ApproxCT: Approximate Clustering Techniques for Energy Efficient Computer Vision in Cyber-Physical Systems

Raja Haseeb Javed*, Ayesha Siddique*, Rehan Hafiz†, Osman Hasan*, Muhammad Shafique‡
*School of Electrical Engineering and Computer Sciences, National University of Sciences and Technology, Islamabad, Pakistan
†Information Technology University (ITU), Lahore, Pakistan
‡Institute of Computer Engineering, Vienna University of Technology, Vienna, Austria
Email: {14mseehjaved,14mseeasiddique,osman.hasan}@seecs.nust.edu.pk, rehan.hafiz@itu.edu.pk, muhammad.shafique@tuwien.ac.at

***Abstract*—The emerging trends in miniaturization of Internet of Things (IoT) have highly empowered the Cyber-Physical Systems (CPS) for many social applications especially, medical imaging in healthcare. The medical imaging usually involves big data processing and it is expedient to realize its clustering after data acquisition. However, the state-of-the-art clustering techniques are compute intensive and tend to reduce the processing capability of battery-driven or energy harvested IoT based embedded devices (e.g., edge and fogs). Thus, there is a desire to perform energy efficient implementation of the machine learning based clustering techniques. Since, the clustering techniques are inherently resilient to noise and thus, their resilience can be exploited for energy efficiency using approximate computing. In this paper, we proposed approximate versions of the widely used K-Means and Mean Shift clustering techniques using the state-of-the-art low power approximate adders (IMPACT). The trade-off between power consumption and the output quality is exploited using five well-known pattern recognition datasets. The experiments reveal that K-Means algorithm exhibits more error resilience towards approximation with a maximum of 10% - 25% power savings.**

***Index Terms*—Cyber-Physical Systems, Internet of Things, Approximate Computing, Clustering, Low Power Approximate Adders, Energy Consumption, Computer Vision.**

I. Introduction

Cyber-Physical Systems (CPS) are the key enablers for the future trends in computer vision due to their exquisite capability of interacting with the physical world through communication, control and computation strategies [1]. The smart homes, transportation and healthcare are some archetypal CPS applications which envision incorporating the smart digital forensics, video surveillance and medical imaging. Such broad applicability of the computer vision envisages an immense expansion of Internet of Things (IoT) or CPS devices and an unpredictable increase in the associated data [2]. According to Gartner's survey [3], the number of such interconnected heterogeneous devices is expected to increase around 75 billion by 2025, which will eventually result into (around) 160 zettabytes of processing data. To handle such huge data, the clustering based machine learning has recently emerged as the de-facto analysis tool [4]. Besides reducing the data volume, it is very helpful in image segmentation for foreground extraction and object recognition. Fig. 1 illustrates the clustering in Magnetic Resonance Imaging (MRI) for dimensionality reduction and an efficient regional grading of gliomas (brain tumors).

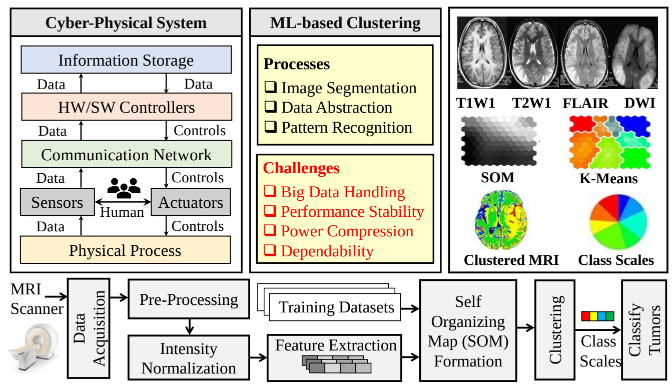

Fig. 1: The Tasks and Challenges in Image Clustering based Machine Learning (ML), using Embedded IoT Devices, for Data Abstraction and Regional Grading of Gliomas (Brain Tumor) in MRI.

Although, the data clustering capitalizes the characteristics of machine learning in IoT based CPS for big data analytics but the recent trend of high image resolution makes the compute-intensive clustering techniques impractical for embedded real-time applications [5]. The CPS devices (e.g edge and fogs) are either battery operated or employ energy-harvesting. They do not possess sufficient computational capabilities for machine learning and high resolution image processing. Thus, it is an open research challenge to develop an efficient clustering technique under low power envelope.

