Comparative study of Unsupervised learning algorithms on Multispectral SatelliteImages

PrasadKaviti[1], Valli Kumari Vatsavayi[2]

[1]Computer Science and Systems Engineering, Andhra University, AP, India
[2]Computer Science and Systems Engineering, Andhra University, AP, India
([1]prasadkaviti@gmail.com, [2]vallikumari@gmail.com)

Abstract**—**In computer vision, one of the evolving fields with lot of applications is image clustering. The challenging problem is to choose appropriate clustering algorithm for a given image data set and the clustering algorithm have a strong impact on clustering accuracy. There arevarious state of art clustering algorithms available with a dependency on parameter tuning. K-Means algorithm is one of such standard algorithms. K-Means has the requirement of prior specification of number of clusters for centroids initialization. But number is unknown in most of the cases.Another popular technique is Agglomerative clustering which depends on connectivity matrix and requires the number of clusters to be specified in prior. The connectivity matrix consumes more space hence more complexity. Another popular clustering technique is Mean-Shift clustering which is nonparametric in nature. Mean-Shift clustering algorithm required bandwidth parameter 'h' to be tuned even though it is nonparametric. Spectral Clustering is another standard clustering algorithmdepends on similarity matrix which leads to a memory constraint. DBSCAN is another standard algorithm which requires two parameters eps,minpts to be tuned and it is density based algorithm. In this paper, a comparative study on all above mentioned techniques is presented. Experimentation is done on a sentinel image for each of these methods.

Keywords—*Clustering; K-Means; DBSCAN; Agglomerative clustering; Mean-Shift; Spectral clustering*

I. INTRODUCTION

One of the interesting fields of Machine Learning is Unsupervised learning or Clustering where similar groups are classifiedfrom the datasets. Image clustering partitions image data into clusters based on similarities. Similarity might be images looking similar or similar size or similar pixel distribution, similar background etc. Similarity definition changes with different set of rules for every methodology. Unlabeled image data is handled by image clustering approach since it is an unsupervised learning method.Specific image vector should be derived for different use cases. There are different clustering algorithms introduced [7]. Each clustering algorithm follows an assumption of grouping data points with similar qualities and features in a feature space. Thus, all nearby points are clustered with some sort of similarities. We cannot make sure of this assumption because high volumes of image data are being collected in real world scenarios. So, it is challenging to handle huge amounted, unstructured and unlabeled image collection.

The clustering algorithms family differ by approach [22] and differ using various metrics for measuring the distance between data objects (Euclidean distance, Manhattan distance, etc.). The criteria to join two clusters must be chosen by the user and this selection may differs from user to user. The main objectives of clustering algorithms are scalability, dealing with different types of attributes, discovering clusters with arbitrary shape, minimal requirements of domain knowledge to determine input parameters, ability to deal with noise and outliers, insensitivity to order of input records, high dimensionality and usability. Also, there are wide ranges of issues that are faced during clustering like: dealing with large number of dimensions and large number of data items which can become an overhead because of time complexity, defining distance measure should be made appropriately which is a tedious task when it comes to multidimensional spaces, etc. Hence care should be taken while applying the clustering algorithm to images.

In computer vision, one of the evolving fields with lot of applications is image clustering. The challenging problem is to choose appropriate clustering algorithm for a given image data set and the clustering algorithm have a strong impact on clustering accuracy. There are various state of art clustering algorithms available with a dependency on parameter tuning.Some of them are K-Means [1][10], DBSCAN [9], Agglomerative clustering [8], Spectral clustering [15][16], Affinity propagation and Mean-Shift [2][3]. But these

algorithms require tuning of some parameters. K-Means algorithm is one of such standard algorithms. K-Means has the requirement of prior specification of number of clusters for centroids initialization. But number is unknown in most of the cases. Another popular technique is Agglomerative clustering which depends on connectivity matrix and requires the number of clusters to be specified in prior. The connectivity matrix consumes more space hence more complexity. Another popular clustering technique is Mean-Shift clustering [3] which is nonparametric in nature [2]. Mean-Shift clustering algorithm required bandwidth parameter 'h' to be tuned [5] even though it is nonparametric. Spectral Clustering [17][18] is another standard clustering algorithm depends on similarity matrix which leads to a memory constraint. DBSCAN [6][11] is another standard algorithm which requires two parameters eps,minpts to be tuned and it is density based algorithm.Affinity propagation [19] is another clustering algorithm depends on similarity matrix.
In this paper, a survey is presented on all the mentioned clustering algorithms. For results comparison, a sentinel image is considered.

