

Supplementary Material of “A Mixed Quantization Network for Computationally Efficient Mobile Inverse Tone Mapping”

Juan Borrego-Carazo^{1*,2}

juan.borrego@uab.cat

Mete Ozay¹

m.ozay@samsung.com

Frederik Laboyrie¹

f.laboyrie@samsung.com

Paul Wisbey¹

p.wisbey@samsung.com

¹ Samsung Research UK

Staines-upon-Thames

United Kingdom

² Universitat Autònoma de Barcelona

Cerdanyola del Vallès

Spain

1 Overview

In the present document, we provide more insights with regards to data, evaluation procedure, ablation studies, and inference examples.

2 Datasets, Architecture and Hyperparameters of the MQN

2.1 Training Datasets

We employ 4 different datasets for training: Ward, Funt, PFSTools, and HDRPlus. Most of them only contain high bit depth images, and thus input LDR images have to be recreated as detailed in Section 4.1, Experimental Setup of the main paper. The four tone mapping operators (TMOs) used are: Drago [1], Mantiuk [2], Reinhard [3] and Exposure obtained from the OpenCV library [4]. For data augmentation, we use a batch of ground-truth HDR images. For each image, a TMO is chosen randomly and the TMO is applied on the image with random parameters.

2.1.1 Ward and PFSTools

The Ward dataset [5] is a collection of 33 HDR images originally intended to compare different HDR formats (OpenEXR, Radiance RGBE and XYZE, 24-bit and 32-bit LogLuv TIFF, and others). The PFSTools [6] is a collection of 8 HDR images of both outside and interior scenes.

2.1.2 Funt

The Funt collection [10] is a set of 105 HDR images built by bracketing 9 differently exposed LDR images. The LDR images have a difference of 1 EV between them, a rate of capture of 5 seconds and the f-stop is not fixed. The Final HDR images are created from raw LDR bursts by alignment and filtering.

2.1.3 HDRPlus

HDR+ [11] is a content enhancement pipeline dataset consisting on 3640 bursts (made up of 28461 images in total) resulting from the Google HDR+ system. The dataset also contains an intermediate DNG burst merge image and the 8-bit image resulting from the pipeline. We use the merged burst image in DNG format as our HDR output and do not use the 8-bit images defined as a result of the pipeline or the raw input images.

2.2 Test Datasets

For testing our method, we choose three datasets that, contrarily to our training data, have LDR-HDR paired images, thus enabling a fair evaluation.

2.2.1 HDR Eye

HDR Eye [12] consists of 46 LDR-HDR pairs taken with the Sony DSC-RX100 II, Sony NEX-5N, and Sony 6000 cameras. The HDR images are generated by combining LDR images with exposures (-2.7, -2, -1.3, -0.7, 0, 0.7, 1.3, 2, 2.7). Evaluation is performed using images with size 256x256 as suggested in [13].

2.2.2 HDR Real

HDR Real [14] is a photographic dataset specially designed for extreme HDR contexts. It consists of 1838 LDR-HDR pairs taken by amateur photographers, employing 42 different cameras, using different exposures and covering the whole range of lighting conditions: from near pitch-black to extremely saturated images. We perform evaluation using 256x256 images, following [14]. Instead of preprocessing the the datasets as employed in [14], we employ the dataset directly without preprocessing.

2.2.3 RAISE 1K

The RAISE-1K [15] consists of a subset of 1000 RAW-TIFF images pairs selected from the original RAISE dataset. Originally intended for digital forensics, it contains high-resolution images captured in diverse scenarios: indoor, outdoor, man-made and natural. Following [15], we consider using unprocessed RAW images (with 12- or 14-bit depth) as ground truth HDR images, and the TIFF images as 8-bit LDR input images. We convert the RAW images to .hdr format and the TIFF images to JPEG. For evaluation we downsize images to a quarter of their original size respecting proportion ratios, resulting in images which are approximately 720p.

