

---

# Change Detection Algorithms for Surveillance in Visual IoT: A Comparative Study

BEENISH AYESHA AKRAM\*, AMNA ZAFAR\*, ALI HAMMAD AKBAR\*, BILAL WAJID\*\*,  
AND SHAFIQUE AHMAD CHAUDHRY\*\*\*

RECEIVED ON 20.10.2016 ACCEPTED ON 21.02.2017

## ABSTRACT

The VIoT (Visual Internet of Things) connects virtual information world with real world objects using sensors and pervasive computing. For video surveillance in VIoT, ChD (Change Detection) is a critical component. ChD algorithms identify regions of change in multiple images of the same scene recorded at different time intervals for video surveillance. This paper presents performance comparison of histogram thresholding and classification ChD algorithms using quantitative measures for video surveillance in VIoT based on salient features of datasets. The thresholding algorithms Otsu, Kapur, Rosin and classification methods k-means, EM (Expectation Maximization) were simulated in MATLAB using diverse datasets. For performance evaluation, the quantitative measures used include OSR (Overall Success Rate), YC (Yule's Coefficient) and JC (Jaccard's Coefficient), execution time and memory consumption. Experimental results showed that Kapur's algorithm performed better for both indoor and outdoor environments with illumination changes, shadowing and medium to fast moving objects. However, it reflected degraded performance for small object size with minor changes. Otsu algorithm showed better results for indoor environments with slow to medium changes and nomadic object mobility. k-means showed good results in indoor environment with small object size producing slow change, no shadowing and scarce illumination changes.

**Key Words:** Change Detection, Video Surveillance, Image Processing, Simulation, Visual Internet of Things.

## 1. INTRODUCTION

The term IoT (Internet of Things) was first used by Lee et. al. [1] defined IoT, also called the Internet of Everything or the Industrial Internet, "is a new technology paradigm envisioned as a global network of machines and devices capable of interacting

with each other". It emerged from amalgamation of ubiquitous computing and WSN (Wireless Sensor Networks). The applications of IoT are humongous, encompassing smart homes, intelligent transportation, health assistance, security, environment monitoring, smart

---

Corresponding Author (E-Mail: beenish.ayesha.akram@uet.edu.pk)

\* Department of Computer Science & Engineering, University of Engineering & Technology, Lahore.

\*\* Department of Electrical Engineering, University of Engineering & Technology, Lahore.

\*\*\* Department of Computer Science, Dhofar University, Salalah, Oman.

shopping and surveillance [2]. VIoT deals with image and video processing in particular [3]. In recent years, IoT and VIoT have garnered huge research interest due to evolving and merging heterogeneous technologies.

Surveillance refers to monitoring of behavior, objects and events whereas DVS (Digital Video Surveillance) is simply surveillance conducted using cameras. DVS has found widespread role in security [4] in urban, industrial and defense sectors. Other applications include monitoring of assets, measuring traffic flow and vehicle trajectories, detection of accidents on highways, logging and monitoring pedestrian congestion in public areas, compiling customer demographics in markets and parks, measuring flow of people in troubled areas, in bio-hazard industries like nuclear reactor for process control or in defense for target surveillance and tracking via drones.

Furthermore, DVS may be performed using close circuits TVs (CCTVs) or DVRs (Digital Video Recorder) which are capable of capturing and storing large amount of video and can generate alerts and emails based on what is seen on these cameras. On the contrary, DVS may be part of homogeneous VWSN (Visual Wireless Sensor Networks) [5] or VIoT containing heterogeneous devices with varying computational and storage capabilities. These devices may include resource constrained visual sensor nodes, with lower cost, smaller size, limited processing power, average quality cameras and smart phones cameras [6]. Computationally powerful device such as gateway is used to relay the information to Back Office. For low-form factor VIoT devices, it is required to employ techniques which pose less computational and communication overhead [7].

A common DVS for VIoT comprises several phases as shown in Fig. 1. ChD is performed as in-network processing on low-form factor VIoT devices. For video surveillance, object classification and tracking is done as Back-office processing on servers.

The efficiency and effectiveness of VIoT based DVS is dependent upon algorithms used at every step. This paper focuses on comparison of various ChD algorithms suitable for resource constrained devices in VIoT.

ChD refers to monitoring temporal changes in images. ChD algorithms can be categorized in several ways. One way to categorize them is based on image representation [8] namely pixel based, region based and edge based. Another categorization is based on algorithm being supervised or unsupervised. ChD algorithms can employ different techniques. Examples include image differencing, image ratioing, image regression, principle component analysis and component vector analysis [9]. Image differencing and rationing are computationally inexpensive and can easily be implemented on low-form factor devices with minimal resources which makes these techniques well suited for VIoT and VWSN.

For VIoT devices, ChD must be implemented using computationally inexpensive techniques to conserve memory, CPU (Central Processing Unit) cycles and battery power. Consider a typical VIoT based DVS system, where multiple VIoT devices such as visual sensor nodes, smart cameras acquire images and execute ChD algorithm. If change is detected, data is transmitted to gateway. The gateway forwards data to Back office for further processing. This process is represented in Fig. 2.

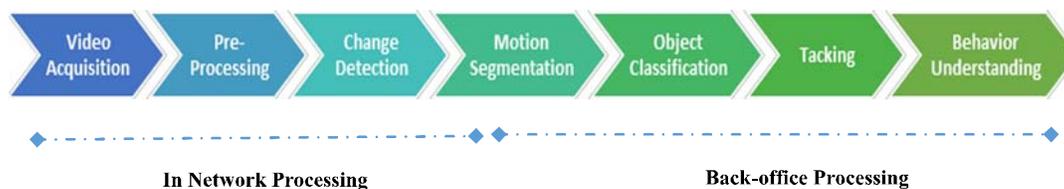


FIG. 1. STEPS EMPLOYED IN VISUAL SURVEILLANCE FOR VIOT

ChD algorithms have been compared previously as well [10-11]. However, as far as the authors know, there is no reported work that compares and suggests appropriate ChD algorithms for VIoT based specific environment settings. The scenarios where slow change and/or rapid change and illumination changes in indoor and/or outdoor are present. This work simulates multiple ChD algorithms and evaluates the performance of each in different environment settings using quantitative measures such as OSR, JC and YC. The algorithms that have been compared are pixel-based, unsupervised using image differencing. The algorithms list includes histogram thresholding algorithms Otsu, Rosin, Kapur and classification methods k-Means and EM [10,12].

The remainder of the manuscript is organized as follows; Section 2 provides a detailed review of related work; Section 3 describes the simulation framework, evaluation parameters along with results and performance analysis. Sections 4 and 5 summarize the results and suggest possible future work.

## 2. RELATED WORK

There have been various studies to compare and evaluate the performance of ChD algorithms to analyze their efficacy using different quantitative measures. Patra et. al. [13] investigated histogram thresholding algorithms for ChD. The authors compared Otsu, Kapur, Kittler, Huang and Liu’s thresholding algorithms on three different image data sets in terms of number of false alarms (unchanged pixels identified as changed ones), number of missed alarms (changed pixels categorized as unchanged ones) and overall error calculated as sum of false and missed alarms. The authors used three diverse multi-temporal datasets consisting of satellite images of terrestrial areas of the Island of Sardinia Italy, Mexico, and the Peloponnesian Peninsula, Greece. All the algorithms evaluated were histogram thresholding based and tested only on satellite images of aforementioned three cities. It was shown by obtained results that Liu’s and Kapur’s algorithms performed better.

