Available Online at www.ijcsmc.com

International Journal of Computer Science and Mobile Computing

A Monthly Journal of Computer Science and Information Technology

ISSN 2320–088X
IMPACT FACTOR: 5.258

IJCSMC, Vol. 5, Issue. 3, March 2016, pg.124 – 139

Content-Based Cartoon Images Retrieval

Mustafa M. Abd Zaid[1], Loay E. George[2], Ghadah Al-Khafaji[3]
[1'2'3] College of Science, University of Baghdad, Iraq
[1] mustafamajeed2014@gmail.com, [2] loayedwar57@yahoo.com, [3] hgkta2012@yahoo.com

Abstract- Cartoon images play essential roles in our everyday lives especially in entertainment, education, and advertisement, that become an increasingly intensive research in the field of multimedia and computer graphics[1-2]. Cartoon image retrieval is bedrock of producer's new cartoon videos, particularly when provides initial planning to be elevated in so as to identify their intention of artistic [3].
Today, a number of researchers have exploited the concepts related to content based images retrieval (CBIR) to search for cartoon images containing particular object(s) of interest [4]
This paper aims to design content-based retrieval system for cartoon images. The proposed image retrieval system consists of four basic stages "(image quantization, object extraction, feature extraction and similarity matching). Given a query image, the system extracts features for the query image which is a percentage of each index value (dominant color), cluster values for three bands of quantization image (RGB), and the weight of each index values (weight of each dominant color) depends on its location in the image, then compares it with the images features in database system in similarity matching stage and then the top 10 images are retrieved as the query results.
Keywords: Objects Extraction, Cartoon images, Cartoon Image Retrieval.

I. Literature Review

Since the early 1990s when Kato use the term CBIR, CBIR has become a very important search area [Rui99]. Several researches have been published in the field of CBIR systems, for instance:
Haseyama and Matsumura, (2003), have introduced a cartoon image retrieval system, where the research introduce fractional features, referred to as regions and aspects in [5], to help compute the similarity of cartoon images. The database for cartoon images that are most like to the query image; based on the partial features are fully returned as a results. The method may only be used to return image for face as a cartoon character because the partial features are not appropriate for cartoon characters gesture recognition. Experimental results show the effectiveness and accurate performance of this method.

© 2016, IJCSMC All Rights Reserved

Krishnan, et al (2007) [6], have introduced a method to retrieve images depends on predominant colors in the front of image, where the front of image gives only semantics compared with the background of the image. The proposed scheme is showed higher rate of recall and precision compared to other scheme that depend on dominant color for image retrieval.

Yang et al, (2008), [7], have proposed an efficient scheme for extracting dominant color that significantly improves the efficiency of computation for dominant color extraction, also researchers introduced new measure of similarity of image that simulates human color perception. Proposed method depended on the biggest (largest percentage) DCs in the image in the term of image retrieval process. To improve their results and reduce the dependencies on the largest (largest percentage) domains colors, researchers proposed mutual colors and appended to their similarity measure. Experiment results showed that the performance enhancement not only regarding to saving features extraction cost, but also cognitively similar image retrieval, research presented quantization method for extracting dominant color, called the linear block algorithm (LBA).

Yang et al (2009) [3]**,** have proposed a gesture recognition technique for cartoon character images with two implementation, named cartoon clip installation and content-based cartoon image retrieval, the proposed technique depended on Motion Direction Features (MDF) and Edge Features (EF) for cartoon images. The features are grouped into two different kinds, named inter-features and intra-features. Experiment results showed retrieval of high precision and the cartoon clip synthesis can be made worthily.

Ahmed, et al (2013), [8] In this research, two of the salient object detection algorithms are used, one for natural images and another simple one for cartoon images because the latter type of images is characterized by the object of cartoons are surrounded by bold dark contours. This salient object detection algorithm assumed that the object mostly located in the image's center, to give weight to each dominant color according its spatial location whether it is belong to the salient object or background (border). Ahmed introduced an adapted semantic dominant color descriptor that can be used in object-based image retrieval. The mechanism of the proposed descriptor is based on assigning weight to each DC (Dominant Color) in the image in accordance with its belonging to the object or to the background. The background colors, which are in contact with the image borders and out of salient object area, will receive a lower weight whereas the object colors, which are located at the salient object area and don't touch the border, will receive a higher weight. Such a method helps alleviate the background effect. Additionally, they also introduced a new modification for the purpose of measure the similarity. This modification represents the mutual color ratio. Experimental results demonstrate that the proposed descriptor with the similarity measure modification performs better than the existing descriptor in content-based image retrieval application. The proposed descriptor considers as step forward to the object-based image retrieval.

In this paper, Design and implement a cartoon images retrieval system based on object extraction in order to minimize the effect of the background in cartoon images that is considered as a backbone to the object-based image retrieval. The implementation and testing of the system would depend on databases with different objects; backgrounds and colors contained in the cartoon images collected as part of the research work.