II. EXISTING CLUSTERING ALGORITHMS

A. K-Means

K-means clustering[6][10] is an unsupervised learning algorithmto cluster unlabeled data which is non categorized and not grouped. The goal of this algorithm is to find groups in the data, with the number of groups represented by the variable K. The algorithm works iteratively to assign each data point to one of K groups based on the features that are provided. Data points are clustered based on feature similarity.
First is the data assignment step in which each centroid defines one of the clusters. In this step, each data point is assigned to its nearest centroid, based on the squared Euclidean distance. The next step is Centroid update. In this step, the centroids are recomputed. This is done by taking the mean of all data points assigned to that centroid's cluster.
The algorithm iterates between steps one and two until a stopping criterion is met (i.e., no data points change clusters, the sum of the distances is minimized, or some maximum number of iterations is reached).
Advantages:

- Easy to implement
- K-Meanscomputationally faster because of its number of variables

Disadvantages:

- Difficult to predict the number of clusters
- Final results could be strongly impacted bye initial seeds.

Algorithm

```
Input: Set of data points $P = \{p_1, p_2, \dots, p_n\}$, K – Number of clusters desired
Output: K number of clusters
    1. Choose K points randomly from P, say $M = \{m_1, m_2, \dots, m_k\}$
    2. Do
        i) Select a point $p_i \in P$
        ii) Calculate distance between $p_i$ and all points of M
        iii) Assign $p_i$ to the nearest point in M

       For all 'n' points
    3. Calculate mean of all newly assigned points of M and update $\{m_1, m_2, \dots, m_k\}$
    4. Repeat steps 2,3 until no change in cluster centres.
```

B. Agglomerative clustering:

The agglomerative clustering [8] is the most common type of hierarchical clustering used to group objects in clusters based on their similarity.
Agglomerative clustering works in a "bottom-up" manner. That is, each object is initially considered as a single-element cluster (leaf). At each step of the algorithm, the two clusters that are the most similar are combined into a new bigger cluster (nodes). This procedure is iterated until all points are member of just one single big cluster (root). The result is a tree-based representation of the objects, named dendrogram.

Advantages:

- No specification of number of clusters.
- Easy to implement and performs well in some cases.
- Easy to decidethe number of clusters by observing the dendrogram

Disadvantages:

- Once the instances are assigned, we cannot undo the assignment. So the instances cannot be moved around after the previous step.
- Time complexity is more for large datasets
- Final resultsare impacted by the initial seeds
- Sensitive to outliers

Algorithm

```
Input: Set of data points $P = \{p_1, p_2, \dots, p_n\}$, A- Adjacency or Connectivity matrix to show distance between points.
Output: A dendogram of ordered points.
```

1. Assume each point $p_i \in P$ as an individual cluster, i.e, the cluster set C be $C = \{\{p_1\}, \{p_2\}, \dots, \{p_n\}\}$.
2. Do
   i) From A, take two closest clusters in C and combine them as a single cluster.
   ii) Update A, C with new clusters.

   Until a single cluster remains in C

C. DBSCAN

The DBSCAN [9] algorithm should be used to find associations and structures in data that are hard to find manually but that can be relevant and useful to find patterns and predict trends.
Based on a set of points, DBSCAN groups together points that are close to each other based on a distance measurement (usually Euclidean distance) and a minimum number of points. It also marks as outliers the points that are in low-density regions.
The DBSCAN algorithm basically requires 2 parameters:
Takes eps - specifies how close points should be to each other to be considered a part of a cluster. It means that if the distance between two points is lower or equal to this value (eps), these points are considered neighbors.
Takes minPoints - the minimum number of points to form a dense region.

