Journal of Big Data

RESEARCH

Open Access

Binary image description using frequent itemsets

Khalid Aznag*, Toufik Datsi, Ahmed El oirrak and Essaid El bachari

*Correspondence:
khalidaznag@gmail.com;
khalid.aznag@ced.uca.ac.ma
Faculty of Sciences
Semlalia, Computer Science
Department, Cadi Ayyad
University, Marrakesh,
Morocco

Abstract

In this paper, a novel method for binary image comparison is presented. We suppose that the image is a set of transactions and items. The proposed method applies along rows and columns of an image; this image is represented by all frequent itemset. Firstly, the rows of the image are considered as transactions and the columns of the image are considered as items. Secondly, we considered rows as items and columns as transactions. Besides, we also apply our technique to color image; firstly we segment the image and each segmented region is considered as a binary image. The proposed method is tested on the MPEG7 database and compared with the moment's method to show its efficiency.

Keywords: Binary image, Apriori algorithm, Frequent itemsets, Moments, Color image

Introduction

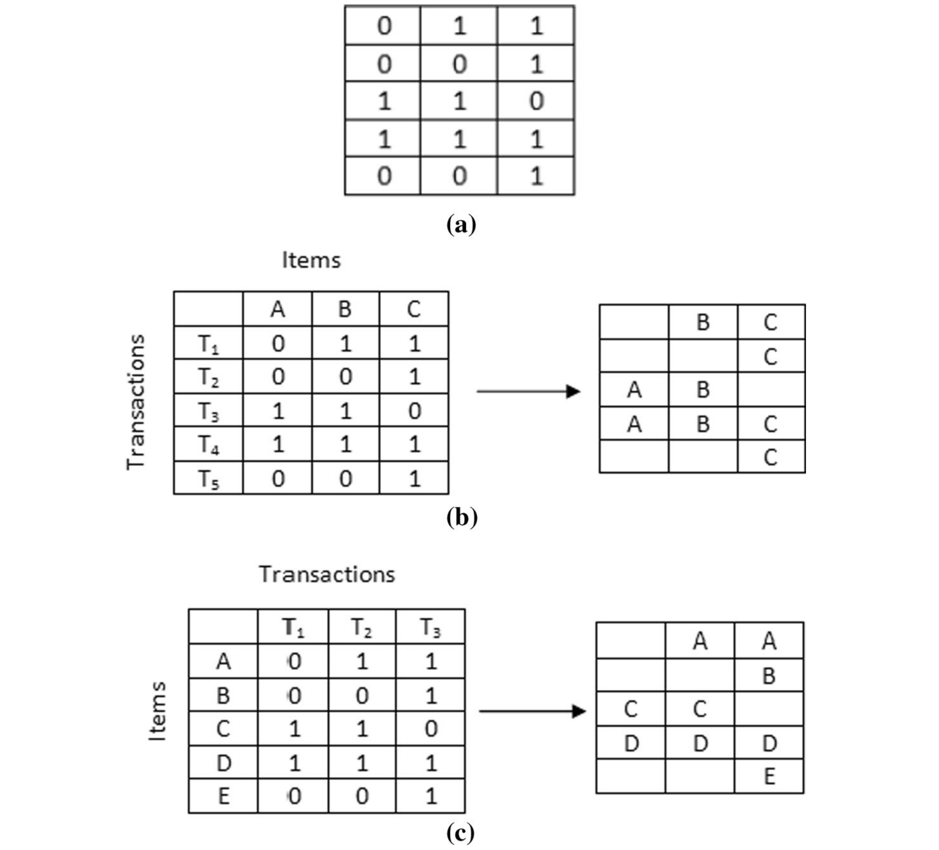
In recent years, due to information technology such as a computer, Internet, software and the development of sensor technology, make present life appeared "everything can be a digital", big data is becoming a hot research topic in the field of all kinds of industrial [1]. Big Data has been created in the fields of business applications such as marketing, social networking, science, and smartphone applications. Nowadays, better computing power is more necessary in the area of data mining [2].

Image comparison is largely used in different fields: shape matching [3], registration [4]. However, comparing binary images that represent the same content is not easy because an image can undergo transformations like rotation, translation, resolution change. In data mining, there are many frequent patterns (FP) mining algorithms e.g FP-growth [5], CFP-growth [6], MISFP-growth (Multiple Item Support Frequent Patterns) [7], Apriori [8]. A comparison of FP mining algorithms is given in [9]. Apriori algorithm is the best-known basic algorithm proposed by R. Agrawal and R. Srikant in 1994. The Apriori algorithm is one of the most important algorithms which is used to retrieve frequent itemsets. For large databases, there exists the algorithms FTWeightedHashT [10] and MISFP-growth [9]. It essentially requires two important things: minimum support and minimum confidence [11, 12]. Most of the commonly used association rule discovery algorithm that utilise the frequent itemset strategy, and which is exemplified by the Apriori algorithm [13]. The Apriori algorithm searches for a large set of items in the initial database analysis. The result of this analysis is then used as a base for discovering other datasets during other passes. The rules having a support level above the minimum threshold are called small itemsets [14]. On the other hand, the algorithm is based on the big itemset property which states: Each subset of a large itemset is large and if an itemset is not wide and then none of its supersets are large [8].