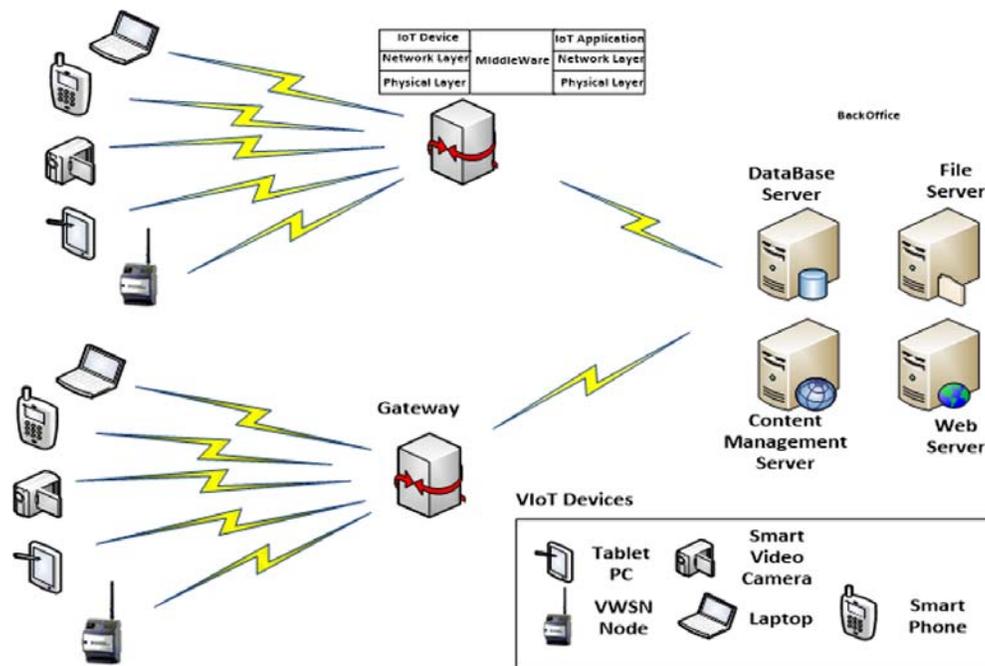


FIG. 2. TYPICAL DVS SYSTEM IN VIOT

Coelho et. al. [14] compared thresholding, classification and merging algorithms for ChD. Three thresholding algorithms namely Ridler-Calvard, Otsu, Mean Pixel Value, three classification algorithms including two variants of Watershed, Srinivasa et al. and a merging algorithm by Lin et. al. was used. The authors utilized biomedical imagery data set consisting of 97 fluorescence microscopy images. The authors employed similarity measures, Rand and JC for performance evaluation. The evaluation results concluded that Lin, et. al. merging algorithm performed unsurpassably. Ursula et. al. [15] evaluated six thresholding algorithms (Isodata, Otsu, Minimum error, Moment-preserving, Pun and fuzzy and a manual method) against k-means to analyze features of bread crumb images. The authors generated image dataset using fifteen white bread loaves sliced in 15mm thick pieces and scanned. The authors concluded that fuzzy, Isodata, Otsu and moment-preserving algorithms generated good and consistent results. Fuzzy algorithm performed well on images with large void areas in particular.

Minu et. al. [16] evaluated performance of ChD algorithms. The authors' perspective was to use multiple techniques namely image differencing, image ratioing, CVA (Change Vector Analysis), TCT (Tasseled Cap Transformation) and PCA (Principal Component Analysis) for change image computation. Furthermore, they applied Ridler-Calvard [17] thresholding algorithm to all five obtained change images. The dataset used was satellite images of an agricultural village belur situated in Karnataka, India over the span of 7 years during which Yagachi dam was commissioned. The satellite imagery captured the agricultural area in dry weather to obtain fogless and cloud free images. The authors concluded that overall best performance was shown by CVA. Also, image differencing was found to be suitable for ChD in biomass content and vegetation health. The authors recommended PCA for ChD in water quality and vegetation type; Tasseled cap transformation in regeneration and deforestation.

However, image rationing did not perform well for ChD in the area under investigation.

Rosin et. al. [18] compared performance of eight different thresholding algorithms for ChD. The algorithms selected included Ridler-Calvard, Tsai, Otsu, Kapur, Normal fitting, Huang-Wang, Yager and Rosin thresholding algorithms. Evaluation parameters chosen for result comparison were PCC (Percentage Correct Classification), YC and JC. But the dataset was limited to uniform floor image sequence and textured floor image sequence only. Kapur algorithm was found to perform well for change detection in presence of shadowing. Normal fitting, Tsai, and Rosin algorithms produced good results except where shadowing was present. Yager fuzziness algorithm performed the worst. Overall best performance was demonstrated by Tsai, Kapur, Rosin and Normal fitting respectively.

One of major applications of VIoT is video surveillance. VIoT based DVS is used for home security, target tracking, monitoring hazardous areas, object identification and outdoor surveillance. Zhang et. al. [3] proposed a VIoT system based on combination of conventional camera color image and depth image. The proposed system performed data fusion on both images to extract features for object identification. Meyer et. al. [19] proposed a framework supporting energy efficient image capture using low complexity wireless cameras, image enhancement, object detection and object state publication on IoT. In work done by Zarghami [20], a VMS (Video Monitoring System) was developed for VIoT incorporating a middleware to support interoperability. For efficient VMS, ChD is the crucial step requiring effective algorithms suitable for low-form factor VIoT devices. Maggiani et. al. [21] designed a histogram extractor algorithm targeting FPGA based smart cameras with low cost and complexity. In a distributed VIoT environment, the proposed algorithm can be used for thresholding based ChD.

Su et. al. [22] proposed a real time adaptive thresholding algorithm by computing different threshold values from local regions of change and then combined these threshold values by taking average. This final global threshold is then used for ChD. They used three different scenarios for evaluation of results and compared their proposed algorithm with other thresholding algorithms namely Otsu, Poisson, Euler, and Kapur. However, their comparisons were more focused on qualitative evaluations rather than quantitative. They used JC (JaccardCoefficient) as a quantitative measure but as it is discussed in section 3.1 Table 4, JC does not respond well to the effects of bad background estimation. They also specified that the average computation time of their proposed method is several times better than Euler and Poisson method but they have not provided any details of experimental or simulation results regarding computation time.

Gargi et. al. [23] conducted performance evaluation and characterization of different video shot-ChD methods for video sequence indexing. They compared algorithms on the basis of detection of cuts and gradual transition in four sequences of MPEG (Moving Picture Experts Group) video. Algorithms evaluated included color histogram methods using nine different color spaces, four different frame measures, three different representations of histogram and six variants of MPEG standards MPEG-A to MPEG-F and three block matching algorithms in different combination for experiments. They used window average thresholding, computed over local regions instead of a global threshold. They concluded that the color-histogram-based methods work well for cut detection with better tradeoff between recall and precision. Specifically, histogram intersection method was the best. They suggested that combined 3D or 1D methods are better than 2D methods which signifies that luminance plays an important role in shot ChD. They recommended Munshell color space and MPEG-E as it

considers all frames in MPEG video stream instead of selective frames. Their work specifically targets shot ChD in colored MPEG video streams and they evaluate the algorithms mainly in terms of recall and precision measures using the notion of false alarms and false detections.