© 2016, IJCSMC All Rights Reserved

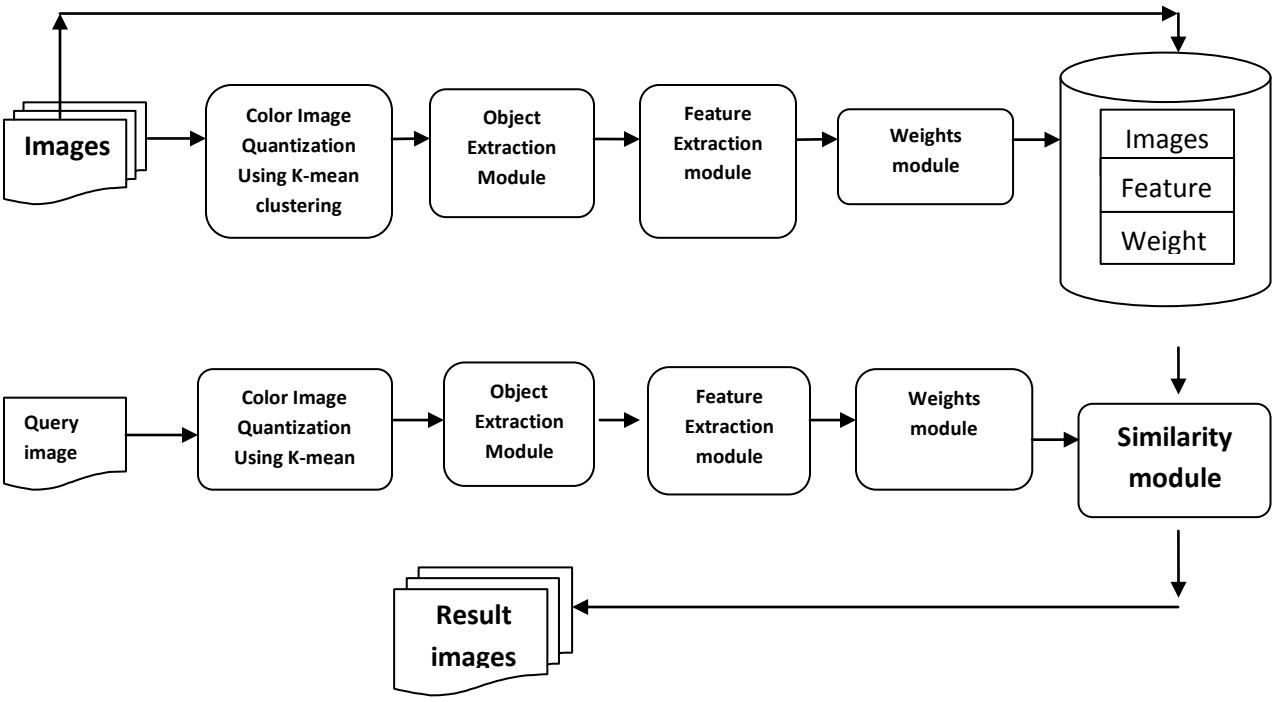


Fig.1: The layout of the proposed

II. The Proposed System

The layout of the proposed system is given in fig.1. The system stages are explained in the following:

Step-1*:* Load the color cartoon input image, *I*, of size *W*×H.

Step-2: Use Cartoon Image Object Extraction in [9,] to extract the cartoon character in input image. The most important step in this research is the process of extracting object, because it reduces the effect of image background on nature of retrieved images and grant significant assistance in the retrieval of the cartoon character corresponding to the cartoon character queries, each color index in image takes prosperity corresponding weight. The difference in weight depends on the index of color place in image weather this index is background index or object index of color.

The process of extracting the object from the cartoon image which is adopted in current PAPER consists of sub stages:

A. Reduce the number of colors in *I* using the K-Means clustering algorithm using the color components of the input color image. The mean (m) and standard deviation (σ) values of each color band (R,G,B) are determined and, then, used to determine initial centroid values of each cluster. Here 8 clusters have been used to initially representing the color contents of the input image, such that:

B.

$$\boldsymbol{C}_{RGB} = \boldsymbol{m}_{RGB} \mp \boldsymbol{\sigma}_{RGB} \quad (1)$$

Where CRGB is initial cluster vector, mRGB is the mean vector of the color bands of whole image, σRGB is the corresponding standard deviation vector.

© 2016, IJCSMC All Rights Reserved

K-Means clustering algorithm attempts to minimize the squared error for all elements in all clusters, where the error (E) equation is:

$$E = \sum_{p \in I} \min_{i=0,1,..,K} |p - C_i| \quad (2)$$

Here E is the sum of the absolute errors for all pixels (p) belong to the image, I, data set; and C_i is the centroid value of i^{th} cluster. The steps of k-means algorithm have been repeated five times to reach the convergence state.

The outcome of applying k-mean is the index map array, M(W,H), each element represent the color index value of certain image pixel; this index value is the cluster index value whose color is the closest one to the pixel color.

C. Removal of Segments Extended along the Border Using Seed Filling method, the following steps are applied:

1. Apply the seed filling method on the produced index map array, All connected index of same value without any cutting for connecting of the current index would be considered as a segment[9].

2. Then a check is made to see if this segment will satisfy the condition to remove or not, the condition to remove the segment is showed in equation (3).

$$Result\ segment = \begin{cases} backgroud\ segment & if\ segment\ size\ in\ borders => 0.04\ for\ up, lift\ and\ right\ edges\ of\ image \\ backgroud\ segment & if\ segment\ \ size\ in\ broder > 0.8\ for\ below\ edges\ of\ image \\ object\ \ segment & else \end{cases} \quad (3)$$

This threshold value is chosen by experiment. If a wrong number is selected for this parameter this will lead to wrong object extract.

3. All these steps are repeated for all index values.

C. Discard the colors not existing in the central part of image [9]

D. Find the ghost for the extracted character from previous steps, where the following steps are applied:

1. Create binary image object by converting the resultant index value of map array from previous steps to binary image, using the following equation (4):

$$Bin(x,y) = \begin{cases} 0 & if\ M(x,y) = 16\ (background\ index\ value) \\ 255 & else \end{cases} \quad (4)$$

Where $Bin(x,y)$ is the obtained binary image contain 255 for pixel classified as being part of background pixel, others it will contain 0 for pixel classified as part of object, Bin will be in the same size of the download image, and $M(x,y)$is the index map array.

2. Fill the Holes of the cartoon character: The goal of this step is to fill the holes that may be found in the cartoon object and finds the ghost for cartoon character.

© 2016, IJCSMC All Rights Reserved

E. Remove the small segments from the previous step, because these segments always cannot be considered as an object and maybe belongs to the background.

