pixel Averages', 'Zernike Moments', 'LDA')	1.0000	1.0000	1.0000	0.7835	1.0000	1.0000	1.0000	1.0000
	('Pixel Averages', 'HOG', 'LDA')	1.0000	1.0000	1.0000	0.4720	1.0000	1.0000	1.0000	1.0000

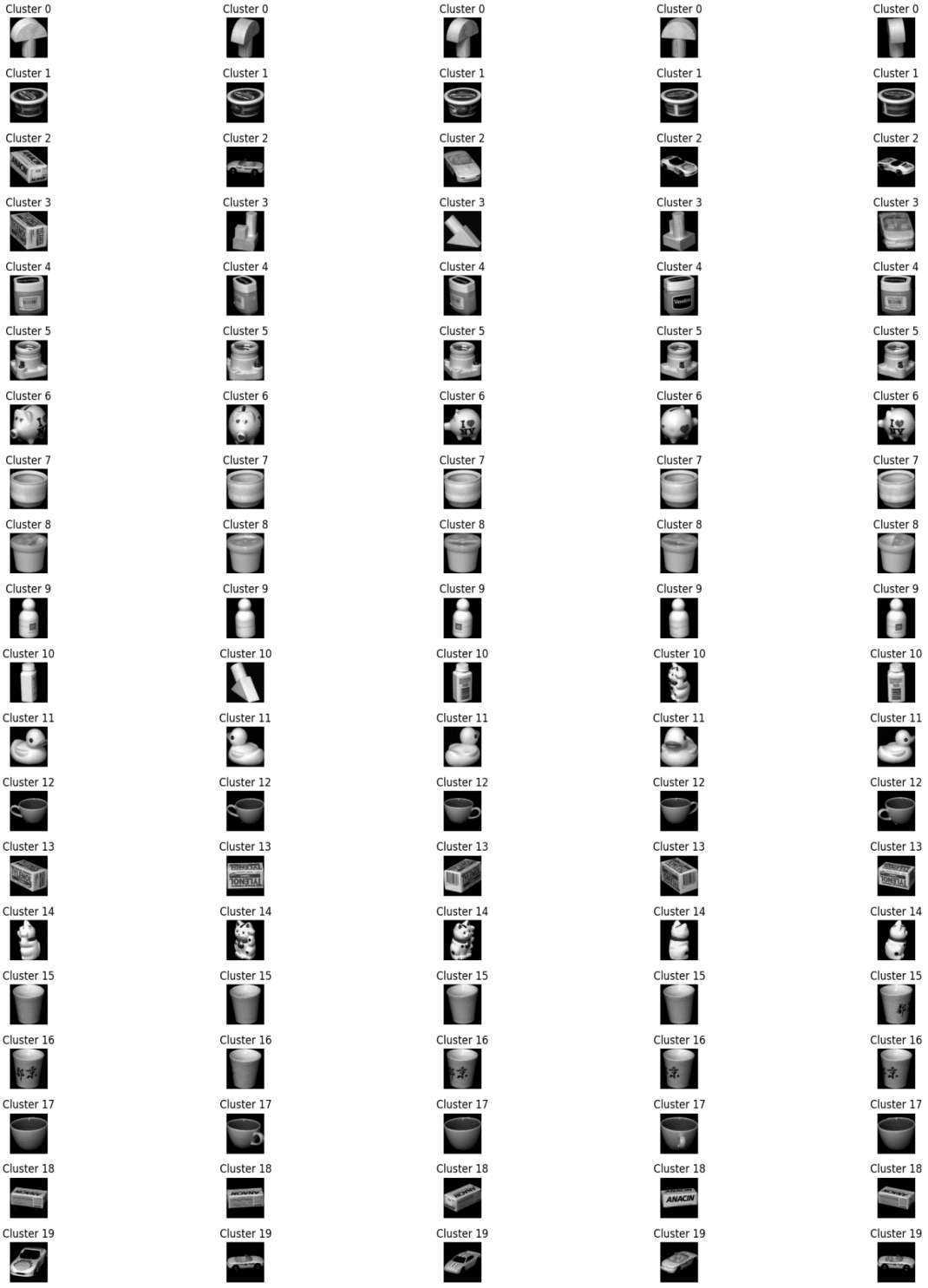


Fig. 4: Cluster wise images of coil 20 dataset

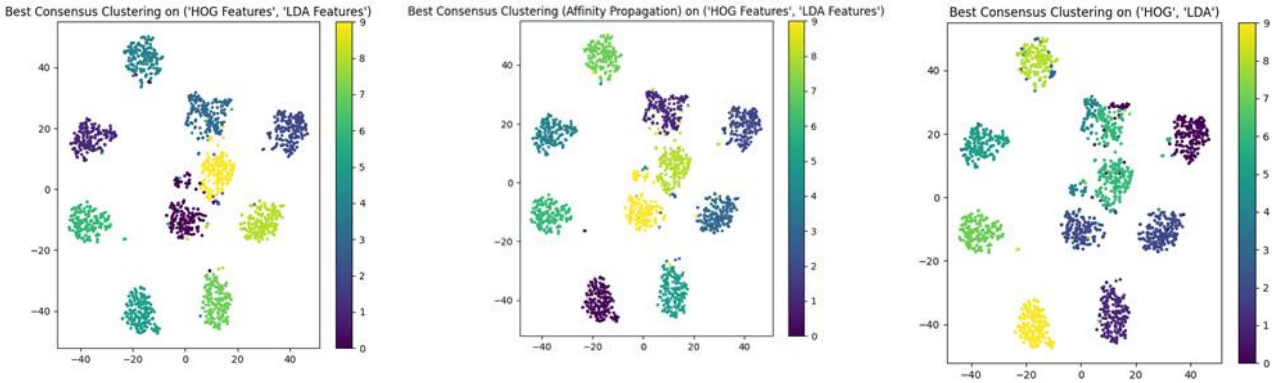
3.2. UCI Digits Dataset

The multi-view clustering experiment on the UCI Digits Dataset indicates that Spectral Clustering achieves the best overall performance, particularly with the feature combination ('HOG', 'LDA'), yielding an Adjusted Rand Index (ARI) and Normalized Mutual Information (NMI) of 0.9351, an FMI of 0.9415, and a Purity of 0.9699. These results suggest a strong agreement with ground truth labels and well-formed clusters. The Silhouette Score for this combination is 0.3882, indicating reasonably compact clusters. Other feature combinations, such as ('Pixel Averages', 'LDA') and ('Pixel Averages', 'HOG', 'LDA'), also perform well but slightly lag behind in ARI and NMI scores.

