Project 1 Report

<Mugeon Kim, Harish Jayanti, Wilson FNU Ronald, Hudanyun Sheng>

# Member Roles

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
|  | Cluster  Alg | Eval  Alg | Cluster  Test | Eval  Test | Cluster  Write | Eval  Write | Cluster  Edit | Eval  Edit |
|  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |

# Experiments

## RGB Experiments and Tabular Results

|  |
| --- |
| RGB experiments |
| Input: preprocessed image |
| Output: connected component image, clustered image |
| 1: for every image in the folder |
| 2: Calculate clustered image using 5 algorithms |
| 3: Calculate connected component based on the clustered image. |
| 4: end for |

|  |
| --- |
| Algorithm Fuzzy C-means |
| 1: Set number of clusters: M, and 1 |
| 2: Initialize cluster centers. |
| 3: repeat |
| 4: for i = 1 to N do |
| 5: for j = 1 to M do |
| 6: if then |
| 7: |
| 8: else |
| 9: |
| 10: end if |
| 11: end for |
| 12: end for |
| 13: for j = 1 to M do |
| 14: Update cluster representatives |
| 15: end for |
| 16: until Change in cluster centers is small |

|  |
| --- |
| Algorithm EM-GMM |
| 1: Initialize all parameters |
| 2: t = 1. |
| 2: While convergence not yet reached: |
| 4: Compute for every and k. |
| 5: Update , , for all k. |
| 6: t = t+1 |
| 7: Check convergence criteria. |

|  |
| --- |
| Algorithm Self Organizing Maps |
| 1: Input grid size |
| 2: Randomize the node weight vectors in the grid |
| 3: For n = 1:max\_epoch |
| 4: Randomly pick an input vector, I |
| 5: For every node in the map |
| 6: Calculate the Euclidean distance between I and the map’s node weight vector |
| 7: Track the node that produces the smallest distance |
| 8: Update the weight vectors of the nodes in the neighborhood of the best matching unit using: |
|  |
|  |
|  |
| 9: End for |
| 10: End for |
| 11: Choose top “numCluster” weights and set rest to 0. |
| 12: Evaluate the membership of the input vector using weights |

|  |
| --- |
| Algorithm Spectral Clustering |
| 1: Input image I |
| 2: Unwrap I to get a vector of length N |
| 3: Generate similarity matrix S |
| 4: For i= 1:N |
| 5: For j = 1:N |
| 6: |
| 7: End for |
| 8: End for |
| 9: Calculate diagonal matrix D |
| 10: For i= 1:N |
| 11: |
| 12: End for |
| 13: Calculate normalized laplacian using |
| 14: Perform eigen-decomposition on L and choose “numClust” lowest eigenvectors, E |
| 15: Generate projected matrix using |
| 16: Cluster the N input vectors in P using K-means algorithm with “numCluster” centroids |
| 17: Generate label by Euclidean distance metric between the centroids and the input vector |

|  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| Algorithm | Kmeans  mean | Kmeans  Std.dev. | SOM  mean | SOM  Std.dev. | FCM  mean | FCM  Std.dev. | Spectral  mean | Spectral  Std.dev. | GMM  mean | GMM  Std.dev. |
| Score | 0.8067 | 0.0755 | 0.756 | 0.1253 | 0.8253 | 0.0742 | 0.7833 | 0.1134 | 0.7864 | 0.0949 |



## Hyperspectral Experiments

|  |
| --- |
| Hyperspectral experiments |
| Input: preprocessed image |
| Output: clustered image |
| 1: for every image in the folder |
| 2: Reduce dimensionality using principal component analysis |
| 3: Calculate clustered image using 5 algorithms |
| 4: end for |

The objective-level consistency error (OCE), which quantifies the similarity (or discrepancy) between a segmented image and the ground truth image at the objective level. To begin, assume is a reference image where is the j-th fragment in . Assume further that is the segmented image where is the i-th fragment in .

A partial error measure is defined as

,

,

,

Where is the delta function whose value equals 1 if the input is 0 and whose value is 0 otherwise. . The OCE is then defined to be

Pseudocode used to calculate the objective-level consistency error:

|  |
| --- |
| Algorithm Calculation of Objective-level Consistency error for Hyperspectral images |
| Input: cluster image, ground truth image, ground truth mask |
| Output: the score for the segmentation result. |
| 1: Multiply the cluster image by the ground truth mask and get the updated cluster image. |
| 2: Calculate by calling function *calculateE*. |
| 3: Calculate by calling function *calculateE*. |
| 4: Return segmentation score by taking the minimum of and . |

|  |
| --- |
| Algorithm Calculation of score |
| Input: cluster image, ground truth image |
| Output: the score |
| 1: Calculate M and N |
| 2: for j:=1 to M |
| 3: Calculate A by finding the location ground truth equal to j |
| 4: for i:=1 to N |
| 5: Calculate B by finding the location cluster image equal to j |
| 6: Calculate : |
| 7: end for |
| 8: Calculate : |
| 9: end for |
| 10: Normalize and :, |
| 11: Calculate : . |

# Observations

## K-means advantages and disadvantages

## Self-Organizing Map advantages and disadvantages

## FCM advantages and disadvantages

## Spectral Clustering advantages and disadvantages

## Gaussian Mixture Models advantages and disadvantages

Relative performance of algorithms

|  |  |  |
| --- | --- | --- |
| Ranks | RGB Rankings | Hyperspectral Rankings |
| 1 | SOM | GMM |
| 2 | Spectral | Kmeans |
| 3 | GMM | FCM |
| 4 | Kmeans | SOM |
| 5 | FCM | Spectral |