In this paper, we propose a novel method to compare the binary image based Apriori algorithm. We transform our image into a set of items and transactions (see Fig. 1). In which, we suppose that image is a set of transactions and items. Our method is applied along rows and columns of an image. Thus, the Apriori algorithm is applied along rows and columns of an image:

1. Apriori algorithm along row: The rows of the image will be considered as transactions. The columns of the image will be considered as items.
2. Apriori algorithm along column: The rows of the image will be considered as items. The columns of the image will be considered as transactions.

The rest of the paper is organized as follows. "Related work" section discusses related work. In the "Methodology" section, we explain the proposed approach. "Results and



0	1	1
0	0	1
1	1	0
1	1	1
0	0	1

(a)

	A	B	C
T_1	0	1	1
T_2	0	0	1
T_3	1	1	0
T_4	1	1	1
T_5	0	0	1

	B	C
		C
A	B	
A	B	C
		C

(b)

	T_1	T_2	T_3
A	0	1	1
B	0	0	1
C	1	1	0
D	1	1	1
E	0	0	1

	A	A
		B
C	C	
D	D	D
		E

(c)

Fig. 1 An example of Apriori execution for binary image. **a** Binary image, **b** Apriori algorithm along row, **c** Apriori algorithm along column

discussion" section shows experimental results and discussion. Finally, the conclusion is drawn, and the future work described, followed by references.

Related works

Different types of techniques are presently used in the domains of big data analytics and image processing [15–19]. The integration and interaction of the two wide fields need more vision to better exploit and explore the benefits of the two techniques. Vahini et al. [20] proposed a state of the art in image processing and big data analytics, this study aims to focus of the recent research progresses in the two broad fields of Big Data analytics and image processing to show out the importance of their interaction and integration. A novel workflow based on big data analytic for biomedical image classification is proposed in [21]. It is composed of two structures; the first is based on the spark and the second on the Hadoop framework. In [22] authors present an efficient fast-response content-based image retrieval (CBIR) Hadoop-based framework that consists of a set of modules working in two layers. Creating a content-based image retrieval system pointing at big data to operate efficiently with real-time data is counted as a critical competition.

There are a lot of methods for image comparison. Current papers aim at reviewing the available tools and techniques for image processing and comparison [23–25]. These tools in relation to big data can create a user-friendly solution in solving the industrial problem [26]. Several approaches to compare binary images are proposed in the literature [27]. Some are based on the comparison of image descriptors and others on the direct comparison of image. In direct approaches, the comparison is done between the images directly from the pixels. Direct comparisons between binary images are difficult to be implemented. This is because unlike the color image, the pixels in the binary image have little information: the amplitude of the variation is 0 or 1. Direct comparisons can be used to assess noise, the segmentation fault [28] and to compare images of faces contours to evaluate the dissimilarity [29–31]. Most are based on an aggregate measure and provide a measure of similarity in the scalar form (a positive real), they are also based on the local study (as part of grayscale images) [32] and provide a similarity index in two dimensions. Indirect methods are based on image processing to represent the image in a new space. The geometrical moments, with two dimensions, which provide descriptors for recognition have many applications in image processing. In robotics, they are used for motion search calculation and guidance for registration; they are also used in image processing for solving some problems such as matching scene [33] and character recognition [34]. The methods based on frontier methods such as Fourier descriptors, space decomposition, contour curvature-scale and decomposition of the principal component outline have many applications especially for the treatment of forms [35]. The methods based on the border are a robust descriptor for a shape, but they are not available for a complex image, which contains several details. Transformed the entire image: global transforms (Fourier and wavelet transform) are unsuitable for the processing of digital images as they are composed of discontinuities. Filtering non-linear methods are adapted for binary images [36]. However, in the case of linear filtering, the comparison of the transformed images can be performed after selecting the most significant coefficients (for the comparison of their standards), which is much less discriminating factor in the case of binary coefficients.

We have presented in this section direct and indirect methods of image comparison. Indirect methods are inadequately appropriate for comparing binary images other than shapes. However, direct methods are actually interesting for measuring a distance between sets of points and not only pixel to pixel. But, they do not generally allow access to the local dissimilarity information or so without adapting to the type of dissimilarity to assess. In this work, we apply the Apriori algorithm, which is an indirect method to find frequent itemsets.

Methodology

In this section, we first describe briefly Hu's moments method [37]. Then, we present our proposed method.

Moments method

Moments invariants are introduced by Hu. In [37], Hu derived six absolute orthogonal invariants and one skew orthogonal invariant based upon algebraic invariants, which are not only independent of position, size and orientation but also independent of parallel projection. The moment invariants have been proved to be the adequate measures for tracing image patterns regarding the translation of the images, scaling and rotation under the assumption of images with continuous functions and noise-free. Moment invariants have been extensively applied to image registration [38] and image pattern recognition [38–41]. The well-known moments include geometric moments [37], zernik moments [42, 43], rotational moments [44] and complex moments [39].