Aach et. al. [24] proposed and evaluated statistical model based video change detection targeted on noise handling. They compared statistical significance tests for ChD based on grayscale difference image. They used local squared sum of differences inside a small sliding window based on Gaussian-distributed camera noise and sum of absolute differences as test statistic, both with tunable significance level. They proposed ChD as a null hypothesis of whether change is detected or not. They used the significance test to mark center pixel of a sliding window either changed or unchanged. Then they used Markov random field based refining method for smoothing and forming well defined boundaries between changed and unchanged areas, sparse small region removal, taking into account actual image data instead of binary change mask.

Ortego et. al. [25] proposed usage of foreground and motion images history for stationary foreground detection. They used both pixel level and frame level analysis of foreground and motion data. They extracted foreground features using background subtraction as well as computed foreground history by measuring the foreground temporal variation. Afterwards, they combined both of these producing an image which depicted foreground and motion variation over time. The resulting image was then thresholded to acquire the stationary foreground mask. They also performed motion analysis and motion analysis history evaluation using median filter over each half of temporal window at a time. The difference result from both halves was then thresholded using Kapur algorithm to extract motion regions. Afterwards results from motion and foreground image history are combined

using thresholding to produce the final static mask. They evaluated their work based on precision, recall and F-measure over four image datasets and showed that it performed well for high motion and occlusion prone scenarios.

In this paper, we analyze the performance of ChD algorithms using diverse datasets which is then shown to represent VIoT application scenarios. This work compares the performance of pixel based and unsupervised ChD algorithms and identifies appropriate algorithm based on IoT scenario requirements for effectiveness and efficiency. To represent indoor and outdoor environment, the datasets selected for simulation contain salient features such as varying number of objects, different lighting and illumination conditions and sudden, rapid and slow change in successive images.

The simulation work focuses on computing 'change image' using image differencing and then compares five ChD algorithms based on OSR, YC, JC measures. Based on the simulation results, a detailed performance analysis of ChD algorithms is presented. The work also highlights the guidelines for applicability of ChD algorithms for video surveillance in IoT.

### 3. PERFORMANCE EVALUATION

Performance of ChD algorithms can be evaluated qualitatively and quantitatively based on the application requirements. For qualitative evaluation, the user is displayed images successively while the 'change mask' produced by the ChD algorithm can be superimposed onto the image to make the change more visible. Furthermore, color codes may be used to differentiate between original image and change image. As for quantitative evaluation, it is based on the comparison of the output produced by ChD algorithm in relation to the 'ground truth'. The ground truth is an application specific reference image which gives the exact output as required

in a particular scenario [12]. As for the comparison of quantitative and qualitative evaluations, quantitative methods are better as they can be reproduced more reliably and are unbiased.

In various studies on image ChD, different evaluation parameters were used. Gonzalo Pajares et. al. [26] evaluated image change detection based on CDR (Correct Detection Rate) and FAT (False Alarm Rate). The authors used two data set; a synthetic dataset generated by commercial IKONOS satellite. The other was real dataset consisting of frames sequence representing varying illumination conditions both in indoor and outdoor. José Sigut et. al. [27] used PCC and YC for image ChD evaluation. Ilsever et. al. [28] used PCC, YC and JC to evaluate performance of ChD. The authors used two video sequences obtained from human-machine interaction scenario with diverse illumination conditions. The terms PCC, CDR and Overall Success Rate have been used alternatively to represent same accuracy measure in text, which is also the most widely used evaluation parameter.

#### 3.1 Performance Evaluation Measures

Confusion matrix for analysis of English alphabets was first proposed by Townsend [29] and its usage in machine learning was popularized by Provost et. al. [30]. Confusion matrix construction has become a de facto standard to use in image processing for result categorization as well. Table 1 shows the confusion matrix for evaluation of ChD algorithms.

Vincent et. al. [28] discussed several categories of accuracy measures for comparison of classifiers including association measures such as: chi-square-based measures (Coefficient, Pearson's, Cramer's), YC, Matthew's correlation coefficient, OSR, sensitivity, producer's accuracy, Dice's asymmetric index, F-measure and JC. The authors divided these accuracy measures into different categories which are depicted in Table 2.

The authors suggested to use simple accuracy measures such as OSR, TPR and TNR as their interpretation in accordance with data is straight forward. They recommended OSR as widely adapted measure and F-measure or JC to target specific class. They also suggested that similar trends are shown by multiple measures such as F-measure and JC, ICSI and Kulczynski's measure.

OSR is the most direct and commonly used parameter to evaluate algorithms in computer vision. However, it is not suitable when change is relatively small in comparison to the overall image itself [18]. So JC and YC help in overcoming the possibility of high rates of TN (True Negative) and small ChD in the image. JC excludes TN resulting in higher values of JC if TP value is large, which means detected change was identified correctly. However, it must be noticed that YC cannot be used when there is no change in the images at all as one of

the denominators will become zero in this case and all pixels will be classified as one class. Therefore, to avoid this exception, 'special condition' must be added to avoid dividing by zero. Consider a WSN deployed unattended in a remote sensing area. To avoid physical manipulation of SN (Sensor Nodes), VS (Visual Sensors) are deployed to periodically capture images of WSN field. This scenario represents a large number of small objects. In this case, the image sequence captured by VS contain small change, requiring the use of JC and YC along with OSR as discussed earlier.

In order to measure effectiveness and efficiency of ChD algorithms, the following measures were selected as shown in Table 3.

The suitability of these evaluation measures is described in Table 4 in form of their strengths, limitations and suitability for evaluating certain kind of image changes.

TABLE 1. CONFUSION MATRIX FOR IMAGE CHANGE DETECTION

Term	Description
TP (True Positive)	Number of changed pixels and correctly detected
FP (False Positive)	Number of unchanged pixels and incorrectly detected
TN True Negative)	Number of unchanged pixels and correctly detected
FN (False Negative)	Number of changed pixels and incorrectly detected

TABLE 2. CATEGORIZATION OF ACCURACY MEASURES [27]

Category	Accuracy Measure	Description
Nominal Association Measures	chi-square based measures ( $\Phi$ coefficient, Pearson's, Cramer's V) Yule's coefficient Matthew's correlation coefficient Mutual information-based measures	Maps similarity between distribution of instances over estimated classes and their distribution over the true classes
An Overall Success Rate	Overall Success Rate (OSR)	Observed proportion of correctly classified instances
Marginal Rates	Sensitivity (TPR) Specificity (TNR) Precision (PPV) Dice's association index (NPV)	Focuses on true classes Focuses on estimated classes
F-Measure and Jaccard Coefficient	F-Measure Jaccard Coefficient	Class specific and symmetric Class specific and symmetric
Classification Success Index	ICSI (Individual Classification Success Index) Kulczynski's measure CSI (Classification Success Index)	One minus the sum of PPV and TPR Overall measure defined by averaging ICSI over all classes

#### 4. SIMULATION

The following histogram thresholding and classification algorithms for ChD were compared: Otsu [32], Kapur [33], Rosin [34], k-means and EM. For simulation, hardware and software specifications are summarized in Table 5.

For performance comparison of ChD algorithms, eleven widely used data sets representing various indoor, outdoor environment with static to fast moving objects demonstrating change and illumination variation were used. The source of datasets is at [35]. The salient features for scene description, identified in selected datasets are summarized in Table 6.

Fig. 3 depicts feature set of datasets pictorially, representing feature wise richness of datasets. It can be seen that dataset No. 11 stands out as it depicts extreme end of feature combination such as high object mobility, image source mobility, illumination changes, shadowing,

fast change, variety of objects and of different sizes (small, medium and large). However, dataset No. 5 represents a common feature combination comprising of indoor environment, static object mobility, no image source mobility, medium change, moderate number of objects and medium object size. These environments are shown particularly to shed some light on individual algorithms in action.

