Optimized Real - Time HDR Tone Mapping Using Hybrid Fixed-Point and Floating-Point Model

M Vamsi Krishna Allu[1]*, B. Bala Murali[2], B. V V N Sai Sree Ram Teja[3], K. Jaswanth[4], B. Phani Lakshman[5], B. Bhavani[6], A. Koataiah[7]
[1-7] Department of ECE, Sir C R Reddy College of Engineering, Eluru-534007, A.P.

Abstract – Our aim is to develop an efficient MATLAB implementation of the Drago tone-mapping operator, demonstrating its potential for real-time HDR processing in resource-constrained environments. Our software simulation combines floating-point precision for critical operations with optimized fixed-point arithmetic for luminance calculations, achieving professional-quality results with reduced computational overhead. The simulation produces excellent tone-mapped images with natural preservation of highlights and shadows, verified through PSNR, GMSD, and TMQI metrics. While currently implemented in MATLAB, our approach demonstrates a practical pathway for hardware implementation, balancing computational efficiency with visual quality. These results provide valuable insights for developers working to implement complex tone-mapping algorithms in power-constrained devices like smartphones and automotive systems.

Keywords - HDR, Tone Mapping, MATLAB, Image Processing Tool Box, Drago, GMSD, PSNR, TMQI, Naturalness, Throughput.

I. INTRODUCTION

High Dynamic Range (HDR) imaging is a cutting-edge technology designed to capture intricate details in the brightest highlights and the darkest shadows of a scene. HDR imaging surpasses the limitations of conventional Low Dynamic Range (LDR) imaging by combining multiple LDR images, taken at varying exposure levels, to produce a single HDR image [2]. The resulting images, with dynamic ranges spanning billions, are widely used in applications such as digital photography, medical imaging, and automotive systems. Fixed-point arithmetic offers a more efficient alternative by reducing resource requirements and accelerating calculations, but it sacrifices precision and is prone to quantization errors, especially in operations with extreme values [5-9].

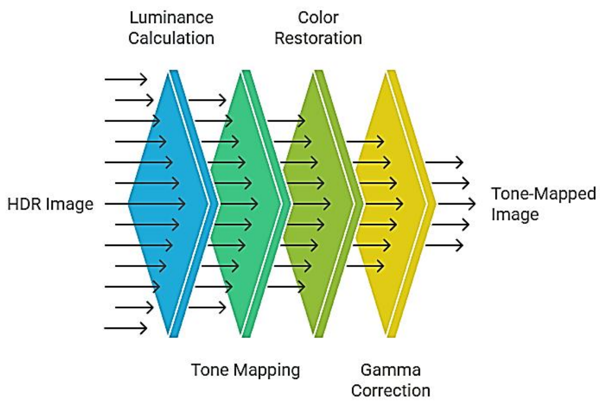

Fig. 1. Workflow diagram of tone mapping algorithm

The Floating-Point and Fixed-Point Arithmetic. This Hybrid model simplifies complex arithmetic operations—such as multiplication, division, and exponentiation—by transforming them into addition and subtraction operations. This reduces computational complexity without incurring conversion costs and provides a balance between resource efficiency and precision. In this work, the Drago tone mapping operator (TMO), a globally adaptive algorithm, is optimized using Hybrid Model. The Drago TMO is well-suited for HDR imaging due to its ability to dynamically adjust logarithmic bases for varying pixel luminance, compressing brightness while preserving details in shadows. By replacing traditional floating-point operations with LNS-based computations, the complexity of the tone mapping process is significantly reduced [10-12]. Additionally, fixed-point arithmetic is employed for luminance calculations to enhance computational efficiency further.

II. HYBRID MODEL IMPLEMENTATIONS

A. Floating – Point Arithmetic:

Floating-point arithmetic is a way of managing numbers with high precision, designed to handle the widest possible range of values efficiently. It represents numbers using a base, an exponent, and a fraction (mantissa), which allows it to tackle both extremely large and minuscule values without losing accuracy [13]. In tone mapping, it's indispensable for operations like logarithmic transformations and gamma correction. For instance, logarithmic transformations condense the brightness spectrum of an image, ensuring that details in both shadows and highlights are preserved [14]. Similarly, gamma correction fine-tunes brightness and contrast, giving images a natural and balanced look.