In his work, Hu proved that each irradiance function f(x, y) (or image) has a one-to-one correspondence to a single set of moments and vice versa. The $(p+q)^{th}$ order geometric moment of f(x,y) can be expressed in the following equation:

$$m_{pq}(x,y) = \int_{-\infty}^{+\infty} \int_{-\infty}^{+\infty} x^p x^q dxdy \tag{1}$$

So, the 7 invariants moments Hu are shown in Eq. (4).

With

$$\mu_{ij} = \int_{-\infty}^{+\infty} \int_{-\infty}^{+\infty} (x-\tilde{x})(y-\tilde{y})I(x,y)dxdy \tag{2}$$

i, j = 0, 1, 2,...

and

$$\theta_{ij} = \frac{\mu_{ij}}{\mu_{00}^{\left(1+\frac{1+j}{2}\right)}} \tag{3}$$

where i+j ≥2, then

$$\begin{cases} \phi_1 = (\theta_{20} + \theta_{02}), \\ \phi_2 = (\theta_{20} - \theta_{02}) + 4\theta_{11}^2, \\ \phi_3 = (\theta_{30} - 3\theta_{12})^2 + (3\theta_{21} - \theta_{03})^2, \\ \phi_4 = (\theta_{30} + \theta_{12})^2 + (\theta_{21} + \theta_{03})^2, \\ \phi_5 = (\theta_{30} - 3\theta_{12})(\theta_{30} + \theta_{12})\left[(\theta_{30} + \theta_{12})^2 - 3(\theta_{21} + \theta_{03})^2\right], \\ \qquad +(3\theta_{12} - \theta_{03})(\theta_{21} + \theta_{03})\left[3(\theta_{30} + \theta_{12})^2 - (\theta_{21} + \theta_{03})^2\right] \\ \phi_6 = (\theta_{20} - \theta_{02})\left[(\theta_{30} + \theta_{21})^2 - (\theta_{21} + \theta_{03})^2\right] + 4\theta_{11} \\ \qquad (\theta_{30} + \theta_{12})^2(\theta_{21} + \theta_{03}), \\ \phi_7 = (3\theta_{21} - \theta_{03})(\theta_{30} + \theta_{12})\left[(\theta_{30} + \theta_{12})^2 - 3(\theta_{21} + \theta_{03})^2\right] \\ \qquad -(\theta_{30} + 3\theta_{12})(\theta_{21} + \theta_{03})\left[3(\theta_{30} + \theta_{12})^2 - (\theta_{21} + \theta_{03})^2\right] \end{cases} \quad (4)$$

Proposed technique

In this section, we present our approach. In Fig. 2, an illustration to explain the proposed approach is presented. As shown in this figure, example image the size 8 × 10. If we take min_sup = 2, then the triplet ((2,3), 2,5) contains three frequent items of length 1, two frequent items with length 2 and one frequent item of large three (see Fig. 2b, c).

In general, we can rewrite a frequent pattern using the triplet (Position First Pixel, Start row, End row, Width) or (Position First Pixel, Start column, End column, Width).

Where

Position_First_Pixel: indicates the position of the first pixel inside a frequent pattern.

- Star_row: denotes the number of starting row,
- End_row: denotes the number of ending row,
- Width: indicates the large of frequent pattern.

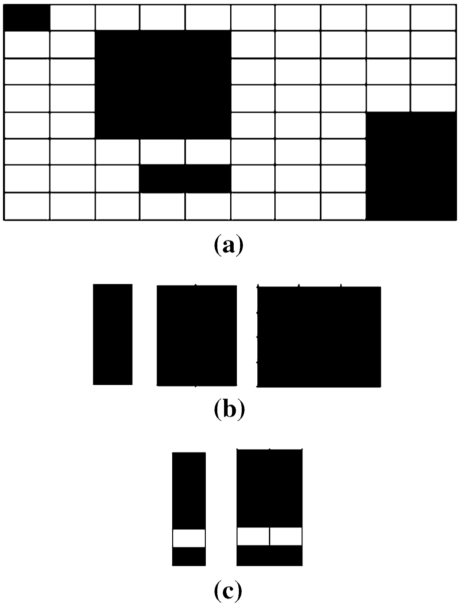


Fig. 2 Example illustrating the proposed approach for binary image. **a** Binary image, **b** three frequent items with length 1, **c** two frequent items with length 2

The principle of the proposed approach is that the image is a set of transactions and items. We apply the Apriori algorithm along rows and columns of an image; this image is represented by all frequent items. Algorithm 1 presents the algorithm Apriori for a binary image. In the case of the row, the rows of the image are considered as transactions and the columns of the image are considered as items. In this case, the column, the rows of the image are considered as items and the columns of the image are considered as transactions.