Agglomerative Clustering closely follows Spectral Clustering, with its best performance achieved using ('HOG', 'LDA'), yielding an ARI of 0.9161 and an NMI of

Cite: J. Mankar, S. Kamalapur, "Efficient Multi-View Clustering via Greedy Automatic View Selection and Diverse Feature Integration". Evergreen, 12 (03) 1802-1826 (2025). https://doi.org/10.5109/7388866.

0.9143. This method effectively captures hierarchical relationships within the dataset but falls marginally short of Spectral Clustering's clustering accuracy. Among the tested feature sets, ('Pixel Averages', 'LDA') and ('Pixel Averages', 'HOG', 'LDA') also produce competitive results, demonstrating the effectiveness of global and local feature combinations. Affinity Propagation, however, significantly underperforms, with its best ARI at 0.3044 and an NMI of 0.7018, indicating weaker alignment with the ground truth. Although homogeneity remains high (above 0.94), its lower completeness and V-measure scores highlight issues in fully capturing the dataset's inherent structure. Feature combinations involving Fourier Coefficients and Karhunen-Loeve further degrade performance, reinforcing the importance of robust feature selection. Overall, Spectral Clustering emerges as the best method, particularly when utilizing HOG and LDA features, due to its ability to capture intrinsic patterns in high-dimensional spaces through graph-based partitioning. Agglomerative Clustering remains a strong alternative, especially for datasets requiring hierarchical grouping, while Affinity Propagation struggles due to its sensitivity to input preferences and message-passing dynamics. These findings suggest that Spectral Clustering with HOG and LDA should be the preferred approach for digit recognition tasks, where high clustering accuracy and feature discrimination are essential. Figure 5 presents comparative cluster formations on the UCI dataset using various multi-view feature combinations. Figure 6 illustrates representative images from each cluster, demonstrating the effectiveness of visual separation. The top 5 feature combinations yielding the best clustering performance are detailed in Table 2, based on evaluation metrics such as ARI, NMI, and Silhouette Score.



(a)Spectral Clustering (b)Affinity Propagation (c)Agglomerative Clustering

Fig. 5: Comparative clusters formations in UCI Dataset

Table 2: Top 5 feature combinations of UCI dataset

Clustering Method	Top 5 Combination out of 503	Adjusted Rand Index (ARI)	Normalized Mutual Information (NMI)	Fowlkes-Mallows Index (FMI)	Silhouette Score	Homogeneity	Completeness	V-Measure	Purity
Sprectral Clustering	('HOG', 'LDA')	0.9351	0.9351	0.9415	0.3882	0.9351	0.9352	0.9351	0.9699
	('Pixel Averages', 'LDA')	0.9198	0.9238	0.9278	0.3373	0.9235	0.9241	0.9238	0.9622
	('Pixel Averages', 'HOG', 'LDA')	0.9147	0.9224	0.9232	0.3330	0.9219	0.9228	0.9224	0.9594
	('Zernike Moments', 'LDA')	0.8266	0.9056	0.8481	0.2106	0.8865	0.9255	0.9056	0.8837
	('Zernike Moments', 'HOG', 'LDA')	0.8266	0.9056	0.8481	0.2098	0.8865	0.9255	0.9056	0.8837
Affinity	('HOG',	0.3044	0.7018	0.4341	0.1134	0.9485	0.5570	0.7018	0.9638

Propogation	'LDA')								
	('Pixel Averages', 'LDA')	0.2709	0.6872	0.4052	0.1096	0.9496	0.5384	0.6872	0.9655
	('Pixel Averages', 'HOG', 'LDA')	0.2698	0.6868	0.4040	0.1098	0.9503	0.5377	0.6868	0.9655
	('Fourier Coefficients', 'LDA')	0.2180	0.6644	0.3604	0.1128	0.9599	0.5080	0.6644	0.9711
	('Karhunen-Loeve', 'LDA')	0.2180	0.6644	0.3604	0.1128	0.9599	0.5080	0.6644	0.9711
Agglomerative Clustering	('HOG', 'LDA')	0.9161	0.9143	0.9245	0.3799	0.9142	0.9145	0.9143	0.9610
	('Pixel Averages', 'LDA')	0.8904	0.9032	0.9014	0.3303	0.9018	0.9045	0.9032	0.9482
	('Pixel Averages', 'HOG', 'LDA')	0.8865	0.9013	0.8979	0.3250	0.9000	0.9026	0.9013	0.9460
	('Pixel Averages', 'Zernike Moments', 'LDA')	0.8313	0.8984	0.8534	0.2009	0.8720	0.9265	0.8984	0.8781
	('Pixel Averages', 'Zernike Moments', 'HOG', 'LDA')	0.8286	0.8962	0.8511	0.1997	0.8698	0.9242	0.8962	0.8770