In the floating – point arithmetic, a number is represented as:

$$N = (-1)^s \times M \times 2^E$$

where:

s: Sign bit (0 for positive, 1 for negative)

M: Mantissa (normalized fractional part, 1.M)

E: Exponent (offset by a bias)

However, this precision comes at a cost: floating-point operations demand significant computational resources, which makes them energy-intensive and less ideal for compact or energy-efficient devices. A unique insight here is the paradoxical nature of floating-point arithmetic—it is both the enabler and the bottleneck. While it allows for unparalleled precision and visual fidelity, it forces developers to think creatively about when and where it should be applied, particularly in hybrid models. This delicate balancing act highlights how engineering solutions often require both technical rigor and artistic sensibility, especially in fields like tone mapping that blend science and visual art.

B. Fixed – Point Arithmetic:

Fixed-point arithmetic is a streamlined approach to numerical calculations, using a fixed distribution of integer and fractional bits [15]. Unlike floating-point systems, it avoids the need for an exponent, making it inherently simpler and more hardware-friendly. In tone mapping, fixed-point arithmetic proves especially advantageous for luminance calculations, where slight rounding errors are inconsequential. This trade-off allows the computational process to remain efficient while maintaining acceptable image quality.

The value of a fixed – point number can be calculated as:

$$\text{Value} = \text{Integer Part} + \left(\frac{\text{Fractional Part}}{2^n}\right)$$

where n is the number of fractional bits.

The efficiency of fixed-point arithmetic makes it a natural fit for applications demanding real-time processing, such as streaming or automotive imaging systems. Operations like addition, subtraction, and multiplication execute rapidly, enabling high throughput. However, this simplicity comes with limitations: fixed-point systems have a reduced dynamic range and precision, which can impact tasks requiring fine-grained detail preservation. By strategically deploying fixed-point arithmetic where precision is less critical, tone mapping systems can allocate computational resources more effectively. This careful partitioning of tasks allows for faster processing and energy savings while still producing visually acceptable results, striking an optimal balance between performance and quality [16].

III. TONE MAPPING OPTIMIZATION

A. Luminance Calculation:

The first step in the tone mapping process is to calculate the luminance (*Lw*) of the HDR image. Luminance represents the brightness of each pixel and is computed as a weighted sum of the RGB channels.

The formula used for computing the luminance L_w is:

$$L_w = 0.27 \cdot R + 0.67 \cdot G + 0.06 \cdot B$$

Where:

- **R, G, B** are the red, green and blue channels of the HDR image.
- The weights 0.27, 0.67, 0.06 correspond to the approximate sensitivity of the human eye to red, green and blue wavelengths

The luminance calculation is performed using fixed-point arithmetic to optimize computational efficiency. Fixed-point arithmetic is faster and requires fewer hardware resources, making it ideal for real-time applications. The result is scaled appropriately based on the bit-width used for fixed-point representation (e.g., 16-bit with 8 integer and 8 fractional bits) to avoid overflow and minimize quantization errors [17].

B. Tone Mapping:

The tone mapping step compresses the dynamic range of the HDR image to make it suitable for display on devices with limited dynamic range, such as LCDs and LEDs. The Drago Tone Mapping Operator (TMO) is used in the hybrid model [3].

The contrast factor is computed by:

$$\text{contrast\_factor} = \frac{\log_2(L_{w_{\max}}) - \log_2(L_{wa})}{\log_2(2^8) - 0.5}$$

where:

- L_{wmax} is the maximum luminance in the image.
- 2^8 represents the display range (eg., for an 8 – bit display) .