DBSCAN will start by dividing the data into n dimensions[11]. It will start at a random point, and it will count how many other points are nearby. DBSCAN will continue this process until no other data points are nearby, and then it will look to form a second cluster.
A point is a core point if it has more than a specified number of points (MinPts) within Eps. These points belong to a dense region and are at the interior of a cluster
A border point has fewer than MinPts within Eps but is in the neighbourhood of a core point.
A noise point is any point that is not a core point or a border point.

Advantages:
- Best at separating high density clusters and low-density clusters of a dataset.
- DBSCAN greatly handles outliers of a dataset.

Disadvantages:
- Clusters with varying densities cannot be dealt greatly.DBSCAN separates high density clusters and low-density clusters but struggles with similar dense clusters.
- Struggle to deal with high dimensionality data.

Algorithm

Input: Set of data points $P = \{p_1, p_2, \dots, p_n\}$, eps – Maximum radius of neighborhood, minpts – Minimum number of points to form a cluster.
Output: Set of clusters formed

```
1. Randomly choose a point $p_i$
2. Form a cluster C with other points $p_j \in P \mid distance(p_i, p_j) < eps$ , i.e, density reachable from $p_i$.
3. If (number of points in C >minpts) then form a cluster.
4. Else $if(p_i == borderpoint || nosiepoint)$ then choose another point from P.
5. Repeat from step 1 to 4 until all the points in P are processed.
```

D. Spectral Clustering

In spectral clustering [15], the data points are treated as nodes of a graph. Thus, clustering is treated as a graph partitioning problem. The nodes are then mapped to a low-dimensional space that can be easily segregated to form clusters. An important point to note is that no assumption is made about the shape/form of the clusters.
To compute a similarity graph, we simply connect all points with each other, and we weight all edges by similarity sij. This graph should model the local neighborhood relationships, thus similarity functions such as Gaussian similarity function are used. Some data points in the same cluster may also be far away–even farther away than points in different clusters. Our goal then is to transform the space so that when the 2 points are close, they are always in same cluster, and when they are far apart, they are in different clusters. We need to project our observations into a low-dimensional space. For this, we compute the Graph Laplacian, which is just another matrix representation of a graph and can be useful in finding interesting properties of a graph. The whole purpose of computing the Graph Laplacian L was to find eigenvalues and eigenvectors for it, in order to embed the data points into a low-dimensional space. To create clusters, compute k-means on the 2nd eigenvalue of the eigenvectors.

Advantages:
- No strong assumptions on the cluster statistics
- Reasonably fast for large and sparse data sets with thousands of elements.

Disadvantages:

- Depending on initial centroids, the clusters may vary in each step.
- Expensive in computation for large datasets, because clustering needs to be performed on vectors obtained by eigenvalues and eigenvectors computation.

Algorithm

Input: Set of data points $P = \{p_1, p_2, \dots, p_n\}$, A – Similarity matrix containing distance between points
Output: Set of clusters

1. Construct Diagonal matrix D, where $D_{ij} = \sum_{j=1}^{n} A_{ij}$ if (i=j), otherwise $D_{ij} = 0$
2. Calculate Laplacian matrix $L = D - A$.
3. Perform Eigen value decomposition on L to solve $min_{F^T F=I} trace(F^T LF)$, where F is the cluster indicator matrix.
4. Apply K-Means clustering algorithm on the eigen values obtained in step 3

E. Mean-Shift

Mean shift considers feature space of data points as a probability density function [12]. Mean shift considers the input data points are sampled from the formed probability density function. The dense regions (or clusters) present in the obtained feature space correspond to the local maxima (or mode) of the probability density function. We can also identify clusters associated with the given mode using Mean-shift [4]. For each data point, Mean-shift associates it with the nearby peak of the dataset's probability density function. For each data point, Mean-shift defines a window around it and computes the mean of the data point. Then it shifts the centre of the window to the mean and repeats the algorithm till it converges. After each iteration, we can consider that the window shifts to a denser region of the dataset. At the high level, we can specify Mean Shift as follows: 1) Fix a window around each data point. 2) Compute the mean of data within the window. 3) Shift the window to the mean and repeat till convergence.