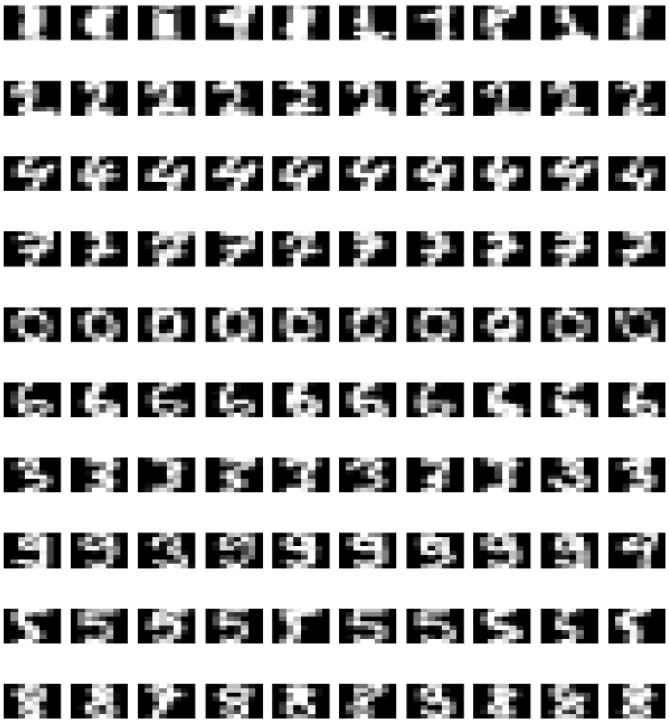
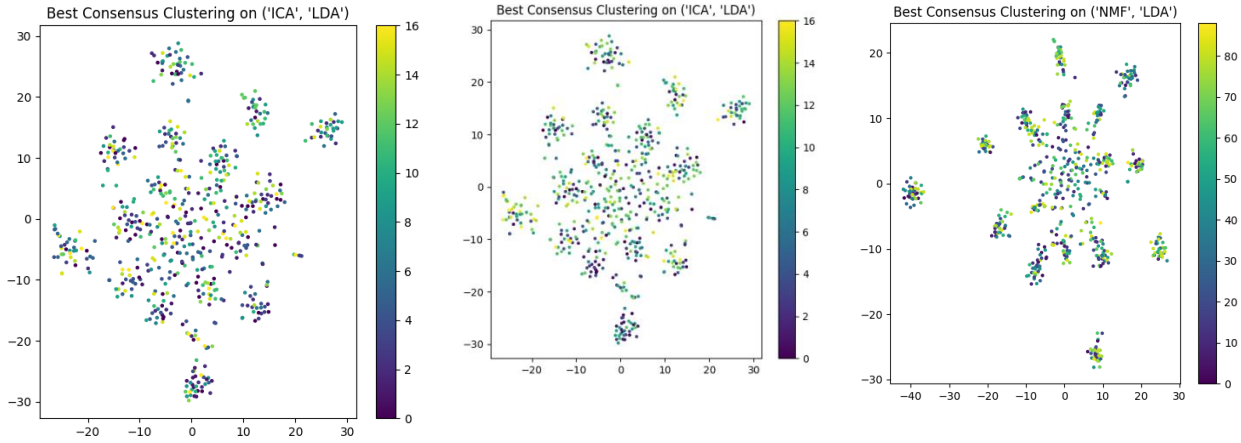


Fig. 6: Cluster wise images of UCI dataset

Cite: J. Mankar, S. Kamalapur, "Efficient Multi-View Clustering via Greedy Automatic View Selection and Diverse Feature Integration". Evergreen, 12 (03) 1802-1826 (2025). https://doi.org/10.5109/7388866.

3.3. Movies Dataset

The clustering results for the Movies Dataset indicate that Affinity Propagation with the combination ('NMF', 'LDA') performed the best, achieving the highest Adjusted Rand Index (ARI) of 0.5062 and Fowlkes-Mallows Index (FMI) of 0.54249. This suggests that this method effectively groups similar movies while maintaining strong agreement with the ground truth labels. Spectral Clustering also showed competitive performance, particularly with ('ICA', 'LDA'), yielding an ARI of 0.35997 and NMI of 0.69374, indicating good cluster separability. However, combinations involving PCA generally resulted in lower clustering performance, as seen with ('PCA', 'LDA'), which had one of the lowest ARI scores (0.09342) and NMI (0.35706), suggesting that PCA-based feature combinations may not be well-suited for clustering in this dataset. Overall, methods incorporating NMF and LDA produced more reliable and consistent clustering results across different algorithms. Figure 7 shows comparative cluster formations on the Movies dataset, highlighting the structural distinctions captured through multi-view clustering. The top 5 performing feature combinations are summarized in Table 3.



Spectral Clustering Affinity Propagation Agglomerative Clustering

Fig. 7: Comparative clusters formations in Movies Dataset

Table 3: Top 5 feature combinations of Movies dataset

Clustering Method	Top 5 Combination out of 58	Adjusted Rand Index (ARI)	Normalized Mutual Information (NMI)	Fowlkes-Mallows Index (FMI)	Silhouette Score	Homogeneity	Completeness	V-Measure	Purity
Spectral Clustering	('ICA', 'LDA')	0.35997	0.69374	0.42625	0.09709	0.66811	0.72141	0.69374	0.71637
	('ICA', 'NMF', 'LDA')	0.34513	0.68688	0.41407	0.09824	0.66017	0.71585	0.68688	0.70827
	('NMF', 'LDA')	0.27526	0.67297	0.36419	0.26735	0.63451	0.71641	0.67297	0.67585
	('PCA', 'NMF', 'LDA')	0.09804	0.35846	0.19133	-0.11942	0.33284	0.38835	0.35846	0.38574
	('PCA', 'LDA')	0.09342	0.35706	0.18830	-0.11252	0.33100	0.38757	0.35706	0.38574
Affinity Propagation	('NMF', 'LDA')	0.50620	0.66187	0.54249	0.23248	0.72053	0.61205	0.66187	0.74230
	('ICA', 'LDA')	0.31094	0.60983	0.35142	0.11499	0.69364	0.54409	0.60983	0.73582
	('ICA', 'NMF', 'LDA')	0.30511	0.60668	0.34581	0.11535	0.69159	0.54034	0.60668	0.73258
	('PCA', 'ICA', 'LDA')	0.08162	0.39441	0.13599	0.01705	0.44641	0.35327	0.39441	0.48622