A. State-of-the-Art and Open Research Problems

The big data clustering is a simple data abstraction technique which maps the input data into different clusters having more intra-cluster and less inter-cluster similarity [10]. It models the image segmentation as pattern recognition by exploiting the relationship between un-labeled data points and heuristically searching for some interesting features in the

TABLE I: Feature Comparison with State-of-the-Art

Related Work	Clustering techniques				Energy Efficient Techniques				Target Parameters			Testing Dataset
	Mean Shift	Gaussian Mixture Models	Regression	K-Means	Others	Approximate Computing: Adders	Multipliers	Mapping	Power	Accuracy	Delay	
[5]	✓				✓				✓		✓	Standard Images
[6]	✓				✓						✓	Segmentation Evaluation Dataset
[7]				✓	✓						✓	USGS, CSN, BIGX, SONG, KDD, WEB
[8]		✓	✓	✓				✓	✓	✓		3cluster, 3d3cluster, 4cluster, HangSeng INDEX, NASDAQ, S&P 500
[9]				✓			✓		✓	✓	✓	Fisher Iris
ApproxCT	✓			✓		✓			✓	✓		Standard Images

dataset. The widely used initial seed (number of clusters) sensitive K-Means and less sensitive Mean Shift algorithms exhibit the best image segmentation quality as compared to other state-of-the-art clustering techniques (e.g., Fuzzy C-Means and Gaussian Mixture Model) [11] [12]. However, their exhaustive learning and iterative behavior impel these algorithm to occupy a larger portion of FPGA and ASIC and drain the battery subsequently. In the past, many efforts have been made for accelerating the hardware designs but they contribute to meager energy gains [5] [6] [7]. Towards this direction, approximate computing has recently emerged as a promising approach for energy efficiency especially, in combination with other state-of-the-art low-power techniques [13]. It simplifies the complex hardware of image processing by relaxing the equivalence margin between specification and implementation. V. K. Chippa et al. exploited the inherent error resilience in clustering techniques [14] and proposed to use approximate computing in image clustering. In [15], an energy efficient framework for all kinds of error-tolerant iterative methods is presented and validated through K-means clustering. However, Q. Zhang et. al. claimed that it is quite impossible to have such generalized framework [8]. The author presented approximate computing as an application-specific energy saving strategy and developed approximate Gaussian Mixture (GMM) model using reconfigurable approximate adders [16]. However, it is quite hard to initialize the clusters and segment the high dimensional data using GMM clustering. Recently, many inexact arithmetic units [17] [9] have been designed which can also be utilized for energy efficiency with slight accuracy loss. I. Alouani et al. designed a low-power approximate multiplier and used K-Means clustering as a case study for validating its performance [9]. However, it is expedient to realize more approximate computing in K-Means clustering for an adequate on-board energy efficiency and compare its performance with a superseding automated clustering, i-e., Mean Shift. Table I highlights the comparison between our proposed approach and prior state-of-the-art.

B. Motivational Analysis

The clustering techniques usually employ certain arithmetic units (e.g., adders) to calculate Euclidean distance, mean, etc. These additions increases with an increase in the number of image pixels and clusters. This precept study infers larger area over head and an exponential increase in the power

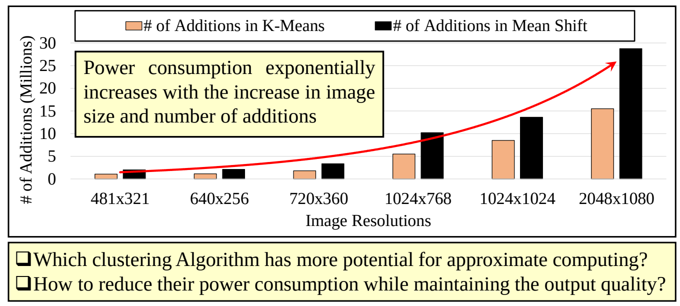

Fig. 2: Motivational Analysis to show the Exponential increase in Power Consumption with respect to Number of Additions and Image Resolution in K-Means and Mean Shift Algorithms.

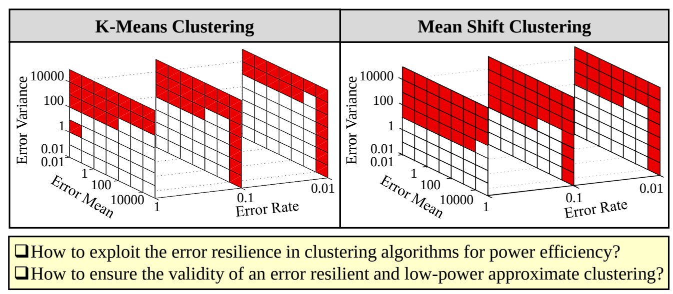

Fig. 3: Error Resilience Analysis of K-Means and Mean Shift Algorithms using Mean Centroid Distance as a Quality Metric [14]. The white spaces in the slice plot shows that algorithms are tolerant to the errors if the error rate is less than 1.

consumption (see Fig. 2). Hence, an energy efficient algorithm for low-power ASIC implementation of clustering technique is required. As illustrated in Fig. 3, *K-Means and Mean Shift algorithms inherit considerable error tolerance towards approximation. So, approximate computing is an optimal solution for substantial power savings in image segmentation based pattern recognition.*

C. Research Challenges

The main research challenges in developing an approximate computing based image clustering technique, for pattern recognition, in IoT based CPS are:

1) **Error Margin Exploration:** Approximate computing is an error inducing practice. So, the first question arises that how much error resilience exist in an approximate algorithm?

