User User

LL

70 □□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□

RESEARCH A

M. S. Ramaiah University Of Applied Sciences

Document Details

Submission ID

trn:oid:::1:2800530804

Submission Date

Jan 17, 2024, 12:20 AM GMT+5:30

Download Date

Jan 17, 2024, 12:23 AM GMT+5:30

File Name

ICDCS_2024_FOV_HDR.pdf

File Size

1.6 MB

12 Pages

9,769 Words

52,236 Characters

How much of this submission has been generated by AI?

***1%**

of qualifying text in this submission has been determined to be generated by AI.

* Low scores have a higher likelihood of false positives.

Caution: Percentage may not indicate academic misconduct. Review required.

It is essential to understand the limitations of AI detection before making decisions about a student's work. We encourage you to learn more about Turnitin's AI detection capabilities before using the tool.

Frequently Asked Questions

What does the percentage mean?
The percentage shown in the AI writing detection indicator and in the AI writing report is the amount of qualifying text within the submission that Turnitin's AI writing detection model determines was generated by AI.

Our testing has found that there is a higher incidence of false positives when the percentage is less than 20. In order to reduce the likelihood of misinterpretation, the AI indicator will display an asterisk for percentages less than 20 to call attention to the fact that the score is less reliable.

However, the final decision on whether any misconduct has occurred rests with the reviewer/instructor. They should use the percentage as a means to start a formative conversation with their student and/or use it to examine the submitted assignment in greater detail according to their school's policies.

How does Turnitin's indicator address false positives?
Our model only processes qualifying text in the form of long-form writing. Long-form writing means individual sentences contained in paragraphs that make up a longer piece of written work, such as an essay, a dissertation, or an article, etc. Qualifying text that has been determined to be AI-generated will be highlighted blue on the submission text.

Non-qualifying text, such as bullet points, annotated bibliographies, etc., will not be processed and can create disparity between the submission highlights and the percentage shown.

What does 'qualifying text' mean?
Sometimes false positives (incorrectly flagging human-written text as AI-generated), can include lists without a lot of structural variation, text that literally repeats itself, or text that has been paraphrased without developing new ideas. If our indicator shows a higher amount of AI writing in such text, we advise you to take that into consideration when looking at the percentage indicated.

In a longer document with a mix of authentic writing and AI generated text, it can be difficult to exactly determine where the AI writing begins and original writing ends, but our model should give you a reliable guide to start conversations with the submitting student.

Foveated HDR: Efficient HDR Content Generation on Edge Devices Leveraging User's Visual Attention

***Abstract*—In recent years, high dynamic range (HDR) content has become increasingly popular since it can represent a wider range of brightness levels compared to traditional low dynamic range (LDR) data. This allows for more details in the bright and shadow regions, leading to more realistic and immersive viewing experience, which is essential for numerous applications on edge devices such as augmented reality/virtual reality (AR/VR) systems. While DNN-based solutions are effective for reconstructing high-fidelity HDR content, due to their high computational demands and memory usage, the DNN-based HDR reconstruction takes up to several seconds to generate one HDR image, making it very challenging to deploy such techniques onto the edge devices.**

Towards this, we propose Foveated HDR to enable more efficient HDR content generation on edge devices by leveraging the inherent foveation nature of human vision system and adaptively distributing the compute resources based on visual importance. Unlike the state-of-the-art works which treat all the pixels equally, we prioritize computational resources for the region of the user's focus and reduce computation for unimportant regions. Specifically, the Foveated HDR can not only dynamically identify the user's focus of view, but also is a user-driven approach that tailors DNN inference speedup to individual user preference. Furthermore, Foveated HDR carefully considers the employed DNN model architecture when performing foveated processing to preserve the quality of the generated HDR content. We implement and evaluate the proposed design using a typical edge GPU SoC (NVIDIA Jetson Orin Nano) board. The experimental results show that Foveated HDR can speedup the DNN inference by 3.xx, which translates to 75% energy saving, with negligible quality degradation.

***Index Terms*—High dynamic range reconstruction, edge devices, efficient, focus of view**

I. Introduction

The human eye perceives a wide range of luminance, encompassing blinding sunlight to moonlit shadows. Capturing this dynamic range has long been a challenge in visual representation, leading to images/videos that often lose details at the extremes. High dynamic range (HDR) imaging has emerged as an effective solution to this issue, thanks to its capability to capture, preserve, and transmit a broad spectrum of luminance. According to [], the global HDR market is projected to rise from $13.76 billion in 2020, to $126.74 billion by 2030. In the meantime, the recent advent and proliferation of edge devices, such as smartphones and VR/AR headsets have led to notable changes in how visual content is consumed, with users increasingly demanding high-quality visual data and immersive experiences on these resource-constrained platforms. Recognizing the potential of HDR imaging to enhance visual experiences, many leading manufacturers such as Samsung and Apple have been actively upgrading their devices with the capability to capture HDR images []. These devices typically achieve HDR imaging by merging/combining multiple low dynamic range (LDR) images captured at different exposure levels. These techniques, though effective for reconstructing HDR content for static scenes, will produce ghosting artifacts when there are large movements in the scene or camera, which is often unavoidable. This has led to increased interest in single-image HDR (SIHDR, also known as inverse tone mapping), which can reconstruct HDR content from only a single LDR input, and is robust to motions.

However, due to limited exposure information contained in a single LDR input, the produced HDR image often lacks details and appears less vivid. To tackle this challenge, recent studies have employed machine learning/deep neural networks to predict details beyond the traditional inverse tone mapping capabilities and enhance the overall quality of HDR image reconstruction [9], [10], [13], [17], [22], [23], [26]. While this opens up the potential for users to experience high-quality HDR on low-end edge devices, the inherent resource constraints of such platforms pose a significant challenge to directly deploying current DNN models, which are typically resource-intensive. For example, based on our experiment, the SingleHDR [23] model (a state-of-the-art DNN model for HDR content reconstruction) requires approximately 4 seconds to generate a single HDR frame in 1080p resolution on a typical edge GPU SoC [6]. Such processing latencies are not suitable for practical use as they will jeopardize user experience and cause fast battery drain, making it critical to optimize such models and thereby enabling high-quality HDR reconstruction on low-end devices.

While there are several prior efforts attempting to optimize DNN-based single image HDR reconstruction via sharing the weights between different modules [22] or reducing the precision for some specific layers [3], however, [22] can only reduce the number of parameters but fail to decrease the computational cost of the DNN model, while [3] requires the machine learning domain knowledge for designing the training the model from scratch. Moreover, these prior works overlook the unique characteristics of the application itself and the associated optimization opportunities. For instance, the reconstructed HDR content will eventually be viewed by the users, given that the users are likely to focus on only a small portion of the visual content (also refer as region of focus, or RoF) [40], one optimization strategy for saving the computation and memory usage is to allocate less amount of computation for the background/non-RoF regions. Motivated by this, instead of treating all the pixels equally (i.e., what has been done by the state-of-the-art works), we propose

to prioritize the computation for regions of focus (which plays a significant role in determining the user experience) and reduce the computation for the background regions via approximations. However, such foveation-based compute reallocation is not straightforward due to the following **challenges**: First, *since the user's focus of view varies across images/video frames, how to* ***adaptively*** *localize the region of focus (RoF) with minimal overhead?* Second, *how to* ***tradeoff*** *between the computational cost and visual quality for the background region to maximize the performance improvement while maintaining the quality?* Third, *processing the RoF and background regions separately may lead to different levels of irradiation, and consequently, creates visual artifacts in the final reconstructed HDR content. How to effectively avoid such visual artifacts to guarantee satisfactory user experience?*

To tackle these challenges, we propose Foveated HDR, a framework for improving the performance and energy-efficiency of DNN-based HDR content reconstruction on edge devices, with the core idea of "leveraging user's visual information to smartly reduce computation for the unimportant regions". As shown in Fig. 1, by employing Foveated HDR, we are able to accelerate the DNN inference execution for HDR reconstruction by 3.6x with minimal quality degradation. To the best of our knowledge, this is the first work that aims to optimize the DNN-based HDR content reconstruction on edge devices via leveraging the users' visual information (i.e., the visual attention). Our major **contributions** are summarized as follows:

- We first conducted a detailed characterization of several popular DNNs for single-image HDR reconstruction to understand their inefficiencies on edge devices, as set our optimization target as *reducing the computation for DNN inference*.
- From a widely-used HDR dataset [28], [29], we identified the foveating nature of the eyes when users view the HDR content, which provides the opportunity to skip/reduce the computation for the "unimportant" portions of the HDR reconstruction (i.e., outside the user's region of focus). Moreover, we observed a clear trade-off between the size of input LDR and the computation demands of the SIHDR DNN models. Such observation offers an optimization chance by first reducing/downscaling the resolution of input LDR, then upscaling the reconstructed HDR to obtain high-resolution HDR content, where the downscaling and upscaling are expected to introduce minimal overheads compared to the expensive DNN inference.
- To utilize these opportunities, we designed and implemented a framework, namely "Foveated HDR", which can adaptively identify the region of focus and facilitate user-driven determination of the downscaling factor for the "unimportant" regions. We also take the target SIHDR DNN model architecture into consideration to eliminate the potential visual artifacts caused by Foveated HDR.
- Finally, we evaluated the proposed Foveated HDR on an edge device – NVIDIA Jetson Orin Nano board [6]. Our experimental results reveal that, compared to the baseline configuration, Foveated HDR can provide 28% reductions in power consumption and 3.78x speedup for DNN inference, which collectively translate to 2.9x end-to-end speedup and 75% total energy savings, with negligible quality drop (i.e., HDR-VDP-2 [25] drops by 0.8).

II. Background and Motivation

In this section, we first briefly introduce the background for HDR imaging in Sec. II-A. We then dive into the issues and potential opportunities associated with the DNN-based HDR reconstruction in Sec. II-B.

A. Background

1) High Dynamic Range Imaging: Dynamic range, which is defined as the ratio between the maximum and minimum luminance in the photography and imaging domain [14], is one of the key metrics that characterize the quality of the visual contents. As depicted in [7], a higher dynamic range can retain more scene details. Typically, the range of luminance that the human eye can handle is 10000:1 in a single view [14], for which the conventional RGB image with 8 bits per channel (also known as low dynamic range (LDR) image), struggles to capture. To bridge this gap and better represent real-world scenes, the high dynamic range (HDR) imaging technique, which aims to produce images or videos with high dynamic range, has been introduced and extensively explored during the past several decades [27], [36], [39]. Unlike LDR images, which store the pixels using 8-bit integers, HDR images utilize 32-bit floating-point data to represent pixel values for each channel and result in 96 bits per pixel. Consequently, HDR images can represent a significantly broader range of luminance.

2) HDR Content Reconstruction: HDR content can be reconstructed from the low dynamic range (LDR) images captured by the conventional camera sensors. This approach can be further categorized into multiple-exposure HDR and single-image HDR. Specifically, in the multi-exposure HDR approach, multiple LDR images with different exposure values are captured within a very short period. These LDR images are then aligned, merged, and fused to generate the HDR images. This technique is commonly used for HDR mode in mobile devices by companies like Apple and Google, for their iPhone and Pixel phones, respectively [11], [16]. While multiple-exposure HDR can effectively generate HDR images for static scenes, it may produce ghost artifacts in the presence of large motions. Moreover, aligning the LDR images, which directly affects the quality of the generated HDR images, requires extra effort. On the other hand, single-image HDR, which constructs the HDR content using only a single LDR image, can overcome the above issues associated with multiple-exposure HDR. However, due to the limited information from a single LDR image, the generated HDR might be of low quality and/or lack details and textures [13]. Fortunately, recent advancements in machine learning have introduced deep neural networks (DNNs) as effective inverse tone mappers for single-exposure HDR reconstruction [9],

(a) Baseline without optimizations. (b) Foveated HDR-2X: achieve 1.9x speedup (c) Foveated HDR-4X: achieve 3.6x speedup

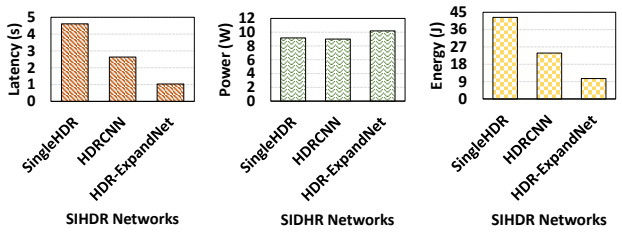
Fig. 1: Qualitative comparison on reconstructed HDR images under various configurations. (a): reconstructed HDR image using SingleHDR [23]; (b): reconstructed HDR image using the proposed optimizations with 2x as target speedup for DNN inference (real speedup=1.9x); (c): reconstructed HDR image using the proposed optimizations with 4x as target speedup for DNN inference (real speedup=3.6x).

[13], [23], [26]. However, these methods, while effective in generating high-quality HDR images, usually involve heavy DNN models and consume significant time and power for inference, as depicted later in Sec. II-B1. Given the unique resource constraints and limited battery budget of edge devices, using these DNN models to generate HDR images on edge devices remains a significant challenge. Although there have been some previous attempts to optimize DNN-based SIHDR for resource-constrained devices [3], [22], however, these works overlooked the application-specific optimization opportunities and require machine learning domain knowledge, which significantly limits the scope of their proposed solutions, as we will discuss in Sec. V.

B. Motivation

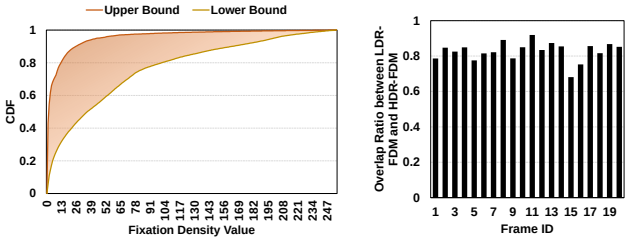
In this section, we first characterize the performance and energy behavior of the single-image HDR (SIHDR) DNNs on an edge GPU SoC, using three popular SIHDR DNN models in Sec. II-B1 to understand their inefficiencies. Following this, we explore the foveation nature of users' gaze direction when viewing the HDR visual contents in Sec. II-B2 and show the potential opportunities for optimizing HDR content generation by leveraging this foveated viewing in Sec. II-B3.

1) Performance Characterization for DNN-based SIHDR: To better understand the performance of the DNN-based HDR content reconstruction, we characterize the latency, power, and energy consumption of three widely-used SIHDR DNN models, including SingleHDR [23], HDRCNN [9] and HDR-ExpandNet [26], on a typical edge SoC platform (NVIDIA Jetson Orin Nano [6]). As shown in Fig. 2a, overall, all three DNN models require at least a few seconds to perform the inference. This extended latency will lead to noticeable processing delays and significantly jeopardize the user's experience. Moreover, as illustrated in Fig. 2b, the power consumption for SingleHDR [23] and HDRCNN [9] is around 9W, with HDR-ExpandNet [26] consuming up to 10W. Considering that the typical use cases such as audio/video playback or phone calls generally consume no more than 1000mW power [5], this level of power consumption is substantially high for edge devices and could lead to faster battery drain and potential overheating



(a) Inference Latency (b) Power Consumption (c) Energy Consmption

Fig. 2: State-of-the-art single-image HDR DNN models profilings.



(a) CDF Envelopes for Fixation Density Values. (b) Overlap Ratio between the FDM of LDR and the FDM of HDR.

Fig. 3: Foveation study. **TODO: Redraw**

issues. The combination of long inference latency and high power consumption results in exceedingly high energy consumption, as depicted in Fig. 2c, which is unacceptable for edge devices. For instance, consider a mobile phone with an average battery capacity of 4000mAh [37] and a voltage of 3.7volts [12], merely processing approximately 1005 frames (only accounts for 33 seconds for 30FPS videos) of inference using HDR-ExpandNet [26] (i.e., the most lightweight model among the tested DNNs), would consume 20% of the battery. This clearly indicates the need for more efficient HDR content generation techniques tailored for edge devices.