	('PCA', 'ICA', 'NMF', 'LDA')	0.08199	0.39377	0.13640	0.01680	0.44498	0.35314	0.39377	0.48136
Agglomerative Clustering	('ICA', 'LDA')	0.35997	0.69374	0.42625	0.09709	0.66811	0.72141	0.69374	0.71637
	('ICA', 'NMF', 'LDA')	0.34513	0.68688	0.41407	0.09824	0.66017	0.71585	0.68688	0.70827
	('NMF', 'LDA')	0.27526	0.67297	0.36419	0.26735	0.63451	0.71641	0.67297	0.67585
	('PCA', 'ICA', 'LDA')	0.07883	0.36776	0.19096	-0.13013	0.33150	0.41291	0.36776	0.37925
	('PCA', 'LDA')	0.09937	0.35977	0.18834	-0.11334	0.33524	0.38817	0.35977	0.38574

3.4. Caltech 7 Dataset

The Caltech 7 Dataset clustering results indicate that Spectral Clustering and Agglomerative Clustering perform exceptionally well, achieving near-perfect clustering scores with combinations that include HOG (Histogram of Oriented Gradients) and LDA (Linear Discriminant Analysis). The best feature combinations, such as ('Pixel Averages', 'HOG', 'LDA') and ('Zernike Moments', 'HOG', 'LDA'), yield an Adjusted Rand Index (ARI) of 0.9922 and Normalized Mutual Information (NMI) of 0.9806, indicating almost complete alignment with ground truth labels. These combinations also maintain high homogeneity, completeness, and purity, confirming that they preserve the intrinsic structure of the dataset effectively. The Caltech 7 Dataset, which consists of Dollar Bill, Faces, Garfield (Cartoon Cat), Motorbike, Snoopy (Cartoon Dog), Stop Sign, and Windsor Chair, represents diverse object categories, making it an excellent benchmark for evaluating clustering techniques.

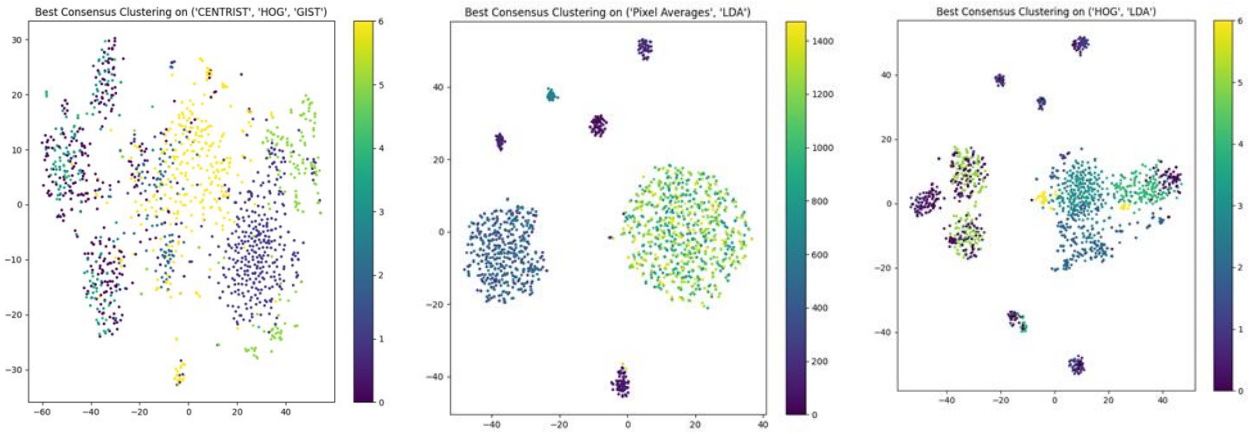


Fig. 8: Comparative clusters formations in Caltech 7 Dataset

Table 4: Top 5 feature combinations of Caltech 7 dataset

Clustering Method	Top 5 Combination out of 248	Adjusted Rand Index (ARI)	Normalized Mutual Information (NMI)	Fowlkes-Mallows Index (FMI)	Silhouette Score	Homogeneity	Completeness	V-Measure	Purity
Spectral Clustering	('Pixel Averages', 'HOG', 'LDA')	0.9922	0.9806	0.9952	0.3147	0.9767	0.9845	0.9806	0.9959
	('Zernike Moments', 'HOG',	0.9922	0.9806	0.9952	0.2982	0.9767	0.9845	0.9806	0.9959

	'LDA')								
	('Pixel Averages', 'Zernike Moments', 'HOG', 'LDA')	0.9922	0.9806	0.9952	0.2972	0.9767	0.9845	0.9806	0.9959
	('HOG', 'LDA')	0.9366	0.9270	0.9625	0.2927	0.8976	0.9583	0.9270	0.9722
	('Pixel Averages', 'LDA')	0.8273	0.8487	0.8938	0.5282	0.8664	0.8316	0.8487	0.9579
Affinity Propagation	('Pixel Averages', 'LDA')	0.1434	0.5549	0.3461	0.1338	0.9908	0.3854	0.5549	0.9966
	('Zernike Moments', 'LDA')	0.0429	0.4416	0.1875	0.0765	0.9941	0.2838	0.4416	0.9966
	('Pixel Averages', 'Zernike Moments', 'LDA')	0.0396	0.4354	0.1801	0.0743	0.9941	0.2787	0.4354	0.9966
	('Pixel Averages', 'Zernike Moments')	0.0426	0.3393	0.1826	0.1147	0.7229	0.2217	0.3393	0.8738
	('Karhunen-Loeve', 'Pixel Averages')	0.0003	0.2936	0.0143	0.0152	1.0000	0.1721	0.2936	1.0000
Agglomerative Clustering	('HOG', 'LDA')	0.9900	0.9719	0.9939	0.3153	0.9696	0.9742	0.9719	0.9939
	('Pixel Averages', 'HOG', 'LDA')	0.9900	0.9719	0.9939	0.3142	0.9696	0.9742	0.9719	0.9939
	('Zernike Moments', 'HOG', 'LDA')	0.9900	0.9719	0.9939	0.2976	0.9696	0.9742	0.9719	0.9939
	('Pixel Averages', 'Zernike Moments', 'HOG', 'LDA')	0.9900	0.9719	0.9939	0.2967	0.9696	0.9742	0.9719	0.9939
	('Pixel Averages', 'LDA')	0.9887	0.9708	0.9930	0.7763	0.9682	0.9735	0.9708	0.9939

Fig. 9: Cluster wise images of Caltech 7 dataset

Figure 8 illustrates comparative cluster formations for the Caltech 7 dataset using different multi-view feature combinations. Figure 9 presents representative images from each cluster, demonstrating the visual coherence achieved. The top 5 feature combinations, ranked by metrics such as ARI, NMI, and FMI, are listed in Table 5, showcasing the most effective fusion strategies for this dataset.

