International Journal of Emerging Trends & Technology in Computer Science (IJETTCS)
Web Site: www.ijettcs.org Email: editor@ijettcs.org, editorijettcs@gmail.com
Volume 3, Issue 1, January – February 2014 ISSN 2278-6856

Image Classification Using K-mean Algorithm

Haval M. SIDQI[1] and Jamal F. KAKBRA[2]

[1]Master in Computer Science, Sulaimani Polytechnic University
Qirga St., Sulaimani, Kurdistan Region of Iraq

[2] Master in Information and Communication Technology, Sulaimani Polytechnic University
Qirga St., Sulaimani, Kurdistan Region of Iraq

Abstract: ***This research aims to investigate the required image operation to extract the features. Clustering analysis plays an important role in the scientific research and commercial application. K-means algorithm is a widely used partition method in clustering techniques. As the dataset's scale increases rapidly, it is difficult to use K-means to deal with massive amount of data. A parallel strategy that incorporated into clustering method and a K-mean algorithm are proposed. For enhancing the efficiency of K-mean, dynamic load balance is introduced.***

The disadvantages behind the algorithm are the cost of time time calculation when the number of cluster is taken high. For example if the number of cluster is 100 then the coding time is equal to 21second for color image and the coding time is equal to 5 second for grey image.

Keywords: Cluster, Pixel, Centroids, RGB, Gray

1- INTRODUCTION

In this paper the results of some tests conducted to assess the K-mean algorithm performance. The K-means method has been shown to be effective in producing good clustering results for many practical applications.
K-means method is well known for its relatively simple implementation and decent results. However, a direct algorithm of k-means method requires time proportional to the product of the number of documents (vectors) and number of clusters per iteration. This is computationally very expensive especially for large datasets[1].
The algorithm is an iterative procedure and requires the number of clusters k to be given a priory. Suppose that the k initial cluster centers are given, and then the algorithm follows the steps as below[2]:

1. Compute the Euclidean distance from each of the documents to each cluster center. A document is associated with a cluster; its distance from that cluster is the smallest one among all of the distances.
2. After this assignment or association of all the documents with one of k clusters is done, each cluster center is recomputed so as to reflect the true mean of its constituent documents.

Steps 1 and 2 are repeated until the convergence is achieved.

2- IMAGE DATA

The pixels values are arranged row by row, starting from the bottom raw toward the top raw. In case of palette driven formats, the listed pixel values represent the color index, while in other cases the blue, green and red components values are grouped (arranged sequentially) to construct the color value. The below code summarizes the implemented steps to load the cell bitmap images[4][5].

```
Input: FileName
Output: Wid, Hgt, Red(), GrnO, Blu()
Open FileName For Binary As #1: Get #1,, Bmp
Wid = Bmp.Wid: Hgt = Bmp.Hgt
Wm = Wid - 1: Hm = Hgt- 1
If Bmp.BitPlan = 8 Then NoColor =
(Bmp.OfsPos -

54) \ 4 - 1 ReDim RGBr(NoColor): Get #1,,
RGBr Wb = Int((Wid * 8 + 31) / 32) * 4: ReDim
A(Wb - 1) ReDim Blu(Wm, Hm), Gm(Wm,
Hm), Red(Wm, Hm)
For Y = 0 To Hm: Get #1,, A: Yy =
Hm - Y For X = 0 To Wm With
RGBr(A(X))
Blu(X, Yy) = .B: Grn(X, Yy) = .G: Red(X, Yy) - .R
End With Next X Next Y Erase RGBr
Elself Bmp.BitPlan = 24 Then Wb = Int((Wid *
24 + 31) / 32) * 4: ReDim A(Wb - 1) ReDim
Blu(Wm, Hm), Gm(Wm, Hm), Red(Wm, Hm)
For Y = 0 To Hm: Get #1,, A: Yy = Hm - Y:
Xx = 0 For X = 0 To Wm: Blu(X, Yy) =
A(Xx)
Grn(X, Yy) = A(Xx + 1): Red(X, Yy) = A(Xx + 2)
Xx = Xx + 3 Next X Next Y
End If
Close #1: Erase A End
```

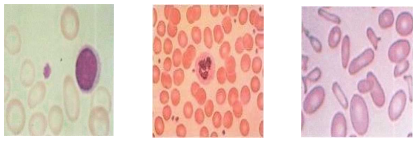

Figure 1 Blood cell image samples

3- PRE-PROCESSING

In this step the color contents of the image are analyzed to assess the dominant color of the background pixels and the target (cells pixels). One of the main problems in this

work is the color variability of both the background and the cells pixels. This variability is due to different kinds of paints used to enhance the visual appearance of the blood test samples.