Advantages:

- Mean shift is suitable for real data analysis due to its application-independent behavior.
- No assumption of any predefined shape on data clusters.
- Arbitrary feature spaces are handled well by Mean-Shift.
- Bandwidth is the only parameter on which the procedure relies.

Disadvantages:

- The window size selection is not trivial.
- Inappropriate window size may lead to merging modes orproduce additional "shallow"modes.

Algorithm

Input: Set of data points $P = \{x_1, x_2, \dots, x_n\}$, bandwidth h.
Output: Set of clusters

1. Take each data point $x \in P$ as a seed
2. Compute the threshold value t as t = 1e-3*h for convergence of Mean Shift algorithm
3. For each seed $x \in P$, find the nearest neighbors N(x) within the bandwidth (h) distance and the distance metric considered is the Euclidean distance$E(x_1, x_2)$ where
$$E(x_1, x_2) = \sqrt{x_1^2 + x_2^2 - 2x_1x_2}$$
4. For each seed $x \in P$, calculate the weighted mean m(x) where
$$m(x) = \frac{\sum_{x_i \in N(x)} K(x_i - x)x_i}{\sum_{x_i \in N(x)} K(x_i - x)}$$
Where,
   - N (x) = Function determines the nearest neighbors of x
   - $K(x_i - x)$ = Kernel Function
5. Update $x \leftarrow m(x)$ for every seed $x \in P$
6. Repeat the steps 3,4,5 for all seeds till $m(x) - x < t$ or until the convergence of m(x)

F. Partition Around Medoids:

K-Medoids [14][20] (also called as Partitioning Around Medoid) algorithm was proposed in 1987 by Kaufman and Rousseeuw. A medoid can be defined as the point in the cluster, whose dissimilarities with all the other points in the cluster is minimum.
The k-medoids algorithm is a clustering approach related to k-means clustering for partitioning a data set into k groups or clusters [13]. In k-medoids clustering, each cluster is represented by one of the data points in the cluster. These points are named cluster medoids.
The term medoid refers to an object within a cluster for which average dissimilarity between it and all the other the members of the cluster are minimal. It corresponds to the most centrally located point in the cluster. These objects (one per cluster) can be considered as a representative example of the members of that cluster which may be useful in some situations. After finding a set of k medoids, clusters are constructed by assigning each observation to the nearest medoid.

The PAM algorithm is based on the search for k representative objects or medoids among the observations of the data set. Each selected medoid m and each non-medoid data point are swapped and the objective function is computed. The objective function corresponds to the sum of the dissimilarities of all objects to their nearest medoid.

K-medoid is a robust alternative to k-means clustering [514]. This means that, the algorithm is less sensitive to noise and outliers, compared to k-means, because it uses medoids as cluster centers instead of means (used in k-means).

The k-medoids algorithm requires the user to specify k, the number of clusters to be generated (like in k-means clustering).

Advantages:

- Understandable and implementation is easy.
- Fast and Fixed number of steps to converge.
- K-Medoids algorithm is robust to outliers than other partitioning algorithms.

Disadvantages:

- K-Mediodstakes more time to execute than K-Means, hence time complexity is more.
- For large datasets, K-Medoids doesn’t scale well.
- Initial partitions determine results and the total runtime

Algorithm

```
Input: Set of data points $P = \{p_1, p_2, \dots, p_n\}$, K – Number of clusters desired
Output: Set of K clusters
1. Initially take K random points from P as medoids, Say $M = \{m_1, m_2, \dots, m_k\}$
2. Do
        iv) Select a point $p_i \in P$
        v) Calculate dissimilarity between $p_i$ and all points of M, $dissimilarity = p_i - M$
        vi) Assign $p_i$ to the point in M with less dissimilarity

        For all ‘n’ points
3. For each medoid in M, for each non-medoid data point P
    a) Randomly choose a point $p_j \in P$, and swap $p_j$ with its medoid $m_{actual} \in M$
    b) Follow step 2, and find configuration cost S.
    c) If $S < 0$ then keep the swap, else undo the swap and form new set of K clusters
4. Repeat steps 3 until no change in cluster centers.
```

III. Variants

K-Means variants:

k-medians clustering uses the median in each dimension instead of the mean, and this way minimizes L1 norm.

