International Journal of Computational Methods and Experimental Measurements
https://www.acadlore.com/journals/IJCMEM

Adaptive Quality-Energy Trade-Offs in Image Processing Through Statistical Priority Classification and Variable-Approximate Computing

Ibrahim Haddadi*

Department of Computer Engineering, Collage of Computer Science and Engineering, Taibah University Janadah, 42353 Madinah, Saudi Arabia

* Correspondence: Ibrahim Haddadi (iHaddadi@taibahu.edu.sa)

Received: 11-03-2025 **Revised:** 12-10-2025 **Accepted:** 12-17-2025

Citation: I. Haddadi, "Adaptive quality-energy trade-offs in image processing through statistical priority classification and variable-approximate computing," *Int. J. Comput. Methods Exp. Meas.*, vol. 13, no. 4, pp. 785–801, 2025. https://doi.org/10.56578/ijcmem130404.

Abstract: Modern image processing systems deployed on embedded and heterogeneous platforms face increasing pressure to deliver high performance under strict energy and real-time constraints. The rapid growth in image resolution and frame rates has significantly amplified computational demand, making uniform full-precision processing increasingly inefficient. This paper presents a significance-driven adaptive approximate computing framework that reduces energy consumption by tailoring computational precision and resource allocation to the spatial importance of image content. We introduce a statistical importance metric that captures local structural variability using low-complexity deviation-based analysis on luminance information. The metric serves as a lightweight proxy for identifying regions that are more sensitive to approximation errors, enabling differentiated processing without the overhead of semantic or perceptual saliency models. Based on this importance classification, the proposed framework dynamically orchestrates heterogeneous CPU–GPU resources, applies variable kernel sizes, and exploits dynamic voltage and frequency scaling (DVFS) to reclaim timing slack for additional energy savings. The framework is validated through two complementary case studies: (i) a heterogeneous software implementation for adaptive convolution filtering on an Odroid XU-4 embedded platform, and (ii) a hardware-level approximate circuit allocation approach using configurable-precision arithmetic units. Experimental results demonstrate energy reductions of up to 60% compared to uniform-precision baselines, while maintaining acceptable visual quality. Image quality is evaluated using both PSNR and the perceptually motivated SSIM metric, confirming that the proposed approach preserves structural fidelity despite aggressive approximation.

Keywords: Frequency scaling; Parallel computing; Hard-ware/software co-optimization; Approximate computing

1 Introduction

Digital imagery systems that capture, manipulate, and interpret real-world visual information are now widely deployed across embedded computing platforms and pervasive technological infrastructures, including mobile devices, edge processors, and real-time vision systems. These systems face two fundamental and often conflicting challenges related to computational performance and energy efficiency. The first challenge arises from continuous advances in imaging sensor technology, which impose sustained pressure toward higher pixel densities and increased frame rates that must be processed within strict temporal constraints [1]. Consequently, the volume of visual data requiring processing within fixed time intervals grows rapidly, as illustrated in Table 1. The second challenge stems from the need to provide sufficient computational capacity to handle this growing data volume, rendering energy-efficient operation increasingly difficult—particularly in systems whose hardware and software architectures are already optimized for low-power execution.

1.1 Prior Research and Related Contributions

Table 1 summarizes representative data throughput rates, storage requirements, and daily energy consumption corresponding to commonly used image resolution formats. These benchmark measurements were obtained on an Odroid-XU4 embedded computing platform operating at 2.0 GHz across multiple resolution configurations. The

results demonstrate that both storage demand and energy expenditure increase significantly as image resolution scales, particularly when processing is performed continuously over extended duty cycles.

Table 1. Transmission rates, storage capacity, and daily energy utilization across various image resolutions

Resolution	Pixels (Kpx)	30fps			60fps		
		Rate (Mbps)	Cap. (TB)	Pwr (MJ)	Rate (Mbps)	Cap. (TB)	Pwr (MJ)
640 × 480	307	74	0.8	0.3	147	1.6	0.6
1024 × 768	786	189	2.0	0.7	377	4.1	1.5
1600 × 900	1440	346	3.7	1.3	691	7.5	2.8
2048 × 1152	2359	566	6.1	2.2	1132	12.2	4.6
4320 × 2432	10506	2521	27.2	9.8	5043	54.5	20.6
8192 × 4608	37749	9060	97.8	35.3	18119	195.7	73.9

Over recent decades, extensive research efforts have investigated power-efficient strategies for time-constrained image processing through a variety of methodological frameworks. Among these, approximate computing has emerged as a particularly effective paradigm by exploiting the inherent tolerance of many image processing applications to controlled computational inaccuracies [2]. Approximate computing has been explored across multiple abstraction layers, including arithmetic circuit design, compiler support, runtime systems, and architectural optimization. Libraries of approximate arithmetic components have been proposed to facilitate design-space exploration and benchmarking [3], while pattern-based and quality-configurable approximation techniques have demonstrated effectiveness for data-parallel workloads [4, 5]. Compiler- and language-level support for variable-accuracy execution has further enabled automated approximation tuning in performance-critical applications [6]. Comprehensive surveys have summarized approximation strategies spanning logic, architecture, and system levels [7, 8]. Power reduction is achieved by replacing computationally expensive hardware and software implementations with reduced-complexity alternatives, enabling faster execution with lower energy consumption [9]. This relaxation of accuracy, however, introduces degradation in output quality, which must remain within acceptable limits defined by subjective perception or quantitative fidelity metrics.

At the hardware level, approximation techniques commonly employ deliberate architectural simplifications in image processing components, reducing gate count and circuit complexity. A prominent class of such approaches focuses on simplifying arithmetic units, particularly multipliers. For example, multiplier-less image processing architectures have been proposed using probabilistic domain transformations for convolution operations 4or shift-and-add techniques for Fast Fourier Transform (FFT) implementations [10]. These methods demonstrate that significant energy savings can be achieved while preserving functionally acceptable results.

Power reduction is accomplished by substituting computationally demanding hardware and software implementations with reduced-complexity alternatives, thereby enabling accelerated execution with diminished energy expenditure [9]. Nevertheless, this approach introduces degradation in output quality to levels considered tolerable according to either subjective perceptual assessments or quantitative fidelity metrics.

Hardware-level approximation techniques employ deliberate architectural simplification in image processing components, achieving reduced gate counts in their implementation. The defining characteristic of such approaches lies in diminishing the structural complexity of arithmetic computational units. For example, multiplier-less image processing units are proposed by the following studies [10, 11]. Replacing multipliers by probabilistic domain transformation in a discrete convolution [10] or by shift-and-add operations in a 16-point Fast Fourier Transformer (FFT) [11] provides similar functionality at reduced complexity and energy consumption.

1.2 Rationale and Contributions

Although approximation has traditionally been applied using application-agnostic strategies, such approaches do not fully exploit domain-specific characteristics that could enable further energy savings while maintaining higher output quality. Image processing represents a class of applications where deeper application-level knowledge can be leveraged to guide approximation decisions more effectively [12]. Visual imagery inherently contains spatial regions that differ in informational relevance. Areas with pronounced inter-pixel intensity variations—such as edges, contours, and textured structures—tend to contribute more strongly to perceptual and structural significance than spatially homogeneous regions [13]. The perceptual salience of such regions has been extensively validated through subjective vision studies [14]. Conversely, smooth or low-variance regions generally exhibit greater tolerance to approximation-induced distortion.

Motivated by this observation, we characterize local pixel-level variability as statistical importance and posit that this property can be exploited to guide adaptive computation. Rather than attempting to model semantic or perceptual saliency—which typically requires computationally intensive feature extraction or learning-based methods—statistical importance provides a low-complexity, structure-oriented proxy suitable for real-time and

energy-constrained systems. While biologically inspired and computational saliency models have been widely studied, they often require complex feature extraction and learning-based inference, making them less suitable for low-power embedded systems [15]. By aligning computational effort with detected importance levels, it becomes possible to reduce energy consumption while preserving visual fidelity in critical regions. We acknowledge that colour-dependent information and high-level semantic saliency are not explicitly captured by the proposed luminance-based statistical importance metric. This limitation is a deliberate design trade-off that prioritizes computational simplicity and real-time applicability over semantic completeness. While smooth but chromatically rich regions may be assigned lower importance, this approach ensures predictable execution cost and energy efficiency on resource-constrained embedded platforms.