**TABLE 5. HARDWARE AND SOFTWARE SPECIFICATIONS FOR SIMULATION**

Feature	Specification
Processor Model	Core2Duo P7350
Processor Speed	2 GHz
RAM Type	DDRII
RAM Size	3 GB
Operating System	Windows Vista Business Service Pack 1
Simulation Software	MATLAB version 7.8.0 release 2009a

**TABLE 3. QUANTITATIVE MEASURES FOR PERFORMANCE EVALUATION OF CHD ALGORITHMS**

	Evaluation Measure	Description	Values Interpretation
Effectiveness	Overall Success Rate	$(TP + TN)/(TP+FP+TN+FN)$	Higher value means better performance
	Yule's Coefficient	$ \frac{TP}{(TP+FP)} + \frac{TN}{(TN+FN)} - 1 $	
	Jaccard Coefficient	$\frac{TP}{(TP+FP+FN)}$	
Efficiency	Memory Consumed	Amount of memory consumed by algorithm execution	Smaller value means better performance
	Time Consumed	Amount of time consumed by algorithm execution	

**TABLE 4. QUANTITATIVE MEASURES STRENGTHS AND LIMITATIONS**

Evaluation Parameters	Strengths	Limitations	Environment uitability
OSR	Efficient processing of images with large change	Misleading results for small change resulting in large TN	Large change
YC	Overcomes shortcomings of OSR and JC by giving equal weight to both TP and TN	Cannot be used for no change in images (denominator becomes 0)	Medium change
JC	Efficient processing of images with small changes. Gives no weightage to TN	Does not consider no change pixels i.e. TN. Unable to take into account bad TN estimation	Small or no change

Fig. 4 illustrates results of all five ChD algorithms for one sample of data set No. 5, demonstrating an indoor environment where two persons are talking and background screen shows different statistics. From Fig.

4(a-b), it can be seen that there are only minor changes in both images. The person on the left slightly turned his head toward the other person. Such image sequence is an example of small change by medium sized objects. The

TABLE 6. FEATURE SET EXTRACTED FROM DATASETS FOR SCENE DESCRIPTION USED IN SIMULATION

Dataset No.	Environment	Object Mobility	Feature Set				No. of Objects	Object Size
			Image Source Mobility	Illumination Changes	Shadowing	Change Type		
1	Outdoor	Static	Yes	Yes	Yes	Slow	Few	Large
2	Indoor	Static	No	Yes	No	Slow	Few	Medium and small
3	Indoor	Nomadic	No	Yes	No	Medium	Few	Medium
4	Indoor	Static	No	No	No	Slow	Few	Medium
5	Indoor	Static	No	Yes	Yes	Medium	Moderate	Medium
6	Indoor	Nomadic	No	No	No	Slow	Moderate	Medium
7	Indoor	Static	No	Yes	No	Slow	Few	Small
8	Outdoor	Static	Yes	Yes	Yes	Fast	Few	Large
9	Indoor	Nomadic	No	No	No	Medium	Moderate	Medium and small
10	Outdoor	Nomadic	Yes	Yes	Yes	Fast	Many	Medium
11	Indoor	high	Yes	Yes	Yes	Fast	Many	Small, medium and Large

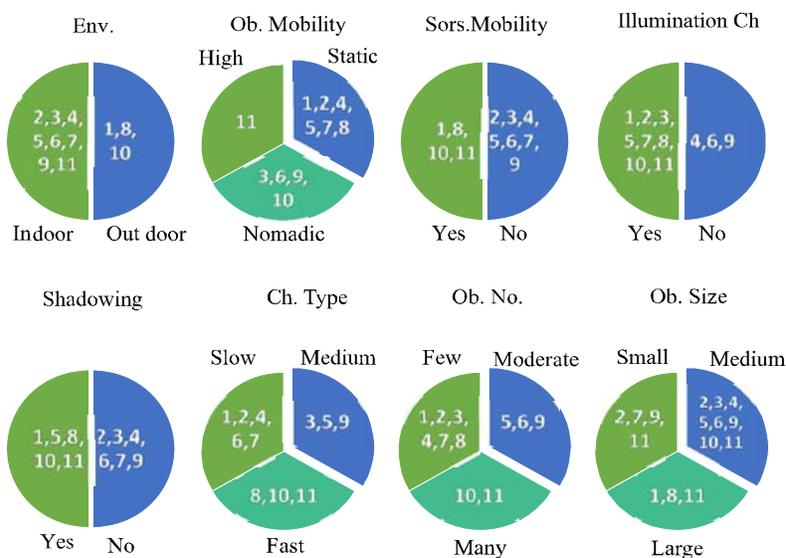


FIG. 3. PICTORIAL REPRESENTATION OF ATTRIBUTES OF DATASETS

image is segmented colored in green and yellow illustrating unchanged and changed regions. According to visual comparison, k-means algorithm generated better result in comparison with the other algorithms as it did not identify any change in background but only in position and posture of two persons which moved marginally in Image-2 from Image-1.

Fig. 5 shows results of all five ChDs algorithms for one sample of data set No. 11, representing movement of table tennis player. Fig. 5(a-b) mark two images from video sequence at two different times. The green area in change image represents regions where no change was identified by the algorithm whereas yellow area signifies regions where change was detected. It can be seen from Fig. 5 that Kapur algorithm produced minimum noise in change image whereas Rosin and Otsu generated change images with slightly higher noise. Change image generated by k-means and EM detected change in regions where actually there was no change.

#### 4.1 Results Discussion

From Fig. 6-7, Overall Kapur's OSR values were consistently highest except for dataset No. 2,6,7 and 9. For datasets 2, 7 k-means achieved maximum value of OSR. From Table 3, high value of OSR depicts large TP and TN computed, which signifies correct classification. Referring to Table 6, it can be observed that datasets 2 and 7 have some common features namely indoor environment with static object mobility, few number of objects, no image source mobility, no shadowing, scarce illumination changes and slow change. It can be inferred that for such environment k-means is a suitable algorithm for ChD. Similarly, Otsu generated highest values for datasets 6 and 9. Datasets 6 and 9 also shared same features namely indoor environment with nomadic object mobility, no image source mobility, moderate number of objects of medium size, no shadowing, no illumination changes and medium change as shown in Table 6. Therefore, it can be ascertained that Otsu is suitable for environments sharing such features.