2) **Accurate Pattern Recognition:** How to exploit the trade-off between output quality and power consumption and ensure the validity of an approximate clustering technique (ApproxCT) in pattern recognition?
3) **Knob Controllability:** Which context aware knobs can be utilized in developing the low-power hardware for an image clustering based pattern recognition?

D. Novel Contributions

This paper makes the following novel contributions:

1) A **Generic Methodology** for ApproxCT with design space exploration and thus, enabling its multiple energy-aware architectural versions.
2) A **Comprehensive Performance Analysis** of ApproxCT framework, employing approximate K-Means and Mean Shift clustering, using context-aware knobs such as quality and power.
3) An **ASIC Implementation** of ApproxCT framework by employing accurate and low-power approximate adders (IMPACT).
4) A **Comparative Analysis** of the initial seed dependent and independent clustering techniques.

In this paper, we exploited approximate computing in the clustering techniques, i.e., K-Means and Mean Shift, using an open-source IMPACT adders' library [17], and comprehensively analyzed their performance on the basis of multiple approximation knobs,like quality and power savings.

E. Paper Organization

The approximate computing based K-Means and Mean Shift algorithms are discussed in Section II. Section III presents our proposed methodology for ApproxCT modeling. Section IV presents its experimental setup and tool flow. It also discusses the performance evaluation of ApproxCT and provides a comprehensive comparative analysis of the approximate and accurate versions of K-Means and Mean Shift algorithms. Finally, Section V concludes the paper.

II. Approximate Image Clustering

Approximate computing is an emerging paradigm which leverages the intrinsic error resilience of an application to find a trade-off between energy efficiency and quality [18] [19]. At the hardware level, the approximate computing can be explored in K-Means and Mean Shift algorithms by replacing the accurate adders with their approximate counterparts.

A. Approximate K-Means Clustering

Like the traditional K-Means clustering method [20], the approximate K-Means algorithm also refers to the most simplest unsupervised machine learning algorithm which uses Euclidean, Manhattan, City Block or Chess Board distance to gather the input samples into multiple asymmetric clusters (see Fig. 4). However, the Euclidean distance is the best possible solution for minimizing the intra-cluster variance. The summary of this iterative approximate clustering is provided below:

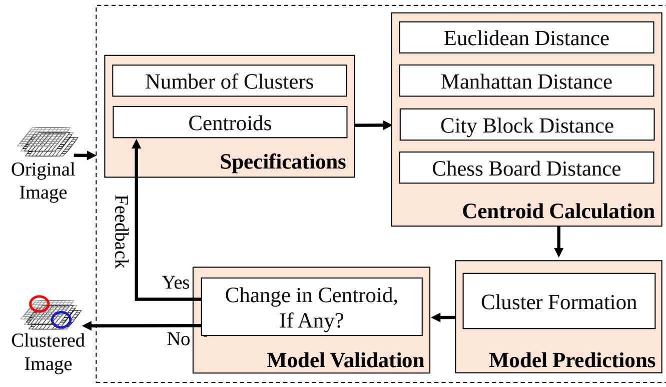

Fig. 4: The Initial Seed Dependent K-Means Algorithm.

1) In the first iteration (t = 1), the user specifies k number of clusters and randomly defines their corresponding centroids 'm' as a set '**A**'.

$$\mathbf{A} = \{ \mathrm{m}_1^{(t)},\ \mathrm{m}_2^{(t)},\ \ldots ,\mathrm{m}_k^{(t)} \} \tag{1}$$

2) Then, the distance between p^{th} sample 'x' and the given i^{th} centroid is calculated. The sample having minimum distance to the given centroid is assigned to the i^{th} cluster 'C'.

$$\mathrm{C}^{(t)} = \{ \mathrm{x_p} : ||\ \mathrm{x_p} \ominus \mathrm{m}_i^{(t)}\ ||^2 \leq ||\ \mathrm{x_p} \ominus \mathrm{m}_j^{(t)}\ ||^2, \forall\, j \} \tag{2}$$

where $\ominus$ represents the signed approximate addition.

3) In the third step, the centroids of the clusters '$\mathrm{S_i}$' are updated as:

$$\mathrm{m}_i^{(t+1)} = \frac{1}{|\ \mathrm{C}_i^{(t)}\ |} \sum_{\mathrm{x_i} \in \mathrm{C}_i^{(t)}} (\ \mathrm{x_1} \oplus \mathrm{x_2} \oplus \ldots \oplus \mathrm{x_n}\ ) \tag{3}$$

where $|\ \mathrm{S}_i^t\ |$ is the number of samples 'x in cluster 'S'.

4) Finally, the standard squared error criteria is used for the convergence [21]. If the error lies below certain threshold (such that the centroids are shifted) then, the algorithm repeat itself from step 2.

$$||\ \mathrm{m}_i^{(t)} - \mathrm{m}_i^{(t-1)}\ ||_2 < \epsilon, \forall\, i \tag{4}$$

B. Approximate Mean Shift Clustering

Since, K-Means algorithm is very sensitive to the seeding value (i.e., k) so, a non-parametric solution, without a-prior knowledge of the input, is required (see Fig. 5). The Mean Shift algorithm is labeled as un-supervised and thus, can be used in autonomous applications. It works on the principle of 'window based mean centroid clustering' as a gradient ascent (on the density function). The summary of the approximate Mean shift clustering method is given below:

1) In the first iteration (t = 1), the user specifies the size of the window. Let, the window 'W' is initialized with n number of samples.