The main contributions of this work are summarized as follows:

• We formalize a deviation-based statistical importance metric for image processing applications and present computationally efficient methods for estimating this metric at run time.

• We propose a parallel image processing framework that employs importance-weighted approximate computation, integrating heterogeneous resource allocation with dynamic voltage and frequency scaling (DVFS) to explore the quality–energy–performance trade-off space.

• We validate the proposed framework through two complementary implementation scenarios: (i) a hardware-accelerated adaptive approximate filtering architecture, and (ii) a heterogeneous CPU–GPU system for variable-kernel parallel convolution on an Odroid-XU4 platform, both demonstrating clear advantages over conventional uniform-precision processing approaches.

2 Statistical Importance in Visual Data Processing

Within the domain of digital imagery, we characterize statistical importance as spatial regions exhibiting elevated deviation magnitudes relative to their neighborhood average values [16]. The fundamental principle involves revealing informational characteristics that emerge from perceptual and visual effect variations. Our investigation commenced by exploring, through software-based experimental tools, whether statistical importance in visual data could be quantified via parallel computation of local mean and deviation statistics across image subdivisions. The methodology for computing mean and deviation metrics was inspired by the integral image framework presented in reference [17]. Such statistical feature extraction techniques are well established in classical image processing literature and form the basis for many spatial-domain analysis methods [18].

Figure 1. Statistical importance across thresholds

Note: Threshold-based deviation masks applied to monochrome images. Black regions indicate low importance (below threshold); grey regions show high importance (meeting or exceeding threshold), across three computational methods.

Computing standard deviation demands substantial computational resources, necessitating squaring operations and root extractions. In pursuit of computationally-efficient importance quantification techniques, we investigated multiple approaches, implemented using an OpenCV3.1 experimental platform, as detailed below:

Approach 1: Standard deviation calculation utilizing Integral Image representations with sum and square sum accumulation matrices applied to 32 × 32 pixel regions. This matrix dimension was selected to maintain mean calculations within 16-bit integer arithmetic constraints;

Approach 2: Absolute Deviation metric replacing standard deviation by employing the magnitude of difference between samples and their local mean. This modification eliminated squaring and root extraction requirements;

Approach 3: Approximate Absolute Deviation technique through direct computation of individual deviation values sampled from 4 neighboring (4 × 4) pixel blocks, effectively utilizing 4 samples from 64 available pixels.

Figure 1 illustrates four processed outputs generated by the experimental demonstrator. The source images, in Figure 1(a), underwent subdivision into smaller (4 × 4) pixel regions with threshold-based deviation masking applied to its grayscale representation (pixels below threshold rendered as black indicating low importance, pixels at or above threshold rendered as grey indicating elevated importance). Adaptive cluster density configurations can be implemented with performance-quality-energy tradeoff considerations, as elaborated in Section 4.

2.1 Approach 1: Standard Deviation Computation Technique

Integral image–based standard deviation calculation provides an efficient mechanism for image feature detection and is widely employed in facial recognition architectures and image processing frameworks [17]. In our implementation, integral and squared-sum accumulation matrices of size 32 × 32 are constructed to enable efficient computation of local mean and variance statistics. This dimensionality ensures that local mean evaluations remain within 16-bit integer precision bounds during exploratory experimentation. Figure 2 illustrates the computational steps required for standard deviation estimation using this approach, together with the absolute deviation alternative discussed in Section 2.2.

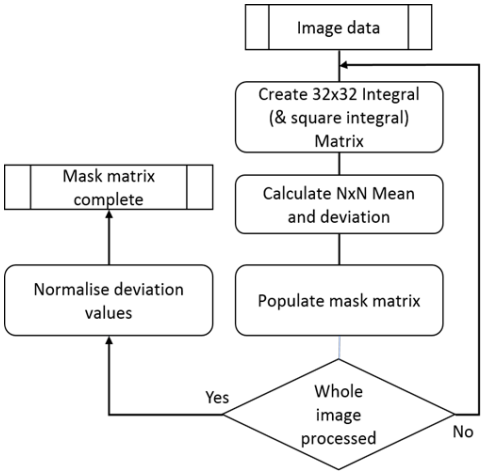

Figure 2. Flowchart for standard (and absolute) deviation utilizing Integral images

46	76	175	332	457	594	974
52	90	251	453	633	791	1279
68	108	448	805	1180	1523	2246
79	124	660	1202	1757	2267	3201
88	134	835	1563	2291	2994	4167
95	148	1032	1960	2887	3756	5181
108	168	1053	1990	2917	3794	5223

Figure 3. A demonstration of deviation computation from an Integral

The 32 × 32 accumulation matrices are further partitioned into configurable subregions of size 4 × 4, 8 × 8, or 16 × 16. For clarity, the following discussion assumes a 4 × 4 configuration. Integral image representations enable efficient computation of local mean and standard deviation metrics, as illustrated in Figure 3. For both the integral and squared-sum representations, the local sum within a rectangular region is obtained using four array lookups combined through simple addition and subtraction operations.

Specifically, the sum of pixel intensities within the highlighted blue 4 × 4 region is computed using the four corner values of the corresponding accumulation matrix, following the standard formulation introduced by Viola and Jones [17]. The local mean is then obtained by dividing this sum by the number of pixels in the region. An analogous computation using the squared-sum matrix yields the local variance, from which the standard deviation is derived.

2.2 Approach 2: Absolute Deviation Computation

Standard deviation using established techniques (Approach 1) incorporates squaring to prevent negative values when assessing variability measures. Root extraction must follow to obtain the deviation statistic. Squaring enhances how outlier values impact the resulting standard deviation metric. Ongoing discussion examines the comparative strengths of standard versus absolute deviation approaches [19]. Absolute deviation offers an alternative using dev = |value - mean|, employing the C standard library's absolute value function. This method sidesteps both squaring and root extraction operations, substantially lowering algorithmic complexity for computational units.

2.3 Approach 3: Approximate Absolute Deviation

This approximation methodology extracts one pixel sample per (4 × 4) block to calculate neighborhood statistics including mean and absolute deviation spanning 4 (2 × 2) zones (4 pixels selected from 64 total). Figure 4 illustrates the processing steps. A two-iteration strategy is implemented: iteration one retrieves two image rows into a 2 × (width/4) buffer, evaluates 2 × 2 neighborhood means plus corresponding absolute deviations from these means for each row, and commits results to a staging array subsequently propagated to matching row indices in the comprehensive mask array.

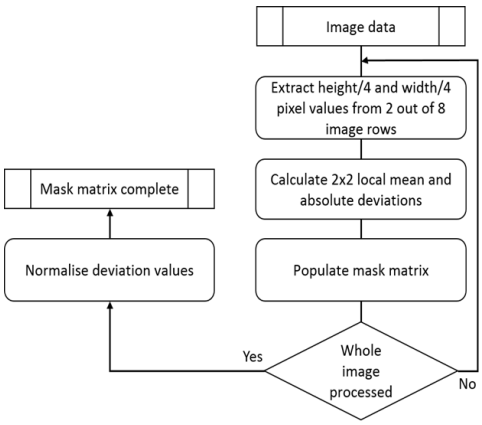

Figure 4. Calculation Flowchart for approximate absolute deviation

2.4 Performance Evaluation

All performance measurements were obtained using 4 × 4 subdivision computations across the three deviation-based importance estimation methodologies to ensure fair and consistent timing comparisons. The resulting execution times are summarized in Table 2. For ultra-high-definition imagery exceeding 2k horizontal resolution, larger subdivisions (e.g., 16 × 16) can yield satisfactory importance estimates; however, the approximate absolute deviation method demonstrates acceptable accuracy with substantially reduced execution time when operating at the finer 4 × 4 granularity. Unless otherwise stated, all subsequent timing analyses were performed on 5184 × 3888 pixel (20.1 MPixel) test images.