2) Foveation in Human Visual System: As discussed in Sec. II-B1, the high energy consumption of the state-of-the-art

(SOTA) DNN models for SIHDR poses a significant challenge for their deployment on edge devices without additional optimizations. This motivates us to explore the techniques that can facilitate the efficient deployment of such models. Note that, while designing an efficient DNN model architecture from scratch could potentially address this inefficiency issue, it requires the developers to have domain-specific knowledge of machine learning and also, substantial resources for training the DNN models. Therefore, our focus in this paper is on improving the execution efficiency of the existing DNN models. One potential approach for achieving this is to leverage the foveation nature of the human visual system (HVS) and reduce computation by allocating less precision to the regions outside the user's fovea. The intuition behind this approach stems from the fact that the spatial resolution of the HVS is highest around the foveation point and decreases rapidly with increasing eccentricity [40]. In fact, foveation-based optimization has been applied in prior works to improve the efficiency for various applications such as AR/VR, cloud gaming, etc. [18], [31], [32], [42]. In order to utilize this foveation feature for optimizing HDR content generation, we need to answer the following two critical questions regarding its validation and feasibility: *1). Will such foveated viewing phenomenon remain when users view HDR content? and 2). Given that a user's visual attention on HDR images is not available until the HDR images are reconstructed, how can we utilize this attention information to optimize the reconstruction process beforehand?*

To verify if such foveation feature remains when viewing HDR content, we analyzed the fixation density maps (FDMs) collected from users viewing HDR images from the HDREye dataset [28], [29] and draw the cumulative distribution function (CDF) envelopes for the fixation density value distributions across all FDMs in Fig. 3a. The density values range from 0 to 255, with a higher fixation density value corresponding to more focus []. As we can observe, over 91% of the fixation density values fall within 30 (which is quite small) for the orange curve, indicating that users primarily focus on a very small portion of the image (i.e., the regions with large fixation density values). Even for the lower envelope (yellow line), over 50% of the fixation density values are within 36, meaning that a substantial area receives minimal attention. These observations confirm the presence of foveality with HDR visual content and validate the potential of foveation-based optimizations for efficient HDR content generation.

To answer the second question, we investigated the correlation between users' fovation on LDR images and the corresponding HDR versions in Fig. 3b. Recall that a higher fixation density value corresponds to more attention from the user, as such, we first define a threshold[1], the regions with fixation density values higher than this threshold are regarded as the foveated regions or regions of focus (RoFs). Following this, we can obtain the foveated regions for both the LDR and HDR images using their FDMs, respectively. Fig. 3b shows the overlap ratio (y-axis) between the foveated regions for LDR and HDR images across different LDR-HDR pairs (x-axis). As can be observed from the figure, the overlap ratio is higher than 80% for most of the frames, suggesting that visual attention is likely to remain the same when users switch from LDR to HDR images. Upon observing this, it is straightforward, but promising, to utilize the obtained visual attention information from LDR images to predict foveated regions for HDR images.

[1]The threshold is set as p50 of the fixation density values of the given FDM.

3) Opportunities: Sec. II-B2 presents the validation and feasibility of leveraging the foveation-based optimizations to improve the performance and energy efficiency of on-device single image HDR. Specifically, we achieve this by prioritizing the inference for RoF and reducing the computation for the unimportant region (i.e., the regions outside user's focus of view). Towards this, one potential strategy is to downscale the resolution of input LDR for these regions. Unlike the traditional classification or object detection models like YOLOv3 [35], MobileNet [15], etc. which usually require a fixed input size, the SIHDR networks allow the variable input size, typically matching the size of the input LDR image. To understand the relationship between the size of the input LDR image and the computational demands, we vary the input size and plot the corresponding GFLOPs values for each input size for Single-HDR [23] in Fig. **??**. As we can observe, the computational complexity decreases dramatically as the input size reduces. Such observation motivates us to first downscale the LDR background region before feeding it to the SIHDR network and upscaling it back after the HDR content is generated. While the downscaling and upscaling can be achieved using conventional imaging processing techniques like bilinear interpolation [19] or cubic interpolation [4], the performance gains and energy savings are expected to be significant.

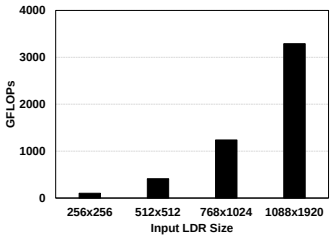


Fig. 4: Input LDR Size vs required inference GFLOPs.

III. Design

We have discussed the potential opportunity of optimizing HDR image reconstruction by reducing the computation of background regions in Sec. II-B, in this section, we will dive into the detailed designs for utilizing such opportunity. We begin by providing the overview of our design in Sec. III-A, followed by a detailed explanation, including the determination of the region of focus (RoF) and the selection of the resolution or downscale factor for the background regions in Sec. III-B. Finally, Sec. III-C addresses the issue of potential quality drops caused by such optimization/approximation to ensure the generated HDR image retains optimal quality.

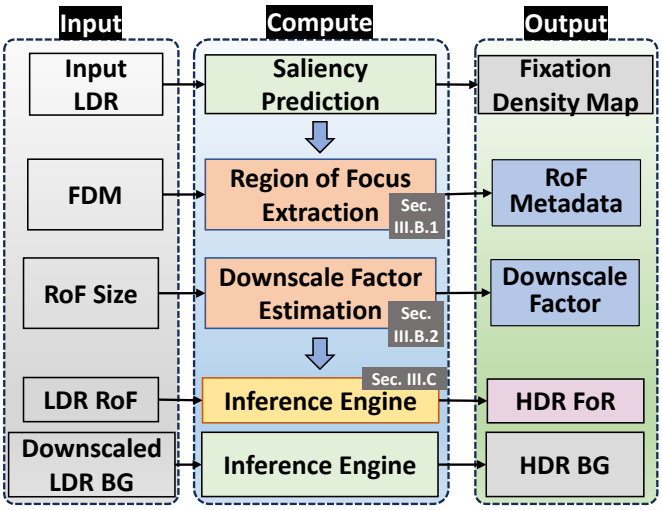


Fig. 5: Foveated HDR overview.

A. Overview

Fig. 5 illustrates the overview the our proposed Foveated HDR, which primarily consists of three steps, including: 1). Saliency prediction using the input LDR to generate the fixation density map (FDM); 2). Leveraging the generated FDM to determine the Region of Focus (RoF) as well as the downscaling factor for the background regions; 3). Downscaling the background accordingly and performing inference for the RoF and the downscaled background.

Saliency Prediction is a relatively well-established field with extensive exploration [2]. Specifically, with the input LDR image, the fixation density map (FDM) can be predicted either using hand-crafted feature extractors [], or using deep learning models []. Due to the recent prevalence of edge/mobile devices, researchers have been tried to design lightweight and mobile-friendly DNN models for saliency predictions [38], [41]. For example, [38] only takes 1.4ms to predict user's fixations, while the model in [41] only requires less than 1GFLOPs to execute. Note that, the saliency prediction is beyond the scope of this work, and we do not claim this as our contribution. Instead, we assume that the FDM has been pre-generated using the lightweight methods in previous works [38], [41] for the next steps (i.e., the contributions of this work).

RoF Determination and Downscaling Factor Selection: Given the obtained fixation density map, we then employ Algo. 1 (discussed in detail later) to extract the metadata for the region of focus (RoF), which includes its center coordinates, as well as the size/dimension. After that, we can leverage this information to estimate the potential inference latency for the RoF, and then select a proper downscaling factor for the background region in conjunction with the user's desired speedup target. The selected background downscaling factor aims to ensure that the total reconstruction time for both RoF and background regions meets, or closely approximates, the user's speedup expectation.

RoF-assisted HDR Reconstruction: The extracted RoF metadata and background downscaling factor allow us to isolate the low dynamic range (LDR) RoF and low-resolution LDR background regions. While simply concatenating the reconstructed HDR RoF and background may appear feasible (as discussed in Section II-B3), this approach may not always yield optimal results. To ensure high-fidelity HDR images, we need to carefully consider the DNN model used for HDR reconstruction and make necessary modifications to ensure the generated HDR image retains optimal quality, as we will elaborate more later in Sec. III-C.