Fuzzy C-Means [26]Clustering is a soft version of k-means, where each data point has a fuzzy degree of belonging to each cluster.

HG-means is a sophisticated extension of K-means which escapes from local minima through iterative recombination steps and K-means improvement phases

Hierarchical variants such as Bisecting k-means [24], X-means clustering and G-means clustering [25] repeatedly split clusters to build a hierarchy and can also try to automatically determine the optimal number of clusters in a dataset.

Hierarchical Clustering variants:

Three variants of hierarchical clustering are BIRCH, CURE, CHAMELEON [23]. BIRCH, developed in 1996, follows macro-clustering idea by using clustering feature tree and incrementally adjust the quality of sub clusters. The second one is CURE, developed in 1998. That method essentially represents the cluster using a set of well-scattered representative points. The third one, CHAMELEON, was developed in 1999. It uses graph partitioning methods on K-nearest neighbor graph of the data.

DBSCAN variants:

DBSCAN separates data to clusters with similar densities. It forces clusters to be in similar density. A Dynamic Method for Discovering Density Varied Clusters (DMDBSCAN) [21], a LocalDensity Based Spatial Clustering Algorithm with Noise (LDBSCAN) and an Enhanced Density Based Spatial Clustering of Applications with Noise (EDBSCAN) are developed for separating data to clusters with different densities.

DMDBSCAN, LDBSCAN, EDBSCAN [21] are developed for finding clusters with different densities. Two of them (DBSCAN-GM, VDBSCAN) are developed for separating data to clusters without input parameters. Two of them (FNDBSCAN, Soft-DBSCAN) are enhanced to ensure robustness. Three of them (Active-DBSCAN, FDBSCAN, Fast ParzenWindow) are developed for speed.

IV. Results

K-Means clustering algorithm required the parameter K to be tuned. For simplicity the K value is taken as 8 clusters. Silhouette score is 0.72. The main disadvantage for K-Means clustering is to define the value K. If K value is known prior, K-Means clustering will give better results. There are some methods to find the value of K like Elbow method and Silhouette valued graph method. But those are applicable when we interpret or assume the range of values for K.

DBSCAN on the other hand required eps and minpts parameters to be tuned. In this project default values of 0.5 and 5 for eps and minpts respectively are taken. Results scored 0.65 Silhouette value. But those default values are suited for the image that is considered. If the image changes, then the performance is also changing.

Mean-Shift clustering technique is dependent on bandwidth parameter. There is no chance of having a global value for bandwidth since it changes with the data set. The Elbow method and Silhouette valued graph methods can also be used in Mean-Shift to determine the value of bandwidth. But before that we should know the range of values where the bandwidth value resides. For results comparison, Silverman rule is used for bandwidth calculation. Mean-Shift clustering with Silverman bandwidth is giving better performance with 0.74 Silhouette value.

Agglomerative clustering depends on the number of clusters which is given as 8. Also the agglomerative clustering algorithm optimized by converting data points into the grid. It scored 0.67 Silhouette score.

Pam on the other hand also required number of clusters to be specified. Here also the standard value 8 is given for K. The silhouette value scored by Pam is 0.72.