F. Fill the ghost region with true color values: each white pixels location in binary image filled by the original color to find the object of cartoon image because the white pixels are the ghost of object.

Step-3. Object Quantization**,** in this process the original image of object (Img_Object) resulted from algorithm (step-2) is quantized using K-Means clustering. Quantization process applies for object only, not for the entire image.

Step-4. **Feature Extraction step,** in this step the features of the images are represented as follows F1 = {(Ci,Pi), i = 0,. . . ,N _ 1} represents image features of that has number of dominant color (DC), C and P is value of colors and DC in the image percentage of any DC in the loaded image. In addition to color value of image quantization and the percentage of each DC in the image, weight for each DC in the image explains in this section.

DCs and The percentage of each index in the image will be extracted to be used in similarity matching process.

Step-5. **Weights of Index Values,** each index of color which contains cluster of quantization image will be given weight based on index location in image.

For the object index values, weight extracted in previous steps can be easily found depending on the percentage of the frequency for each index in that object; on the other hand, although the best result for the method which is used to extract the object, it may not give all the index values of clusters which form the object or may be some indexes, related to the background and it is not removed. Therefore, it is not possible to depend neither on the object weight only nor neglect the other indexes of clusters in image and depend only in extract object index. The number of each index value of object is divided on number of object's pixel called the salient object weight (SOWindex). The salient object weight indexes are explained in equation (5).

The second coefficient is the index's percentages (weights) in the resulted array index (all image) (DindexW) which computed in equation (6):

$$\text{SalientObjectWeight}_{\text{index}} = \frac{\text{Freq}_{\text{index}}}{\text{No: of Pixel(Object)}} \quad (5)$$

$$\text{DindexW} = \frac{\text{Freq index}}{\text{weight} * \text{height}} \quad (6)$$

For each of two coefficients (SOWindex and DindexW), there are two symbols, either large "L" or small "S", depending on any specific thresholds, The thresholds that extracted experimentally for cartoon image and can be considered in this work are 0.07 for SOWindex and 0.4 FOR DindexW.

The symbol "L" indicates that the index of color belongs to the coefficient with high percentage, otherwise it refers to the index that belongs to the coefficient with low percentage.

Since there are two input "two coefficients", it would be four possible cases to produce final weight of an index (FinalW). The index value of cluster will be given value (≈1); if it belongs to the object and must be considered or assigned with the value (≈0); if it belongs to background, , Other values range between (0 and 1). The importance of index depends on the two inputs to result final weight (FinalW).

All 4 possible cases of two inputs in order to result final weight of index can be stated as the following cases:

Case1: LL: refers to the high index in the two inputs weight (the object and the image). The index may be a large index object covers a large area in the image; therefore there is a need to find the border weight as stated equation (7 and 8). Final weight will be (1) if BW<0.3 (chooses experimentally) , on the other hand, if the index is also found in border in high percentage, it means the weight value will be maximum of the two inputs weights, because this index makes the decision to be confusing: whether to decide it belongs to the object or background.

$$\text{Border}_{\text{index}} = \frac{\text{Freq}_{index}(\text{Border})}{\text{BorderLength}} \quad (7)$$

© 2016, IJCSMC All Rights Reserved

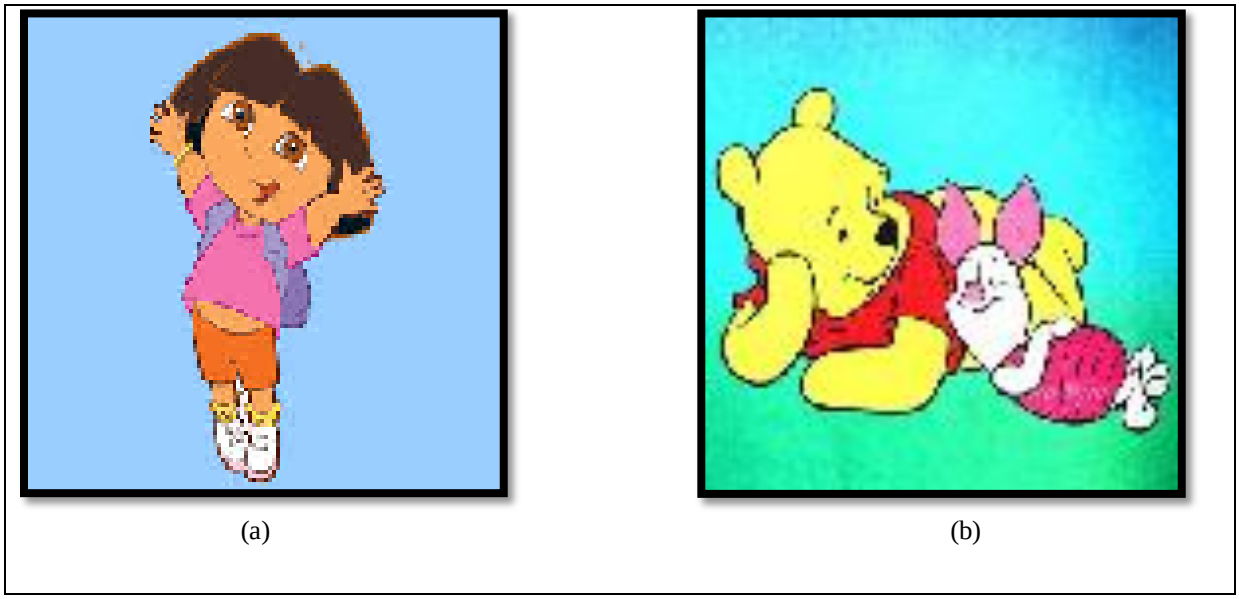
$$\text{BorderLength} = (\text{width} + \text{height}) * 2 \quad (8)$$

Case2: LS: refers to the large index in the object and low in the image, in this case, the weight will be the highest of the final indexes weights.
Case3: SL: refers to the small index in the object and high in the image. In this case the index will take the percentage of this index in image if it is not founded in borders of high percentage, if it is founded, then final weight will be "1-BW".
Case4: SS: refers to the index small in the images and the objects, but removing it from consideration may cause a mistake because it may be belong to object's index, so it is assigned with (DindexW), but if this index founded in high percentage on borders, it can be thought as line surrounded the image therefore it is assigned with "0" weight.