B. Extracting Region of Interest and Adjusting Background Resolution

1) Extract Region of Focus: As we have discussed in Sec. II-B2, higher fixation density values in the FDM indicate more focus from the user. Driven by this observation, we propose a two-step algorithm (Algo. 1) for determining the center coordinates and size for the region of focus (RoF), respectively. The first loop (line#3 - #9) aims to determine the center coordinates for the region of focus (RoF) by traversing the input fixation density map $M_H \times W$, where H and W denote the height and width of the FDM, respectively (matching those of the input low dynamic range image). Specifically, in each iteration, we first extract a region of size $D_L \times D_L$ from the input FDM and calculate its density sum (line#5 - #6), where D_L is the pre-defined lower bound for the RoF size. If this sum exceeds the maximum density sum obtained so far, we will update the maximum density sum (max_sum) as well as the RoF center coordinates ($center$) accordingly (line #7 - #9). After obtaining the RoF center coordinates in the first loop, the average fixation density value within the current RoF region of size $D_L \times D_L$ is calculated (line#10). The RoF size is then iteratively expanded outward from its initial size D_L. As shown in line#12 in Algo. 1, in each iteration, we increase the RoF size by the fixed step size (i.e., S), and extract the corresponding sub-region from the FDM (line#13 - #17). The average fixation density value for the expanded area is then calculated (line#18 - #19). If this average value falls below a threshold compared to the average obtained previously (line #10), this means that the current RoF size is sufficient and has already included regions with smaller fixation density values, which may not be of interest to the users. Therefore, the expansion process is stopped and the corresponding RoF size is returned (line#20 - #21). Otherwise, the RoF continues to expand until reaching the pre-defined upper bound.

2) Determine Background Downscale Factor: As depicted in Sec. II-B3, reducing the resolution of the background region presents an opportunity to save the computation and energy consumption for HDR reconstruction. However, the optimal downscaling factor for the background remains unknown and requires careful selection. While a larger downscaling factor can preserve the reconstructed HDR image quality at the cost of decreased performance improvement, a smaller factor can lead to significant speedups but potentially at the expense of

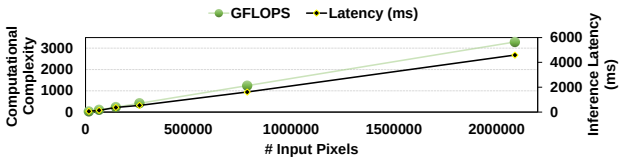
Algorithm 1: Computation of Region of Interest Using Fixation Density Map

Input : $M_{H\times W}$: Fixation density map
Input : $(D_L,\ D_U)$: Lower and upper bounds for the RoI dimension
Input : S: Step size for RoI expansion
Input : T: Threshold for terminating RoI expansion
Output: $center$=$(h,\ w)$: Center coordinates of the RoI
Output: d: Dimension of the RoI

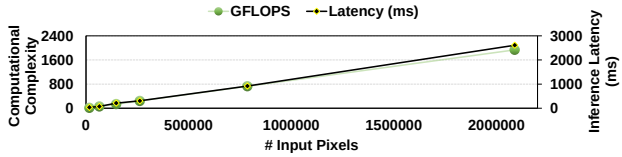
Procedure `ComputeRoI`$(M_{H\times W},\ D_L,\ D_U,\ S,\ T)$
  Init: $max\_sum = 0,\ center = (0,0),\ d = D_L$
  for $h \leftarrow 0$ **to** H **by** D_L **do**
    for $w \leftarrow 0$ **to** W **by** D_L **do**
      $roi = M[h : h + D_L,\ w : w + D_L]$
      $density\_sum = sum(roi)$
      if $density\_sum > max\_sum$ **then**
        $max\_sum = density\_sum$
        $center = (h + D_L/2, w + D_L/2)$
  $avg_0 = max\_sum/D_L^2$
  while $d < D_U$ **do**
    $d = d + S$
    $start\_h = max(0,\ center[0] - d/2)$
    $start\_w = max(0,\ center[1] - d/2)$
    $end\_h = min(H,\ center[0] + d/2)$
    $end\_w = min(W,\ center[1] + d/2)$
    $roi = M[start\_h : end\_h,\ start\_w : end\_w]$
    $density\_sum = sum(roi)$
    $avg = (density\_sum - max\_sum)\ /\ (d^2 - D_L^2)$
    if $avg/avg_0 < T$ **then**
      return $center$, d
  return $center$, d

quality. Rather than randomly selecting a downscaling factor without any prior knowledge, we propose a user-driven design and allow the user to determine this factor. It is important to note that, we do not expect users to provide the exact downscaling factor, which would be user-unfriendly. Instead, the only required parameter from user-end is their desired speedup[2] by employing our Foveated HDR proposal. Based on this target speedup as well as the estimated execution latency required for inferencing the RoF, we can predict the downscaling factor for the background region such that the desired speedup can be achieved. Towards this, a critical question arises – *how do we estimate the execution latency for RoF with only its center coordinates and size given?* To answer this question, we analyze the relationship between RoF size

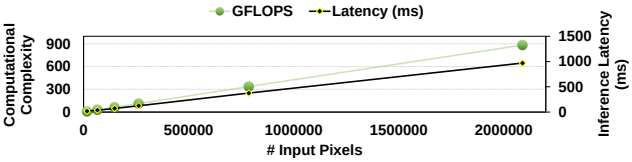
[2]We are referring to the speedup for the forward pass computation during DNN inference, while there are extra overheads for searching for RoF and downscaling/upscaling the background region included in our proposed Foveated HDR, the final end-to-end HDR content generation speedup is expected to be lower than the target speedup specified by the user.



(a) SingleHDR



(b) HDRCNN



(c) HDR-ExpandNet

Fig. 6: Computational complexity and inference latency for LDR images with varying sizes.

TABLE I: Tested input low dynamic range (LDR) image size and the number of pixels.

LDR Size	256×256	384×384	512×512	768×768	1088×1920
# Pixels	65536	147456	262144	786432	2088960

and the inference latency through profiling as shown in Fig. 6. Specifically, the left y-axis shows the required GFLOPs while the right y-axis shows the inference latency on Jetson Orin Nano edge board [6], with various input LDR image sizes as listed in Table. I. As we can observe, for all of the three tested SIHDR DNN models, both the computational complexity (i.e., GFLOPs) and the execution latency increase *linearly* along with the total number of pixels of the input LDR image. This observation motivates us to use the following equation (Eq. 1) to estimate the execution latency for the RoF, where the S_{RoF} and $S_{Input\ Image}$ denote the area for the input RoF and the original image, respectively, and are equivalent to their total number of pixels.

$$T_{RoF} = \frac{S_{RoF}}{S_{Input\ Image}} * T_{Input\ Image} \tag{1}$$

Similarly, we can estimate the inference latency for reconstructing the downscaled background using Eq. 2:

$$\begin{aligned} T_{Background} &= \frac{S_{Background}}{S_{Input\ Image}} * T_{Input\ Image} \\ &= factor^2 * T_{Input\ Image} \end{aligned} \tag{2}$$

where $factor$ is the downscaling factor for the background region, and ranges from 0 to 1. Combining the definition for inference speedup (Eq. 3) with Eq. 1 and 2, we can derive the downsampling factor for the background region using Eq. 4.

(a) Straightforward concatenation. (b) Model-aware concatenation

Fig. 7: Eliminate the visual artifacts by taking the DNN model-architecture into consideration to maintain quality.**TODO: Update figures**

$$\frac{T_{Input\ Image}}{T_{RoI} + T_{Background}} = speedup \tag{3}$$

$$factor = \sqrt{\frac{1}{speedup} - \frac{S_{RoI}}{S_{Input\ Image}}} \tag{4}$$

C. How to Preserve Quality?

As illustrated in Fig. 5, Foveated HDR requires to process the RoF and background regions separately. While this approach is effective in most cases (as we will show in Sec. IV-C4), sometimes it may not work. For instance, as shown in Fig. 7a, where both the RoF and background are inferenced using the HDR-ExpandNet [26] DNN model while the background is downscaled by a factor of 0.5 (RoF remains at original resolution). This will save 62% inference computation, however, as shown in the figure, visible artifacts appear around the edges where the RoF and background connect. To understand the reason causing these artifacts, we delve into the detailed architecture of HDR-ExpandNet [26]. Specifically, HDR-ExpandNet consists of global, local, and post-processing modules. The global module considers the entire image, while the local module focuses on generating local details. Simply feeding the RoF to HDR-ExpandNet allows it to treat the RoF as the global image. When the visual characteristics (e.g., luminance) between the RoF and the whole input LDR are significant, the generated HDR RoF (based on a "fake" global image, i.e., the RoF itself) may not be compatible with the HDR generated for the whole image. This incompatibility leads to visible artifacts at the edges. To address this issue, we propose to skip the global module computation for the RoF inference. Instead, we reuse the output of the global module obtained during background inference. This will ensure that the global image information is captured and utilized when processing the RoF, thereby eliminating the artifacts. As shown in Figure 7b, the proposed modifications towards the HDR-ExpandNet [26] effectively remove the artifacts observed in Figure 7a. To summarize, while applying the same SIHDR DNN model to both RoF and background regions offers an effective solution to eliminate artifacts, we also need to carefully consider the SIHDR model architecture, especially, when handling models like HDR-ExpaneNet which utilizes the global context for HDR construction.