Algorithm 1 Apriori algorithm for binary image.

```
im = ReadImage(I); // I: binary image
w: width; h: height; m: minsup;
data = new int [h][w];
elm = new String [h];
for(int i=0 ;i≺h;i++)
  for(int j=0;j≺w;j++)
    if((im.getRGB(j,i) & 0xff) ≤ 128)
    ng=1;
    data [i][j] = ng;
      elm [i] = line;
    endFor
endFor
for(int i=0; i≺w; i++)
emp=0;
for(int j=0;j≺h;j++)
    if(data[j][i]==1)
    emp++;
endFor
buffer.put(“ ” + (i+1), emp);
if(emp ≥m)
Filtredbuffer.put(“ “+ (i + 1), emp);
displayTM(buffer, “Items”);
displayTM(buffer, “Transactions”);
```

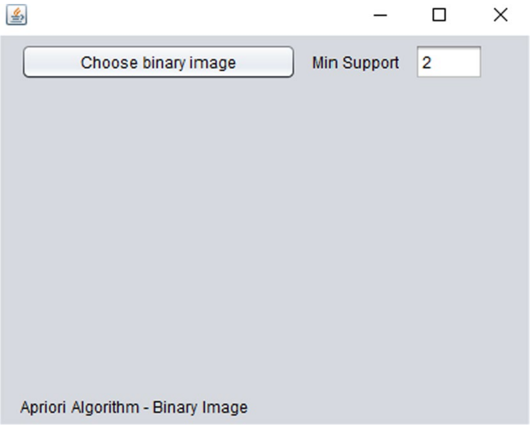
Apriori algorithm for color image

The main idea is that we segment the image, and each segmented region is considered a binary image. The segmentation [45] is the first step in analyzing an image. It is a separation of the image elements into homogenous regions having the same property. These regions can be characterized by their borders, in this case, they are characterized by the pixels that compose those borders.

In general methods of image segmentation can be one of the following four cases:

1. Segmentation based on regions,
2. Segmentation based on contours,
3. Segmentation based on the classification of pixels as a function of intensity,
4. Segmentation based on cooperation of these three methods.

For simplicity in our study, we apply the first case which is based on dividing the image into small regions [46].



Min Support 2

Fig. 3 Java Interface implementing Apriori algorithm

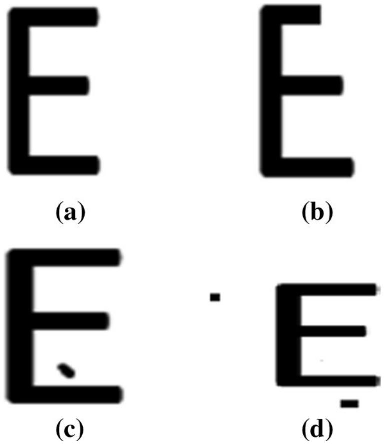


Fig. 4 Images for experiments

Results and discussions

We conducted an experiment to measure the performance of the proposed method. In the experiment, we present the tools used to evaluate our approach. Then we compare the proposed approach with Hu moments approach and we test its robustness to noise. Finally, we apply our approach to color image, in addition to an application on the MPEG7 database is presented.

We developed a java application that is used to implement our approach and a simple interface that makes it possible for the user to import a binary image for the Apriori algorithm (see Fig. 3). The user imports the binary image, an interface allows him to view the image and also in the same interface to find the Apriori algorithm result. The editor was used in this study is the Eclipse IDE. It is an integrated development environment and open source. It is also characterized by open plugin-based architecture. This is one of the most used editors by the developer's Java IDE. Our application runs under the Windows platform, using 2.20 GHz and 1024 MB memory.

In this study, we extract seven moment invariants from our test image and noisy image. In Fig. 4, image (a) is the original image, image (b) is a noisy image obtained

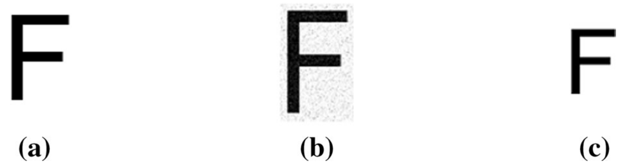


Fig. 5 An example of noising and scaling operations on the letter "F." Left: the original image. Middle: noised image. Right: image scaling

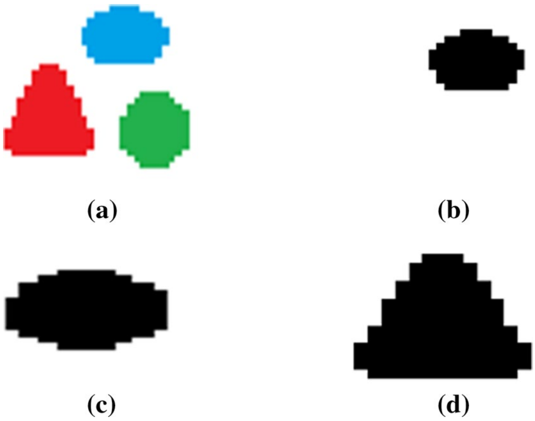


Fig. 6 **a** Original image, **b** blue image, **c** green image, **d** red image

using randn() function to the image (a), and the two images (c) and (d) are two noisy image obtained by adding noise manually.