To handle this task, the well-known clustering algorithm, *called k-means algorithm*, was utilized to distribute the pixel's color around two dominant colors (centroids). One of these centroids will be very close to the dominant color of the background pixels. The implemented steps could be summarized as follows:

1. Scan the pixels lay within the strip (of width=w; say w= 5) surrounding the image boundary area, here we assume that the most of the pixels lay within this surrounding area belong to the background, so the average values of the color components of pixels belong to strip will construct the initial centroid of the background color.
2. In the same way, the pixels in the interior region of the images are scanned and the average values of their colors are determined and considered as initial centroids for the target (cells) pixels.
3. Increase the distance between the initial background and target centroids by applying the following equations[3]:

$$R_c = R_h + a(R_c - R_b) \ldots\ldots \quad (1)$$
$$G_e = G_b + a(G_e - G_b) \ldots\ldots, \quad (2)$$
$$B_c = B_b + a(B_c - B_b) \ldots\ldots\ldots, \quad (3)$$

Where,

(R_C,G_C,B_C) are the red, green, and blue components of the cell centroid.

(R_b,Gb,B_b) are the red, green, and blue components of the background centered, a is the separation parameter (a is greater than 0 and equal or less than 1)

4. Distribute all the image pixels among the two clusters (background or cell) according to their distances from the two corresponding centroids (background or cell). The following distance criteria (i.e., nearest distance) was utilized:

If Distance (Pixel color, Background centroid)< Distance (Pixel color, Cell centroid)

THEN

Set Pixel Index = Background

Else

Set Pixel Index = cell

Where,

Distance (PixelColor , BackGroundCentroid) =

$$\sqrt{(R-Rb)^2 + (G-Gb)^2 + (B-Bb)^2} \quad \text{----------4}$$

Distance (PixelColor , CellsCentroid) =

$$\sqrt{(R-Rc)^2 + (G-Gc)^2 + (B-Bc)^2} \quad \text{----------5}$$

Where (R,G,B) are the color components of the tested pixel.

5. Determine the average color components for all pixels indexed as background using the following equations:

$$R'b = \frac{1}{nb}\sum_{i\in b} Ri \qquad G'b = \frac{1}{nb}\sum_{i\in b} Gi$$
$$B'b = \frac{1}{nb}\sum_{i\in b} Bi \quad \text{------6}$$

Where

b is the set of image pixels indexed as background. n_b is the size of the set (b).

6. Determine the average color components for all pixels indexed as target (or cell) using the following equations:

$$R'c = \frac{1}{nc}\sum_{i\in c} Ri \qquad G'c = \frac{1}{nc}\sum_{i\in c} Gi$$

Where

c is the set of pixels indexed as cell pixel n_c is the length (or size) of the set (c).

7. Compare the difference (D) between the vectors (R'_b,G'_b,B'_b) and (R_b, G_b,B_b), and between the vectors (R'_C,G'_C,B'_C) and (R_C,G_C,B_C) by using the following distance measure[3]:

$$D = \frac{1}{6}[(R'_b - R_b)+(G'_b - G_b)+(B'_b - B_b)+(R'_c - R_c)+(G'_c - G_c)+(B'_c - B_c)] \quad \text{------8}$$

8. Set,
R_b=R'$_b$; G_b=G'$_b$; B_b=B'$_b$
R_C=R'_C ; G_c = G; ; B_C=B'_C ' ----------**9**
9. If D is greater than a pre-defined threshold value (D_{min}) then repeat steps (4 to 8), otherwise exit.

4- TEST RESULTS

Firstly, the implemented algorithm is applied on a famous 'Lena' gray scale image (256x256). This test is depending on the number of clusters and the require calculation time. The performance parameter that used to evaluate our algorithm is based on fidelity measure

PSNR. Table (1) presents the test results performed on the gray image.

Table 1: Gray image test results

Time	*Number of*	*PSNR/d*
5	100	61
4	50	53
5	30	52
3	15	47
3	10	45
1	5	40
.5	2	32
.5	1	27

Figure 2 shows the output image based on PNSR when the number of clustering is equal 100 and PSNR=.61.8

Figure 2 The output Gray image NC=100

Figure 3 The output Gray image NC=15,PSNR=47.05

And table (2) presents the test results performed on the color image.