TABLE II: Dataset

Name	HDR-Eye	Bistro_01	Carousel	Fishing	Playground
#Frames	36	151	339	371	222
Resolution	1920×1080	1920×1080	1920×1080	1920×1080	2048×1080

IV. Evaluation

In this section, we compare our proposed Foveated HDR design against the baseline setting where all of the HDR frames are generated with equal computational resources allocated to all pixels. We evaluate four critical metrics relevant to HDR content generation applications: execution latency, power and energy consumption, as well as the quality of the generated HDR images. Towards this, we first describe the experimental platform, datasets (Sec. IV-A), and configurations (Sec. IV-B), and then analyze the collected results in Sec. IV-C.

A. Methodology

1) Experimental Platform: To evaluate the effectiveness of our proposed Foveated HDR, we use the NVIDIA Jetson Orin Nano [6] board as the experimental platform, which is an edge development board, and is well-known to simulate the realistic edge development environment. Specifically, it is equipped with a 1024-core NVIDIA Ampere GPU with 32 Tensor Cores, a 6-core Arm Cortex-A78AE v8.2 64-bit CPU, 8GB 128-bit LPDDR5 memory and uses a 256GB MicroSD card as the storage. In our evalutaion, we config the Jetson Orin Nano board to operate in 15W compute mode and monitor its power consumption using the tegrastats tool [30].

2) HDR Dataset: We employ two datasets, namely HDR-Eye [28], [29] and DML-iTrack-HDR [1], [8], for our evalutation. The HDR-Eye is an image dataset consists of 46 HDR images and their corresponding LDR version. Each of the LDR and HDR images has its corresponding fixation density map (FDM) showing the users' view of focus when viewing the HDR/LDR images. [3] The DLM-iTrack-HDR dataset [1] contains several HDR videos and the corresponding fixation density maps showing the user's visual attention for each frame when viewing the HDR videos. However, it does not include the corresponding LDR videos or the FDMs for the LDR videos. To address this issue, we manually generated LDR versions of the HDR videos using a tone mapper, as commonly done in prior works like []. Furthermore, since our previous results in Sec. 3 indicate large overlaps between the focus areas when users watch LDR and HDR contents, we decide to use the fixation density maps collected on the HDR videos to simulate the FDMs for the LDR inputs. Table. II summarizes the details for each dataset and videos used in this evaluation, including the total number of frames and the resolution.

[3]Only 36 of the LDR/HDR image pairs are considered as valid for evaluation since they contain all the necessary componenets such as the LDR images, HDR images, and FDM or fixation points for the LDRs.

3) SIHDR DNN Models: To demonstrate the generalizability of our proposed foveated HDR design, we evaluated it on three widely-used SIHDR DNN models with varying characteristics and computational demands: SingleHDR [23], HDR-CNN [9], and HDR-ExpandNet [26]. The SingleHDR [23] and HDRCNN [9] are implemented using TensorFlow v1 [24], while HDR-ExpaneNet [26] utilizes the Pytorch v1.12 []. We leverage the pretrained weights provided by the original authors for all models.

B. Design Configurations

We evaluate our proposal against the baseline setting under two different configurations:

Baseline: We implement the baseline HDR content generation on an edge GPU board (NVIDIA Jetson Orin Nano [6]), where every HDR frame is generated using the SIHDR DNN inference and all the pixels are treated equally, regrdless of their importance to the user's visual attention.

Speedup_2X: As discussed in Sec. III-B2, Foveated HDR is user-driven, where the resolution for background region is adaptively decided based on the user-specified speedup. Specifically, after determining the computational resources required for the regoin of focus (RoF) using the algorithm described in Sec. III-B1, the background is downscaled by a factor before performing the inference, so that we can save the computation and achieve the target speedup specified by the user. We first set the target speedup as 2, which aims to provide a moderate speedup with minimal quality loss.

Speedup_4X: In this configuation, we set a larger target speedup to achieve greater latency savings and energy consumption reduction. This setting prioritizes reducing latency and energy consumption over the HDR content quality and is suitable for applications where content is consumed quickly, such as fast playback of videos, and users are less sensitive to quality changes in the background due to their focus on the RoF. For example, mobile devices with limited battery budgets may benefit from this configuration , as it saves significant energy while maintaining acceptable visual quality for the users' attention focus.

C. Results

We compare the execution latency, power and energy consumption, as well as the HDR content quality in Fig. 8 - Fig. 10 and Table. III, when using various design configurations as described in Sec. IV-B.

1) Execution Latency:

Latency for Inference Computation: We first compare the inference latency consumed by the SIHDR models during forward pass computation under difference configurations. Specifically, as shown in Fig. 8a, the baseline SingleHDR [23] model requires an average of 4.66s to generate one HDR frame, while with our proposed Foveated HDR optimization, the inference latency can be decreased to 2.41s and 1.34s per HDR frame for Speedup_2X and Speedup_4X configurations, respectively. These reductions translate to speedups of 1.93x and 3.48x compared to the baseline, respectively. Note that, these achieved speedups are slightly lower than the user-specified speedups (i.e., 2x and 4x). This discrepancy arises because the input LDR image needs padding to ensure its dimensions are a multiple of 64 before being fed into the SingleHDR model [23], which will add additional computations. For the Speedup_2 configuration, the moderate target speedup allows the background region to remain relatively large. Therefore, the additional computational cost introduced by inferencing the padded regions has a minimal impact on the inference latency, yielding the real speedup closely matches the expected speedup estimated using Equ. 3 in Sec. III-B2. However, for the Speedup_4X configuration, the significant background downscaling leads to a substantial reduction in total computation. As a result, the extra computation introduced by inferencing the padded background region becomes non-negligible, since it now accounts for a larger portion of the overall computation. This explains why the real speedup is lower than the configured speedup. Similarly, as shown in Fig. 9a, the inference latency for HDRCNN [9] can be reduced from 2.66s/frame to 1.34s/frame (1.985x speedup) for Speedup_2X and 0.73s/frame (3.64x speedup) for Speedup_4X configuration, respectively. While for HDR-ExpandNet [26] (Fig. 10a), our proposed Foveated HDR is able to reduce the inference latency for generating one HDR frame from 1.04s in baseline setting to 0.54s and 0.28s for Speedup_2X and Speed_4X configurations, respectively, corresponding to speedups of 1.93x and 3.71x.

Latency for End-to-end HDR Generation: Fig. 8b, Fig. 9b and Fig. 10b compare the end-to-end execution time for generating one HDR frame using the baseline configuration and our proposed Foveated HDR design. While our approach introduces additional overhead due to the Region-of-Focus (RoF) search, LDR background downscaling, and HDR background upscaling (shown in the yellow bars), this overhead is minimal compared to the performance gains achieved. Specifically, the Foveated HDR overhead accounts for approximately 1.6%, 3.0% and 8.2% of the baseline end-to-end inference latency for SingleHDR [23], HDRCNN [9], and HDR-ExpandNet [26] models, respectively.

2) Power Consumption: To study the impact of our proposal in terms of power consumption, we measure the power consumed during HDR content generation for different configurations in Fig. 8c, Fig. 9c and Fig. 10c. As we can observe, compared with the baseline inference, the Foveated HDR can achieve 10%, 9% and 18.5% power saving for SingleHDR [23], HDRCNN [9] and HDR-ExpandNet [26], respectively, under the Speedup_2X configuration. These power savings primarily stem from two aspects: 1. the amount of computation allocated for the background region is reduced, which leads to lower power consumption by the compute units (e.g., CPU/GPU); and 2). downscaling the LDR background translates to a smaller input size for the SIHDR model, which reduces data movement and memory consumption and leads to further power savings in memory operations. Moreover, with the more aggressive downscaling employed by the Speedup_4X configuration, we can achieve an additional 7%

- 10% reduction in power consumption.