The display luminance L_d is computed by:

$$L_d = \left(\frac{\log(1 + L_w)}{\log(1 + L_{wa})}\right) \cdot \left(1 + \left(\frac{L_w}{L_{w_{\max}}}\right)^{\text{bias}}\right)$$

where, bias is an adaptive bias adjustment based on the contrast factor.

The luminance values (Lw) are computed using fixed-point arithmetic, while the logarithmic transformations and adaptation parameters ($Lwa, Lwmax$) are handled with floating-point arithmetic [1][6]. The log-average luminance (Lwa) and maximum luminance ($Lwmax$) are precomputed for each frame using floating-point arithmetic to ensure accurate tone mapping.

C. Color Restoration:

After tone mapping, color restoration is performed to adjust the RGB channels based on the modified luminance. This step ensures that the color fidelity of the original HDR image is preserved. The division operations involved in color restoration are performed using floating-point arithmetic to ensure accurate color fidelity.

The tone-mapped luminance (*Ld*) is computed using fixed-point arithmetic, while the division and multiplication operations for color restoration are handled with floating-point arithmetic.

The restored color channels R_d, G_d, B_d are calculated as:

$$R_d = \frac{R}{L_w + \epsilon} \cdot (L_{d_{\text{final}}})^{\text{color\_exponent}}$$

$$G_d = \frac{G}{L_w + \epsilon} \cdot (L_{d_{\text{final}}})^{\text{color\_exponent}}$$

$$B_d = \frac{B}{L_w + \epsilon} \cdot (L_{d_{\text{final}}})^{\text{color\_exponent}}$$

where:

- color_exponent is adaptively determined as 0.96+0.02·(contrast_factor>1.2).
- L_{dfinal} is the final luminance after detail enhancement and normalization.

The division operations involved in color restoration are performed using floating-point arithmetic to ensure accurate color fidelity. Floating-point arithmetic ensures that the color channels are accurately restored, maintaining the original color balance and visual quality [1].

D. Gamma Correction:

Gamma correction is applied to adjust the luminance of the image to match the non-linear response of human vision. This step ensures that the final image is visually accurate across different display devices.

Formula:

$$\begin{bmatrix} R_{out}(x,y) \\ G_{out}(x,y) \\ B_{out}(x,y) \end{bmatrix} = \begin{bmatrix} R_d(x,y)^{1/\gamma} \\ G_d(x,y)^{1/\gamma} \\ B_d(x,y)^{1/\gamma} \end{bmatrix}$$

$R_{out}(x,y)$, $G_{out}(x,y)$, $B_{out}(x,y)$: Final output color channels after gamma correction.

γ: Gamma value (typically 2.2).

The power function involved in gamma correction [2] is computed using floating-point arithmetic to ensure high precision. The tone-mapped luminance (*Ld*) and color channels like (*Rd*, *Gd*, *Bd*) are computed using fixed-point arithmetic, while the gamma correction is handled with floating-point arithmetic. Floating-point arithmetic ensures that the gamma correction is performed accurately, enhancing the perceptual quality of the final image.

5.3 Pipeline Parallel Processing:

Pipelined parallel Processing divides the tone mapping process into multiple stages, where each stage performs a specific task. These stages are executed in parallel, with data flowing from one stage to the next in a pipeline fashion. While one stage is processing a frame, the next stage can start processing the previous frame, allowing for continuous and efficient processing of multiple frames [1].

Work Flow of Pipeline Parallel Processing:

- **Stage 1 (Luminance Calculation):**

  The luminance (*Lw*) of the HDR image is calculated using fixed-point arithmetic. This stage is computationally intensive but optimized for speed using fixed-point operations. Once the luminance calculation for a frame is complete, the results are passed to Stage 2, while Stage 1 starts processing the next frame.

- **Stage 2 (Tone Mapping - Drago TMO):**

  The tone-mapped luminance (*Ld*) is computed using the Drago TMO. This stage involves logarithmic transformations and adaptation parameter calculations, which are performed using floating-point arithmetic for high precision. The results from this stage are passed to Stage 3, while Stage 2 starts processing the next frame.