Concerning our approach, our technique is presented by:

Apriori algorithm along the row. In this case, the rows of the image are considered as transactions and the columns of the image are considered as items. We take minsup row = 50%, minsup is arbitrarily set.

Apriori algorithm along the column. In this case, it is the same procedure applied along row except Apriori algorithm is applied to columns. The rows of the image are considered as items and the columns of the image are considered as transactions.

Figure 5 gives a binary image of the English letter "F." Figure 5a denotes the original image, Fig. 5b denotes the noised image, and Fig. 5c denotes the scaled image. Different kinds of imaging systems might give us different noises. In this study, we used Gaussian noise [47].

In the case of color images, we segment the image; the color image is presented in RGB space, so we apply our technique using the following steps:

1. Segmentation of the image into regions using k_means,
2. Each segmented region is considered as binary image to which we apply Apriori algorithm,
3. Extract the frequent items for each image.

In this study, we test our technique in image color, as shown in Fig. 6.

Table 1 The value of Hu moment invariants

Image	$\phi1$	$\phi2$	$\phi3$	$\phi4$	$\phi5$	$\phi6$	$\phi7$
(a)	0.1747	0	0	0	0	0	0
(b)	0.1740	0	0	0	0	0	0
(c)	0.1754	0	0	0	0	0	0
(d)	0.1752	0	0	0	0	0	0

Table 2 Results found by our approach along rows

(a)	(8 9 10 11 12 13 14)
(b)	(8 9 10 11 12 13 14)
(c)	(8 9 10 11 12 13 14)
(d)	(8 9 10 11 12 13 14)

Table 3 Results found by our approach along columns

(a)	(4 5 6 7 8 9 10 11 12 13 14 15 16)
(b)	(4 5 6 7 8 9 10 11 12 13 14 15 16)
(c)	(4 5 6 7 8 9 10 11 12 13 14 15 16)
(d)	(4 5 6 7 8 9 10 11 12 13 14 15 16)

Table 4 Results found by our approach along rows and columns for binary images with scaling and Gaussian noise

	(a)	(b)	(c)
Rows	(40 41 42 43 44 45)	(40 41 42 43 44 45)	(40 41 42 43 44 45)
Columns	(43 43 43 43 43 43 43)	(43 43 43 43 43 43 43)	(43 43 43 43 43 43 43)

Table 1 shows the values of seven moment invariants for the images (a–d) of Fig. 4. We remark that only the value of the moment is different for the original image and the corresponding noisy image.

Tables 2 and 3 show the obtained results when we use our method. We note that the results are the same for the three images, so there is no influence of the noise.

Table 2 shows only the maximum length frequent items but in our experience, the program displays all levels of frequent items. Rows 8, 9, 10, 11, 12, 13 and 14 constitute a frequent itemset of length 7 for each image. In addition, Table 4 gives the experimental results for the binary image with scaling and with added noise. The scale is solved by normalizing image to a fixed size.

Notice that, Hu moment invariants depend on noise. Despite the presence of noise two images have the same frequent columns or rows concerning the proposed approach.

From these results, it can be observed that the proposed approach is better than Hu moment invariants.

Images in Fig. 7 are used to show that our approach can be used to compare binary images of different levels. The results found with the Apriori algorithm along rows for

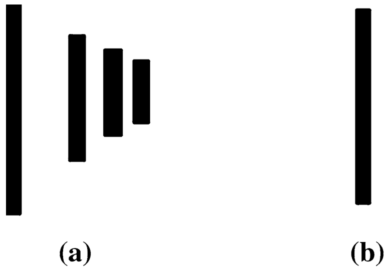


Fig. 7 Two images line used in experiments

images given in Fig. 7 are shown in Table 5. Image (a) depicts four frequent items in different length, from the longest to the shortest item. Image (b) has one frequent item having a maximum length. If we take only items of maximum length, images (a) and (b) will be similar.

The results found with the Apriori algorithm along rows for binary images Fig. 6b–d are shown in Table 6.

Application on MPEG7 database

In order to test our approach; we apply the Apriori algorithm on the MPEG7 database [48]. In this database there are 70 classes of shapes, each one has 20 members as shown in Fig. 9. We extract the frequent items from the image by counting the frequency of non black pixel. An MPEG7 image of size 256 × 256 is presented in Fig. 8.

We have found that is easy to extract frequent item from image. In Figs. 10 and 11 the authors show the frequency of non black pixel respectively in each row and each column in the image. As can be seen in Figs. 10 and 11 we can notice that those graphs present mostly the shape of 'Apple' image. Frequency of non black pixel for each row and frequency of non black pixel for each column are given in Tables 7 and 8 respectively. As row '141' contains '222' white pixel it can be considered as frequent if min-sup is fixed to 10.