Table 2. Comparative timing analysis of the three significance methods

Task	Time Period	Method 1 - Standard Deviation	Method 2 - Absolute Deviation	Method 3 - Approximate Deviation
Integral Image array creation	ms	46.3	24.3	0
Calculation time	ms	142.9	139.1	6.9
Sub total	ms	189.2	163.4	6.9
Image mask creation	ms	971.9	438.1	411.8
Total time	ms	1161.1	601.5	418.7

As shown in Table 2, Method 1 (standard deviation) incurs the highest computational cost due to the use of squaring and square-root operations, resulting in significant overhead during both deviation calculation and image mask generation. Method 2 (absolute deviation) reduces arithmetic complexity by eliminating these operations, leading to moderate performance improvements while retaining sensitivity to local structural variation. Nevertheless,

both methods remain dominated by full neighborhood processing costs, limiting their scalability for high-resolution or real-time applications.

In contrast, Method 3 (approximate absolute deviation) achieves a substantial reduction in execution time by sampling a limited subset of pixels within each subdivision. This approximation introduces a degree of uncertainty in importance estimation, particularly in regions containing fine-grained textures or weak intensity gradients. However, the proposed framework does not rely on exact importance ranking; instead, it employs coarse-grained classification into low, medium, and high importance levels, for which Method 3 provides sufficient discrimination.

Experimental observations indicate that minor inaccuracies in importance estimation primarily result in conservative processing decisions—assigning slightly higher precision to marginal regions—rather than visually disruptive artefacts. Subsequent case studies confirm that the use of approximate absolute deviation preserves acceptable PSNR and SSIM values in the final output while enabling significant reductions in execution time and energy consumption. It should be noted that Methods 1 and 2 provide more precise estimates of local deviation by exhaustively processing neighborhood statistics, at the cost of significantly higher computational overhead. Method 3 intentionally sacrifices fine-grained importance accuracy in favor of execution efficiency by sampling a limited subset of pixels. Since the proposed framework relies on coarse-grained classification into low, medium, and high importance levels rather than exact importance ranking, this approximation is sufficient for guiding adaptive computation, as confirmed by downstream PSNR and SSIM evaluations.

Based on this balance between computational efficiency and importance detection fidelity, Method 3 is selected as the default importance estimation technique in this work. This choice aligns with the overarching objective of enabling energy-efficient, real-time adaptive approximation in embedded and heterogeneous image processing systems.

3 Proposed Significance Driven Computation Approach

Figure 5 Illustrates our original phased approach utilising Method 3 to highlight significance in the image and permit an adaptive computational and energy efficient approach to Image processing.

The phases involved are as follows:

Phase 1. Acquire the image which may be a choice of static (stored) image a stored video sequence or a live camera input image. Divide the whole image work-area into suitablesized Work-groups e.g. 16 × 16, 32 × 32, 64 × 64 etc. Further divide the image into a choice of 2 × 2, 4 × 4, 16 × 16 sub-area Work-items. Determine the statistical importance for each computational work-item through single-pixel sampling from work-items within 2 × 2 neighboring clusters, enabling local mean computation and subsequent importance quantification via absolute deviation metrics.

Phase 2. Classify each work-group in the image work-area by Pooling the values in each work group into either a maximum or Dynamic range (maximum - minimum) selection, reducing a group of values into a single value. Utilise a two threshold system to divide the work-groups into three distinct levels, high, medium and low significance, to allow varying levels of kernel filtering to these three levels for each work-group.

Phase 3. Allocate individual available heterogeneous elements, e.g. GPU, Neon FPU, to effect the processing required. This allocation strategy could, for instance, assign the highest-importance regions to complex 5 × 5 convolution operations executed on the GPU, intermediate-importance regions to 3 × 3 convolution processing utilizing the Neon FPU, and minimal-importance regions to 1 × 1 operations performed on the CPU.

Phase 4. Utilising such available resources, as outlined in Phase 3, along with the low complexity significance calculation, renders results much quicker and energy efficiently than the available inter-frame time, which offers slack-time that further allows DVFS to be applied separately to both the GPU and the CPU/Neon combination thereby offering greater energy efficiency in the process.

Phase 5. Evaluates the output imagery and result quality using metrics such as PSNR, SSIM, or alternative methodologies to establish a feedback mechanism through machine learning processes, enabling Phase 1 threshold parameter adaptation for optimizing the overall importance detection framework.

4 Case Study 1: Variable Kernel Sizing for Convolution Filters

To assess the viability of the importance-weighted methodology and furnish preliminary validation of the research advancement, the case study adapts significance based software resources allocation for a real-time convolution filtering application running on the Odroid-XU4. The estimation of significant data and convolution filter are designed for implementation on the Mali-T628 GPU using OpenCL framework. The application execution model is shown in Figure 6.

4.1 The Estimation of Significance (Phase 1)

Two proposed statistical analysis techniques were implemented for the modulation of significant data. The timing results for absolute and approximate absolute deviation method in Figure 7 demonstrates the comparative execution

time for the two methods. This shows that the execution time decreased by roughly a factor of two upon increasing work-group granularity from 2 × 2 to 16 × block configurations, maintaining this trend across all tested image dimensions.

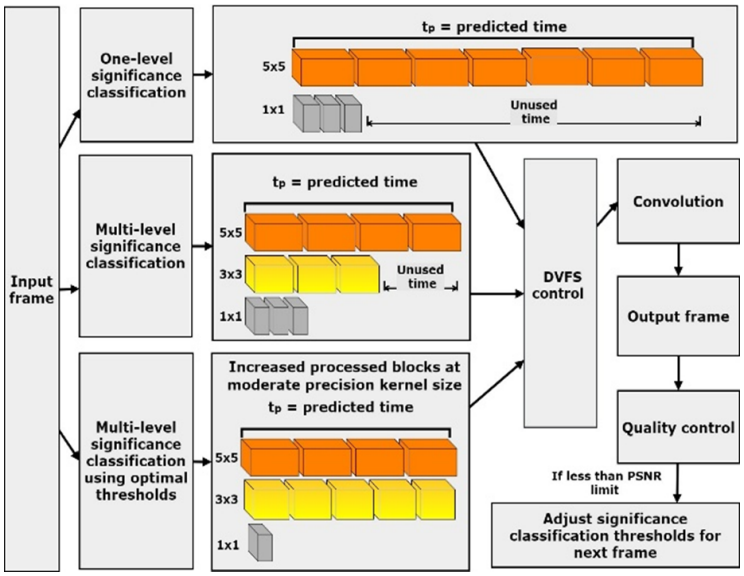

Figure 5. Proposed adaptive approximate computing approach

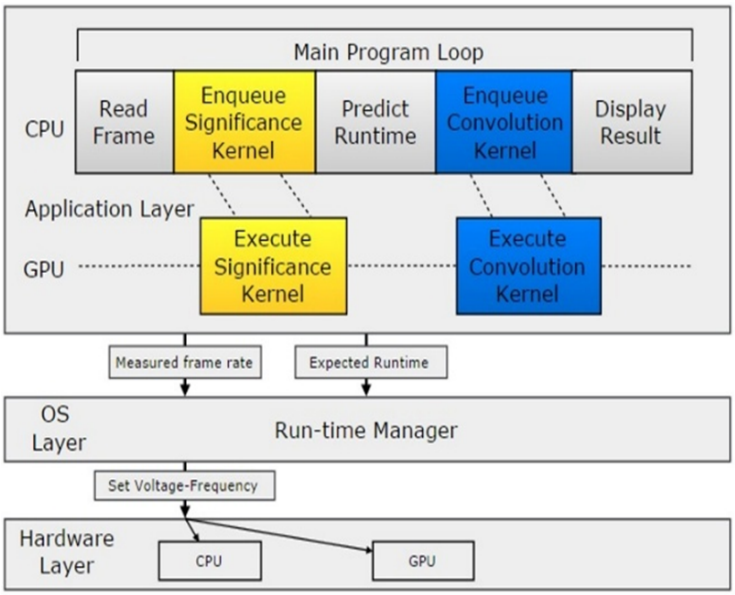

Figure 6. The execution model of the convolution filter application

4.2 Software Resources Allocation (Phase 2)

After the significance levels are generated, the higher significance levels are assigned to a larger convolution kernel size (e.g., 5 × 5), while the lower importance image regions are processed using 3 × 3 kernel or remain unprocessed, shown in Figure 8. Here the significance classification is displayed as different colour regions: grey-low or no significance, white and black, indicate high and moderate significance, respectively.