- **Stage 3 (Color Restoration):**

  The RGB channels are adjusted based on the modified luminance using floating-point arithmetic for division operations. This ensures accurate color fidelity. The results are passed to Stage 4, while Stage 3 starts processing the next frame.

- **Stage 4 (Gamma Correction):**

Gamma correction is applied to adjust the luminance of the image to match the non-linear response of human vision. This stage uses floating-point arithmetic for the power function involved in gamma correction. The final output image is produced, while Stage 4 starts processing the next frame.

IV. RESULTS

The proposed hybrid tone-mapping algorithm, which combines the fixed-point and floating-point arithmetic, was implemented and tested using MATLAB's Image Processing Toolbox. The simulation results demonstrate the efficiency and effectiveness of the proposed method in terms of performance, image quality, and suitability for real-time HDR image processing. Hybrid Drago and Hybrid Popović methods stand out as the best-performing, with near-perfect TMQI scores and enhanced naturalness.

A. HYBRID – Based Drago TMO:

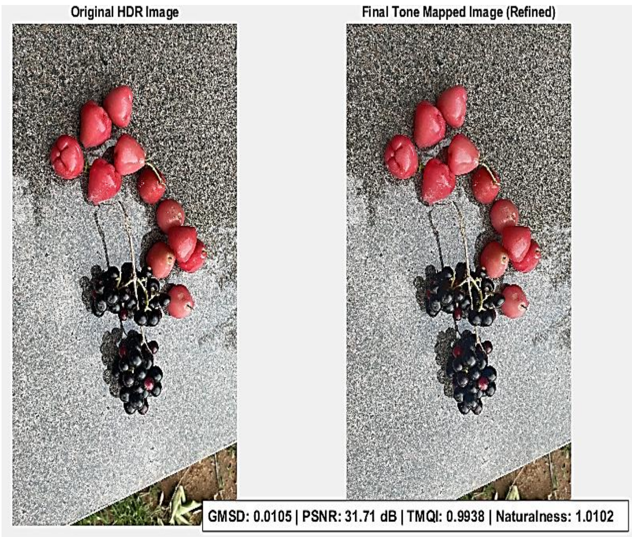

Fig 6.2(a): Simulation of Hybrid based Drago TMO

PSNR of 31.71 dB: Indicates very low distortion compared to the original HDR image. GMSD of 0.0105: Shows excellent preservation of gradient information and edge details. TMQI of 0.9938: Demonstrates near-perfect overall quality in terms of structural fidelity and naturalness. Naturalness of 1.0102: Suggests that the image is visually natural and perceptually enhanced.

B. HYBRID – Based Hassan Carletta TMO:

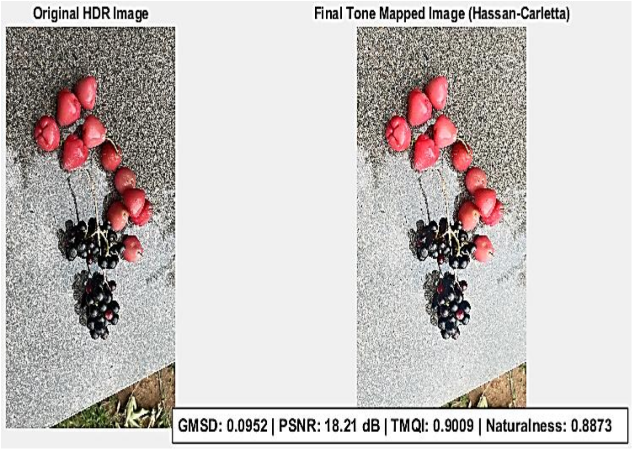

Fig 6.2(b): Simulation of Hybrid based Hassan Carletta TMO

PSNR of 18.21 dB: Indicates moderate distortion control, though not as high as the hybrid Drago and AmbalathaKandy methods. GMSD of 0.0952: Shows good preservation of gradient information and edge details. TMQI of 0.9009: Demonstrates good overall quality in terms of structural fidelity and naturalness. Naturalness of 0.8873: Suggests that the image is visually natural and perceptually pleasing.