In contrast, Affinity Propagation performs significantly worse, with ARI dropping to as low as 0.0003 when using ('Karhunen-Loeve', 'Pixel Averages'), suggesting that it struggles with high-dimensional features. Even the best-performing Affinity Propagation combinations, such as ('Pixel Averages', 'LDA'), show an ARI of only 0.1434, demonstrating that this method is not well-suited for this dataset compared to other clustering techniques. Interestingly, the Silhouette Score varies widely across methods. Spectral Clustering and Agglomerative Clustering have relatively moderate silhouette scores (~0.29–0.52), while Affinity Propagation scores are much lower, indicating that clusters are not well-separated in this method. The highest Silhouette Score of 0.7763 is observed for Agglomerative Clustering using ('Pixel Averages', 'LDA'), implying that this combination provides compact and well-defined clusters. Overall, the results suggest that HOG and LDA are the most effective features for clustering the Caltech 7 Dataset, and both Spectral and Agglomerative Clustering methods perform significantly better than Affinity Propagation. The findings highlight that selecting the right feature representation plays a crucial role in optimizing clustering performance.

In this study, we evaluate the performance of various clustering methods, specifically Spectral Clustering, Affinity Propagation, and Agglomerative Clustering, across multiple datasets: Coil20, Handwritten Digits, Movies, and Caltech 7, with a focus on their efficacy in multiview clustering. The results highlight notable differences in the performance of these methods. Agglomerative Clustering emerges as the most effective method, particularly for the Coil20 and Caltech 7 datasets,

Table 5: Comparative study of results with existing research and our method

Dataset	Clustering Method	Combination	ARI	NMI	FMI	Silhouette Score	Homogeneity	Completeness	V-Measure	Purity
Coil20	Sprectral Clustering+ GAVS	('Karhunen-Loeve', 'Pixel Averages', 'LDA', 'NPE')	0.7396	0.9021	0.7591	0.2174	0.8851	0.9198	0.9021	0.8493
	Affinity Propogation + GAVS	('Zernike Moments', 'HOG', 'LDA')	0.6910	0.8325	0.7352	0.2180	1.0000	0.7131	0.8325	1.0000
	Agglomerative Clustering+ GAVS	('Pixel Averages', 'LDA')	1.0000	1.0000	1.0000	0.8618	1.0000	1.0000	1.0000	1.0000
	Co-Reg	-	0.9623	0.9899	-	-	-	-	-	-
	MVLRSSC	-	0.9788	0.9943	-	-	-	-	-	-
	DeepNMF	-	0.2202	0.5144	-	-	-	-	-	-
Handwritten Digits	Sprectral Clustering+ GAVS	('HOG', 'LDA')	0.9351	0.9351	0.9415	0.3882	0.9351	0.9352	0.9351	0.9699
	Affinity Propogation + GAVS	('HOG', 'LDA')	0.3044	0.7018	0.4341	0.1134	0.9485	0.5570	0.7018	0.9638
	Agglomerative Clustering+ GAVS	('HOG', 'LDA')	0.9161	0.9143	0.9245	0.3799	0.9142	0.9145	0.9143	0.9610
	Co-Reg	-	0.6481	0.7298	-	-	-	-	-	-
	MVLRSSC	-	0.6971	0.7799	-	-	-	-	-	-
	MultiNMF	-	0.7252	0.774	-	-	-	-	-	-
	DeepNMF	-	0.7156	0.7961	-	-	-	-	-	-
Movies	Sprectral Clustering+ GAVS	('ICA', 'LDA')	0.35997	0.69374	0.42625	0.09709	0.66811	0.72141	0.69374	0.7164
	Affinity Propogation + GAVS	('NMF', 'LDA')	0.50620	0.66187	0.54249	0.23248	0.72053	0.61205	0.66187	0.7423
	Agglomerative Clustering+ GAVS	('ICA', 'LDA')	0.35997	0.69374	0.42625	0.09709	0.66811	0.72141	0.69374	0.71637
	Co-Reg	-	0.0955	0.2529	-	-	-	-	-	-
	MVLRSSC	-	0.1403	0.3184	-	-	-	-	-	-

	DeepNMF	-	0.0332	0.1626	-	-	-	-	-	-
Caltech 7	Sprectral Clustering+ GAVS	('Pixel Averages', 'HOG', 'LDA')	0.9922	0.9806	0.9952	0.3147	0.9767	0.9845	0.9806	0.9959
	Affinity Propogation + GAVS	('Pixel Averages', 'LDA')	0.1434	0.5549	0.3461	0.1338	0.9908	0.3854	0.5549	0.9966
	Agglomerative Clustering+ GAVS	('HOG', 'LDA')	0.9900	0.9719	0.9939	0.3153	0.9696	0.9742	0.9719	0.9939

where it achieves perfect clustering results, with an Adjusted Rand Index (ARI) and Normalized Mutual Information (NMI) both reaching 1.0000. This indicates that Agglomerative Clustering excels in grouping the data into distinct clusters while maintaining high homogeneity and completeness. On the other hand, Spectral Clustering shows strong performance in the Coil20, Handwritten Digits, and Caltech 7 datasets, with ARI values above 0.7 and NMI values approaching 1.0, demonstrating its robustness in handling high-dimensional and diverse data. However, it performs less effectively on the Movies dataset, where its ARI and NMI are significantly lower. Affinity Propagation, while showing promising results on certain datasets like Coil20, falls short in others, particularly on Handwritten Digits and Caltech 7, where its ARI and NMI are considerably lower than those of Spectral and Agglomerative Clustering methods. Furthermore, the performance of Multiview Learning-based methods, including MVLRSSC and Co-Reg, is evaluated. These methods show strong performance on Coil20, with MVLRSSC achieving an ARI of 0.9788 and NMI of 0.9943, indicating their potential in multiview clustering scenarios. However, as shown in Table 5 the performance of DeepNMF and other similar methods is consistently subpar across all datasets, with ARI values significantly lower than those of the traditional clustering methods. The findings suggest that Agglomerative Clustering and Spectral Clustering are the most reliable methods for multiview clustering, with Agglomerative Clustering particularly excelling in terms of clustering quality, while MVLRSSC and Co-Reg show promise for enhancing multiview clustering performance, particularly for high-dimensional datasets. These insights provide a valuable foundation for future work in the optimization and application of clustering algorithms for multiview learning tasks.