4.3 Parallel Convolution Run-Time Prediction Using Generated Significance (Phase 3)

By using a pre-computed significance data of the acquired frame from a video stream, the information of allocated significance levels is used to predict the run-time of convolution. Assuming two convolution kernel sizes are used, 5 × 5 and 3 × 3, and indexes α, β, γ, ranging from 0–2, respectively, represent an element of workload array: α-1 × 1, i.e. not processed, β-3 × 3, γ-5 × 5.

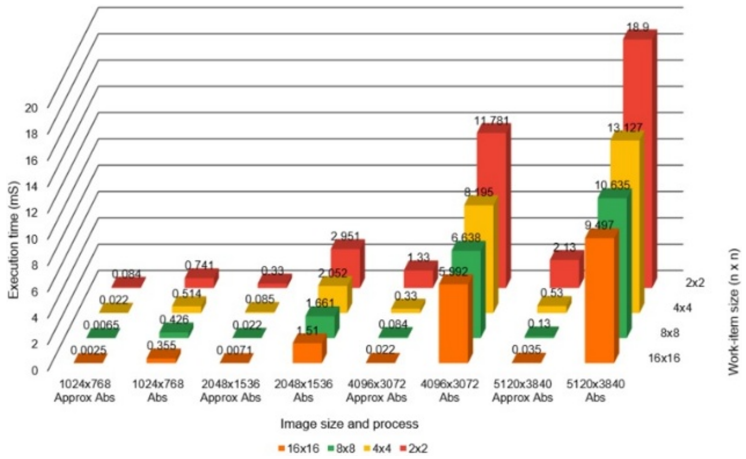

Figure 7. Odroid-XU4 Approximate and absolute deviation estimation time at 600 MHz GPU frequency

Figure 8. (a) Single level significance and (b) Multi level significance

4.4 DVFS Control to Meet Specified Performance Requirement (Phase 4)

In the case of significance-unware applications, the variation of run-time is minimal and both GPU and CPU resources can be iteratively adjusted for minimum available voltage-frequency configurations until the performance requirement is satisfied. However, by using the significance-driven approach, the convolution filter run-time executed on the GPU is a function of detected significant data. Therefore, the prediction explained in step 3, is implemented to deal with run-time changes and more accurate V-F allocation. Figure 9 displays the effect of prediction correction based on collected actual run-time and energy consumption per frame by applying DFVS control to meet performance requirement of 14 fps. Similar coordinated energy-management strategies for heterogeneous systems have been explored in prior work using model-based and learning-assisted optimization techniques [20]. Approximation-aware coordinated power and performance management strategies have been proposed for heterogeneous multi-core systems [21]. The current DVFS control strategy relies on deterministic runtime prediction derived from observed workload distributions and measured execution profiles. While effective in practice, the present implementation does not explicitly model prediction uncertainty or provide confidence intervals for runtime estimates. Incorporating probabilistic prediction techniques, such as Bayesian regression or Monte Carlo-based uncertainty estimation, together with guard-band policies, represents an important direction for future work. Such extensions would enable explicit trade-offs between energy savings and deadline-miss risk under highly dynamic workload conditions.

4.5 Peak Signal-to-Noise Ratio-Based Image Quality Characterization (Phase 5)

Following adaptive approximate processing, the output image quality is evaluated to ensure that approximation-induced distortions remain within acceptable limits. Given the known limitations of relying on a single fidelity metric, image quality assessment in this work employs a combination of Peak Signal-to-Noise Ratio (PSNR) and the Structural Similarity Index Measure (SSIM). This dual-metric strategy enables both analytical optimization and perceptual validation of the proposed framework.

PSNR is adopted as a lightweight metric suitable for closed-loop optimization and real-time feedback control due to its low computational overhead. The mean squared error (MSE) between a reference image processed using full-precision computation and the approximated output image is defined as:

$$\text{MSE} = \frac{1}{MN}\sum_{i=1}^{M}\sum_{j=1}^{N}(f(i,j) - g(i,j))^2$$

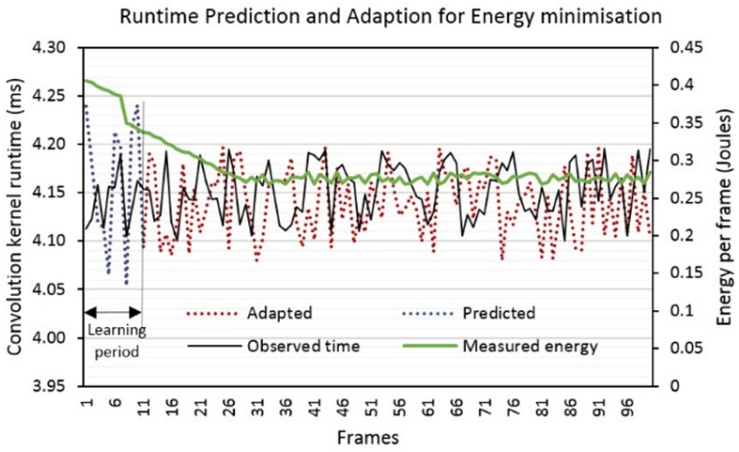

Figure 9. Predicted and Actual convolution run-time with energy consumption at 14 fps

where, f and g denote the reference and approximated images of dimensions $M \times N$, respectively. The PSNR is then computed as:

$$\mathrm{PSNR} = 10 \log_{10} \left(\frac{255^2}{\mathrm{MSE}} \right)$$

While PSNR provides a convenient quantitative measure, it does not always correlate well with perceived image quality, particularly in the presence of approximation artefacts affecting smooth or low-variance regions. A wide range of perceptual and full-reference image quality metrics have been proposed to address these limitations, particularly for evaluating visually meaningful distortions in imaging and display systems [22, 23]. To address this limitation, SSIM is additionally employed as a perceptually motivated metric that evaluates similarity based on luminance, contrast, and structural components. SSIM is defined as [24]:

$$\mathrm{sSIM}(f, g) = \frac{(2\mu_f \mu_g + C_1)\,(2\sigma_{fg} + C_2)}{\left(\mu_f^2 + \mu_g^2 + C_1\right)\left(\sigma_f^2 + \sigma_g^2 + C_2\right)}$$

where, μ_f, μ_g, σ_f^2, σ_g^2, and σ_{fg} represent local means, variances, and covariance, respectively, and C_1, C_2 are small constants introduced to ensure numerical stability.

Dynamic voltage and frequency scaling has been widely investigated as a mechanism for exploiting execution-time slack to reduce energy consumption in both single-core and parallel computing systems. Prior studies have demonstrated the effectiveness of DVFS-based scheduling under deadline constraints in clustered and chip-multiprocessor environments [25, 26], as well as logic- and architecture-level optimizations aimed at minimizing static and dynamic power dissipation [27].

Within the proposed framework, PSNR is primarily used as a feedback signal to adapt importance thresholds and approximation levels during run-time operation, while SSIM is used to validate that structural fidelity is preserved. If the evaluated quality metrics fall below predefined limits, the significance thresholds are dynamically adjusted to increase the proportion of high-precision processing in subsequent frames. This feedback mechanism enables the system to balance energy efficiency and image quality in a principled and adaptive manner. Figure 10 summarizes the proposed significance-driven adaptive computing framework, illustrating the interaction between importance estimation, adaptive approximation, and energy-aware control mechanisms.