C. HYBRID – Based Ambalathakandy TMO:

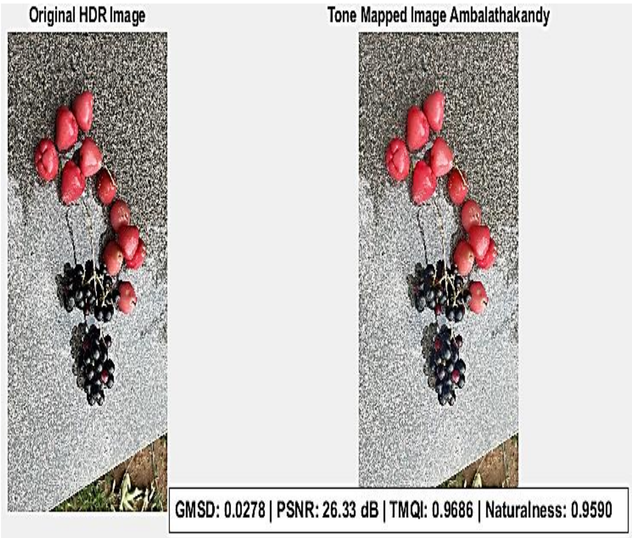

Fig 6.2(c): Simulation of Hybrid based Ambalathakandy TMO

PSNR of 26.33 dB: Indicates good distortion control, though not as high as the hybrid Drago method. GMSD of 0.0278: Shows excellent preservation of gradient information and edge details. TMQI of 0.9686: Demonstrates excellent overall quality in terms of structural fidelity and naturalness. Naturalness of 0.9590: Suggests that the image is visually natural and perceptually pleasing.

D. HYBRID – Based Yang et.al TMO:

Fig 6.2(d): Simulation of Hybrid based Yang et.al TMO

PSNR of 19.88 dB: Indicates moderate distortion control, though not as high as the hybrid Drago and Ambalathakandy methods. GMSD of 0.0628: Shows good preservation of gradient information and edge details. TMQI of 0.9088: Demonstrates good overall quality in terms of structural fidelity and naturalness. Naturalness of 0.8358: Suggests that the image is visually natural and perceptually pleasing.

E. HYBRID – Based Popović et.al TMO:

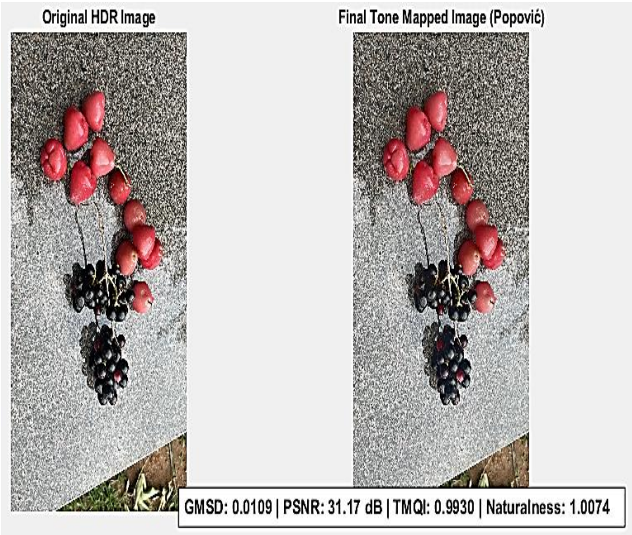

Fig 6.2(e): Simulation of Hybrid based Popović et.al TMO

PSNR of 31.17 dB: Indicates excellent distortion control, very close to the hybrid Drago method. GMSD of 0.0109: Shows excellent preservation of gradient information and edge details. TMQI of 0.9930: Demonstrates near-perfect overall quality in terms of structural fidelity and naturalness. Naturalness of 1.0074: Suggests that the image is visually natural and perceptually enhanced.