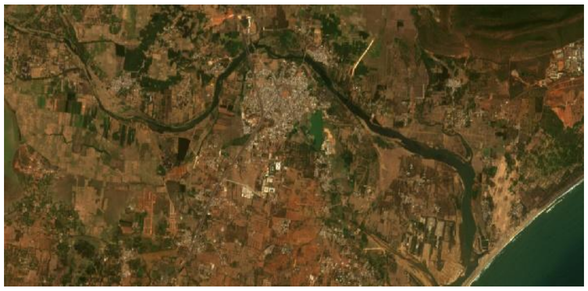

Fig 1: Original Image

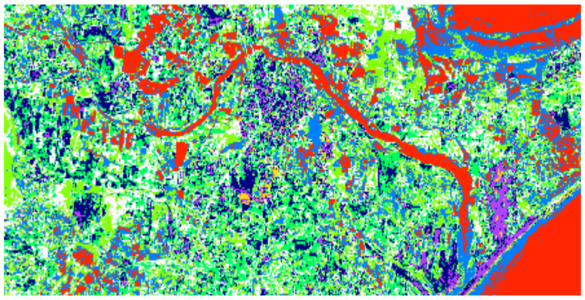

Fig 2: K-Means Clustering

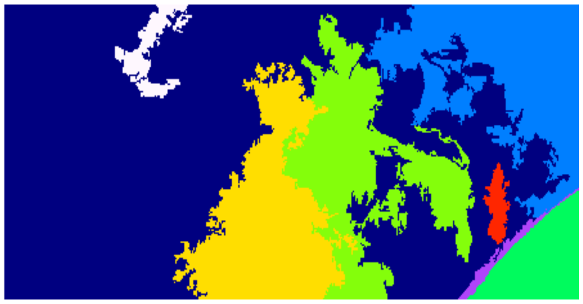

Fig 3: Agglomerative Clustering

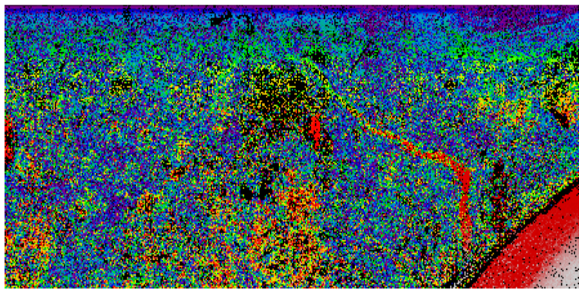

Fig 4: Meanshift Clustering

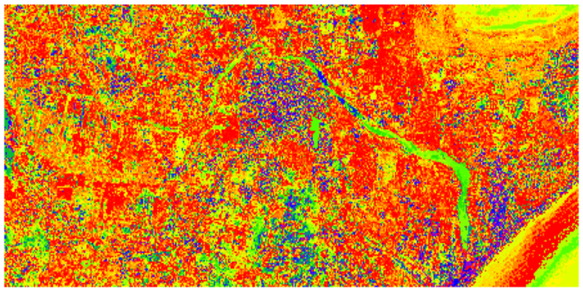

Fig 5: DBSCAN Clustering

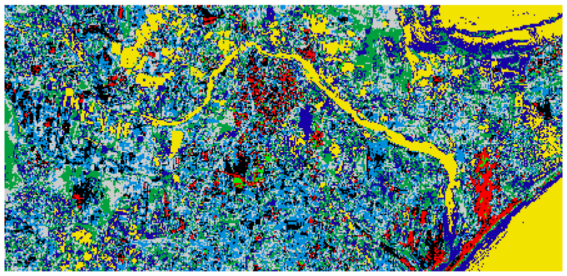

Fig 6: PAM clustering

TABLE I. SILHOUETTE VALUE COMPARISON

Clustering Algorithm	**Silhouette Value**
K-Means	0.72
DBSCAN	0.65
Mean-shift	0.74
Agglomerative Clustering	0.67
PAM	0.72

REFERENCES

[1] "Web Scale K-Means clustering" D. Sculley, Proceedings of the 19th international conference on World wide web (2010)

[2] Biplab Banerjee, Surender Varma G, Krishna Mohan Buddhi Raju, "Satellite Image Segmentation: A Novel Adaptive Mean-Shift Clustering Based Approach", Ieee Igrss,2012.

[3] D. Comaniciu and P. Meer, "Mean-shift: A robust approach toward feature space analysis ". IEEE Trans. Pattern Anal. Machine Intell., 24:603–619, 2002.

[4] Johannes Jordan, Elli Angelopoulou , "Meanshift Clustering For Interactive Multispectral Image Analysis".

[5] ZHAO Yunji1, PEI Hailong2, LIU Baoluo3 , "Mean-shift algorithm based on kernel bandwidth adaptive adjust ".