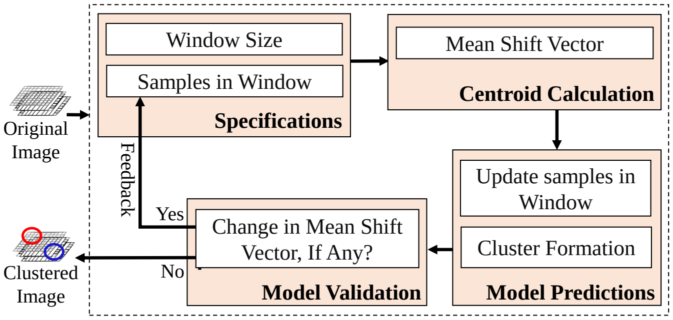

Fig. 5: An Overview of A-priori Mean Shift Algorithm.

$$w = \{\ x_1^{(1)},\ x_2^{(1)},\ \dots,x_n^{(1)}\ \} \tag{5}$$

2) Next, the mean shift vector $m(w)$ is calculated as:

$$m(w) = \frac{\sum_{i=1}^{n} g\ (\ d^2\ (\ w^{(t)},\ x_n,\ H\ )\ )\ x_i\ )}{\sum_{i=1}^{n} g\ (\ d^2\ (\ w^{(t)},\ x_n,\ H\ )\ )} - x^t \tag{6}$$

where $\sum$ and $\ominus$ indicates approximate addition, In addition, 'd' and and 'g' refers to the Mahalanobis distance and a weight function derived from the Epanechnikov kernel [22].

3) Now, the window '$w^{(t)}$' is updated to '$w^{(t+1)}$' with reference to above written $m(w)$.
4) Repeat the second step until the following condition is true for a certain threshold.

$$d\ (\ x^{(t)},\ x^{(t+1)},\ H\ ) < \epsilon \tag{7}$$

III. Proposed Methodology for ApproxCT

Fig. 6 illustrates the proposed methodology for ApproxCT which consists of following three key steps:

A. Error Resilience Analysis

As approximate computing itself adds some imprecision to the algorithms so, it is quite important to pre-evaluate the resilience of the algorithmic computations for any random noise. This analysis can be done by using the Application Resilience Characterization Framework [14]. It partitions the algorithm into error resilient and sensitive parts and identifies the potential error resilient design parts for applying approximate computing.

B. Fixed Point Hardware Approximation

Since, the embedded systems, especially ASICs and FPGA, do not support the floating point computations so, the foremost step in designing an approximate clustering technique is *fixed point quantization*. The algorithm is exhaustively tested for different fixed point word lengths until the outcome of fixed point implementation matches to that of the floating point algorithm. The matching word length is used as a required bit width for the approximate adders.

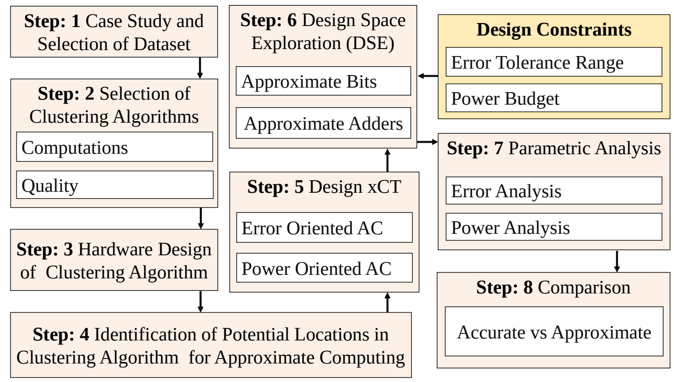

Fig. 6: The Proposed Methodology for Approximate Clustering Techniques.

C. Approximate Adders based Hardware Approximation

For the simple approximate hardware implementation, the approximate adders serve as the best optimal solution. Each IMPACT adder posses unique performance characteristics and exhibits different error probabilities with varying number of approximate bits [23]. So, the accuracies for all the possible adders and bits configurations are obtained by exhaustively evaluating all possibilities. Here, the number of approximate bits are set as the percentage of the total number of bits and initialized in a configuration array, which is given as an input to the approximate adders library.

IV. Results and Discussions

A. Experimental Tool Flow

For simulations and hardware designs, we used an Intel Core i7-6700T Quad-Core server operating at 3.06 GHz with 32 GB RAM. Fig. 7 presents our integrated tool flow. The precision of the approximate clustering framework is first evaluated for an application using MATLAB and then, its corresponding behavioral design is developed in ModelSim. The synthesis of the Verilog code generates a signal activity file (.saif) which is fed into the Synopsys Primetime Design Compiler for generating the power reports.