Comparison of LNS and HYBRID models:

When comparing the two methods, HYBRID clearly delivers sharper and cleaner images, with better overall quality and fewer distortions than LNS. But while HYBRID excels in technical performance, LNS tends to keep the images looking more natural and visually pleasing. It's a bit of a trade-off—HYBRID gives you a crisper picture, but LNS keeps things looking more lifelike. Interestingly, with the POPOVIC method, HYBRID manages to strike the perfect balance, improving both quality and naturalness, which isn't seen with the others.

	Performance Metrics	DRAGO	HASSAN CARLETTA	AMABALA THANKANDY	YANG	POPOVIC
LNS	PSNR (dB)	20.73	15.79	17.25	17.25	15.61
	GMSD	23.45	41.40	34.99	34.99	42.27
	TMQI	0.92	0.87	0.93	0.87	0.83
	Naturalness	7.89	7.73	7.57	7.90	7.94
HYBRID	PSNR (dB)	31.71	18.21	26.33	19.88	31.17
	GMSD	0.10	0.09	0.03	0.06	0.01
	TMQI	0.99	0.90	0.97	0.91	0.99
	Naturalness	1.01	0.89	0.96	0.83	1.01

Table 1.3: Comparison of Performance Metrics in LNS & HYBRID Models

The following graph shows the average values for the above mentioned LNS model and HYBRID model are shown below:

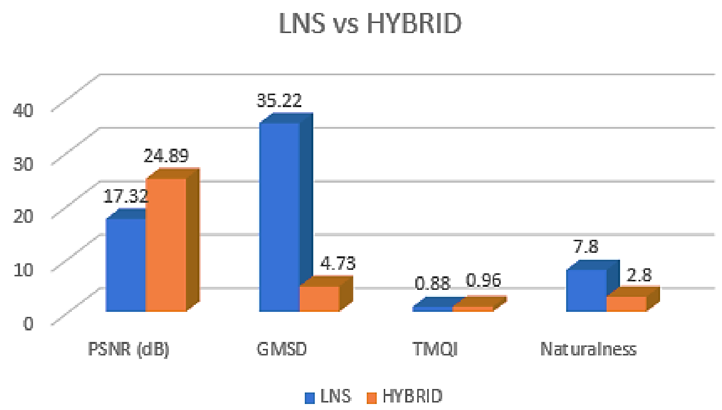

Fig 6.3: Graph showing comparison between LNS & HYBRID Model (Avg. values)

V. CONCLUSION

Capturing high-quality HDR images that preserve detail in both bright and dark areas requires a balance between performance and precision. Floating-point arithmetic offers the accuracy needed but comes with high computational costs, increased memory usage, and greater power consumption, making it less suitable for real-time or resource-limited systems. Fixed-point arithmetic is faster and more efficient but can lead to image quality issues due to quantization errors and limited dynamic range. This study introduces a hybrid model that combines the strengths of both approaches, reducing processing overhead while maintaining visual quality. Experimental results show that this method delivers high-speed performance and effective tone mapping, making it a practical solution for real-time HDR applications and paving the way for future advancements in efficient image processing.