4.6 Convolution and Variable Kernel Sizes

The implemented convolution filter kernel coefficients are selected for sharpening operation. By having only one significance level, which is mapped to a single kernel size, leaves image blocks either processed or not processed. Using multiple kernel sizes, allows an increase in the number of processed image blocks, such that work-groups of lower significance are computed at moderate precision, thereby increasing the overall PSNR. Table 3 summarizes the execution time and PSNR results obtained for different combinations of convolution kernel sizes, illustrating the quality–performance trade-offs enabled by the proposed significance-driven processing approach. The results reported in Table 3 correspond to a sharpening filter applied to a 640 × 480 grayscale image using variable convolution kernel sizes and workload combinations. Image quality is quantified using PSNR as a signal-fidelity metric, while perceptual quality is additionally evaluated using the Structural Similarity Index Measure (SSIM), which remains consistently high (greater than 0.90) for all configurations that preserve high-significance regions.

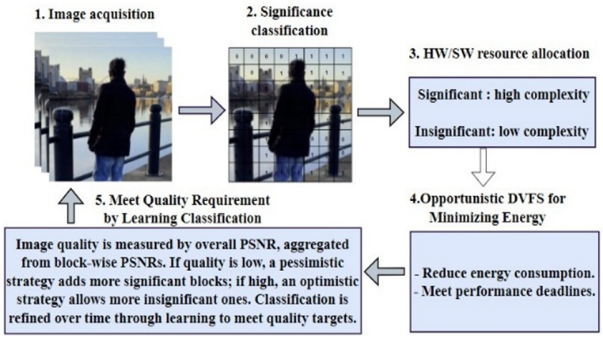

Figure 10. Significance-driven approach

Table 3. Performance and quality results for variable convolution kernel sizes

Kernel Size			Time	PSNR
5 × 5	3 × 3	1 × 1	ms	dB
100	0	0	3.44	NA
0	100	0	2.18	NA
95	0	5	3.27	35.3
0	95	5	2.08	36.2
60	30	10	2.05	30.78
30	60	10	1.30	31.23

5 Case Study 2: Adaptive Approximate Hardware Allocation

Minimizing power consumption for computational workloads comprising parallel multi-threaded applications represents a critical objective across diverse real-time information processing domains. Such processing is extremely vital in many emerging image, video and audio applications. However, the great conflict between meeting the real-time performance and improving the energy-efficiency is considered as a serious challenge in such real-time applications.

In this section, we present a run-time coordinated power-performance management approach for allocating "just-sufficient" hardware circuits required to execute a multi-task of image/frame workload, while meeting a given quality requirement. Our proposed approach optimizes the energy-efficiency by applying two major techniques. The first is by utilizing a significance-driven hardware allocating technique to assign the most relevant approximate circuits to compute a particular task of image processing workload depending on significance-driven classification. The second is performing an effective slack reclamation technique to allow for utilizing dynamic voltage frequency scaling (DVFS).

5.1 Significance-Driven Hardware Technique

For a given application, we define the workload as the volume of computations required to process an input image or an acquired frame. Figure 11 shows the relevant stages in the proposed design. The approach optimizes the total energy consumption required to process each workload together with satisfying the performance requirement. Approximate hardware design techniques have been extensively studied to reduce arithmetic complexity and power consumption. Comparative evaluations of approximate multipliers highlight substantial energy savings with bounded computational error [28], while reliability-aware approximation techniques introduce lightweight error analysis to improve robustness [29]. Emerging memory-centric and accelerator-oriented approximate architectures further demonstrate the potential of approximation in energy-efficient heterogeneous systems [30].

The proposed approach can wisely allocate a particular hardware from a variety of standard and approximate circuits to process each task of image/frame block based on its level of significance. Instead of allocating traditional complex and energy-wasteful circuits to process the lower-significance data blocks, our approach treats such blocks by executing them with low-complexity approximate circuits.

5.2 Slack Reclamation by Applying DVFS Technique

In fact, designing circuits using different approximate building blocks results in disparate critical paths. Therefore, executing a number of equal tasks on different approximate and exact implementations can lead to various task

execution times. The proposed approach exploits the variations in execution time by using dynamic voltage and frequency scaling (DVFS), whenever a circuit is able to accomplish executing a task before a given deadline.

Figure 12 compares the power characteristics associated with processing a single image block across four distinct hardware implementation types. Figure 12 illustrates the baseline power consumption prior to the application of dynamic voltage and frequency scaling (DVFS) and shows the effect of applying DVFS to approximate hardware implementations. The results demonstrate that DVFS-enabled approximate designs significantly reduce aggregate power consumption while tolerating increased execution time, yet still satisfying overall performance constraints.

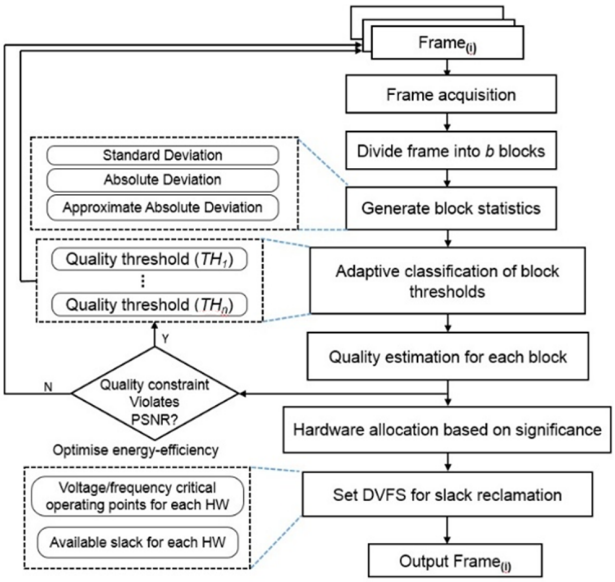

Figure 11. Flow chart showing the main steps of adaptive approximate hardware allocation

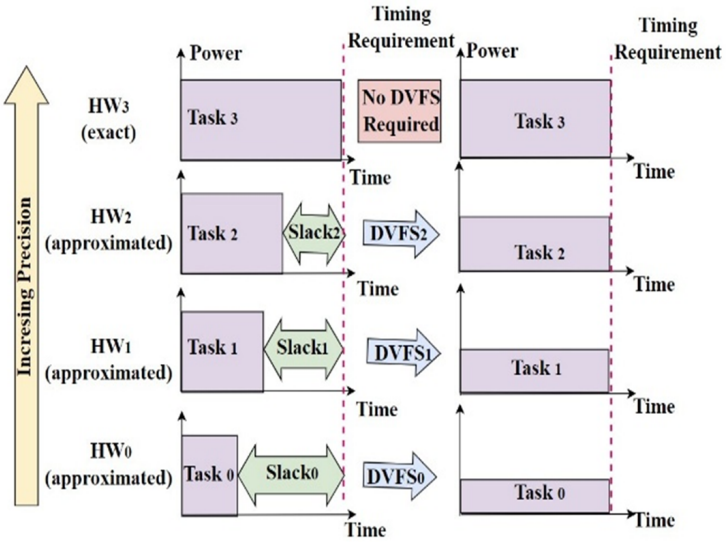

Figure 12. Power characteristics of task execution across different hardware implementations

In CMOS-based systems, dynamic power dissipation $P\_dynamic$ serves as the primary contributor, given by:

$$P\_dynamic = C\_\text{eff} \cdot V^2\_dd \cdot f$$

representing the product of effective capacitance, voltage squared, and frequency. Figure 13 illustrates the proposed Configurable Precision Multiplier (CPM) architecture, highlighting the mode selection between exact and approximate multipliers. The design emphasizes three key attributes: energy efficiency and configurability, integration with dynamic voltage and frequency scaling (DVFS) for timing slack reclamation, and algorithmic scalability to arbitrary bit widths. Similar energy-efficiency strategies have been explored in broader computing

contexts, including heterogeneous object detection pipelines [31] and predictive optimization frameworks for cloud and distributed systems [32], reinforcing the general applicability of coordinated power–performance management approaches.