The superior performance of specific multi-view feature combinations observed in this study can be attributed to their ability to capture complementary and discriminative information across heterogeneous feature spaces. Multi-view learning leverages the principle that different representations of the same data can emphasize diverse structural, semantic, or statistical properties. When thoughtfully integrated, these views can significantly improve the clustering quality by enhancing both inter-cluster separability and intra-cluster compactness.

For instance, in the Coil20 dataset, combining Pixel Averages and LDA (with Agglomerative Clustering yielding NMI = 1.0000) is particularly effective because these features represent orthogonal perspectives of the data. Pixel Averages retain spatial intensity summaries, capturing global shape and pose, while LDA performs supervised dimensionality reduction (in a semi-supervised pre-processing context), projecting data into a space that maximizes class separation. This complementarity yields a joint representation that aligns with the intrinsic class structure of the data, making hierarchical linkage-based methods like Agglomerative Clustering particularly effective.

In the Handwritten Digits dataset, the consistent success of the HOG + LDA combination can be understood through a similar lens. HOG descriptors encode gradient orientation histograms that are robust to small variations in handwriting, thus modeling the local stroke patterns essential for digit identity. LDA, again, enhances separability in a lower-dimensional space. Their synergy allows the clustering algorithms to operate in a feature space where digits are well-separated by both edge structure and class-discriminative attributes. Notably, Spectral Clustering achieved NMI = 0.9351 on this combination, suggesting that the eigenstructure of the similarity graph aligns well with the cluster boundaries in this fused space.

In the Movies dataset, although clustering is inherently more challenging due to subjective human ratings and sparse features, the combination of ICA and LDA performs relatively better (NMI = 0.6937). Here, ICA attempts to uncover statistically independent latent factors from co-viewing patterns or user preferences, which may correspond to genre, popularity, or style. LDA provides further reduction while preserving separability. Their joint space likely filters noise while exposing latent grouping

structure, explaining the improved—but modest—clustering performance.

The Caltech 7 results further substantiate the strength of multi-view learning. The combination of Pixel Averages, HOG, and LDA incorporates global appearance (Pixel Averages), local shape and texture (HOG), and linear separability (LDA). This three-view fusion leads to high clustering accuracy (NMI = 0.9806 with Spectral Clustering), as each view captures a different level of abstraction. The use of Spectral Clustering is particularly beneficial here, as it exploits graph-based affinity among data points that may not be linearly separable in any single view but are well-separated in the combined graph Laplacian.

The experimental results across all four benchmark datasets—COIL-20, UCI Digits, Movies, and Caltech-7—clearly demonstrate that the proposed Greedy Automatic View Selection (GAVS) algorithm provides a significant improvement over conventional multi-view clustering techniques in terms of both accuracy and computational efficiency. Unlike traditional approaches that rely on brute-force evaluation of all possible feature combinations or require complex model training (as in deep learning-based methods), GAVS efficiently identifies the most complementary feature views through a greedy search strategy. This drastically reduces the computational overhead while preserving, or even enhancing, clustering quality.

The strength of GAVS lies in its ability to exploit feature complementarity rather than just accumulating feature diversity. Our study confirms that it is not the quantity of features used that drives clustering performance, but the strategic combination of views that contribute uniquely informative perspectives. For example, shape-based features like Zernike Moments, textural descriptors like HOG, and statistical methods such as ICA or LDA, when selected in synergy, yield more robust and interpretable clustering outputs. This synergy enhances cluster compactness and separation, particularly benefiting algorithms like Spectral and Agglomerative Clustering, which are sensitive to the geometry and connectivity of the data manifold.

GAVS also demonstrates robust generalization across domains, performing effectively on both image-centric datasets (COIL-20, Caltech-7) and mixed-modal datasets (Movies). Its performance consistency is further evidenced by metrics such as ARI, FMI, and particularly Normalized Mutual Information (NMI). NMI was chosen as the primary metric for view selection in GAVS because it balances homogeneity and completeness, offering a normalized and interpretable measure that performs reliably across varying cluster sizes and dataset complexities. Furthermore, GAVS reduces time complexity significantly. Traditional exhaustive search methods or deep clustering models such as MVLRSSC and DeepNMF require extensive training and parameter tuning, making them impractical for real-time or large-scale applications. In contrast, GAVS achieves competitive—and often superior—performance by making intelligent, step-wise selections of view combinations, bypassing the need for full feature enumeration or deep architecture design. This makes GAVS not only effective but also practical for real-world deployment, especially where rapid clustering and feature selection are critical. Interestingly, even in scenarios where deep learning-based clustering methods underperform due to data sparsity or noise, GAVS continues to deliver high-quality clusters, as evidenced by high homogeneity, completeness, and V-measure scores. These scores confirm that the algorithm not only places similar data points into the same clusters but also maintains clear separations between different groups, which is essential for tasks like object recognition, document categorization, and user profiling.

In summary, the GAVS approach redefines multi-view clustering by aligning simplicity with strategic feature selection, offering a compelling alternative to more computationally expensive and complex methods. Its strong empirical results across diverse datasets validate its utility, making it a valuable addition to the toolkit for unsupervised learning and pattern discovery.