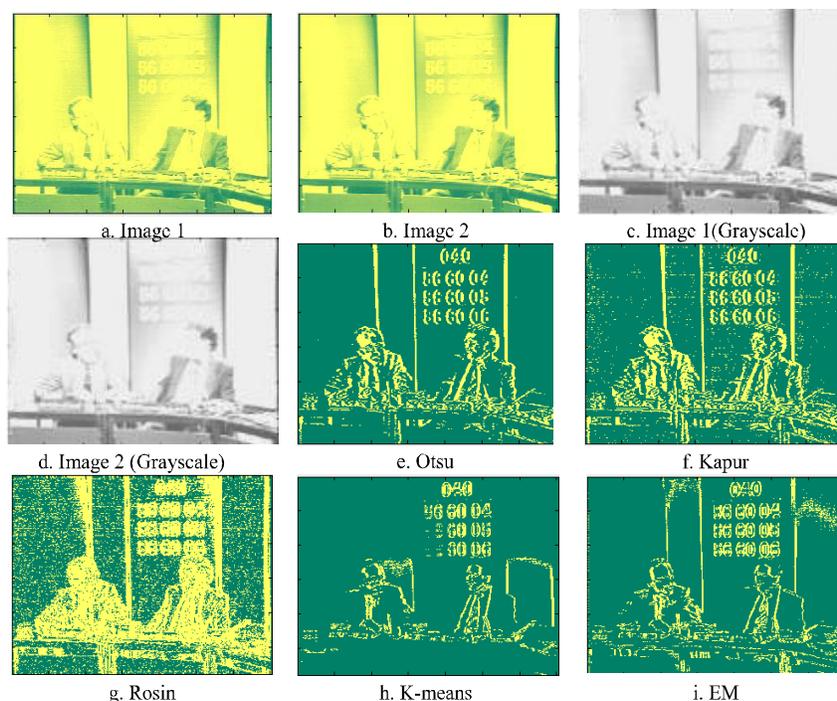


FIG. 4. RESULTS OF CHD ALGORITHMS ON ONE SAMPLE OF DATA SET NO. 5 REPRESENTING SMALL CHANGE

It is evident from Fig 7, that Otsu has the highest average value followed by Kapur, Rosin, k-means and EM respectively. However, it must be noticed that Otsu and Kapur’s overall averages across all eleven data sets are practically comparable. Kapur shows consistently good performance across all data sets whereas Otsu did not perform well in most of the data sets. But, in case of data sets 6 and 9, Otsu outperformed

all other algorithms hence leveraging its average value of OSR.

For similarity measure YC, Otsu and Kapur achieved similar results to OSR as shown in Figs. 8-9. Maximum value of YC is 1. Overall Kapur’s YC values were consistently highest except for datasets 2 and 7 in which Rosin achieved maximum value of YC. Otsu generated higher values for datasets 6 and 9.