All the above description about weights for each index in image can be explained briefly in table (1). Figure (2) shows tow examples and weight of index values in table (2).

Table (1): Weights of Index Values Method.

SOWindex	DindexW	FINALW
L	L	(1 If BW<=0.3) or MAX(SOW, DindexW) if BW>0.3
L	S	1
S	L	((1-BW)If BW>0.3) or (DindexW if BW<0.3)
S	S	(0 if BW>0.3) or (DindexW if BW<=0.3)



(a) (b)

Fig.3: Two cartoon images example to show weight's way.

© 2016, IJCSMC All Rights Reserved

Table (2): Two example images show weights of index values method.

Image(a)										
	Index0	Index1	Index2	Index3	Index4	Index5	Index6	Index7	Index8	Index9
SOWindex	0	0	0.2999	0.0558	0.076	0.0507	0.068	0.1367	0.2704	0.0425
DCWindex	0.00037	0.7259	0.07768	0.01527	0.0209	0.01308	0.0186	0.0376	0.074	0.0166
FINALW	0.00037	0	1	0.01527	1	0.01308	0.0186	1	1	0.0166
Image(b)										
	Index0	Index1	Index2	Index3	Index4	Index5	Index6	Index7	Index8	Index9
SOWindex	0	0	0	0.108	0.03	0.017	0.154	0.069	0.485	0.124
DCWindex	0.0774	0.201	0.309	0.041	0.01184	0.0069	0.059	0.0288	0.187	0.047
FINALW	0.0774	0	0	1	0.01184	0.0069	1	0.0288	1	1

***Step-6*: Similarity matching,** this module computes the distances between query image and the database images to find the most similar images according to the query image depends on extracted features. [8] Described similarity matching modification which was explained in chapter two. It is highly suitable in this research for cartoon images retrieval system.

First compute di,j using equation (9), for all index values in image, finds the sum of absolute difference between query image clusters and target image clusters.

$$d_{i,j} = ||C_i^R - C_j^R| + |C_i^G - C_j^G| + |C_i^B - C_j^B| \qquad (9)$$

A. Then finds dmax, if the dmax is more than the chosen threshold value, Thd is the distance between the two colors to be similar. In this paper, the threshold adopted is selected experimentally 40 which mean two colors are considered similar. If a wrong number is selected for this threshold will lead to wrong retrieval images with different colors for query image), the $a_{i,j}$ (refers to in chapter two) value will be (0) else will be (1-$d_{i,j}/d_{max}$), such as in equation (10), where d_{max}=α Thd where α=1.2 [8].

$$a_{ij} \begin{cases} 1 - \frac{d_{ij}}{d_{max}} & if\ d_{ij} \le Th_d \\ 0 & if\ d_{ij} > Th_d \end{cases} \qquad (10)$$

B. Find the minimum between the two compared weight indexes (indexes of query image and target image), as computed in equation (11).

© 2016, IJCSMC All Rights Reserved

$$W_{ij} = \min(W_i, W_j) \quad (11)$$

C. Use the equation (12) to compute similarity between indexes percentages of two images (S_{ij}), where $p_q(i)$ refers to the percentage of the index in query image, and $p_t(j)$ refers to the percentage in index of target image. The min in equation (2.10) (($p_q(i)$, $p_t(j)$) represent the junction of $p_q(i)$ and $p_t(j)$, which describes the similarity of index in percentages, where, 1 - | $p_q(i)$- $p_t(j)$ |, the distance between two colors index values in their percentage. The index of similarity increases when absolute differences of the two indexes are decreased, while it decreases when the absolute difference increases.

$$S_{ij} = [1 - |p_q(i) - p_t(j)|] \times \min(p_q(i), p_t(j)) \quad (12)$$

D. Now use equation (13) to find similarity between two compared images [8].

$$SIM^w(I_1, I_2) = \sum_{i=0}^{N-1} \sum_{j=0}^{M-1} a_{ij} S_{ij} W_{ij} \quad (13)$$

E. Measures the dissimilarity that based on the color indexes' weight, can be computed utilizing equation (14) [8].

$$D_w(I_1, I_2) = 1 - SIM^w(I_1, I_2) \quad (14)$$

Step-7: Retrieval module, this module is concerned with retrieving the most similar images listed in the database to the end user's query image; the retrieval is done according to images' features in the proposed system, the most 10 similar images are displayed to the user.

III. Experimental and Results

This part is dedicated to present and discuss the results of experimental tests which conducted to study the retrieval performance of the suggested Cartoon Images Retrieval system.