REFERENCES

1. Logarithmically Optimized Real – Time HDR Tone Mapping with Hardware Implementation, *Siddhartha Kashyap, Pushpa Giri and Ashish Kumar Bhandari, IEEE Transactions (2023).*Drago, F., Myszkowski, K., Annen, T., & Chiba, N. (2003). Adaptive logarithmic mapping for displaying high contrast scenes. *Computer Graphics Forum*, 22(3), 419-426.
2. Allu, K. (2021). Simulation Approach to Design High Sensitive Nems Based Sensor for Molecular Bio-Sensing Applications. *European Journal of Molecular & Clinical Medicine*, *8*(03), 2021.
3. Medagam, Avinash Reddy, and Miranji Katta. "Design and implementation of area efficient hybrid PPA adder architecture." *Journal of Critical Reviews*, vol. 7, no. 18, 2020, pp. 2361–2367.
4. Katta, M., & Sandanalakshmi, R. (2021). Simultaneous tropical disease identification with PZT-5H piezoelectric material including molecular mass biosensor microcantilever collection. *Sensing and Bio-Sensing Research*, *32*, 100413.
5. Kaufmann, M. (2010). High Dynamic Range Imaging: Acquisition, Display, and Image-Based Lighting. *Morgan Kaufmann Publishers*.
6. Katta, M., & Sandanalakshmi, R. (2018). A technology overview and future scope of bio-MEMS in tropical disease detection. *Rev Int J Eng Technol*, *7*, 648-651.
7. Drago, F., Myszkowski, K., Annen, T., & Chiba, N. (2003). Adaptive logarithmic mapping for displaying high contrast scenes. *Computer Graphics Forum*, 22(3), 419-426.
8. Katta, Miranji, R. Satish, Komati Yamini, and R. Sandanalakshmi. "An Efficient Electricity Management System Using IOT-Enabled Elife Devices." *i-manager's journal on embedded systems* 11, no. 2 (2023).

9. Yamini, K., K. S. N. Raju, and K. Miranji. "Low power data compression algorithm for wireless sensor networks using VHDL." *International Journal Of Engineering And Computer Science* 3, no. 11 (2014).
10. Katta, M., Kinnera, A., Yamini, K., & Srilakshmi, G. (2025). An Iot-Based Wearable Device For Ensuring Social Distancing Using Rssi Technology. *i-Manager's Journal on Electronics Engineering*, *15*(2).
11. Kumar, P. A., & Katta, M. (2025). Optimized Beamforming Via Gwo-L2 And Hybrid Algorithms For High-Performance Wireless Communications. *Computer*, *25*(7).
12. Arithmetic Operations in The Logarithmic Number System. IEEE Transactions on Computers. – Peter J. Schwander, Earl E. Swartzlander Jr., Vassil Dimitrov Fattal, R., Lischinski, D., & Werman, M. (2002). Gradient domain high dynamic range compression. *ACM Transactions on Graphics (TOG)*, 21(3), 249-256.
13. Katta, Miranji, Sandanalakshmi Rajendran, Srilakshmi Gubbala, and Madhavarao Chunduri. "Classified disease detection from MEMS based biosensor array." In *AIP Conference Proceedings*, vol. 3217, no. 1. AIP Publishing, 2024.
14. Ram Kumar, M. K. A., Sudhakar, H., & Komati, Y. (2022). Optimized design of cascaded and tree-based n-bit comparators using majority logic with enhanced linearity and reduced delay. *Journal of Civil and Environmental Engineering*, *12*(9), 205–213.
15. Katta, M., Sandanalakshmi, R., Srilakshmi, G., & Adireddi, R. (2022). An efficient learning model selection for dengue detection. In *Intelligent and Cloud Computing: Proceedings of ICICC 2021* (pp. 439-453). Singapore: Springer Nature Singapore.
16. Katta, M., Sandanalakshmi, R., Ravikumar, M., Komati, Y., & Gubbala, S. (2024). Microcantilever fabrication and characterization for biosensing applications. In *Futuristic Trends in Electronics & Instrumentation Engineering* (Vol. 3, Book 1, pp. 217–228). IIP Series.
17. Rajesh, G., Katta, M., Nagasai, D., Raja, C., Bhagyasri, B., & Rajya Lakshmi Devi, G., & Manoj Kumar, D. (2025). Design and analysis of iterative detection algorithms for VLSI-based massive MIMO systems. *Computer Research and Development*, *25*(4), 20–30.
18. Katta, Miranji, T. Venkateswara Rao, R. Venkata Sai Muralinadh, Arun Rongala, and Srinidhi V. Ballullaya. "Automation of Gutta-percha Disinfection." *i-Manager's Journal on Embedded Systems* 11, no. 2 (2023): 1.