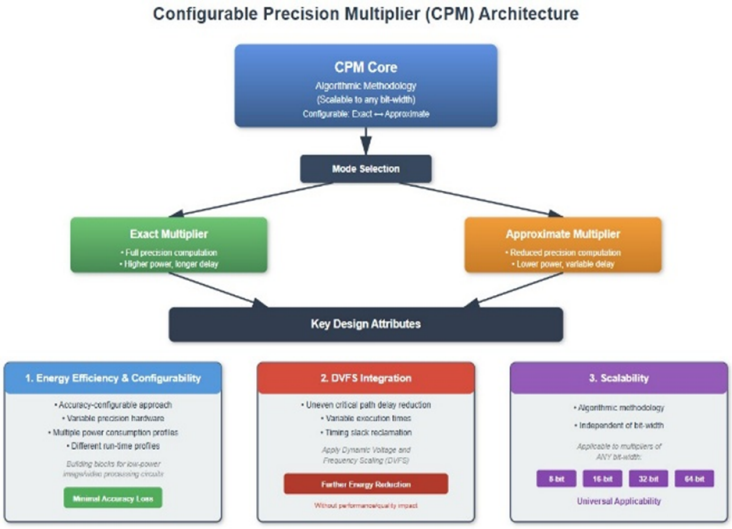

Figure 13. Configurable Precision Multiplier (CPM) architecture

We introduce a Configurable Precision Multiplier (CPM) methodology to reduce hardware complexity in multiplier designs. This approach demonstrates substantial effectiveness for approximating arithmetic and signal processing circuits.

5.3 Quality of Outcome

The quality of the output images produced by the significance-driven adaptive hardware allocation framework is evaluated using both PSNR and SSIM metrics to ensure that energy savings do not come at the cost of unacceptable visual degradation. These metrics are computed by comparing the output images obtained using approximate hardware configurations against reference images generated using exact arithmetic circuits.

PSNR serves as an analytical indicator of signal fidelity and is used to quantify the distortion introduced by approximate computation. However, given its limited sensitivity to perceptual artefacts, SSIM is additionally employed to assess the preservation of structural information, which is particularly important in image processing applications involving edge enhancement and feature extraction.

Experimental results demonstrate that allocating approximate circuits to low-significance image regions results in minimal degradation of SSIM values, indicating that structural content is largely preserved despite reduced arithmetic precision. High-significance regions, which contribute more strongly to perceptual quality, are consistently processed using higher-precision hardware, thereby maintaining visual fidelity.

The combined use of PSNR and SSIM confirms that the proposed significance-driven hardware allocation strategy effectively balances energy efficiency and output quality. Even under aggressive approximation configurations, the framework maintains acceptable perceptual quality while achieving substantial reductions in power consumption and execution time. These results highlight the robustness of the proposed approach and its suitability for energy-constrained image and video processing systems.

5.4 Further Experimental Evaluation

Following on from the experiments in case study 1, it was decided to perform similar experiments to emulate the concept of SDLC by generating a program in OpenCL on an Odroid XU4 board utilising the hardware of the T628 Mali GPU. The target was to demonstrate the effects of various combinations of 100, 75, 50 and 25% allocation of calculation accuracy levels for up to four different levels of significance, 3 down to 0, in the processed image. Table 4 shows the results of these experiments.

6 Significance Driven Adaptive Computing Model

A software model has been developed to demonstrate application of approximate significance on still and video images, utilising a dual threshold system to identify areas of significance for variable level processing, these values being held in a smaller matrix than the original image. Further processing applies pooling of the significance levels within a workgroup to render a yet smaller matrix.

Table 4. Performance at various accuracy levels

Hardware Allocation Ratio (%)				PSNR (dB)	% age Reduction cf Exact		
Sig 3 (Exact)	Sig 2 (2-bit)	Sig 1 (3-bit)	Sig 0 (4-bit)		Power	Delay	Energy
100	0	0	0	60.0	0.0	0.0	0.0
0	100	0	0	42.9	42.0	38.0	67.0
0	0	100	0	38.3	63.0	40.0	78.0
0	0	0	100	30.3	74.0	55.0	88.0
72	28	0	0	48.7	11.8	10.6	18.7
72	0	28	0	43.9	17.6	11.2	21.8
72	0	0	28	35.2	20.7	15.4	24.6
49	51	0	0	46.3	21.2	19.2	33.8
49	0	51	0	41.4	31.8	20.2	39.4
49	0	0	51	32.9	37.4	27.8	44.5
24	24	23	25	34.5	45.7	34.0	59.2
25	46	27	0	41.7	37.1	28.8	52.8
25	46	0	27	34.8	40.1	33.0	55.6
25	27	46	0	40.8	40.9	29.1	54.8
25	0	46	27	34.3	49.9	33.9	60.7
25	0	27	46	33.0	51.8	36.6	62.5
25	27	0	46	33.2	45.9	36.0	59.4
0	72	27	0	40.0	47.9	38.6	70.1
0	72	0	27	34.7	51.0	42.8	72.9
0	27	72	0	39.2	57.1	39.4	74.9
0	0	72	27	33.9	66.1	44.2	80.8
0	0	27	72	31.2	70.9	50.8	85.2
0	27	0	72	31.4	65.1	50.2	82.1
0	47	24	27	34.4	56.0	43.2	75.5
0	24	47	27	34.1	61.0	43.7	78.2
0	25	25	50	32.5	63.4	47.1	80.3

Case study 1 in Section 4 explained the application of a single threshold algorithm that performed a dual kernel processing operation, either a 3 × 3 or 5 × 5 kernel for workgroups above the threshold and 1 × 1 or 3 × 3 kernel for those below the threshold. This section describes the software demonstrator implementation, utilizing a dual-threshold architecture based on configurable percentage allocations of workgroup quantities for processing across three generated hierarchy levels. Significance-aware computing has also been explored at the programming model and runtime level to guide energy-efficient execution in heterogeneous systems [33].

6.1 Matrix Reduction

The main performance and energy efficiency benefits of the software demonstrator are achieved by a reduction in the dimensions of the significance matrix relating to the image and a further reduction by pooling the significance values into a single value per work-group related matrix (Figure 14). This facilitates referencing and processing higher definition images by selecting regions of interest (RoI) for specific processing.

6.2 Application to Images (Phase 1)

A preset spatial offset of [1,1] is employed to select two non-adjacent image rows, specifically the second and sixth rows, for processing. From each selected row, every fourth pixel value is sampled and stored in the array Pixel [2:8}. The sampled pixel values are subsequently used to compute a local mean intensity, which serves as a coarse statistical representation of the corresponding neighborhood. Specifically, four pixel samples, denoted as A, B, C, and D, are accumulated and divided by four to obtain the local mean μ_L. To minimize computational overhead, this division is implemented using two right-shift operations, thereby avoiding explicit division hardware and reducing arithmetic complexity (Figure 15).

6.3 Dynamic Percentage Thresh-Holding (Phase 2)

If PPQ0 is selected, Table 5 entry shows a percentage processing ratio of 90%:5%:5% for each of the three levels, so this selection will need to render 90% = 900 workgroups at level0 and 5% = 50 workgroups for each of level 1 and level 2. The pooled significance matrix is now processed by the histogram() function to create a vector of

binned significance values. The PPQ threshold configurations and percentage allocations employed in this study are empirically selected to demonstrate the feasibility of the proposed framework. These parameters should be regarded as heuristic design choices rather than universally optimal settings. Automated threshold selection or learning-based adaptation strategies constitute promising directions for future work.