3) Energy Savings: As shown in Fig. 8d, compared to the baseline, the Foveated HDR generation approach only consumes 47.9% energy on average for the SingleHDR [23] model, under the Speedup_2X configuration. The energy saving can be further increased to 74.9% with the aggressive downscaling of background region in Speedup_4X configuration. Similar trends can also be observed for the HDRCNN [9] and HDR-ExpandNet [26] models in Fig. 9d and Fig. 10d, where the Speedup_2X configuration can yield 51.1% and 51.0% energy saving, while Speedup_4X can deliver 74.5% and 75.2% energy saving for HDRCNN [9] and HDR-ExpandNet [26], respectively.

4) Quality: To study the quality impact of our proposed Foveated HDR, we plot the HDR-VDP-2 [25] for the three tested models under various configurations in the right axis of Fig. 8b-10b. Specifically, HDR-VDP-2 [25], or High Dynamic Range Visual Difference Predictor, is an objective method for predicting the visibility of differences in HDR images, and has been widely used by the previous works for measuring the quality of HDR images. Note that, as discussed in Sec. III-C, Foveated HDR is a model-aware framework which takes the SIHDR model architecture into consideration for maintaining the quality. Towards this, we modify the compute flow for HDR-ExpandNet [26] following the logic in Sec. III-C, while the SingleHDR [23] and HDRCNN [9] remain the orginal model architectures since they do not include the "global module". As illustrated in Fig. 8b, with our proposal, on average, the HDR-VDP-2 only drops by 0.27 for the 2X configuration. Even for the 4X configuration, the quality degradation is still negligible (i.e., 0.73). Similar trend can also be observed in Fig. 9b and Fig. 10b, which clearly demonstrates the effectiveness of our proposal in preserving the HDR content quality.

We also use the Photomatix commad line tool [33] to convert the HDR images to LDR versions, and quantitatively compare their quality with the LDR images converted from the ground truth HDR images in Table. III. We measure and compare three major metrics critical to LDR visual data, including peak-signal-noise-ratio (PSNR, higher is better), structured similarity index metric (SSIM, higher is better) and learned perceptual image patch similarity (LPIPS, lower is better). As shown, for the tested workloads, with 2X configuration, our proposed Foveated HDR can deliver same quality images with the baseline setting. Even for 4X configuration, the PSNR and SSIM value degradation is minimal, while the LPIPS increases by a moderate value, indicating that users may not want to choose such setting if they prefer the image quality more than the performance saving.

V. Related Work

A. DNN-based SIHDR

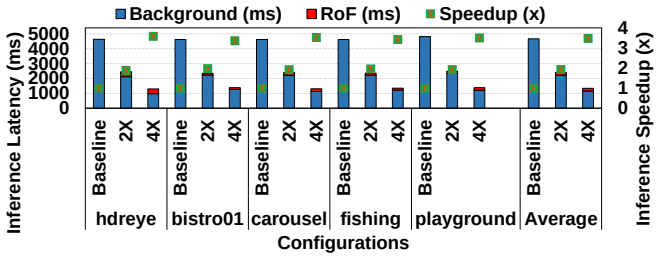
Deep neural networks (DNNs) have been extensively utilized for single-image HDR content reconstruction. For example, HDRCNN [9] leverages the conventional neural network (CNN) to predict the lost information due to sensor saturation, and thereby is capable of producing the HDR image from an arbitrary LDR input. In [26], the authors propose a 3-branch network architecture to handle the local, medium, and global details for HDR imaging, respectively. Marnerides et al. achieve the HDR reconstruction via modeling and reversing the LDR image formation pipeline, where three specialized CNNs are designed to simulate the specific aspects of the reversal process and therefore, recover the missing details in under- or over-exposed regions. FHDR [17] employs a feedback network with a recurrent neural network (RNN) to refine low-level features iteratively using high-level features. Other works like [34] utilizes generative adversarial networks (GANs) to synthesize HDR predictions. Another indirect approach for SIHDR content reconstruction is to first generate multiple LDRs with various exposures using DNNs, then merge these LDRs to produce the final HDR image. There are several representative works using this approach such as [20], [21]. Specifically, in Deep chain HDRI [21], a CNN with dilated convolutional layers is employed to infer LDR images with different exposures and illumination from a single LDR image. Similarly, Le et al. [20] utilize neural networks to invert the camera response and reconstruct pixel irradiance before synthesizing multiple exposures and hallucinating details in under- and over-exposed regions from the single input image. Although these prior works can produce high-quality HDR images, however, due to the compute-intensive nature of the employed DNN models, it is very challenging to deploy them to the edge devices.

B. Lightweight SIHDR

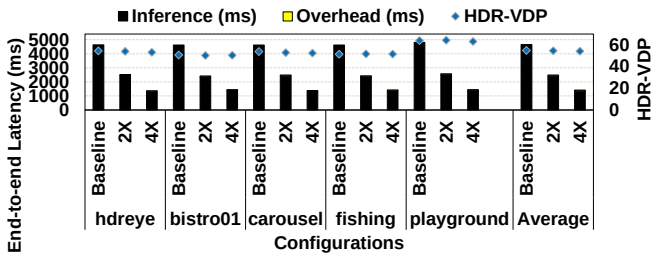
To tackle the performance and/or memory usage issues, several prior efforts have attempted to optimize the DNN-based single-image HDR imaging. For example, in [22], the authors propose to share the parameters between the DNN modules, thereby reducing the memory usage. However, this approach can only yield less number of parameters, while the computational costs remain high. Borrego-Carazo et al. design a mixed quantization network where the precision of some layers are reduced from float32 to int8, while in [3], most of the computations are moved to the pre- and post-processing steps, and the DNN is only utilized for fusion purpose. These prior works are effective in reducing the computation and memory burden to some extent, however, they ignore the unique characteristic of the application itself (i.e., the users are likely to focus on only partial of the HDR contents) and the potential opportunities behind such property (i.e., instead of uniformly distributing the computation resources across the entire image, such resources can be allocated based on the importance of the image regions). Note that, our proposed Foveated HDR is orthogonal with these prior works, therefore, can be applied on top of these works to facilitate even more efficient HDR content reconstruction.

VI. Conclusion

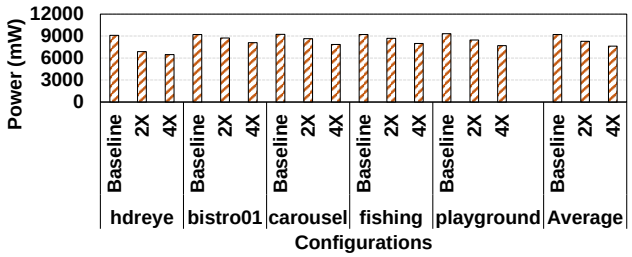
High dynamic range (HDR) content can represent a much wider range of luminance than the low dynamic range (LDR)



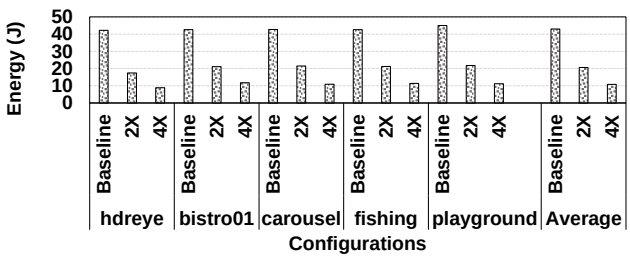
(a) Inference Latency



(b) End-to-end Latency and Quality



(c) Power Consumption



(d) Energy Consumption

Fig. 8: Main results for SingleHDR [23].