Conclusion and future work

In this work, we have proposed a technique to compare binary images. We considered an image as a set of transactions and items. The proposed technique depends on the Apriori algorithm, we apply Apriori along rows or columns of an image. In addition, we

Table 5 Results found by our approach along rows for binary images

(a)	(7 13 22 27)
(b)	(7)

Table 6 Results found by our approach along rows for segmented binary image

(b)	(14 15 16 17 18 19 20 21 22 23 24 25)
(c)	(17 18 19 20 21 22 23 24 25 26)
(d)	(5 6 7 8 9 10 11 12 13 14 15 16 17)

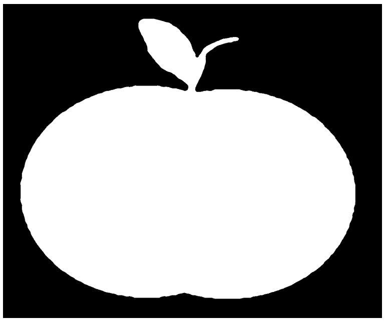


Fig. 8 'Apple' MPEG 7 image

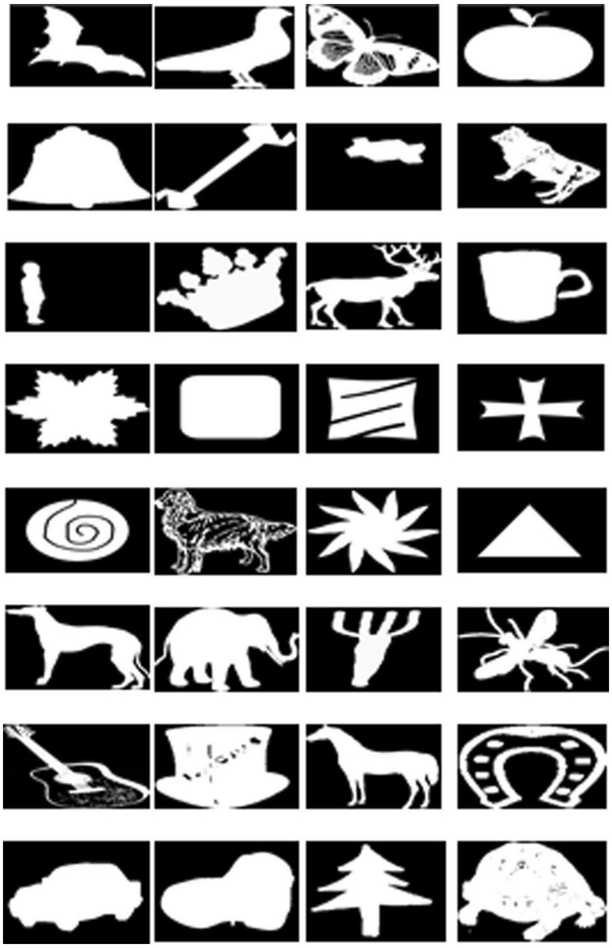


Fig. 9 Example of MPEG7 image database

apply our technique to color image. The result shows that our proposed approach based on the Apriori algorithm is agreeable particularly this technique is robust for noise.

In the future, we will integrate this technique to solve the rotation problem. Then, we will integrate methods, such as FP-growth, FTWeightedHashT Apriori and MISFP-growth to solve the problem of large data (Large Binary Image).

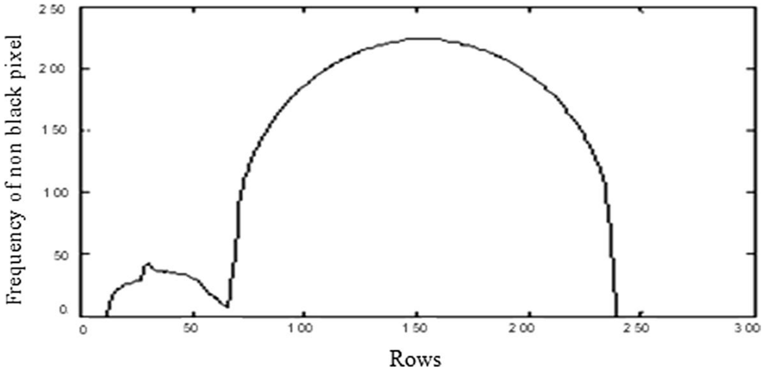


Fig. 10 Frequency of non black pixel in each row of the image

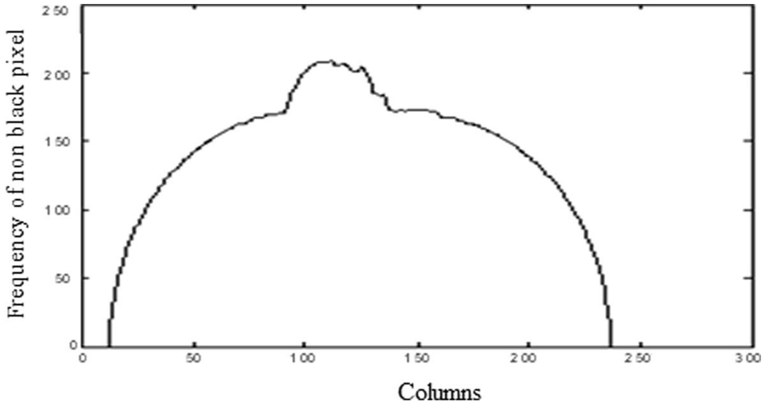


Fig. 11 Frequency of non black pixel in each column of the image

Table 7 Frequency of non black pixel for row

Row number	Frequency of non black pixel
141	222
161	224
181	215
201	195
221	156
241	0

Table 8 Frequency of non black pixel for column

Column number	Frequency of non black pixel
141	172
161	167
181	159
201	137
221	97
241	0

Abbreviations

I: Image; RGB: Red, green and blue; IDE: Integrated development environment; MPEG7: Multimedia Content Description Interface; MB: Megabyte; minsup: Minimum support threshold; FP: Frequent pattern.