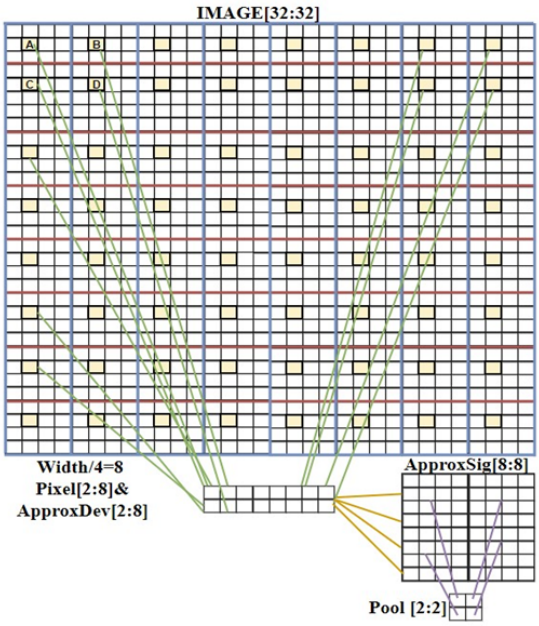

Figure 14. Relationship of image matrix dimensions for a 32 × 32 image with 4 × 4 sub groups and a 16 × 16 work-group

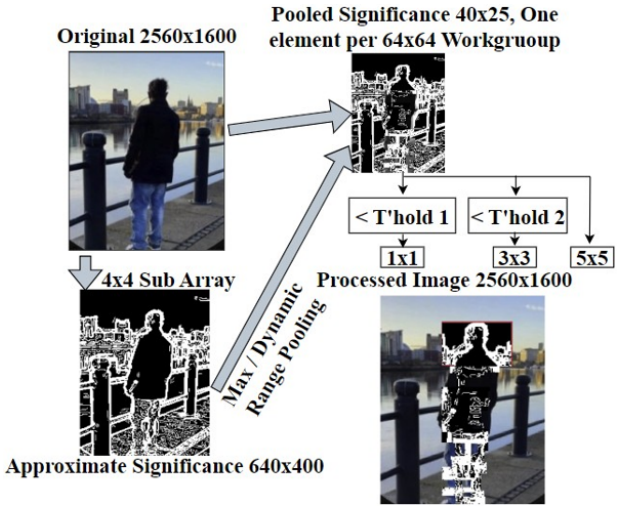

Figure 15. Interconnection of image matrix structures, sub-array partitions, and work-group elements in the demonstrator software implementation

6.4 Image Filtering (Phase 3)

For demonstration purposes, the software applies a 3 × 3 sharpening filter to image regions classified as having intermediate significance, while regions identified as highly significant are processed using a 5 × 5 Sobel filter. The kernel coefficients of both filters were intentionally adjusted to enhance visual contrast, allowing the effects of adaptive filtering to be more clearly observed. In the resulting output, regions processed with the 3 × 3filter appear with lighter intensity, whereas regions processed with the 5 × 5 filter appear darker. Figure 16 illustrates the outcome of this adaptive filtering strategy for an image processed under a simulated video workload at 10 fps, using a PPQ2 configuration with a target processing distribution of 70:15:15 across the three significance levels.

Table 5. Performance at various accuracy levels

PPQ Configuration Levels For Demonstrator Application			
PPQ Setting	Tier 0 Allocation %	Tier 1 Allocation %	Tier 2 Allocation %
0	90	5	5
1	80	10	10
2	70	15	15
3	60	20	20
4	50	30	20
5	0	100	0
6	0	0	100

Figure 16. Resultant image processing with adaptive approximate significance selected filtering. PPQ level 2 applied, 70:15:15%

7 Discussions and Conclusions

The two case studies presented in this work validate the same underlying significance-driven adaptive computation model at different abstraction layers. Case Study 1 demonstrates the effectiveness of importance-aware approximation combined with dynamic voltage and frequency scaling (DVFS) in a heterogeneous software environment, whereas Case Study 2 applies the same principles to hardware-level circuit allocation and power management. Together, these studies confirm the generality of the proposed framework and highlight its applicability across software-only, hardware-only, and hardware–software co-designed systems. Specifically, this work makes the following contributions: first, it introduces a low-cost statistical significance inference algorithm for image processing applications, which is employed within a novel adaptive approximation framework capable of tailoring filtering levels proportionally to regional significance in order to reduce energy consumption while maintaining acceptable visual quality; second, it provides extensive experimental validation of the proposed adaptive approximation approach using both hardware-based implementations and hardware–software co-design strategies applied within an image processing context. The Approximate Absolute Deviation method is shown to be an effective mechanism for highlighting significant features in image frames, offering configurable spatial granularity through the use of 2 × 2, 4 × 4, 8 × 8, or 16 × 16 regions to extract a representative approximate value. While 4 × 4 regions are employed for images with approximately 2K horizontal resolution, experimental results indicate that 8 × 8 regions remain viable for 4K imagery, as demonstrated using a 20-megapixel 5K still image. As image resolutions continue to increase toward 8K, further opportunities emerge for additional significance approximation using larger spatial regions, such as 16 × 16 blocks, which will be explored in future work pending access to suitablehigh-resolution video capture hardware.there is possible potential for further significance approximation based on 16 × 16 areas.

A significance driven adaptive computation software model has demonstrated a novel concept of using image significance in a frame, to localise computation around significant features and adapt the level of filtering based on the level of that significance. While the demonstrator utilised two thresholds to provide three levels of single convolutions to be executed in the relevant areas, it can easily be seen that this could be expanded to multiple thresholds and combinations of more accurate single or more complex multiple convolutions in selected areas.

This technique, processing only significant areas, offers a relatively faster frame processing time thereby providing greater energy efficiency than traditional whole frame image processing. Additionally shorter execution time yields extra frame slack time which can be further exploited utilising DVFS to exploit further energy efficiencies. The experimental evaluation primarily compares the proposed framework against uniform-precision and uniform-kernel

baselines. While these baselines clearly demonstrate the benefits of significance-driven adaptive computation, further isolation of the individual contributions of approximate computation and DVFS would provide additional insight. Such an analysis is left for future work.

Case study 1 indicated the extra performance and energy reductions that could be achieved when integration of OpenCV, OpenCL and ACL co-functionality becomes achievable. Currently ACL utilises C++11 compilation whilst OpenCV 3.3 utilises C++98 which does not support chrono and thread library facilities thereby hindering attempts to integrate the two libraries. The recent release of OpenCV 4.0 supports C++11 compilation and may well facilitate integration of the two libraries.

Data Availability

The data used to support the findings of this study are available from the corresponding author upon request.

Conflicts of Interest

The authors declare that they have no conflicts of interest.

References

[1] V. Joshi and P. Mane, "Novel approximate adaptive carry lookahead adder for error-resilient applications with a generic method for error analysis," *Sci. Rep.*, vol. 15, no. 1, p. 19215, 2025. https://doi.org/10.1038/s41598-025-03865-0

[2] A. Vendhan, S. E. Ahmed, and S. Gurunarayanan, "Design of approximate adder with reconfigurable accuracy," *IEEE Access*, vol. 13, pp. 17 030–17 042, 2025. https://doi.org/10.1109/ACCESS.2025.3531943

[3] V. Mrazek, R. Hrbacek, Z. Vasicek, and L. Sekanina, "EvoApprox8b: Library of approximate adders and multipliers for circuit design and benchmarking of approximation methods," in *Proceedings of the Design, Automation and Test in Europe Conference (DATE), Lausanne, Switzerland*, 2017, pp. 258–261. https://doi.org/10.23919/DATE.2017.7926993

[4] M. Samadi, D. A. Jamshidi, J. Lee, and S. Mahlke, "Paraprox: Pattern-based approximation for data-parallel applications," in *Proceedings of the International Conference on Architectural Support for Programming Languages and Operating Systems (ASPLOS), Salt Lake City, USA*, 2014, pp. 35–50. https://doi.org/10.1145/2541940.2541948

[5] A. Raha, S. Venkataramani, V. Raghunathan, and A. Raghunathan, "Quality-configurable reduce-and-rank for energy-efficient approximate computing," in *Proceedings of the Design, Automation and Test in Europe Conference (DATE), Grenoble, France*, 2015, pp. 665–670. https://doi.org/10.7873/DATE.2015.0569

[6] J. Ansel, Y. L. Wong, C. Chan, M. Olszewski, A. Edelman, and S. Amarasinghe, "Language and compiler support for auto-tuning variable-accuracy algorithms," in *Proceedings of the International Symposium on Code Generation and Optimization (CGO), Chamonix, France*, 2011, pp. 85–96. https://doi.org/10.1109/CGO.2011.5764677

[7] S. Mittal, "A survey of techniques for approximate computing," *ACM Comput. Surv.*, vol. 48, no. 4, pp. 1–33, 2016. https://doi.org/10.1145/2893356