3.1 Cartoon Images Retrieval system evaluation

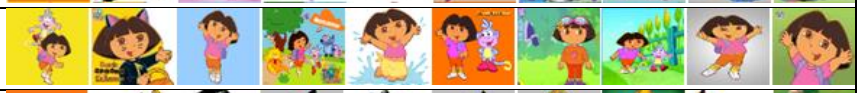
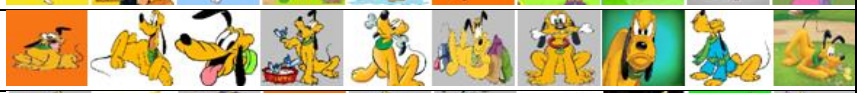
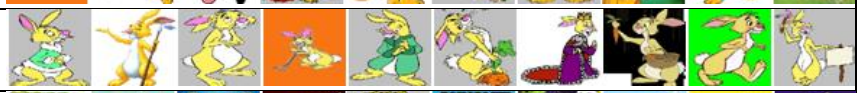
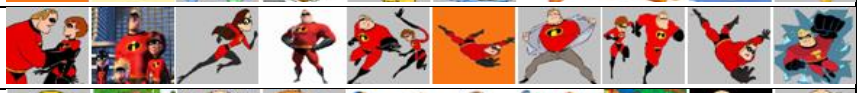
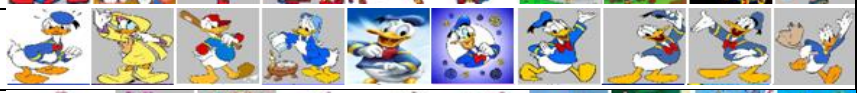
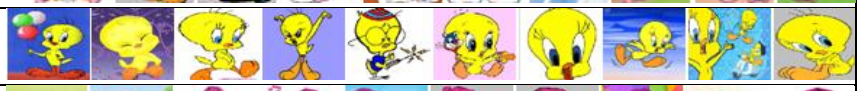
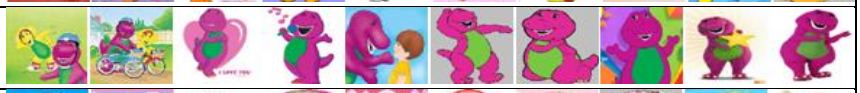
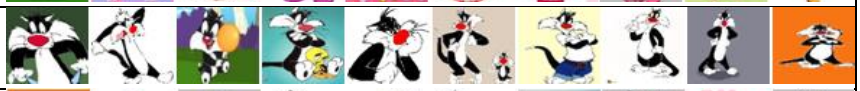
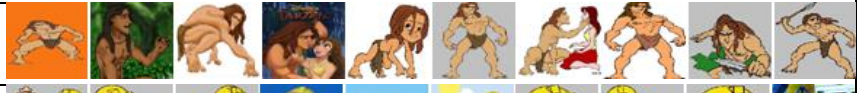
This section is built to present the evaluation of the proposed system that was introduced in the previous chapters. A database selected to test proposed system, and a comparison makes between performance for the system with and/or without weight (object extraction) to show the effect of colors weight to the proposed cartoon images retrieval system.

Time is important in retrieval system especially in query image and feature extraction. In the suggested cartoon images retrieval system, the time required to extract feature for one image of size 200×200 is 3 second, and the maximum retrieval time for Cartoon 3-k is 4 second.

3.1.1 Cartoon 3-k Database

Cartoon 3-k database is a cartoon images' database that have been selected as dataset that contains 3010 cartoon images collected and selected from the cartoon 11k [8] and the Internet. Cartoon dataset has 40 classes (40 different characters); each class contains at least 36 images. Fig4 shows 10 sample images in each image class.

© 2016, IJCSMC All Rights Reserved

Eeyore	
Dora	
Pluto	
Rabbit	
Sponge bob	
Winnie	
Mr. incredible	
Kayu	
Donald duck	
Fanatics	
Tweety	
Pink	
Strawberry	
Sylvester	
Gopher	
Smurf	
Smurfs2	
Spiderman	
Tarzan	
Bob	
Puzzle	