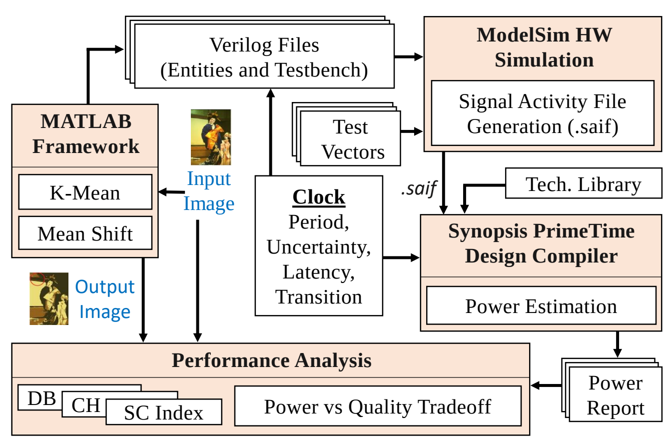

Fig. 7: Experimental Tool Flow of Approximate Clustering in Image Segmentation.

B. Performance Measures

To evaluate the performance of the K-Means and Mean Shift clustering methods we use the following three indices:

1) Silhouette Coefficient: If $\mathrm{d_i}$ is the average distance of a sample to other samples in the same cluster and $\mathrm{d_i^{'}}$ corresponds to the average distance of a sample to other samples in its nearest neighboring cluster (see Fig. 8) then, the Silhouette Coefficient (SC) can be defined as:

$$\mathrm{s_i} = \frac{\mathrm{d_i^{'}} - \mathrm{d_i}}{max\ [\ \mathrm{d_i},\ \mathrm{d_i^{'}}\ ]} \tag{8}$$

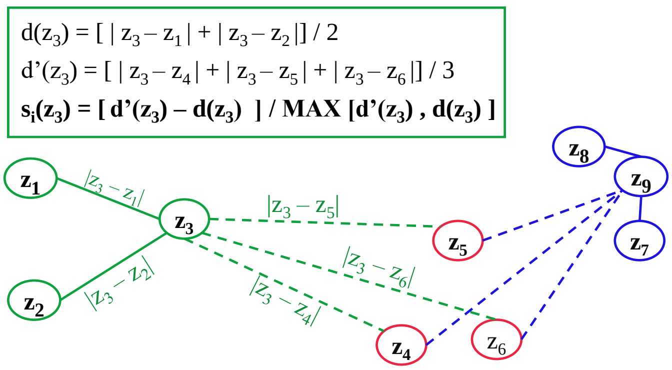

Fig. 8: An Example for Computing the Silhouette Coefficient.

The quality of clustering is determined by the average $\mathrm{s_i}$ across all samples. The range of mean SC varies from -1 to 1, where $SC \approx 1$ and $SC \approx -1$ depicts the good and poor quality of the clustering result, respectively [24].

2) Davies Bouldin Index: The Davies Bouldin (DB) index is the ratio of the sum of within-cluster scatter to between-cluster separation [24]. Let, the scatter within the $\mathrm{i^{th}}$ cluster 'C' is defined as:

$$\mathrm{S_i} = \frac{\sum_{\mathrm{x}\epsilon\mathrm{C_i}} ||\ \mathrm{x} - \mathrm{m_i}\ ||}{|C_i|} \tag{9}$$

and the distance between cluster C_i and C_j, denoted by d_{ij}, is expressed as:

$$\mathrm{d_{ij}} = ||\ \mathrm{m_i} - \mathrm{m_j}\ || \tag{10}$$

where $\mathrm{m_i}$ and $\mathrm{m_j}$ represents the mean centroids of $\mathrm{i^{th}}$ and $\mathrm{j^{th}}$ clusters respectively. Then, the DB is defined as:

$$\mathrm{DB} = \frac{\sum_{i=1}^{K} \mathrm{R_{i,(t)}}}{\mathrm{K}} \tag{11}$$

where

$$\mathrm{R_{i,(t)}} = \mathrm{max_{j,j\neq i}}(\ \frac{\mathrm{S_{i,q}} + \mathrm{S_{j,q}}}{\mathrm{d_{ij,t}}}\ ) \tag{12}$$

Smaller the value of DB index, more better would be the quality of the clustering methods.

3) Calinski Harabasz Index: For n number of samples 'x' and k clusters, the Calinski Harabasz (CH) index [24] is written as:

$$\mathrm{CH} = (\frac{\mathrm{trace\ B}}{\mathrm{k\ -\ 1}})\ /\ (\frac{\mathrm{trace\ W}}{\mathrm{n\ -\ k}}) \tag{13}$$

where B and W represent between and within-cluster scatter, respectively. The trace B and trace W can be calculated using the following equations:

$$\mathrm{trace\ B} = \sum_{\mathrm{i=1}}^{\mathrm{k}} \mathrm{n_i}\ ||\ \mathrm{m_i}\ -\ \mathrm{M}\ ||^2 \tag{14}$$

where M is the centroid of the entire dataset.

$$\mathrm{trace\ W} = \sum_{\mathrm{i=1}}^{\mathrm{k}} \sum_{\mathrm{j=1}}^{\mathrm{n}} \mathrm{k}\ ||\ \mathrm{x_i}\ -\ \mathrm{m_k}\ ||^2 \tag{15}$$

Higher the CH index, more better is the clustering result.