TABLE III: Quality

		hdreye			bistro_01			carousel			fishing			playground		
SIHDR Model	Config.	PSNR(↑)	SSIM(↑)	LPIPS(↓)	PSNR(↑)	SSIM(↑)	LPIPS(↓)	PSNR(↑)	SSIM(↑)	LPIPS(↓)	PSNR(↑)	SSIM(↑)	LPIPS(↓)	PSNR(↑)	SSIM(↑)	LPIPS(↓)
SingleHDR	**Baseline**	**23.09**	**0.82**	**0.16**	**18.23**	**0.78**	**0.35**	**19.54**	**0.80**	**0.18**	19.55	**0.87**	**0.22**	**23.47**	**0.92**	**0.08**
	2X	22.71	0.81	0.20	17.14	0.76	0.38	18.56	0.77	0.23	20.00	**0.87**	0.24	23.38	**0.92**	**0.08**
	4X	22.45	0.80	0.27	16.82	0.75	0.40	17.63	0.73	0.29	**20.93**	0.86	0.28	23.17	0.90	0.11
HDRCNN	**Baseline**	18.30	**0.70**	**0.25**	**29.41**	**0.85**	**0.30**	27.32	0.86	**0.11**	**31.50**	**0.93**	**0.18**	27.60	**0.95**	0.05
	2X	**18.37**	**0.70**	0.26	27.31	**0.85**	0.32	27.53	**0.87**	**0.11**	30.66	0.92	0.20	27.32	**0.95**	**0.04**
	4X	18.36	0.69	0.31	28.03	**0.85**	0.35	**27.54**	0.86	0.13	30.72	0.92	0.23	**27.75**	0.94	0.06
HDR-ExpandNet	**Baseline**	**21.41**	**0.79**	**0.21**	**21.28**	**0.81**	**0.35**	**22.57**	**0.84**	**0.16**	**23.66**	**0.89**	**0.23**	**26.72**	**0.93**	**0.07**
	2X	21.24	**0.79**	0.22	20.93	0.80	0.36	22.36	**0.84**	0.17	23.46	**0.89**	0.26	26.49	**0.93**	**0.07**
	4X	20.91	0.78	0.28	20.98	0.80	0.38	22.19	0.83	0.20	23.34	0.88	0.28	25.54	0.92	0.08

images, thus providing more realistic and immersive viewing experiences. Due to such property, HDR visual content is increasingly gaining attention from various emerging applications on the edge devices, such as the AR/VR systems. While the DNN-assisted HDR image reconstruction is capable of producing high-fidelity HDR content, however, due to the complex computations and large data movements it involves, deploying such technique to the resource-constrained edge devices remains a significant challenge. Towards this, this paper explores the foveation nature when users viewing the HDR content and proposed Foveated HDR to adaptively reduce the computation for the unimportant regions (i.e., the regions outside the user's focus of view). By incorporating the SIHDR model architecture, Foveated HDR can largely avoid the visual artifacts in the reconstructed HDR images and thereby minimize the quality degradation. Our evaluation results demonstrate that Fovated HDR can speedup the DNN inference by 3.6x, which translates to 3.3x end-to-end HDR reconstruction speedup and 75% energy saving, with negligible quality loss.

REFERENCES

[1] "Dml-itrack-hdr," http://dml.ece.ubc.ca/data/DML-iTrack-HDR/, 2023, accessed: 2023-12-25.

[2] A. Borji, "Saliency prediction in the deep learning era: Successes and limitations," *IEEE transactions on pattern analysis and machine intelligence*, vol. 43, no. 2, pp. 679–700, 2019.

[3] J. Borrego-Carazo, M. Ozay, F. Laboyrie, P. Wisbey, and C. del Vallès, "A mixed quantization network for computationally efficient mobile inverse tone mapping," *arXiv preprint arXiv:2203.06504*, 2022.

[4] P. Breeuwsma, "Cubic interpolation," https://www.paulinternet.nl/?page=bicubic, 2023, accessed: December 25, 2023.

[5] A. Carroll and G. Heiser, "An analysis of power consumption in a smartphone," in *2010 USENIX Annual Technical Conference (USENIX ATC 10)*, 2010.

[6] N. Corporation, "Jetson orin modules and developer kits," https://www.nvidia.com/en-us/autonomous-machines/embedded-systems/jetson-orin/, 2023.

[7] D. Djudjic, "WHY DYNAMIC RANGE IS MORE IMPORTANT THAN MEGAPIXELS," https://www.diyphotography.net/dynamic-range-important-megapixels/, 2017.

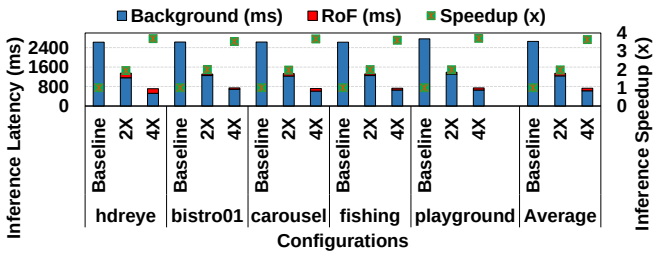
[8] Y. Dong, E. Nasiopoulos, M. T. Pourazad, and P. Nasiopoulos, "High dynamic range video eye tracking dataset," in *2nd International Conference on Electronics, Signal processing and Communications, Athens*, 2015.

[9] G. Eilertsen, J. Kronander, G. Denes, R. K. Mantiuk, and J. Unger, "Hdr image reconstruction from a single exposure using deep cnns," *ACM transactions on graphics (TOG)*, vol. 36, no. 6, pp. 1–15, 2017.

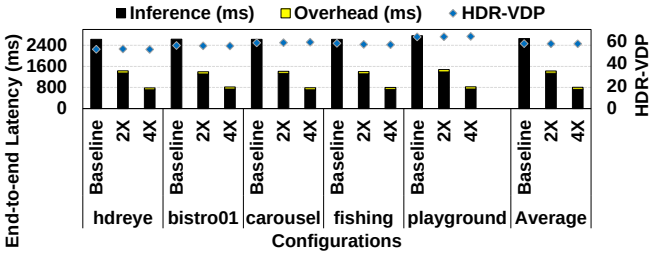
[10] Y. Endo, Y. Kanamori, and J. Mitani, "Deep reverse tone mapping." *ACM Trans. Graph.*, vol. 36, no. 6, pp. 177–1, 2017.

[11] M. Ernst and B. Wronski, "Hdr+ with bracketing on pixel phones," https://blog.research.google/2021/04/hdr-with-bracketing-on-pixel-phones.html, 2021.

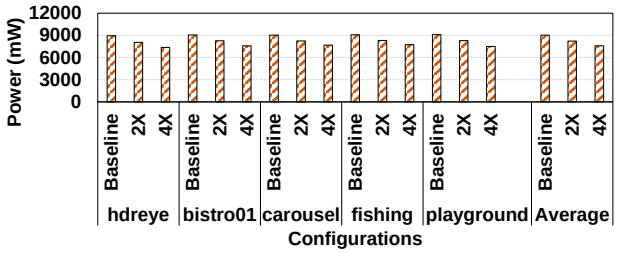
[12] J. Frank, "How many voltage does a phone have?"



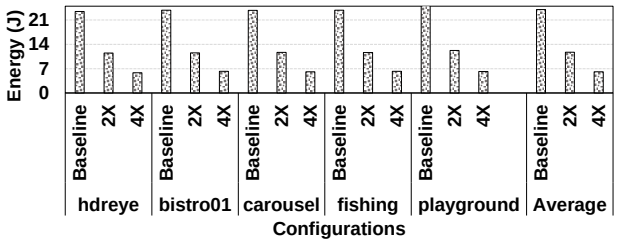
(a) Inference Latency



(b) End-to-end Latency and Quality



(c) Power Consumption



(d) Energy Consumption

Fig. 9: Main results for HDRCNN [9].

http://tinyurl.com/mrxexmtw, 2022, [Online; accessed 25-December-2023].

[13] C. Guo and X. Jiang, "Lhdr: Hdr reconstruction for legacy content using a lightweight dnn," in *Proceedings of the Asian Conference on Computer Vision (ACCV)*, December 2022, pp. 3155–3171.

[14] HDRsoft, "What is the unit of dynamic range?" https://www.hdrsoft.com/resources/dri.html#unit, 2023.

[15] A. G. Howard, M. Zhu, B. Chen, D. Kalenichenko, W. Wang, T. Weyand, M. Andreetto, and H. Adam, "Mobilenets: Efficient convolutional neural networks for mobile vision applications," *arXiv preprint arXiv:1704.04861*, 2017.

[16] A. Inc, "Adjust hdr camera settings on iphone," https://support.apple.com/guide/iphone/adjust-hdr-camera-settings-iph2cafe2ebc/17.0/ios/17.0, 2023.