© 2016, IJCSMC All Rights Reserved

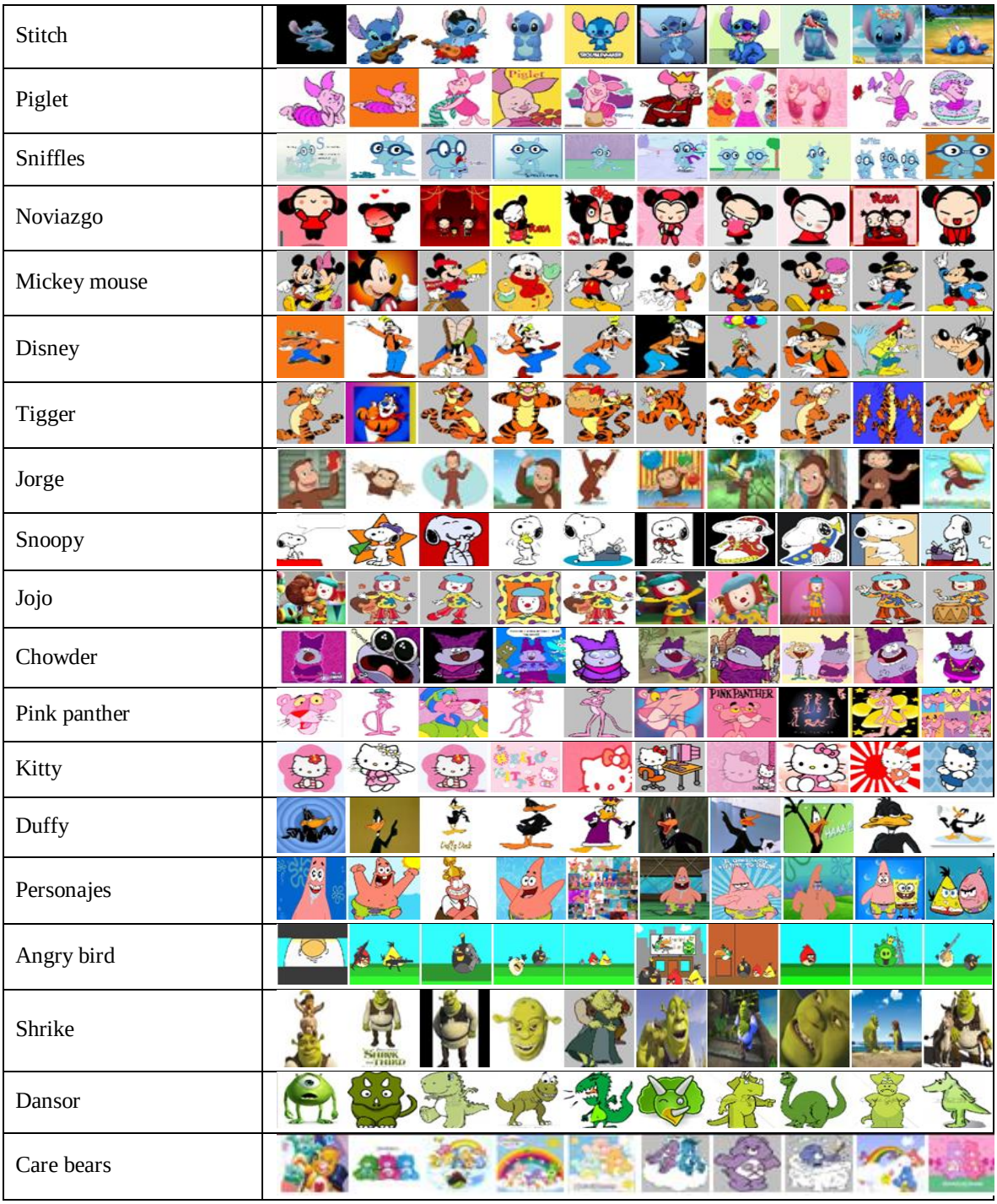


Fig.4: Example Images from each of the 40 Classes of Cartoon 3k Database.

3.1.2 Object extraction

The performance of object extraction step was tested on a data set of cartoon images contains more than 3000 cartoon images for 40 different objects (characters); such as Dora, Sniffer and Pluto, which are calculated from web and cartoon-11k [4]. Fig5 shows some examples of cartoon objects which are extracted well from the cartoon images samples have simple or complex background. Fig6 shows some of failed results; for example the Sniffer object is extracted well, but the background extraction was failed, this is due to the similarity colored regions. While, on the other hand, the extraction of Dora object was failed, because the object is too extended and

© 2016, IJCSMC All Rights Reserved

hit the image boundaries with high hit rate. Also, the Tweety object was wrongly extracted because the object is spreading at all parts of the image. Lastly, the Dora and Pluto over extracted object because the wrong extracted region is found in the centre of image with object and nor the distribution neither touch in borders. Table (3) shows some true extracted rate for some choosing cartoon images within different objects

Table (3) True extracted ratio of some tested images

Character Name	Test Images #	True Ratio
Eeyore	92	93.4%
Dora	113	92.9%
Pluto	92	94.5%
Rabbit	44	95.4%
Spongebob	79	89.8
Winnie	88	90%%
Mr. incredible	20	86.9%
Kayu	53	94.3%
Donald duck	100	96%
Fanatics	76	92.1%
Tweety	100	90%

© 2016, IJCSMC All Rights Reserved

Fig.5: True extracted object

Fig.6: False extracted object under or over

3.1.3 Effectiveness

To do deep evaluation for the proposed Cartoon Retrieval system, 20 images are randomly selected as examples from each of the 40 classes in the cartoon image set (cartoon 3k). For any example, the retrieval precision at each stage of the recall (5, 10, 15, 25, and 30) is acquired by maximize the number of recovered images. The 800 are given the final chart of fig7 by compute average for a retrieval result of precision/recall.

© 2016, IJCSMC All Rights Reserved

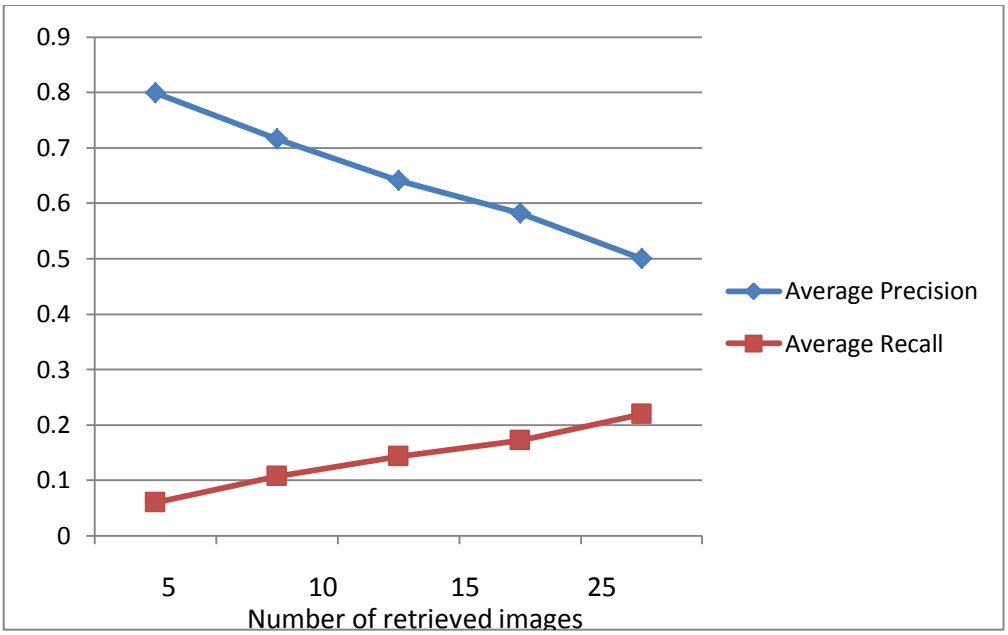


Fig.7: Chart for the average precision and recall for 800 randomly selected queries.

3.1.4 The Effect of Weight on the Proposed System

This subsection discusses the evaluation of the retrieval accuracy of the suggested system and compares it with and without using weight. For 20 class in the image set, randomly selected 10 images as queries The average precision for each group depend on the returned Top (10) images were registered. See fig (8 and 9) which showing weight get better retrieval result.