Acknowledgements

Not applicable.

Authors' contributions

KA developed the methodology and designed the experiment. KA and TD also analysed the result and wrote the manuscript under the supervision of AE and EE as academic supervisors. All authors read and approved the final manuscript.

Funding

Not applicable.

Availability of data and materials

The datasets used during the current study are available in https://drive.google.com/file/d/1WtkCufTBzYP-vjNpkl6TDUTTQJ0H-IFJ/view?usp=sharing.

Competing interests

The authors declare that they have no competing interests.

Received: 21 September 2019 Accepted: 1 May 2020
Published online: 12 May 2020

References

1. Fan X, Li Z, Zhou L. Literature review on Big Data and its application fields. In: Journal of physics: conference series. vol. 1176. IOP Publishing; 2019. p. 042043.
2. Ansari A, Parab A, Kadam S. Apriori-a Big Data analysis—a review. Int J Recent Innov Trends Comput Commun. 2001;2(11):3517–20.
3. Veltkamp RC, Hagedoorn M. Shape similarity measures, properties and constructions. In: International conference on advances in visual information systems. Springer; 2000. p. 467–76.
4. Antani S, Kasturi R, Jain R. A survey on the use of pattern recognition methods for abstraction, indexing and retrieval of images and video. Pattern Recogn. 2002;35(4):945–65.
5. Han J, Pei J, Yin Y. Mining frequent patterns without candidate generation. In: ACM sigmod record. vol. 29. ACM; 2000. p. 1–12.
6. Hu YH, Chen YL. Mining association rules with multiple minimum supports: a new mining algorithm and a support tuning mechanism. Decis Support Syst. 2006;42(1):1–24.
7. Darrab S, ErgenÇ B. Frequent pattern mining under multiple support thresholds. Methods. 2016;10:11.
8. Agrawal R, Srikant R, et al. Fast algorithms for mining association rules. In: Proceedings of 20th international conference for very large data bases, VLDB. vol. 1215; 1994. p. 487–99.
9. Wang CS, Chang JY. MISFP-growth: hadoop-based frequent pattern mining with multiple item support. Appl Sci. 2019;9(10):2075.
10. Ammar SM, Ba-Alwi FM. Improved FTWeightedHashT apriori algorithm for Big Data using Hadoop-MapReduce model. J Adv Math Comput Sci. 2018;. https://doi.org/10.9734/JAMCS/2018/39635.
11. Bhandari A, Gupta A, Das D. Improvised apriori algorithm using frequent pattern tree for real time applications in data mining. Procedia Comput Sci. 2015;46:644–51.
12. Khalili A, Sami A. SysDetect: a systematic approach to critical state determination for Industrial Intrusion Detection Systems using Apriori algorithm. J Process Control. 2015;32:154–60.
13. Agrawal R, Imieliński T, Swami A. Mining association rules between sets of items in large databases. In: ACM sigmod record. vol. 22. ACM; 1993. p. 207–16.
14. Chen MS, Han J, Yu PS. Data mining: an overview from a database perspective. IEEE Trans Knowl data Eng. 1996;8(6):866–83.
15. Elakkia K, Narendran P. Survey of medical image segmentation using removal of gaussian noise in medical image. Int J Eng Sci. 2016;6(6).
16. Yao GL. A survey on pre-processing in image matting. J Comput Sci Technol. 2017;32(1):122–38.
17. Arcondara J, Himmi K, Guan P, Zhou W. Value oriented big data strategy: analysis & case study. In: Proceedings of the 50th Hawaii international conference on system sciences; 2017.
18. Paulchamy B, Archana K, Bruno A, Divya I, Krishnan M. A novel approach towards road safety based on investigational survey using image processing and user interface system. Int Educ Res J. 2017;3(3):105–8.
19. Hu H, Wen Y, Chua TS, Li X. Toward scalable systems for big data analytics: a technology tutorial. IEEE Access. 2014;2:652–87.
20. Ezhilraman SV, Srinivasan S. State of the art in image processing & big data analytics: issues and challenges. Int J Eng Technol. 2018;7(2.33):195–9.
21. Kouanou AT, Tchiotsop D, Kengne R, Zephirin DT, Armele NMA, Tchinda R. An optimal big data workflow for biomedical image analysis. Inform Med Unlocked. 2018;11:68–74.
22. Sakr NA, ELdesouky AI, Arafat H. An efficient fast-response content-based image retrieval framework for big data. Comput Electr Eng. 2016;54:522–38.
23. Shrivas A, Gupta P. Analysis and comparison of data mining tools and techniques for classification of banknote authentication. Int J Adv Res Comput Sci. 2017. https://doi.org/10.26483/ijarcs.v8i5.4051.