Table 2: Colour image test results

Time /second	*Number of cluster*	*PSNR/dB*
21	100	57
20	50	53
7	30	51
6	15	49
5	10	46
2	5	43
1	2	38
.2	1	33

Figure 4 shows the output color image based on PSNR when the number of clustering is equal 100 and PSNR=57.13

Figure 4 The output color image NC=100

Figure 5 The output color image NC=15,PSNR=49.7

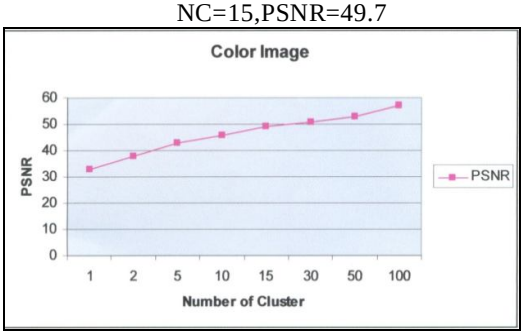

Figure 6 PSNR versus number of cluster (color image)

Figure 7 Computation versus NC (Gray image)

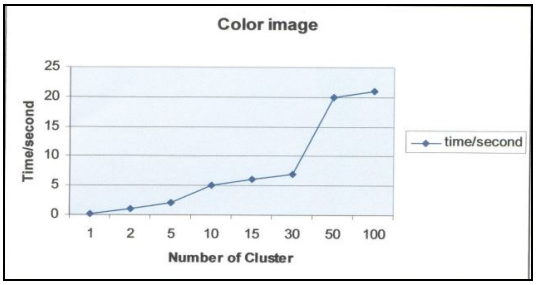

Figure 8 Computation versus NC (Color image)

5- CONCLUSIONS

In the previous chapters, the aspect for establishing K-mean algorithm has been discussed, starting from the k-

mean algorithm steps and implementing the algorithm by visual basic tools and testing the number of cluster on the image depending on the performance parameter PSNR computation time.

1. The main contribution work is that the well-known k-mean algorithm needs not necessarily be considered an impractically slow algorithm, even with many records. In this research empirical results for a new fast implementation on k-mean has been described and analyzed.
2. The implemented algorithm is applied on a famous 'Lena' gray scale image (256x256). This test is depending on the number of clusters and the require calculation time. The performance parameter that used to evaluate our algorithm is based on fidelity measure PSNR.

6- FUTURE SUGGESTIONS

Among the different suggestions that stimulated during the discussion of the test results, the following suggestions were made as material for future research work:

1. Using hybrid algorithm to improve the performance of clustering.
2. Using k-mean algorithm to classify blood cell.
3. Using k-mean algorithm to identify a person by matching an image pre stored in a DB with k-mean algorithm.

REFERENCES

[1] A. Gersho and R. Gray. Vector quantization and signal compression. Kluwer Academic Publisher;Dordrecht, Netherland, 1992.

[2] J.L. Bentlaey. Multidimensional Divide and Conquer. Communications of the ACM, 23(4):214-229, 1980.

[3] Andrew W.moore. Very fast EM-based mixture model clustering using multi resolution kd-trees. In Neural information Processing Systems Conference, 1998.

[4] Andrew W. moore and Mary Soon lee. Cached sufficient statistics for efficient machine learning with large database. Journal of Artificial Intelligence Research, 8:67-91,1998.

[6] J. Mac Queen, (1967) Some methods for classification and analysis of multivariate observations," Proc. 5th Berkeley Symposium on Mathematical Statistics and Probability.

[7] J.-M. Jolion,et al (1991) Robust clustering with applications in computer vision,Trans. Pattern Analysis and Machine Intelligence,

[8] Robert Harrison et al (2005) improved K-Means clustering algorithm .

[9] Carlos Ordonez (2006) Integrating K-Means clustering with a relational DBMS .

[10]Zhongyang Xiong ,et al (2006) The study of Parallel K-Means Algorithm.

[11]Zejin Ding , et al (2007) A new Improved K-mean Algorithm with Penalized Term, K-means Algorithm is popular method in cluster analysis.

[12]Chun-Wei Tsai (2007) A Time Efficient Pattern Reduction Algorithm for K-means Based Clustering.

AUTHOR

Haval M. SIDQI received the B.S.degree in Math , Higher Diploma and M.S. degrees in Computer Science from Mosul and Sulaimani University in 1992, 2008 and 2011. During 1997 to present, he works as assist lecturer in Computer Science Institute led by Sulaimani Polytechnic University, Kurdistan Region of Iraq.