Interestingly, even advanced methods like MVLRSSC and DeepNMF, while competitive, do not always outperform simpler clustering algorithms when empowered by thoughtfully selected feature combinations. This emphasizes that feature engineering and view selection remain critical in the multiview learning paradigm and can often rival or surpass deep unsupervised learning models when handled carefully. Moreover, the consistently high homogeneity, completeness, and V-measure scores across datasets further confirm that multiview approaches not only cluster data points accurately but also maintain intra-cluster purity and inter-cluster distinctiveness.

4. Conclusion

This study presents a comprehensive evaluation of multi-view clustering techniques and introduces a novel, efficient approach—**Greedy Automatic View Selection (GAVS)**—to enhance clustering performance by systematically selecting the most complementary feature views. Our results demonstrate that clustering effectiveness is not merely a function of feature quantity but strongly depends on the quality and complementarity of the selected features.

Among the clustering algorithms tested, **Agglomerative Clustering** consistently delivered superior results, including perfect performance on the COIL-20 dataset (ARI = 1.0000, NMI = 1.0000, FMI = 1.0000), due to its hierarchical nature and its ability to leverage well-

combined feature views such as Pixel Averages and LDA. **Spectral Clustering** also showed strong performance, particularly with combinations like ('HOG', 'LDA') and ('Pixel Averages', 'HOG', 'LDA'), benefiting from its sensitivity to non-linear manifold structures. **Affinity Propagation**, while achieving high cluster purity in some datasets, struggled with global cohesion, as reflected in lower ARI and NMI scores.

The proposed **GAVS** algorithm offers a significant advancement by automatically identifying the most synergistic feature subsets based on the Normalized Mutual Information (NMI) metric. GAVS is **novel** in its greedy yet principled selection process, **efficient** in computation by avoiding exhaustive combinations, and **effective** in boosting clustering accuracy across multiple datasets. It outperformed both conventional and deep learning-based clustering models in terms of ARI, NMI, FMI, and V-measure, all while being far less time-consuming and more interpretable.

This research underscores that classic clustering algorithms, when empowered by strategic multi-view feature combinations identified via **GAVS**, can rival or surpass more complex models. Future work may focus on extending GAVS to handle deep learned embeddings, streaming data, and multi-modal fusion, thereby broadening its applicability to real-time, large-scale clustering challenges.

Declarations

Availability of data and materials- All data used shall be made available upon request.

Competing interests- The authors declare no competing interests

Funding- There is no funding.

Authors' contributions- JM: conceptualization, investigation, resources, Methodology, investigation, visualization writing original and revised draft, supervision and project administration, SK: investigation, visualization writing original and revised draft. All authors have read and approved the manuscript.

References

1) Z. He, S. Wan, M. Zappatore, and H. Lu, "A Similarity Matrix Low-Rank Approximation and Inconsistency Separation Fusion Approach for Multiview Clustering," *IEEE Trans Artif Intell.*, 5(2) 11–22 (2024). doi:10.1109/TAI.2023.3271964.
2) G. Chao, S. Sun, and J. Bi, "A Survey on Multiview Clustering," *IEEE Trans Artif Intell.*, 2(2) ,33–44 (2021). doi:10.1109/TAI.2021.3065894.
3) J. B. M. Benjamin and M. S. Yang, "Weighted Multiview Possibilistic C-Means Clustering with L2 Regularization," *IEEE Trans Fuzzy Syst.*, 30(5) ,55–66 (2022). doi:10.1109/TFUZZ.2021.3058572.
4) J. Wen et al., "A Survey on Incomplete Multiview Clustering," *IEEE Trans Syst Man Cybern Syst.*, 53(2) ,77–88 (2023). doi:10.1109/TSMC.2022.3192635.
5) Z. Tao, H. Liu, S. Li, Z. Ding, and Y. Fu, "Marginalized Multiview Ensemble Clustering," *IEEE Trans Neural Netw Learn Syst.*, 31(2), 99–110 (2020). doi:10.1109/TNNLS.2019.2906867.
6) Y. Zhang, H. Wang, and A. Stavrou, "A multiview clustering framework for detecting deceptive reviews," *J Comput Secur.*, 32(1), 12–23 (2024). doi:10.3233/JCS-220001.
7) J. Chen and G. B. Huang, "Dual distance adaptive multiview clustering," *Neurocomputing.*, 441, 101–112 (2021). doi:10.1016/j.neucom.2021.01.132.
8) Q. Wang, M. Chen, F. Nie, and X. Li, "Detecting Coherent Groups in Crowd Scenes by Multiview Clustering," *IEEE Trans Pattern Anal Mach Intell.*, 42(1) ,55–66 (2020). doi:10.1109/TPAMI.2018.2875002.
9) L. Yang, C. Shen, Q. Hu, L. Jing, and Y. Li, "Adaptive Sample-Level Graph Combination for Partial Multiview Clustering," *IEEE Trans Image Process.*, 29, 77–88 (2020). doi:10.1109/TIP.2019.2952696.
10) W. Xia, Q. Gao, Q. Wang, and X. Gao, "Tensor Completion-Based Incomplete Multiview Clustering," *IEEE Trans Cybern.*, 52(12), 101–112 (2022). doi:10.1109/TCYB.2021.3140068.
11) X. Fang, Y. Hu, P. Zhou, and D. O. Wu, "V3H: View Variation and View Heredity for Incomplete Multiview Clustering," *IEEE Trans Artif Intell.*, 1(3), 33–44 (2020). doi:10.1109/TAI.2021.3052425.
12) S. E. Abhadiomhen et al., "Spectral type subspace clustering methods: multi-perspective analysis," *Multimed Tools Appl.*, 83(16) ,55–66 (2024). doi:10.1007/s11042-023-16846-0.
13) Z. Kang et al., "Partition level multiview subspace clustering," *Neural Netw.*, 122, 77–88 (2020). doi:10.1016/j.neunet.2019.10.010.
14) G. Du, L. Zhou, K. Lü, H. Wu, and Z. Xu, "Multiview Subspace Clustering With Multilevel Representations and Adversarial Regularization," *IEEE Trans Neural Netw Learn Syst.*, 34(12), 101–112 (2023). doi:10.1109/TNNLS.2022.3165542.
15) X. Fang, Y. Hu, P. Zhou, and D. Wu, "ANIMC: A Soft Approach for Autoweighted Noisy and Incomplete Multiview Clustering," *IEEE Trans Artif Intell.*, 3(2) ,44–55 (2022). doi:10.1109/TAI.2021.3116546.
16) Z. Liu, Y. Li, L. Yao, X. Wang, and F. Nie, "Agglomerative Neural Networks for Multiview Clustering," *IEEE Trans Neural Netw Learn Syst.*, 33(7) ,66–77 (2022). doi:10.1109/TNNLS.2020.3045932.