24. Ahsan R, Ebrahimi M. Image processing techniques represent innovative tools for comparative analysis of proteins. Comput Biol Med. 2019;117:103584.
25. Carminati MC, Boniotti C, Fusini L, Andreini D, Pontone G, Pepi M, et al. Comparison of image processing techniques for nonviable tissue quantification in late gadolinium enhancement cardiac magnetic resonance images. J Thorac Imaging. 2016;31(3):168–76.
26. Katukam R, Sindhoora P. Image comparison methods & tools: a review. Image. 2015;35:42.
27. Baudrier E, Millon G, Nicolier F, Ruan S. A fast binary-image comparison method with local-dissimilarity quantification. In: 18th international conference on pattern recognition (ICPR'06). vol. 3. IEEE; 2006. p. 216–9.
28. Xianguo L, Lifang S, Zixu M, Can Z, Hangqi J. Laser-based on-line machine vision detection for longitudinal rip of conveyor belt. Optik. 2018;168:360–9.
29. Dang CN, Do TH. A modification of line Hausdorff distance for face recognition to reduce computational cost. Sci Technol Dev J. 2017;20(K3):152–8.
30. Ribera J, Güera D, Chen Y, Delp E. Weighted hausdorff distance: a loss function for object localization. arXiv preprint arXiv:180607564. 2018.
31. Zhanga Z, Lia J, Lic X, Lina Y, Zhanga S, Wanga C. A fast method for measuring the similarity between 3d model and 3d point cloud. International Archives of the Photogrammetry, Remote Sensing and Spatial Information Sciences; 2016. p. 1.
32. Wang Z, Bovik AC, Sheikh HR. Structural similarity based image quality assessment. In: Digital Video image quality and perceptual coding. CRC Press; 2017. p. 225–42.
33. Fu Y, Sun K. Adaptability simulation and analysis of scene matching algorithm. In: 2nd IEEE advanced information management, communicates, electronic and automation control conference (IMCEC). IEEE. 2018;2018:2562–6.
34. Ayyaz MN, Javed I, Mahmood W. Handwritten character recognition using multiclass SVM classification with hybrid feature extraction. Pak J Eng Appl Sci. 2016;10:57–67.
35. Rohé MM, Datar M, Heimann T, Sermesant M. Pennec X. SVF-Net: learning deformable image registration using shape matching. In: International conference on medical image computing and computer-assisted intervention, vol. 10433. Springer; 2017. p. 266–74.
36. Aggarwal V, Patterh MS. ECG signal compression using morphological haar wavelet transform. Int J Eng Sci. 2016;17:263–5.
37. Hu MK. Visual pattern recognition by moment invariants. IRE Trans Inf Theory. 1962;8(2):179–87.
38. Flusser J, Suk T. A moment-based approach to registration of images with affine geometric distortion. IEEE Trans Geosci Remote Sens. 1994;32(2):382–7.
39. Vargas-Vargas H, Camacho-Bello C. Complex moments for the analysis of metal-mechanical parts. In: Applications of digital image processing XLI. vol. 10752. International Society for Optics and Photonics; 2018. p. 107522Y.
40. Flusser J, Zitová B, Suk T. Invariant-based registration of rotated and blurred images. In: IEEE 1999 international geoscience and remote sensing symposium. IGARSS'99 (Cat. No. 99CH36293). vol. 2. IEEE; 1999. p. 1262–4.
41. Chen Q, Petriu E, Yang X. A comparative study of Fourier descriptors and Hu's seven moment invariants for image recognition. In: Canadian conference on electrical and computer engineering 2004 (IEEE Cat. No. 04CH37513). vol. 1. IEEE; 2004. p. 103–6.
42. Qader HA, Ramli AR, Al-Haddad SAR. Fingerprint recognition using zernike moments. Int Arab J Inf Technol. 2007;4(4):372–6.
43. Schlemmer M, Heringer M, Morr F, Hotz I, Hering-Bertram M, Garth C, et al. Moment invariants for the analysis of 2D flow fields. IEEE Trans Vis Comput Graph. 2007;13(6):1743–50.
44. Hosny KM, Darwish MM. Feature extraction of color images using quaternion moments. In: Recent advances in computer vision. Springer; 2019. p. 141–67.
45. Sivakumar P, Meenakshi S. A review on image segmentation techniques. Int J Adv Res Comput Eng Technol (IJARCET). 2016;5(3):641–7.
46. Chitade AZ, Katiyar S. Colour based image segmentation using k-means clustering. Int J Eng Sci Technol. 2010;2(10):5319–25.
47. Helou ME, Susstrunk S. Blind Universal Bayesian image denoising with gaussian noise level learning. arXiv preprint arXiv:190703029. 2019.
48. Bai X, Yang X, Latecki LJ, Liu W, Tu Z. Learning context-sensitive shape similarity by graph transduction. IEEE Trans Pattern Anal Mach Intell. 2009;32(5):861–74.

Publisher's Note

Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.