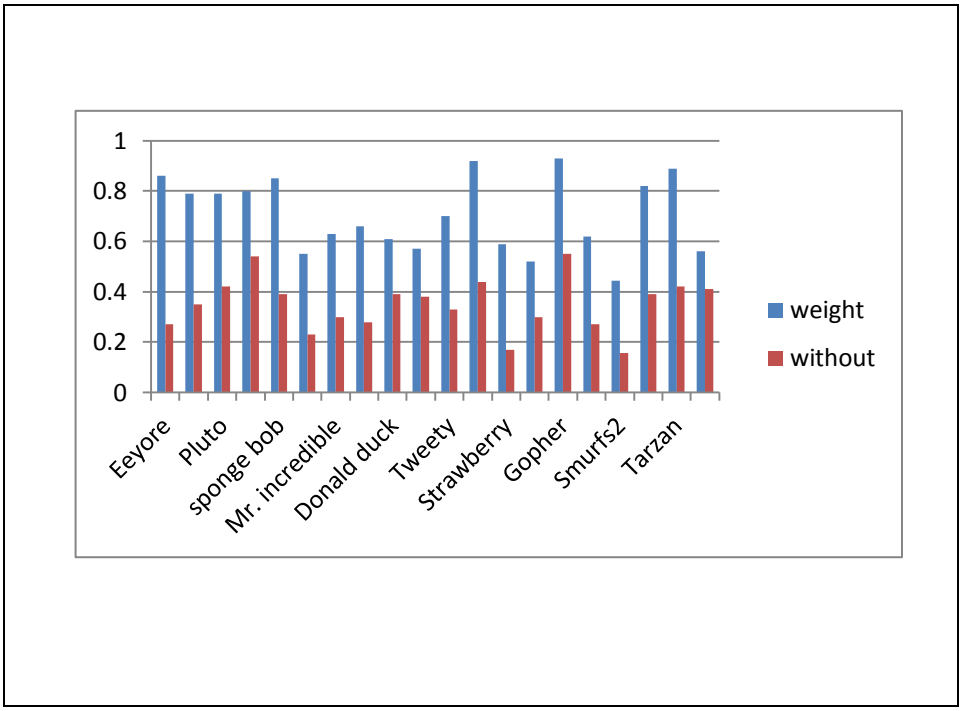


Fig.8: Chart of compression of average precision (top10) of proposed system used weight and without weight

© 2016, IJCSMC All Rights Reserved

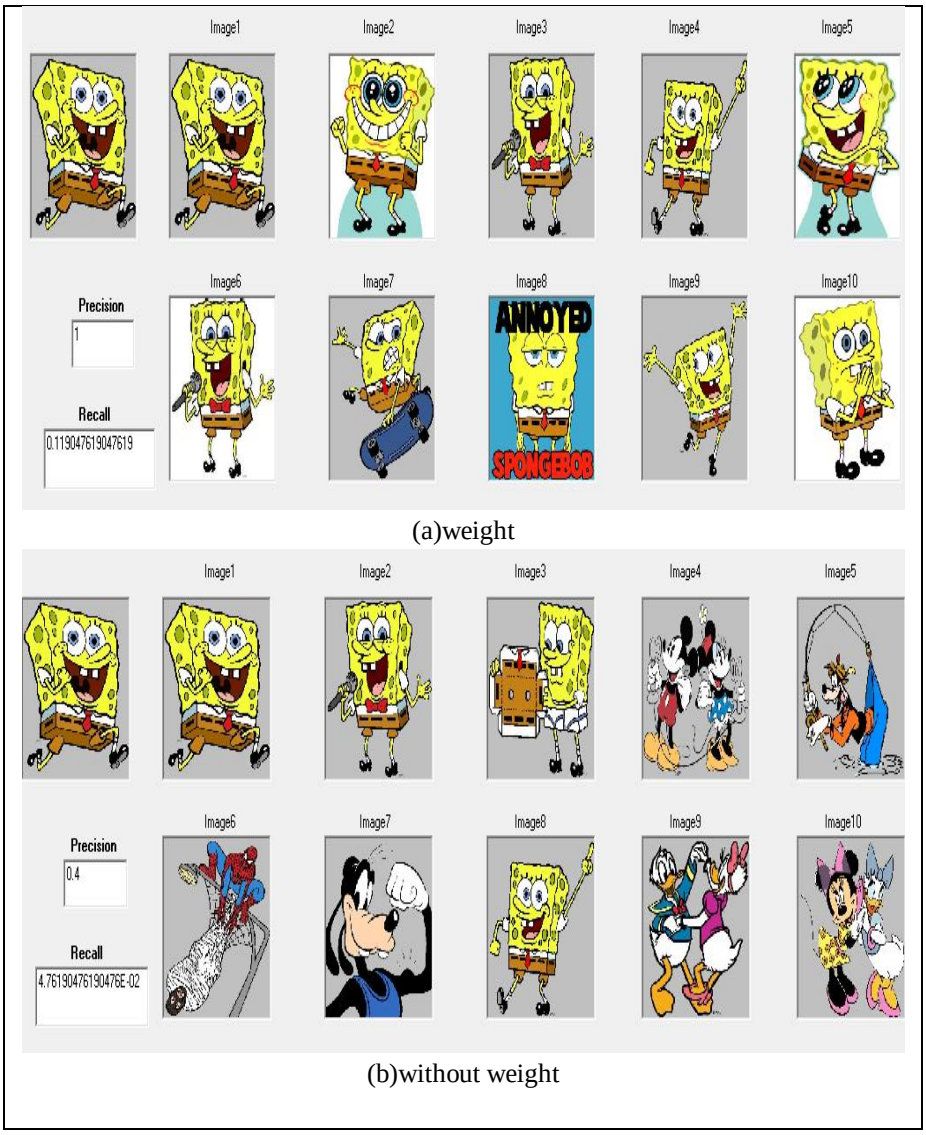


Fig.9: The visual comparison of the proposed weight proposed system and without weight system.