3 Evaluation Procedure

For evaluation of methods using Peak Signal Noise Ration (PSNR) and Structural Similarity (SSIM), we use tone mapped images and predictions. Inspired by [10], and unlike our training procedure where we use OpenCV tone mapping operators, we tone map images using four tone mapping operators from the Photomatix suite [13] for evaluation: (1) detailed, (2) balanced, (3) realistic and (4) photographic.

Regarding HDR-VDP metric evaluation, we use version 2.2.2 for evaluation. Our predictions take values from $[0, 1]$ (relative luminance). We rescale them to a display range of 1000 cd/m^2 and align the 0.01 and 0.99 percentiles of both prediction and ground truth. For a fair comparison, we use the same parameters as utilized in [10, 14] to obtain the *pixel-per-degree* parameter, which are 24-inch display, 0.5 distance and 1080p display resolution.

With regards to latency measurements, we test models on both a desktop GPU, NVIDIA GTX 1080 Ti, and a mobile platform, Samsung Note 20 Exynos 990. The latency calculations are performed on the desktop platform taking into account the process between the reading of the LDR image and the output of the final HDR image, that is to say that the reading and writing computational costs are not considered. So, for those methods, such as [13] or ours, that use the input mixed with the output to create the final HDR image, the combination procedure is also included in the latency computation. In the mobile platform, latency is tested through the native benchmark application for the arch64 architecture offered by Tensorflow [16], always running on a CPU with 4 threads for 300 runs to average the results using images of size 256x256.

3.1 Training and Network Details

The entire training takes approximately 5 days on a machine with an Intel Core i7-6850K and an Nvidia GTX 1080 Ti.

For the encoder of the backbone, we use a MobileNetV2 (MBV2) with width factor $\alpha = 0.35$ and skip connections applied at the activations of layers (1, 3, 6, 13) and output of batch normalization at layer 16. Further details of each of the layers are given in Figure 9.

4 Additional Quantitative Evaluation and Analyses

4.1 Extended State-of-the-art Comparison

4.1.1 Histograms of HDRVDP Q Values

In Figure 1, we plot the distribution of the Q value scores obtained from HDRVDP for HDRCNN [13], SingleHDR [10], ExpandNet [14], and our method for all three test sets: HDR-Eye, HDR-Real and Raise 1K. As the results show, our method performs similarly or better than HDRCNN and ExpandNet, and slightly worse than SingleHDR, even though our method is purposed for fast inference and not for performance quality, as other competing methods are.

4.1.2 Quantitative Evaluation on Tone Mapped Images

We also evaluate our method and competing state-of-the-art methods using PSNR and SSIM. To provide HDR images for these metrics, we tone map the images with the use of Pho-