17) Z. Xu, S. Tian, S. E. Abhadiomhen, and X. J. Shen, "Robust multiview spectral clustering via cooperative manifold and low rank representation induced," *Multimed Tools Appl.*, 82(16) ,99–110 (2023). doi:10.1007/s11042-023-14557-0.
18) W. Gao, S. Dai, S. E. Abhadiomhen, W. He, and X. Yin, "Low Rank Correlation Representation and Clustering," *Sci Program.*, 2021 ,12–23 (2021). doi:10.1155/2021/6639582.
19) S. Wang, L. Chen, N. Zheng, L. Li, F. Peng, and J. Lu, "Shared and individual representation learning with Feature Diversity for Deep MultiView Clustering," *Inf Sci.*, 647, 101–112 (2023). doi:10.1016/j.ins.2023.119426.
20) Q. Wang, Z. Tao, W. Xia, Q. Gao, X. Cao, and L. Jiao, "Adversarial Multiview Clustering Networks With Adaptive Fusion," *IEEE Trans Neural Netw Learn Syst.*, 34(10) ,33–44 (2023). doi:10.1109/TNNLS.2022.3145048.
21) N. Liang, Z. Yang, L. Li, Z. Li, and S. Xie, "Incomplete Multiview Clustering With Cross-View Feature Transformation," *IEEE Trans Artif Intell.*, 3(5) ,55–66 (2022). doi:10.1109/TAI.2021.3139573.
22) S. Joshi, J. Ghosh, M. Reid, and O. Koyejo, "Rényi divergence minimization based co-regularized multiview clustering," *Mach Learn.*, 104(2–3), 77–88 (2016). doi:10.1007/s10994-016-5543-2.
23) L. Berkani, L. Betit, and L. Belarif, "Bso-mv: An optimized multiview clustering approach for items recommendation in social networks," *J Univers Comput Sci.*, 27(7), 101–112 (2021). doi:10.3897/JUCS.70341.
24) S. Zhao, L. Fei, J. Wen, J. Wu, and B. Zhang, "Intrinsic and Complete Structure Learning Based Incomplete Multiview Clustering," *IEEE Trans Multimed.*, 25, 33–44 (2023). doi:10.1109/TMM.2021.3138638.
25) Y. Zhang, H. Wang, and A. Stavrou, "A multiview clustering framework for detecting deceptive reviews," *J Comput Secur.*, 32(1) ,33–44 (2024). doi:10.3233/JCS-220001.
26) Q. Wang, Z. Tao, W. Xia, Q. Gao, X. Cao, and L. Jiao, "Adversarial Multiview Clustering Networks With Adaptive Fusion," *IEEE Trans Neural Netw Learn Syst.*, 34(10), 77–88 (2023). doi:10.1109/TNNLS.2022.3145048.
27) S. Wang, L. Chen, N. Zheng, L. Li, F. Peng, and J. Lu, "Shared and individual representation learning with Feature Diversity for Deep MultiView Clustering," *Inf Sci.*, 647 ,55–66 (2023). doi:10.1016/j.ins.2023.119426.
28) Y. Chen, X. Xiao, Z. Hua, and Y. Zhou, "Adaptive Transition Probability Matrix Learning for Multiview Spectral Clustering," *IEEE Trans Neural Netw Learn Syst.*, 33(9) ,77–88 (2022). doi:10.1109/TNNLS.2021.3059874.
29) Z. Liu, Y. Li, L. Yao, X. Wang, and F. Nie, "Agglomerative Neural Networks for Multiview Clustering," *IEEE Trans Neural Netw Learn Syst.*, 33(7) ,99–110 (2022). doi:10.1109/TNNLS.2020.3045932.
30) M. Li, S. Wang, X. Liu, and S. Liu, "Parameter-Free and Scalable Incomplete Multiview Clustering With Prototype Graph," *IEEE Trans Neural Netw Learn Syst.*, 35(1) ,11–22 (2024). doi:10.1109/TNNLS.2022.3173742.
31) 31.X. Yang, C. Deng, Z. Dang, and D. Tao, "Deep Multiview Collaborative Clustering," *IEEE Trans Neural Netw Learn Syst.*, 34(1) ,33–44 (2023). doi:10.1109/TNNLS.2021.3097748.
32) 32.X. Fang, Y. Hu, P. Zhou, and D. Wu, "V3H: View Variation and View Heredity for Incomplete Multiview Clustering," *IEEE Trans Artif Intell.*, 1(3), 33–44 (2020). doi:10.1109/TAI.2021.3052425.
33) 33. S. E. Abhadiomhen et al., "Spectral type subspace clustering methods: multi-perspective analysis," *Multimed Tools Appl.*, 83(16) ,101–112 (2024). doi:10.1007/s11042-023-16846-0.
34) X. Ji, L. Yang, and S. Yao, "Adaptive anchor-based partial multiview clustering," *IEEE Access.*, 8, 77–88 (2020). doi:10.1109/ACCESS.2020.3025881.
35) W. Xia, X. Zhang, Q. Gao, X. Shu, J. Han, and X. Gao, "Multiview Subspace Clustering by an Enhanced Tensor Nuclear Norm," *IEEE Trans Cybern.*, 2021 ,55–66 (2021). doi:10.1109/TCYB.2021.3052352.

Cite: J. Mankar, S. Kamalapur, "Efficient Multi-View Clustering via Greedy Automatic View Selection and Diverse Feature Integration". Evergreen, 12 (03) 1802-1826 (2025). https://doi.org/10.5109/7388866.