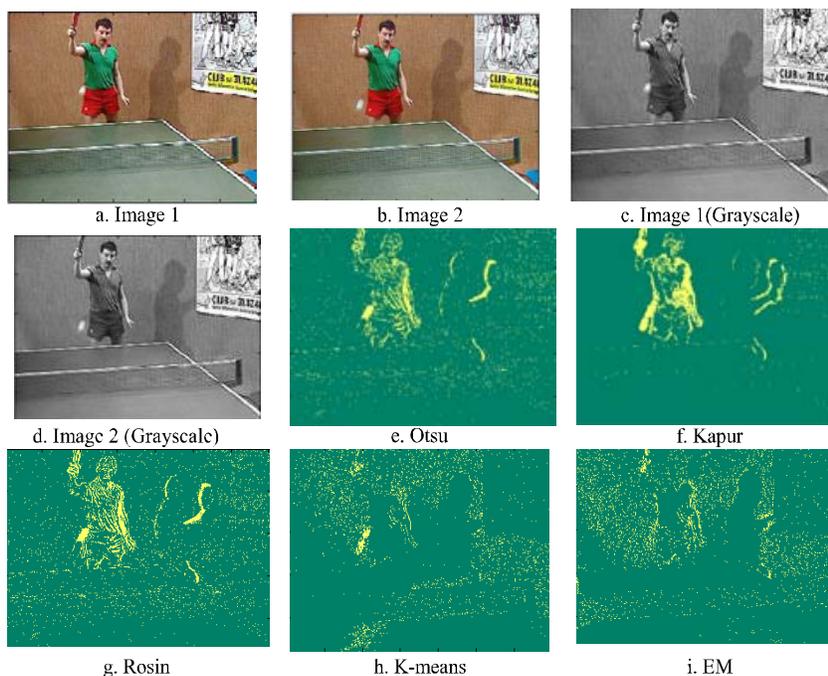


FIG. 5. RESULTS OF CHD ALGORITHMS ON ONE SAMPLE OF DATA SET NO. 11 REPRESENTING FAST CHANGE BY SMALL TO MEDIUM OBJECTS

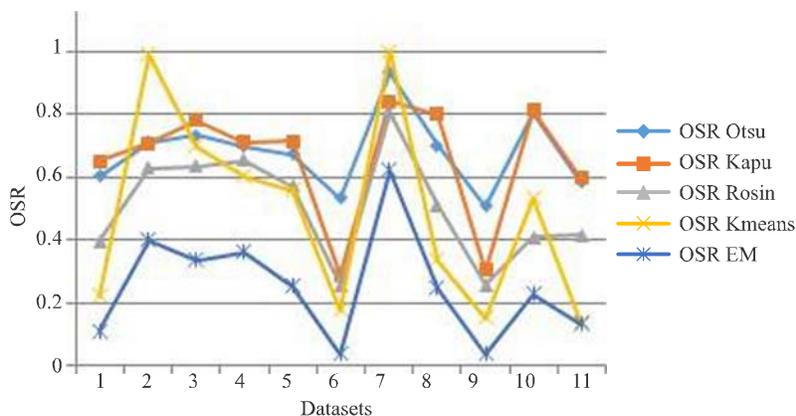


FIG. 6. COMPARISON ACCORDING TO OSR

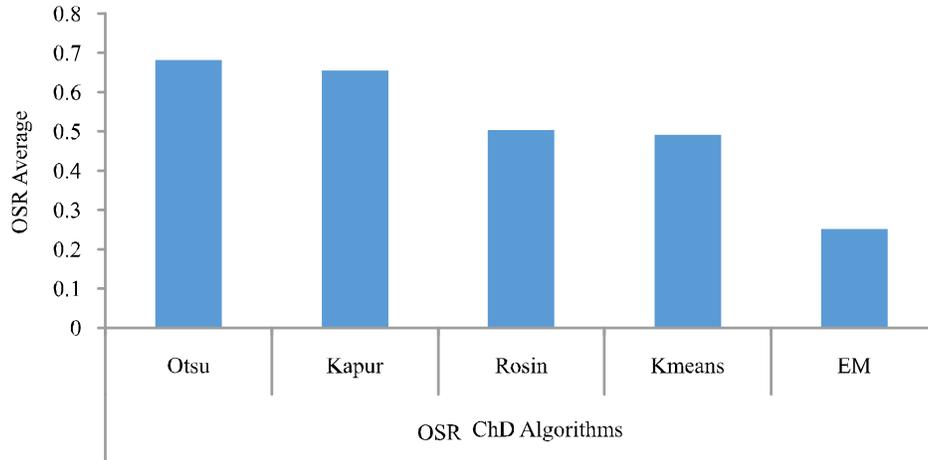


FIG. 7. OSR AVERAGE ACROSS ALL ELEVEN DATASETS

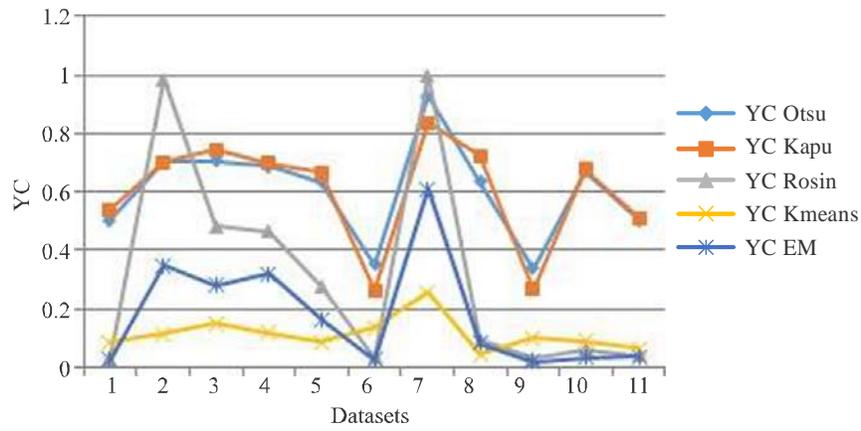


FIG. 8. PERFORMANCE COMPARISON USING YC

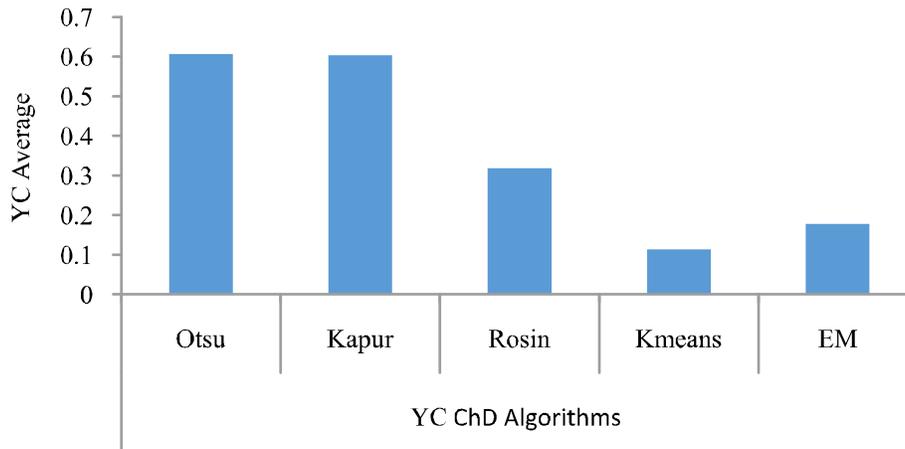


FIG. 9. YC AVERAGE ACROSS ALL ELEVEN DATASETS

In case of similarity measure JC, the results produced by ChD algorithms exhibit significant variation as k-means outperformed all other algorithms in five datasets 2, 3, 4, 5, 7 followed by Otsu, Rosin, EM and Kapur respectively, as illustrated in Fig. 10. These five datasets shared some common features such as indoor environment, with static or nomadic object mobility, varying illumination, no image source mobility and slow or medium change. Shadowing was missing in all cases except in dataset 5. Kapur generated lowest JC values indicating that for smaller number of TPs and large number of TNs in change image, change is very small in consecutive images. In this case, Kapur algorithm

showed poor performance and unable to detect such small change. k-means algorithm performed well in case of small change as compared to whole image.

According to JC measure, the maximum average value was attained by k-means algorithm, followed by Otsu, Rosin, EM and kapur in descending order, as shown in Fig. 11. In case of OSR and YC, similar trend was shown by Otsu and Kapoor across most of the data sets as discussed earlier. However, for JC unlike results were obtained with overall best performance shown by k-means, specifically for datasets with small changes in consecutive images.

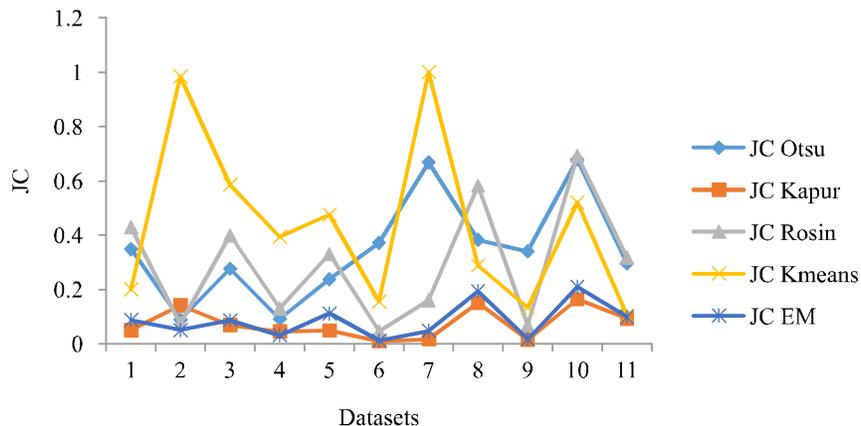


FIG. 10. COMPARISON OF CHD ALGORITHMS ACCORDING TO JC

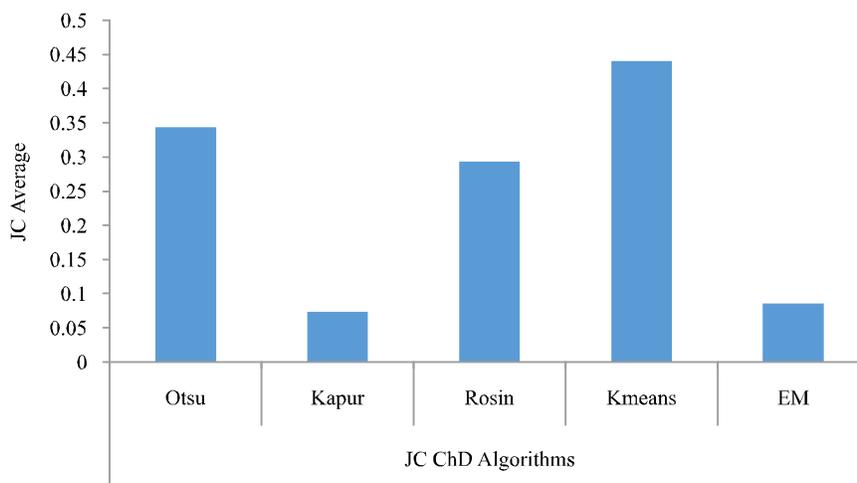


Fig. 11. JC average of ChD algorithms across all datasets

Fig. 12 shows execution time for individual datasets. Execution time of Otsu, Kapur and k-means was approximately similar for all datasets. Rosin consumed slightly higher execution time for dataset 8. EM consistently used up greater execution time in comparison with other algorithms and again showed high spike at dataset 8. In this particular dataset, an outdoor environment with static object mobility, image source mobility, illumination changes, shadowing and fast change were present.

Fig. 13 represent average execution time for all ChD algorithms. Otsu, kapur and k-means displayed approximately similar results with averages being 0.000103, 0.000271 and 0.000294, respectively over all datasets. EM showed worst performance in terms of execution time with average of 11.64427. Rosin algorithm performed better than EM with average 1.409745 seconds,

reflecting that except EM algorithm remaining algorithms consumed almost similar amount of execution time.

This is evident from Figs. 14-15 that k-means consumed maximum amount of memory space during execution. Otsu and Kapur showed comparable result for memory consumption. Best memory efficiency is shown by Kapur and Otsu utilizing an average of 2271.789KB and 2271.97 KB. Subsequently, Rosin, EM and k-means utilized 2873.485KB, 5279.185KB and 5921.769KB, respectively. Thresholding algorithms were memory efficient requiring least amount of memory, particularly Kapur and Otsu. Rosin also consumed lesser memory than k-means and EM algorithm. It can be concluded from result produced that for low-form factor VIOT devices thresholding algorithms are most suitable due to high time and memory efficiency.