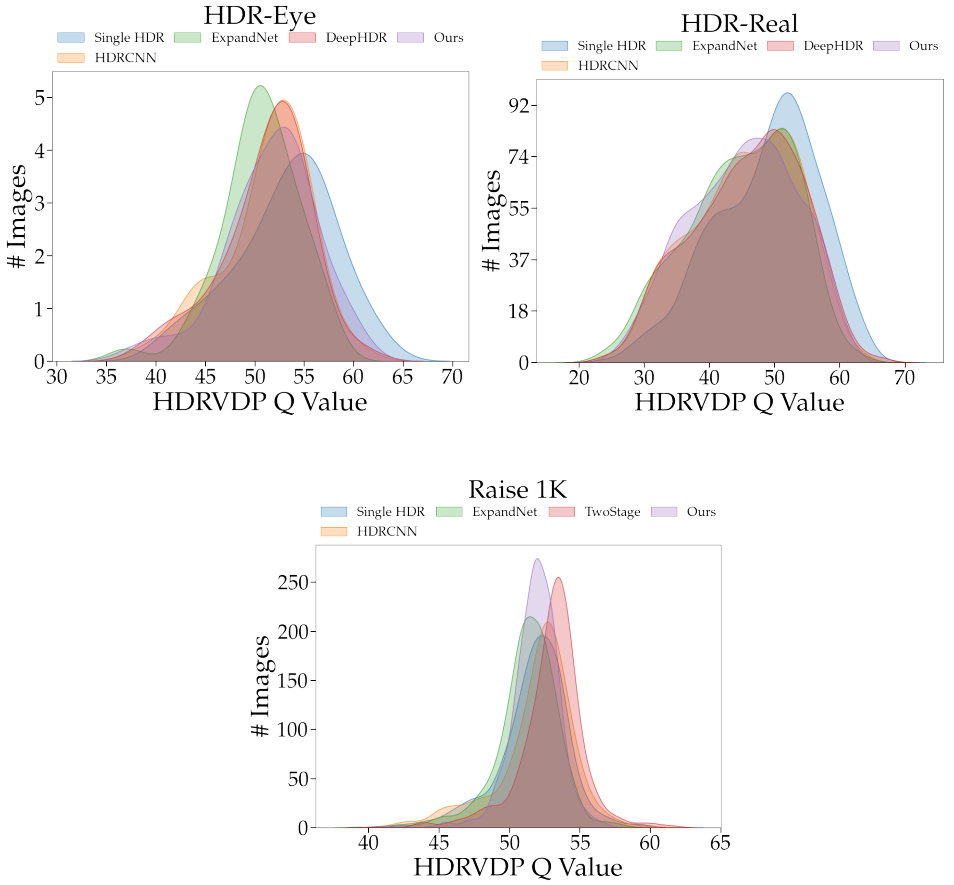


Figure 1: Distribution of HDRVDP Q score values for three test datasets and 5 best competing methods in each case.

tomatix suite and four tone mapping operators: Detailed, Balanced, Realistic and Photographic. Results are shown in Table 1. The results show that our method performs better than ExpandNet and HDRCNN in HDR Eye and Raise, and always after Single HDR. However, note that our method is 100x faster than SingleHDR both on mobile and GPU, and almost 10x faster than ExpandNet on mobile.

4.1.3 Comparative Analysis of Latency and Accuracy

Our method is intended for fast inference, while all competing methods focus on accuracy. We compare the latency and accuracy trade-off in Figure 3. We can see that our model is almost 100x faster than SingleHDR, while providing only 3% accuracy loss, and while being 10x faster than ExpandNet it achieves a 1 Q point more on accuracy.



Figure 2: Visual comparison of our method and state-of-the-art ITM methods. From left to right: input LDR, HDRCNN [15], ExpandNet [16], SingleHDR [17] and our MQN.

Table 1: Comparison with state-of-the-art single image HDR reconstruction methods for PSNR and SSIM performance metrics. Values reproduced with the same evaluation criteria and original code. Blue indicates the best and red indicates the second best accuracy.

Model	HDR Eye		HDR Real		Raise 1K	
	PSNR-T	SSIM-T	PSNR-T	SSIM-T	PSNR-T	SSIM-T
HDRCNN [15]	18.82 \pm 3.49	0.7754 \pm 0.1044	16.53 \pm 5.65	0.6378 \pm 0.2303	17.25 \pm 2.81	0.5950 \pm 0.1213
FHDR [16]	20.30 \pm 5.40	0.7794 \pm 0.1897	16.47 \pm 5.83	0.6436 \pm 0.2305	17.68 \pm 3.67	0.5653 \pm 0.1315
ExpandNet [17]	19.85 \pm 2.96	0.7854 \pm 0.0954	16.24 \pm 5.99	0.6175 \pm 0.2564	16.58 \pm 2.77	0.5264 \pm 0.1306
SingleHDR [18]	21.70 \pm 4.50	0.8259 \pm 0.1244	21.16 \pm 5.33	0.7409 \pm 0.2150	15.12 \pm 3.44	0.5688 \pm 0.1253
DeepHDR [19]	19.38 \pm 3.38	0.7723 \pm 0.1131	16.49 \pm 6.15	0.6252 \pm 0.2591	17.22 \pm 2.86	0.5861 \pm 0.1243
TwoStage [20]	17.97 \pm 4.16	0.7519 \pm 0.1057	14.57 \pm 4.52	0.5660 \pm 0.2235	19.12 \pm 2.81	0.6162 \pm 0.1257
HDRUnet [21]	18.58 \pm 4.11	0.7724 \pm 0.1139	14.59 \pm 4.73	0.5778 \pm 0.2478	17.70 \pm 3.07	0.5989 \pm 0.1263
Ours Best	19.97 \pm 4.21	0.7990 \pm 0.1160	15.95 \pm 5.76	0.6162 \pm 0.2436	17.71 \pm 2.73	0.5776 \pm 0.1150