[6] M. Daszykowski, B. Walczak, D. L. Massart, "Density-based clustering for exploration of analytical data", Analytical and Bioanalytical Chemistry, October 2004, Volume 380, Issue 3, pp 370–372.

[7] Rokach, Lior, and Oded Maimon. "Clustering methods." Data mining and knowledge discovery handbook. Springer US, 2005. 321-352.

[8] Zhang, et al. "Graph degree linkage: Agglomerative clustering on a directed graph." 12th European Conference on Computer Vision, Florence, Italy, October 7–13, 2012.

[9] Ester, Martin; Kriegel, Hans-Peter; Sander, Jörg; Xu, Xiaowei (1996). Simoudis, Evangelos; Han, Jiawei; Fayyad, Usama M., eds. "A density-based algorithm for discovering clusters in large spatial databases with noise". Proceedings of the Second International Conference on Knowledge Discovery and Data Mining (KDD-96).

[10] Hamerly, Greg; Drake, Jonathan (2015). "Accelerating Lloyd's algorithm for k-means clustering".

[11] Sander, Jörg (1998). Generalized Density-Based Clustering for Spatial Data Mining. München: Herbert Utz Verlag. ISBN 3-89675-469-6.

[12] "Mean shift: A robust approach toward feature space analysis." D. Comaniciu and P. Meer, IEEE Transactions on Pattern Analysis and Machine Intelligence (2002)

[13] "Analysis of K-Means and K-Medoids Algorithm For Big Data", PreetiArora, DeepaliDr., ShipraVarshney, 1st International Conference on Information Security & Privacy 2015, Volume 78, 2016, Pages 507-512.

[14] H.S. Park, C.H. Jun, "A simple and fast algorithm for K-medoids clustering, Expert Systems with Applications",36, (2) (2009), 3336–3341.

[15] Jianbo Shi and Jitendra Malik, "Normalized Cuts and Image Segmentation", IEEE Transactions on PAMI, Vol. 22, No. 8, Aug 2000.

[16] Ng, Andrew Y and Jordan, Michael I and Weiss, Yair (2002). "On spectral clustering: analysis and an algorithm". Advances in Neural Information Processing Systems.

[17] Wang Chongjun ; Li Wu jun ; Ding Lin ; Tian Juan ; Chen Shifu, "Image segmentation using spectral clustering", 17th IEEE International Conference on Tools with Artificial Intelligence (ICTAI'05)

[18] Bo Chen, Bin Gao, Tie-Yan Liu, Yu-Fu Chen, Wei-Ying Ma, "Fast Spectral Clustering of Data Using Sequential Matrix Compression", European Conference on Machine Learning ECML 2006: Machine Learning: ECML 2006 pp 590-597

[19] Brendan J. Frey; Delbert Dueck (2007). "Clustering by passing messages betweendata points". Science. 315 (5814): 972–976

[20] H.S. Park , C.H. Jun, A simple and fast algorithm for K-medoids clustering, Expert Systems with Applications, 36, (2) (2009), 3336–3341.

[21] Fatma GünseliYaşar, GözdeUlutagay, "CHALLENGES AND POSSIBLE SOLUTIONS TO DENSITY BASED CLUSTERING", 2016 IEEE 8th International Conference on Intelligent Systems

[22] Seema Wazarkar, Bettahally N. Keshavamurthy, "A Survey on Image Data Analysis through Clustering Techniques for Real World Applications", J. Vis. Commun. Image R. (2018), doi: https://doi.org/10.1016/j.jvcir. 2018.07.009

[23] S.Saraswathi, Dr. Mary Immaculate Sheela, "A Comparative Study of Various Clustering Algorithms in Data Mining", IJCSMC, Vol. 3, Issue. 11, November 2014, pg.422 – 428

[24] "A comparison of document clustering techniques", M. Steinbach, G. Karypis and V. Kumar. Workshop on Text Mining, KDD, 2000.

[25] Greg Hamerly, Charles Elkan, "Learning the k in k-means", 2004

[26] "Fast and Robust Fuzzy C-Means Clustering Algorithms Incorporating Local Information for Image Segmentation", Weiling Cai, Songcan Chen and Daoqiang Zhang.