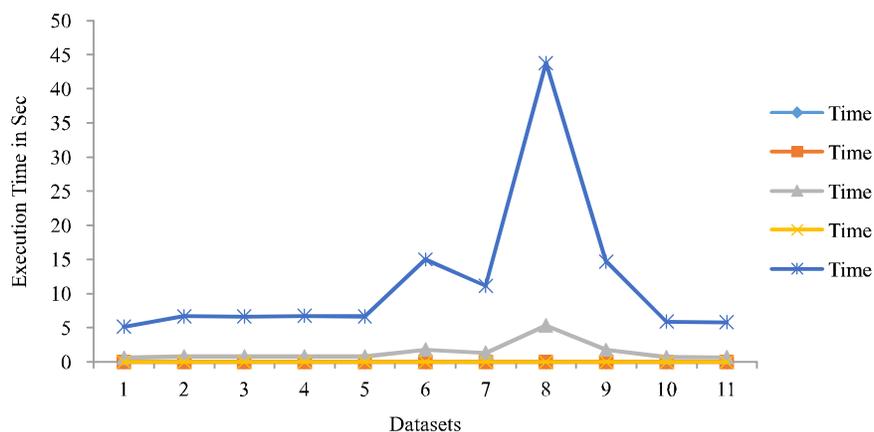


FIG. 12. EXECUTION TIME (IN SEC) OF ALL FIVE CHD ALGORITHMS

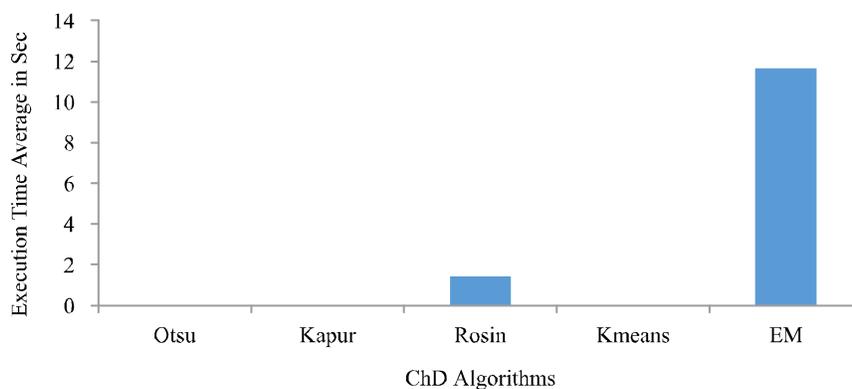


FIG. 13. EXECUTION TIME AVERAGE ACROSS ALL DATASETS

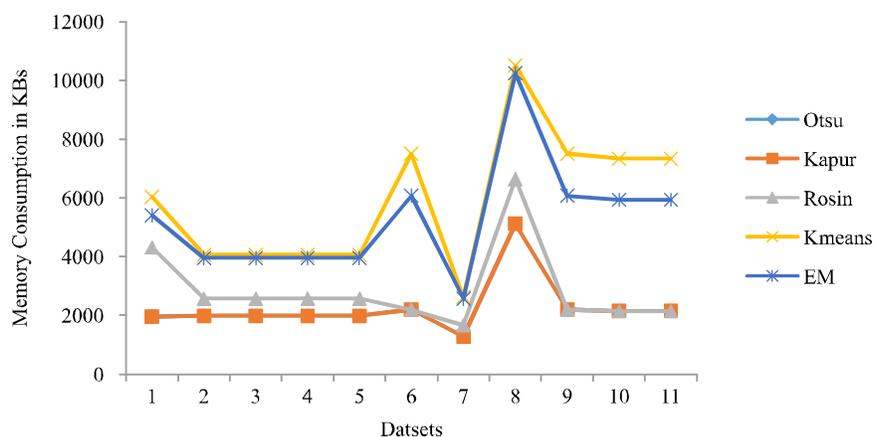


FIG. 14. MEMORY CONSUMPTION (KBS) ACROSS ALL DATASETS

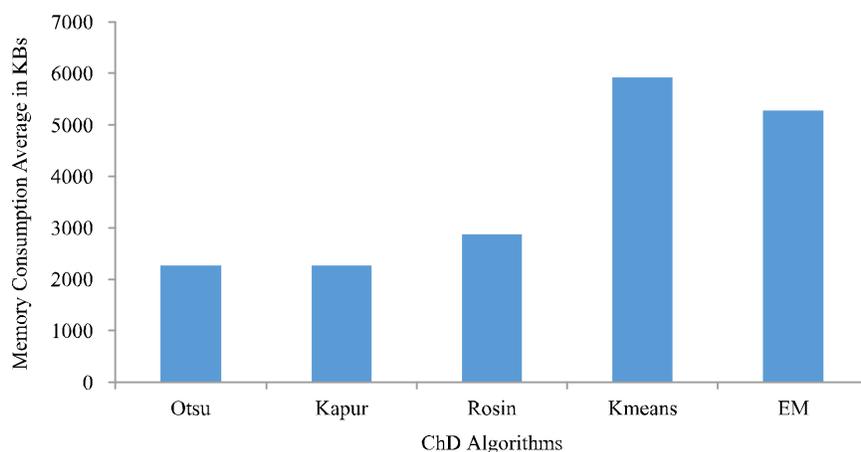


FIG. 15. MEMORY CONSUMPTION AVERAGE (KBS) ACROSS ALL DATASET

## 5. CONCLUSION

In this paper, we have carried out quantitative comparison of various image thresholding and classification techniques based on multitude of accuracy measures for change detection in VIoTs. It is concluded that Kapur algorithm performs well in presence of image source mobility, illumination and shadowing changes, varying number and sizes of object in both indoor and outdoor environments. Hence, it can be applied for ChD in numerous VIoT scenarios involving slow to fast change. However, Kapur algorithm is not suitable for scenarios where

change is small. It was observed from the performance evaluation that Otsu algorithm is suitable for indoor environment with nomadic object mobility and slow to medium change. k-means algorithm was found to be appropriate for indoor environment with small and slow change, provided it has no shadowing, few illumination changes and low or no mobility of objects as well as image source. Considering time and memory efficiency, thresholding algorithms are more efficient as compared to classification methods. Hence, thresholding algorithms are well suited for resource constrained VIoT devices. However, for VIoT scenarios with small change

and stringent memory resources, k-means is not suitable due to higher memory consumption. In such cases, Rosin algorithm provides better tradeoff providing small ChD and low computational complexity of thresholding.

With the emergence of middleware for VIOts, the scope of the study can be extended to solve the ChD problem. As ANN (Artificial Neural Networks), principal component analysis, and change vector analysis can be run at the BackOffice to explore further possibilities. Relationship among OSR, YC and JC can be further studied to gain more insight in the results obtained along with inclusion of other accuracy measures as suggested for better evaluation of ChD algorithms.

## ACKNOWLEDGEMENT

The authors acknowledge Prof. Dr. Muhammad Mian Saleem, Department of Electrical Engineering, University of Engineering & Technology, Lahore, Pakistan, for his valuable guidance and support.