C. Selection of Dataset

For demonstrating the effect of approximate K-Means and Mean Shift Algorithms in image processing applications, we randomly selected fifteen 481 x 321 images (see Fig. 9) from the '$\mathrm{D_1}$' Berkeley Segmentation database [25] and used '$\mathrm{D_2}$' Segmentation Evaluation [26], '$\mathrm{D_3}$' Stanford background [27], '$\mathrm{D_4}$' INIRA Holidays and '$\mathrm{D_5}$' INIRA Copydays [28] datasets for extensive testing.

D. Performance Evaluation

1) Accuracy Analysis: For the qualitative and visual comparison of approximate K-Means and Mean-Shift algorithms, the proposed model was tested to group the image pixels using their density profile. The accuracy of this clustering approach is calculated using a statistical error analysis approach [23]. Lets consider the left most image in Fig 9 which consists of a lady and a child. Since, the number of K-Means clusters are limited to the choice of the user so, it is fixed here (e.g., k = 3). However, the Mean shift algorithm itself divided the image into 5 clusters. Fig. 10 shows that there is no reasonable change in K-Means clustered images, in-spite of the imprecise computations, while the affected region in Mean shift clustered images are marked by red circles. The marked area shows that the data points are distributed differently as compared to the reference image. This analysis depicts that Mean shift does not perform good at lower accuracies.

However, we cannot conclusively judge the quality of the clustering results of the images in this way. For the quantitative analysis, we exhaustively tested the approximate clustering design, with different configurations of IMPACT adders and the approximate bits, while sweeping the accuracy from 75% to nearly 12%. In Fig.11, the average result (of fifteen images in Fig. 9) is plotted for the SC as well as DB and CH indices. A brief summary of this analysis is given below:

1) The accurate as well as approximate K-Means clustering technique has SC index very close to 1 as compared

Fig. 9: Testing Sample Images (Resolution: 481 x321) from Berkley Segmentation Dataset.

Fig. 10: The Qualitative Comparison of K-Means and Mean Shift Algorithms for multiple Accuracies Ranging from 12.5 % to 100 %.

to the Mean Shift segmentation. This confirms the best clustering performance of approximate K-Means clustering model.

2) The DB index is smaller for K-Means as compared to Mean shift clustering in case of accurate as well as inaccurate additions.
3) Similarly, the objective function for CH index is maximized (low CH index) in case of K-Means which also supports our visual claim.

For K-Means, the variation in the results is 4.73 $\pm$ 3.23, 4.27 $\pm$ 3.63 and 7.31 $\pm$ 6.50 for SC, DB index and CH index, respectively. Likewise, the Mean Shift clustering has 10.816 $\pm$ 3.34, 19.91 $\pm$ 6.36 and 9.73 $\pm$ 3.62 variations in the results of SC index, DB index and CH index, respectively. It can be concluded from these observations that the variation in the results of K-Means clustering is less for all of the three indices as compared to Mean shift. It means that K-Means is more resilient towards approximation. These results are further consolidated by calculating the average of results for 15 images. In this case, K-Means undergoes 3.96 $\pm$ 1.74, 6.02 $\pm$ 2.01 and 9.69 $\pm$ 3.64 variations for SC, DB index and CH index respectively. Similarly, the variation in the Mean Shift results of SC, DB index and CH index is 9.78 $\pm$ 4.12, 14.2 $\pm$ 4.62 and 10.55 $\pm$ 4.14 respectively.

2) Power Analysis: For the extensive testing of the proposed approximate clustering strategy, we used fifteen scenic images from each selected dataset and analyzed the results at 25%, 20%, 15% and 10% power savings by choosing different combinations of the IMPACT adders and approximate bits. In case of the K-Means algorithm, it is evident from Fig. 12 that the SC, CH index, DB index indicate poor quality with 25% power saving. However, it is much better at 10% power saving. So, it is evident that the results are compromised a bit with approximation in K-Means algorithm but higher power savings are guaranteed on contrary to Mean-Shift clustering method. In other words, this trade-off between power savings and accuracy can be used in accordance with the user specifications.

V. Conclusion

In this paper, we proposed approximate K-Means and Mean Shift clustering algorithms to synthesize power efficient image processing applications. For demonstration, the open-source IMPACT adders library is used instead of accurate adders in the hardware design and the performance is evaluated for pixel clustering, using density profile, in five widely adopted image databases. From the results, it can be concluded that *both approximate K-Means and Mean Shift algorithms have acceptable clustering quality with large power savings, (25%, 20%, 15% and 10%), despite of the design inaccuracies. However, the approximate K-Means clustering experiences comparatively less variations in the image quality. Therefore, it is more advantageous to exploit approximate computing in K-Means clustering with an appropriate initial seed.*

References

[1] M. Shafique, F. Khalid, and S. Rehman, "Intelligent security measures for smart cyber physical systems," in *21st Euromicro Conference on Digital System Design (DSD)*. IEEE, 2018, pp. 280–287.