3.1.5 Proposed **System Comparison with Other Systems**

This section compares the proposed system with some existing CBIR systems'. The other systems that selected for comparison utilize DC to act images, and they apply on the Cartoon 5K in [8] database to evaluate their proposed systems which contains 5128 images grouped into 85 class. Fig10 presents the quantity comparisons that are apply utilizing the precision as metrics for 106 queries (2.1% from database size [8]). The proposed system shows high precision compared to other system.

© 2016, IJCSMC All Rights Reserved

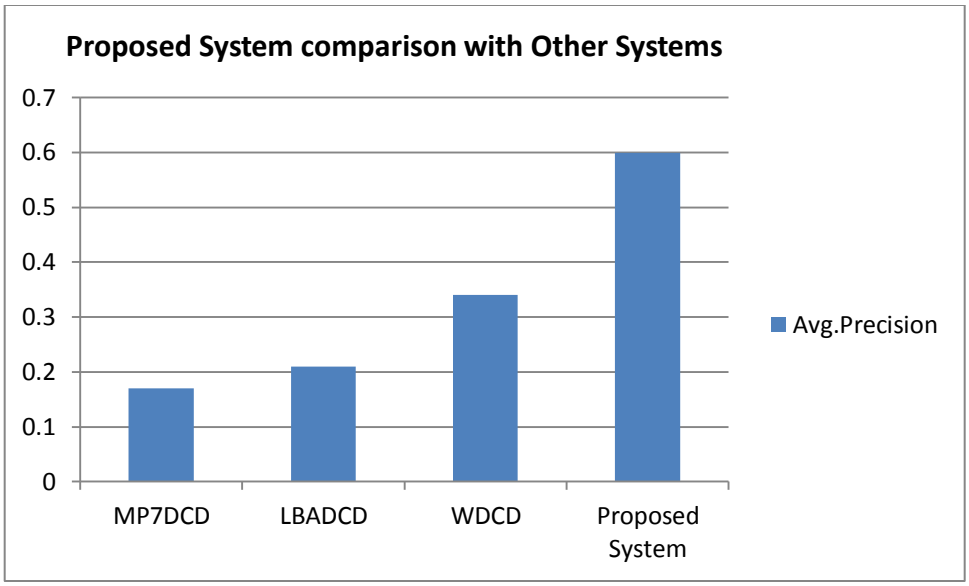


Fig.10: Chart of Proposed System comparison with Other Systems.

IV. Conclusion

The performance of cartoon images retrieval system has been evaluated using two measurements: precision and recall. The results of tests indicate the best rate of 0.965 precision and 0.224 recalls. The mechanism of the proposed cartoon images retrieval is based on assigning weight to each index value of clusters in quantized image "DC in image " in accordance with its belonging (i.e. to the object or to the background). The background index values (background colors) will receive low weight to reduce the effect of those colors in retrieval operation. The object index values (object colors) will receive high weight. A weighted dominant color for cartoon images retrieval is not quite enough to perform object retrieval; on the other hand, it is useful to filter out huge number of images and keeps those which have the same color of object. The results of test indicate that the domain color is not enough to retrieve cartoon images (such as images of different object color for the same character).

References

[1] Yi, Y., Yueting, Z., Dacheng, T., Dong, X., Jun, Y., and Jiebo, L.; "Recognizing Cartoon Image Gestures for Retrieval and Interactive Cartoon Clip Synthesis", IEEE Transactions On Circuits And Systems For Video Technology, Vol. 20, No. 12, Pp. 1745-1756, December 2010

[2] Zhang, L., Yueting, Z. , Yi ,Y. and Jun, X.; "Retrieval-Based Cartoon Gesture Recognition and Applications Via Semi-Supervised Heterogeneous Classifiers Learning ", Pattern Recognition, 2013.

[3]: Yang, Y., Zhuang, Y., Xu, D., Pan, Y., Tao, D. and Maybank, S. "Retrieval Based Interactive Cartoon Synthesis Via Unsupervised Bi-Distance Metric Learning", in Proc. ACM Multimedia, Pp. 311–320, 2009.

[4] Sungyoung, K., Soyoun, P., and Minhwan, K.; "Central Object Extraction for Object-Based Image Retrieval", E. M. Bakker et al. (Eds.): CIVR 2003, LNCS 2728, Pp. 39-49, 2003.

[5]: Haseyama, M., and Matsumura, A., "A Trainable Retrieval System for Cartoon Character Images", in *Proc. ICME*, Pp. 393–396, Jul. 2003.

© 2016, IJCSMC All Rights Reserved

[6]: Krishnan, N., Sheerin, B. and Callins, C., "Content Based Image Retrieval using Dominant Color Identification Based on Foreground Objects", International Conference on Computational Intelligence and Multimedia Applications, Pp.190-194, 2007.
[7]: Yang, N..,. Chang, W.,. Kuo, C., Li, T., "A Fast MPEG-7 Dominant Color Extraction with New Similarity Measure for Image Retrieval", J. Visual Commun. Image Rep. 19 (20), Pp. 92–105, 2008.
[8]: Ahmed, T., Massudi, M., Husniza H., and Loay E. G, "A Weighted Dominant Color Descriptor for Content-Based Image Retrieval", J. Vis. Commun. Image R., (Elsevier), Pp. 345-360, 2013.
[9]: Mustafa, M., Loay, E., Ghadah, A., "Cartoon Image Object Extraction",IJCSMC, Vol. 3, Issue. 12 , pg.482 – 491, 2014.

© 2016, IJCSMC All Rights Reserved