[8] M. Shafique, R. Hafiz, S. Rehman, W. El-Harouni, and J. Henkel, "Cross-layer approximate computing: From logic to architectures," in *Proceedings of the Design Automation Conference (DAC), Austin, USA*, 2016, p. 99. https://doi.org/10.1145/2897937.2906199

[9] V. Lakshmi, V. Pudi, and J. Reuben, "In-memory implementation of an approximate adder with reduced latency and error," *IEEE Trans. Circuits Syst. I Regul. Pap.*, vol. 72, no. 5, pp. 2128–2138, 2024. https://doi.org/10.1109/TCSI.2024.3511955

[10] R. Prasanthi, V. Anuradha, S. K. Sahoo, and C. Shekhar, "Multiplier-less FFT processor architecture for signal and image processing," in *Proceedings of the International Conference on Intelligent Sensing and Information Processing, Chennai, India*, 2005, pp. 326–330. https://doi.org/10.1109/ICISIP.2005.1529470

[11] M. Alawad, Y. Bai, R. DeMara, and M. Lin, "Energy-efficient multiplier-less discrete convolution through probabilistic domain transformation," in *Proceedings of the ACM/SIGDA International Symposium on Field-Programmable Gate Arrays (FPGA), New York, USA*, 2014, pp. 185–188. https://doi.org/10.1145/2554688.2554769

[12] M. Alioto, "Energy–quality scalable adaptive VLSI circuits and systems beyond approximate computing," in *Proceedings of the Design, Automation and Test in Europe Conference (DATE), Lausanne, Switzerland*, 2017. https://doi.org/10.23919/DATE.2017.7926970

[13] S. Hartwig, D. Engel, L. Sick, H. Kniesel, T. Payer, P. Poonam, M. Glöckler, A. Bäuerle, and T. Ropinski, "A survey on quality metrics for text-to-image generation," *IEEE Trans. Vis. Comput. Graph.*, vol. 31, no. 10, pp. 9464–9483, 2025. https://doi.org/10.1109/TVCG.2025.3585077

[14] K. B. Quast, K. Ung, E. Froudarakis, L. Huang, I. Herman, A. P. Addison, J. Ortiz-Guzman, K. Cordiner, P. Saggau, A. S. Tolias, and B. R. Arenkiel, "Developmental broadening of inhibitory sensory maps," *Nat. Neurosci.*, vol. 20, no. 2, pp. 189–199, 2017. https://doi.org/10.1038/nn.4467

[15] A. Borji and L. Itti, "State-of-the-art in visual saliency modeling," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 35, no. 1, pp. 185–207, 2013. https://doi.org/10.1109/TPAMI.2012.89

[16] F. Godtliebsen, J. S. Marron, and P. Chaudhuri, "Statistical significance of features in digital images," *Image Vis. Comput.*, vol. 22, no. 13, pp. 1093–1104, 2004. https://doi.org/10.1016/j.imavis.2004.05.002

[17] P. Viola and M. Jones, "Robust real-time face detection," *Int. J. Comput. Vis.*, vol. 57, no. 2, pp. 137–154, 2004. https://doi.org/10.1023/B:VISI.0000013087.49260.fb

[18] J. C. Russ, *Introduction to Image Processing and Analysis*. CRC Press, 2008.

[19] C. Leys, C. Ley, O. Klein, P. Bernard, and L. Licata, "Detecting outliers: Do not use standard deviation around the mean, use absolute deviation around the median," *J. Exp. Soc. Psychol.*, vol. 49, no. 4, pp. 764–766, 2013. https://doi.org/10.1016/j.jesp.2013.03.013

[20] S. Yang, R. A. Shafik, G. V. Merrett, E. Stott, J. M. Levine, J. Davis, and B. M. Al-Hashimi, "Adaptive energy minimization of embedded heterogeneous systems using regression-based learning," in *Proceedings of the IEEE International Workshop on Power and Timing Modeling, Optimization and Simulation (PATMOS), Salvador, Brazil*, 2015, pp. 103–110. https://doi.org/10.1109/PATMOS.2015.7347594

[21] A. Kanduri, A. Miele, A. M. Rahmani, P. Liljeberg, C. Bolchini, and N. Dutt, "Approximation-aware coordinated power and performance management for heterogeneous multi-cores," in *Proceedings of the Design Automation Conference (DAC), New York, USA*, 2018. https://doi.org/10.1145/3195970.3195994

[22] J. Wang, J. Wang, S. Wang, and Y. Zhang, "Deep learning in pediatric neuroimaging," *Displays*, vol. 82, p. 102583, 2023. https://doi.org/10.1016/j.displa.2023.102583

[23] M. Pedersen and J. Y. Hardeberg, "Full-reference image quality metrics: Classification and evaluation," *Found. Trends Comput. Graph. Vis.*, vol. 7, no. 1, pp. 1–80, 2012. http://doi.org/10.1561/0600000037

[24] Z. Wang, A. C. Bovik, H. R. Sheikh, and E. P. Simoncelli, "Image quality assessment: From error visibility to structural similarity," *IEEE Trans. Image Process.*, vol. 13, no. 4, pp. 600–612, 2004. http://doi.org/10.1109/TIP.2003.819861

[25] K. H. Kim, R. Buyya, and J. Kim, "Power-aware scheduling of bag-of-tasks applications with deadline constraints on DVS-enabled clusters," in *Proceedings of the IEEE International Symposium on Cluster Computing and the Grid (CCGrid), Rio de Janeiro, Brazil*, 2007, pp. 541–548. http://doi.org/10.1109/CCGRID.2007.85

[26] J. Li and J. F. Martinez, "Dynamic power-performance adaptation of parallel computation on chip multiprocessors," in *Proceedings of the IEEE International Symposium on High-Performance Computer Architecture (HPCA), Austin, USA*, 2006, pp. 77–87. http://doi.org/10.1109/HPCA.2006.1598114

[27] C. Piguet, C. Schuster, and J. L. Nagel, "Optimizing architecture activity and logic depth for static and dynamic power reduction," in *Proceedings of the IEEE Northeast Workshop on Circuits and Systems (NEWCAS), Montreal, Canada*, 2002, pp. 41–44. http://doi.org/10.1109/NEWCAS.2004.1359011

[28] H. Jiang, C. Liu, N. Maheshwari, F. Lombardi, and J. Han, "A comparative evaluation of approximate multipliers," in *Proceedings of the IEEE Conference on Nanoscale Architectures, Beijing, China*, 2016, pp. 191–196. https://doi.org/10.1145/2950067.2950068

[29] B. Grigorian and G. Reinman, "Dynamically adaptive and reliable approximate computing using light-weight error analysis," in *Proceedings of the NASA/ESA Conference on Adaptive Hardware and Systems (AHS), Leicester, UK*, 2014, pp. 248–255. http://doi.org/10.1109/AHS.2014.6880184

[30] A. Rahimi, A. Ghofrani, K. T. Cheng, L. Benini, and R. K. Gupta, "Approximate associative memristive memory for energy-efficient GPUs," in *Proceedings of the Design, Automation and Test in Europe Conference (DATE), Grenoble, France*, 2015, pp. 1497–1502. http://doi.org/10.7873/DATE.2015.0579

[31] E. Totoni, M. Dikmen, and M. J. Garzarán, "Easy, fast, and energy-efficient object detection on heterogeneous on-chip architectures," *ACM Trans. Archit. Code Optim.*, vol. 10, no. 4, pp. 1–25, 2013. https://doi.org/10.1145/2541228.2555302

[32] D. M. Bui, Y. Yoon, E. N. Huh, S. Jun, and S. Lee, "Energy efficiency for cloud computing systems based on predictive optimization," *J. Parallel Distrib. Comput.*, vol. 102, pp. 103–114, 2017. https://doi.org/10.1016/j.jpdc.2016.11.011

[33] V. Vassiliadis, K. Parasyris, C. Chalios, C. D. Antonopoulos, S. Lalis, N. Bellas, H. Vandierendonck, and D. S. Nikolopoulos, "A programming model and runtime system for significance-aware energy-efficient computing," *ACM SIGPLAN Not.*, vol. 50, no. 8, pp. 275–276, 2014. https://doi.org/10.1145/2858788.2688546