[2] Q. Xu, K. M. M. Aung, Y. Zhu, and K. L. Yong, "Building a large-scale object-based active storage platform for data analytics in the internet of things," *Journal of Supercomputing*, vol. 72, pp. 2796–2814, 2016.

[3] G. Inc., "Gartner-Survey," 2018. [Online]. Available: https://www.gartner.com/newsroom/id/2636073

[4] R. Elshawi, S. Sakr, D. Talia, and P. Trunfio, "Big data systems meet machine learning challenges: Towards big data science as a service," *Big Data Research*, 2018.

[5] S. Craciun, R. Kirchgessner, A. D. George, H. Lam, and J. C. Principe, "A real-time, power-efficient architecture for mean-shift image segmentation," *Journal of Real-Time Image Processing*, vol. 14, pp. 379–394, 2018.

[6] A. Tehreem, S. G. Khawaja, A. M. Khan, M. U. Akram, and S. A. Khan, "Multiprocessor architecture for real-time applications using mean shift clustering," *Journal of Real-Time Image Processing*, pp. 1–14, 2017.

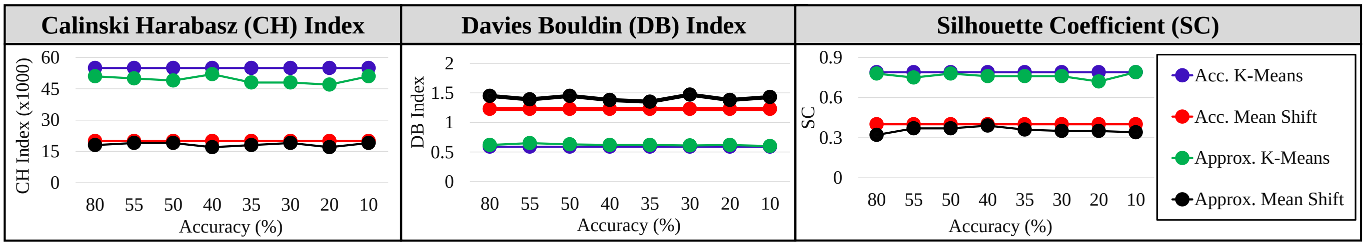

Fig. 11: Comparative Analysis of Accurate (Acc.) and Approximate (Approx.) K-Means and Mean Shift Algorithms based Image Segmentation with respect to performance metrices, i.e., Average Calinski-Harabasz(CH) Index, Davies Bouldin (DB) Index and Silhouette coefficient (SC).

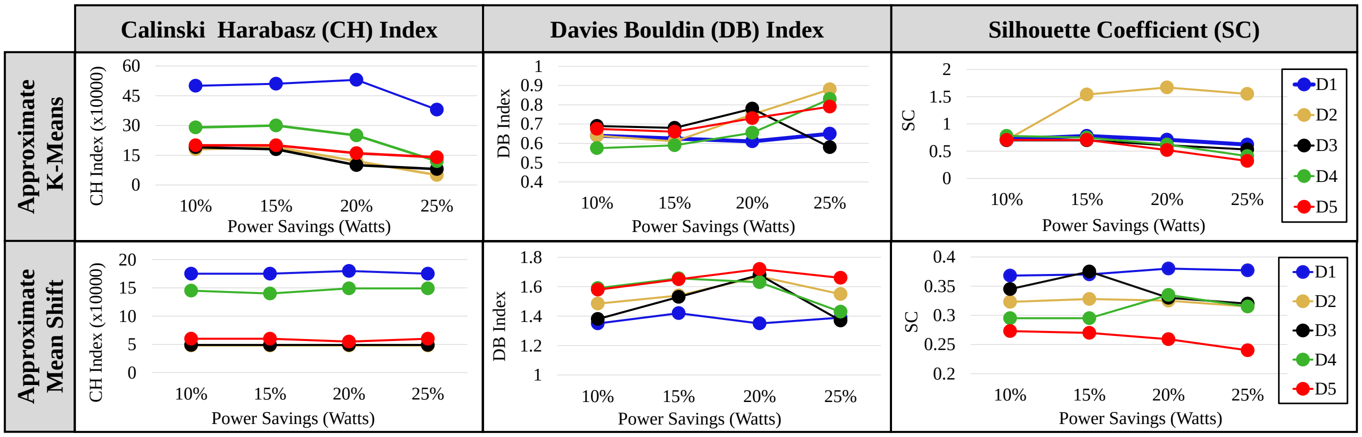

Fig. 12: Power and Quality Analysis of Approximate K-Means and Mean Shift Algorithms using Five Different Image Datasets.