## REFERENCES

- [1] Lee, I., and Lee, K., "The Internet of Things (IoT): Applications, Investments, and Challenges for Enterprises", *Business Horizons*, Volume 58, No. 4, pp. 431-440, Kelley School of Business, Indiana University, USA, 2015.
- [2] Gubbi, J., Buyya, R., Marusic, S., and Palaniswami, M., "Internet of Things (IoT): A Vision, Architectural Elements, and Future Directions", *Future Generation Computer Systems*, Volume 29, No. 7, pp. 1645-1660, Netherlands, September, 2013.
- [3] Zhang, X., Wang, X., and Jia, Y., "The Visual Internet of Things System Based on Depth Camera", *Proceedings Conference on Chinese Intelligent Automation*, pp. 447-455, Springer Berlin Heidelberg, China, 2013.
- [4] Melo, J., Naftel, A., Bernardino, A., and Santos-Victor, J., "Viewpoint Independent Detection of Vehicle Trajectories and Lane Geometry from Uncalibrated Traffic Surveillance Cameras", *International Conference on Image Analysis and Recognition*, pp. 454-462, Springer Berlin Heidelberg, Portugal, September, 2004.
- [5] Jabbar, S., Akbar, A.H., Zafar, S., Quddoos, M.M., and Hussain, M., "VISTA: Achieving Cumulative Vision through Energy Efficient Silhouette Recognition of Mobile Targets through collaboration of Visual Sensor Nodes", *EURASIP Journal on Image and Video Processing*, Volume 1, pp.1-24, USA, 2014.
- [6] Li, S., Da Xu, L., and Zhao, S., "The Internet of Things: A Survey", *Information Systems Frontiers*, Volume 17, No. 2, pp. 243-259, USA, 2015.
- [7] Sehgal, A., Perelman, V., Kuryla, S., and Schonwalder, J., "Management of Resource Constrained Devices in the Internet of Things", *IEEE Communications Magazine*, Volume 50, No. 12, pp. 144-149, USA, 2012.
- [8] Stringa, E., "Morphological Change Detection Algorithms for Surveillance Applications", *BMVC*, pp. 1-10, England, September, 2000.
- [9] Melgani, F., Moser, G., and Serpico, S.B., "Unsupervised Change-Detection Methods for Remote-Sensing Images", *Optical Engineering*, Volume 41, No.12, pp. 3288-3297, USA, 2002.
- [10] Pacifici, F., "Change Detection Algorithms: State-of-the-Art", 2007 URL: [http://www.disp.uniroma2.it/earth\\_observation/pdf/CD-Algorithms.pdf](http://www.disp.uniroma2.it/earth_observation/pdf/CD-Algorithms.pdf) (Accessed on 4<sup>th</sup> November, 2011).
- [11] Zhu, Z., and Huang, T.S., "Multimodal Surveillance: An Introduction", *IEEE Conference on Computer Vision and Pattern Recognition*, pp. 1-6, USA, June, 2007.
- [12] Radke, R.J., Andra, S., Al-Kofahi, O., and Roysam, B., "Image Change Detection Algorithms: A Systematic Survey", *IEEE Transactions on Image Processing*, Volume 14, No. 3, pp. 294-307, USA, 2005.

- [13] Patra, S., Ghosh, S., and Ghosh, A., "Histogram Thresholding for Unsupervised Change Detection of Remote Sensing Images", *International Journal of Remote Sensing*, Volume 32, No. 21, pp. 6071-6089, England, 2011.
- [14] Coelho, L.P., Shariff, A., and Murphy, R.F., "Nuclear Segmentation in Microscope Cell Images: A Hand-Segmented Dataset and Comparison of Algorithms", *IEEE International Symposium on Biomedical Imaging: From Nano to Macro*, pp. 518-521, USA, June, 2009.
- [15] Gonzales-Barron, U., and Butler, F., "A Comparison of Seven Thresholding Techniques with the k-Means Clustering Algorithm for Measurement of Bread-Crumb Features by Digital Image Analysis", *Journal of Food Engineering*, Volume 74, No. 2, pp. 268-278, England, 2006.
- [16] Minu, S., and Shetty, A., "A Comparative Study of Image Change Detection Algorithms in MATLAB", *Aquatic Procedia*, Volume 4, pp. 1366-1373, 2015.
- [17] Ridler, T.W., and Calvard, S., "Picture Thresholding Using an Iterative Selection Method", *IEEE Transactions Systems Man Cybernetics*, Volume 8, No. 8, pp. 630-632, USA, 1978.
- [18] Rosin, P.L., and Ioannidis, E., "Evaluation of Global Image Thresholding for Change Detection", *Pattern Recognition Letters*, Volume 24, No. 14, pp. 2345-2356, Netherlands, 2003.
- [19] Meyer, B., Mietz, R., and Romer, K., "LoCaF: Detecting Real-World States with Lousy Wireless Cameras", *IEEE 8th International Conference on Distributed Computing in Sensor Systems*, pp. 58-66, China, May, 2012.
- [20] Zarghami, S., "Middleware for Internet of Things", Iran, 2013.
- [21] Maggiani, L., Salvadori, C., Petracca, M., Pagano, P., and Saletti, R., "Reconfigurable Architecture for Computing Histograms in Real-Time Tailored to FPGA-Based Smart Camera", *IEEE 23<sup>rd</sup> International Symposium on Industrial Electronics*, pp. 1042-1046, Turkey, June, 2014.
- [22] Su, C. and Amer, A., "A Real-Time Adaptive Thresholding for Video Change Detection", *IEEE International Conference on Image Processing*, pp. 157-160, Georgia, USA, October, 2006.
- [23] Gargi, U., Kasturi, R., and Strayer, S.H., "Performance Characterization of Video-Shot-Change Detection Methods", *IEEE Transactions on Circuits and Systems for Video Technology*, Volume 10, No. 1, pp.1-13, USA, 2000.
- [24] Aach, T., Kaup, A., and Mester, R., "Statistical Model-Based Change Detection in Moving Video", *Signal Processing*, Volume 31, No. 2, pp. 165-180, Netherlands, 1993.
- [25] Ortego, D., and SanMiguel, J.C., "Stationary Foreground Detection for Video-Surveillance Based on Foreground and Motion History Images", *IEEE 10<sup>th</sup> International Conference on Advanced Video and Signal Based Surveillance*, pp. 75-80, August, China, 2013.
- [26] Pajares, G., Ruz, J.J., and de la Cruz, J.M., "Performance Analysis of Homomorphic Systems for Image Change Detection", *Iberian Conference on Pattern Recognition and Image Analysis*, pp. 563-570, Springer Berlin Heidelberg, Portugal, June, 2005.
- [27] Sigut, J.F., Sidha, S.A.O., Díaz, J., and González, C.S., "A New Set of Features for Robust Change Detection", *VISAPP*, Volume 2, Portugal, 2008.
- [28] Olsever, M., and Ünsalan, C., "Locating the Urban Area in Satellite Images to Detect Changes in Them", *IEEE 6th International Conference on Recent Advances in Space Technologies*, pp. 109-114, Turkey, June, 2013.
- [29] Townsend, J.T., "Theoretical Analysis of an Alphabetic Confusion Matrix", *Perception & Psychophysics*, Volume 9, No. 1, pp. 40-50, USA, 1971.
- [30] Provost, F.J., Fawcett, T., and Kohavi, R., "The Case Against Accuracy Estimation for Comparing Induction Algorithms", *Proceedings of 15<sup>th</sup> International Conference on Machine Learning*, Volume 98, pp. 445-453, USA, July, 1998.

- [31] Labatut, V., and Cherifi, H., "Accuracy Measures for the Comparison of Classifiers", arXiv preprint arXiv: 1207.3790, Cornell University, New York, USA, 2012.
- [32] Otsu, N., "A Threshold Selection Method from Gray-Level Histograms", *Automatica*, Volume 11, pp. 23-27, USA, 1975.
- [33] Kapur, J.N., Sahoo, P.K., and Wong, A.K., "A New Method for Gray-Level Picture Thresholding using the Entropy of the Histogram", *Computer Vision, Graphics, and Image Processing*, Volume 29, No. 3, pp.273-285, 1985.
- [34] Rosin, P.L., "UnimodalThresholding", *Pattern recognition*, Volume 34, No.11, pp. 2083-2096, England, 2001.
- [35] [http://see.xidian.edu.cn/vips1/database\\_Video.html](http://see.xidian.edu.cn/vips1/database_Video.html)