[17] Z. Khan, M. Khanna, and S. Raman, "Fhdr: Hdr image reconstruction from a single ldr image using feedback network," in *2019 IEEE Global Conference on Signal and Information Processing (GlobalSIP)*. IEEE, 2019, pp. 1–5.

[18] J. Kim, Y. Jeong, M. Stengel, K. Aksit, R. A. Albert, B. Boudaoud, T. Greer, J. Kim, W. Lopes, Z. Majercik *et al.*, "Foveated ar: dynamically-foveated augmented reality display." *ACM Trans. Graph.*, vol. 38, no. 4, pp. 99–1, 2019.

[19] E. J. Kirkland and E. J. Kirkland, "Bilinear interpolation," *Advanced computing in electron microscopy*, pp. 261–263, 2010.

[20] P.-H. Le, Q. Le, R. Nguyen, and B.-S. Hua, "Single-image hdr reconstruction by multi-exposure generation," in *Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision*, 2023, pp. 4063–4072.

[21] S. Lee, G. H. An, and S.-J. Kang, "Deep chain hdri: Reconstructing a high dynamic range image from a single low dynamic range image," *IEEE Access*, vol. 6, pp. 49 913–49 924, 2018.

[22] ——, "Deep recursive hdri: Inverse tone mapping using generative adversarial networks," in *proceedings of the European Conference on Computer Vision (ECCV)*, 2018, pp. 596–611.

[23] Y.-L. Liu, W.-S. Lai, Y.-S. Chen, Y.-L. Kao, M.-H. Yang, Y.-Y. Chuang, and J.-B. Huang, "Single-image hdr reconstruction by learning to reverse the camera pipeline," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2020.

[24] G. LLC, "Tensorflow 1," http://tinyurl.com/y9aej4p3, 2023.

[25] R. Mantiuk, K. J. Kim, A. G. Rempel, and W. Heidrich, "Hdr-vdp-2: A calibrated visual metric for visibility and quality predictions in all luminance conditions," *ACM Transactions on graphics (TOG)*, vol. 30, no. 4, pp. 1–14, 2011.

[26] D. Marnerides, T. Bashford-Rogers, J. Hatchett, and K. Debattista, "Expandnet: A deep convolutional neural network for high dynamic range expansion from low dynamic range content," in *Computer Graphics Forum*, vol. 37, no. 2. Wiley Online Library, 2018, pp. 37–49.

[27] J. J. McCann and A. Rizzi, *The art and science of HDR imaging*. John Wiley & Sons, 2011.

[28] E. MMSPG, "Hdr-eye: dataset of high dynamic range images with eye tracking data," https://www.epfl.ch/labs/mmspg/downloads/hdr-eye/, 2023, accessed: 2023-12-25.

[29] H. Nemoto, P. Korshunov, P. Hanhart, and T. Ebrahimi, "Visual attention in ldr and hdr images," in *9th International Workshop on Video Processing and Quality Metrics for Consumer Electronics (VPQM)*, no. CONF, 2015.

[30] NVIDIA Corporation, "tegrastats Utility," "http://tinyurl.com/2hzsnzh6", 2019.

[31] A. Patney, J. Kim, M. Salvi, A. Kaplanyan, C. Wyman, N. Benty, A. Lefohn, and D. Luebke, "Perceptually-based foveated virtual reality," in *ACM SIGGRAPH 2016 emerging technologies*, 2016, pp. 1–2.

[32] A. Patney, M. Salvi, J. Kim, A. Kaplanyan, C. Wyman, N. Benty, D. Luebke, and A. Lefohn, "Towards foveated rendering for gaze-tracked virtual reality," *ACM Transactions on Graphics (TOG)*, vol. 35, no. 6, pp. 1–12, 2016.

[33] Photomatix, "Photomatixcl documentation," https://www.hdrsoft.com/support/doc/PhotomatixCL/reference.html, 2023.

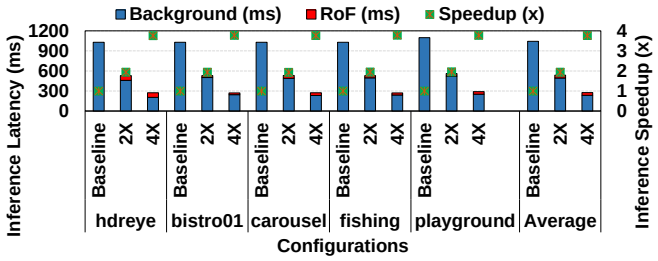
[34] P. Raipurkar, R. Pal, and S. Raman, "Hdr-cgan: single ldr to hdr image translation using conditional gan," in *Proceedings of the Twelfth Indian Conference on Computer Vision, Graphics and Image Processing*, 2021, pp. 1–9.

[35] J. Redmon and A. Farhadi, "YOLOv3: An Incremental Improvement," *CoRR*, 2018.

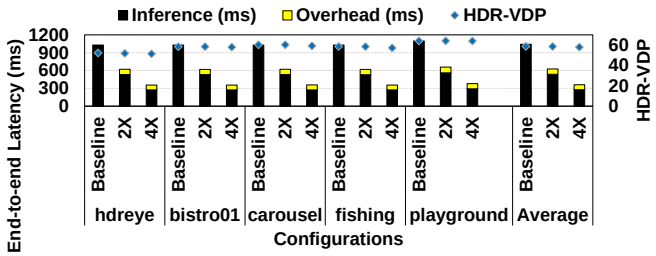
[36] E. Reinhard, W. Heidrich, P. Debevec, S. Pattanaik, G. Ward, and K. Myszkowski, *High dynamic range imaging: acquisition, display, and image-based lighting*. Morgan Kaufmann, 2010.

[37] A. Sharma, "You told us: This is the battery capacity of most of your smartphones right now," https://www.androidauthority.com/smartphone-battery-size-poll-results-1221015/, April 2021.

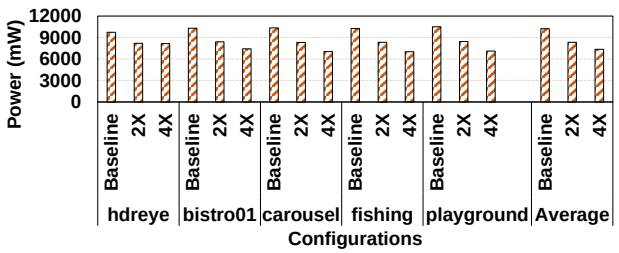
[38] S. Velusamy, R. Radarapu, A. Hegde, and N. Kothari, "A light-weight human eye fixation solution for smartphone applications," in *Proceed-*



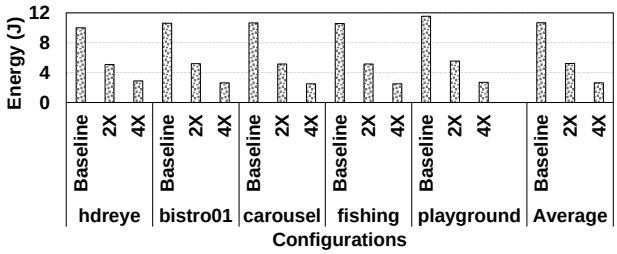
(a) Inference Latency



(b) End-to-end Latency and Quality



(c) Power Consumption



(d) Energy Consumption

Fig. 10: Main results for HDR-ExpandNet [26].

ings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, 2023, pp. 5674–5679.

[39] L. Wang and K.-J. Yoon, "Deep learning for hdr imaging: State-of-the-art and future trends," *IEEE transactions on pattern analysis and machine intelligence*, vol. 44, no. 12, pp. 8874–8895, 2021.

[40] Z. Wang and A. C. Bovik, "Embedded foveation image coding," *IEEE Transactions on image processing*, vol. 10, no. 10, pp. 1397–1410, 2001.

[41] S. Zabihi, H. R. Tavakoli, A. Borji, and E. Mansoori, "A compact deep architecture for real-time saliency prediction," *Signal Processing: Image Communication*, vol. 104, p. 116671, 2022.

[42] S. Zhao, H. Zhang, C. S. Mishra, S. Bhuyan, Z. Ying, M. T. Kandemir, A. Sivasubramaniam, and C. Das, "Holoar: On-the-fly optimization of 3d holographic processing for augmented reality," in *MICRO-54: 54th Annual IEEE/ACM International Symposium on Microarchitecture*, 2021, pp. 494–506.