[7] O. Bachem, M. Lucic, S. H. Hassani, and A. Krause, "Approximate k-means++ in sublinear time," in *Proceedings of the Thirtieth AAAI Conference on Artificial Intelligence*, 2016, pp. 1459–1467.

[8] Q. Zhang, F. Yuan, R. Ye, and Q. Xu, "Approxit: An approximate computing framework for iterative methods," in *Proceedings of the 51st Annual Design Automation Conference*. ACM, 2014, pp. 1–6.

[9] I. Alouani, H. Ahangari, O. Ozturk, and S. Niar, "A novel heterogeneous approximate multiplier for low power and high performance," *Embedded Systems Letters*, vol. 10, pp. 45–48, 2018.

[10] K. Chowdhury, D. Chaudhuri, and A. K. Pal, "Seed point selection algorithm in clustering of image data," in *Progress in Intelligent Computing Techniques: Theory, Practice, and Applications*. Springer, 2018, pp. 119–126.

[11] O. Vincent, A. Makinde, O. Salako, and O. Oluwafemi, "A self-adaptive k-means classifier for business incentive in a fashion design environment," *Applied computing and informatics*, vol. 14, pp. 88–97, 2018.

[12] Y. Guo, A. Şengür, Y. Akbulut, and A. Shipley, "An effective color image segmentation approach using neutrosophic adaptive mean shift clustering," *Measurement*, vol. 119, pp. 28–40, 2018.

[13] F. Samie, L. Bauer, and J. Henkel, "Iot technologies for embedded computing: A survey," in *Proceedings of the Eleventh International Conference on Hardware/Software Codesign and System Synthesis*. ACM, 2016, p. 8.

[14] V. K. Chippa, S. T. Chakradhar, K. Roy, and A. Raghunathan, "Analysis and characterization of inherent application resilience for approximate computing," in *Proceedings of the 50th Annual Design Automation Conference*. ACM, 2013, p. 113.

[15] V. K. Chippa, K. Roy, S. T. Chakradhar, and A. Raghunathan, "Managing the quality vs. efficiency trade-off using dynamic effort scaling," *Transactions on Embedded Computing Systems (TECS)*, vol. 12, p. 90, 2013.

[16] R. Ye, T. Wang, F. Yuan, R. Kumar, and Q. Xu, "On reconfiguration-oriented approximate adder design and its application," in *International Conference on Computer-Aided Design (ICCAD)*. IEEE, 2013, pp. 48–54.

[17] V. Gupta, D. Mohapatra, S. P. Park, A. Raghunathan, and K. Roy, "Impact: imprecise adders for low-power approximate computing," in *Proceedings of the 17th International Symposium on Low-Power Electronics and Design*. IEEE, 2011, pp. 409–414.

[18] M. Shafique, R. Hafiz, S. Rehman, W. El-Harouni, and J. Henkel, "Cross-layer approximate computing: From logic to architectures," in *Proceedings of the 53rd Annual Design Automation Conference*. ACM, 2016, p. 99.

[19] A. Siddique, O. Hasan, F. Khalid, and M. Shafique, "Approxcs: Near-sensor approximate compressed sensing for iot-healthcare systems," *arXiv preprint arXiv:1811.07330*, 2018.

[20] D. Napoleon and S. Pavalakodi, "A new method for dimensionality reduction using k-means clustering algorithm for high dimensional data set," *International Journal of Computer Applications*, vol. 13, pp. 41–46, 2011.

[21] H. Yu, G. Wen, J. Gan, W. Zheng, and C. Lei, "Self-paced learning for k-means clustering algorithm," *Pattern Recognition Letters*, 2018.

[22] T. Li, "Contributions to mean shift filtering and segmentation: Application to mri ischemic data," Ph.D. dissertation, INSA de Lyon, 2012.

[23] M. K. Ayub, O. Hasan, and M. Shafique, "Statistical error analysis for low power approximate adders," in *Proceedings of the 54th Annual Design Automation Conference*. ACM, 2017, p. 75.

[24] K. Chowdhury, D. Chaudhuri, and A. K. Pal, "A novel objective function based clustering with optimal number of clusters," in *Methodologies and Application Issues of Contemporary Computing Framework*. Springer, 2018, pp. 23–32.

[25] D. Martin, C. Fowlkes, D. Tal, and J. Malik, "A database of human segmented natural images and its application to evaluating segmentation algorithms and measuring ecological statistics," in *Eighth International Conference on Computer Vision*, vol. 2. IEEE, 2001, pp. 416–423.

[26] S. Alpert, M. Galun, A. Brandt, and R. Basri, "Image segmentation by probabilistic bottom-up aggregation and cue integration," *Transactions on Pattern Analysis and Machine Intelligence*, vol. 34, pp. 315–327, 2012.

[27] S. Gould, R. Fulton, and D. Koller, "Decomposing a scene into geometric and semantically consistent regions," in *12th International Conference on Computer Vision*. IEEE, 2009, pp. 1–8.

[28] H. Jégou, M. Douze, and C. Schmid, "Hamming embedding and weak geometry consistency for large scale image search-extended version," 2008.