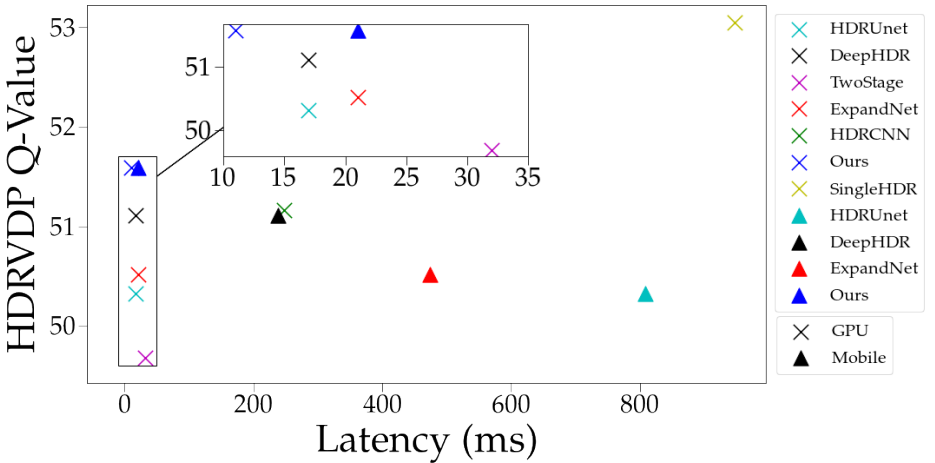


Figure 3: Latency and accuracy trade-off comparison with the HDR Eye dataset between competing methods and our proposed solution. Accuracy is measured by HDR-VDP Q value score. Latency is computed both in the mobile (CPU) and GPU platforms.

Table 2: Results for the different quantization schemes. Latency is measured in the deployment platform (SN20E990) with 4 CPU threads and averaged over 300 rounds.

Q. Scheme	L. BA (ms)	L. HP (ms)	PSNR-T	SSIM-T
MQ	11.52 ± 0.82	9.63 ± 0.12	21.25 ± 3.11	0.8782 ± 0.052
Int 16	857.67 ± 18.2	-	17.23 ± 3.87	0.7612 ± 0.1016

4.1.4 Extended Qualitative Comparison

In Figure 2, we show additional examples of predictions of our model as well as a comparison with state-of-the-art models. We see that our model is able to reconstruct well both under- and over-exposed regions. Our model shows similar accuracy to ExpandNet and shows slightly less reconstruction capacity on over-exposed regions compared to SingleHDR. Further, our model is 10x and 100x faster than ExpandNet and SingleHDR respectively.

4.1.5 Analyses for Extreme Over-exposed Cases

In Figure 4, we illustrate the difficulty of ITM for some of the samples belonging to the HDR Real with three extremely over-exposed inputs. As seen, all networks struggle to recreate the ground truth, failing in most of the content.

4.2 Extended Ablation Studies

4.2.1 Int16 Scheme Comparison

An alternative solution to the MQ scheme, already available in some network optimization suits [24], is quantizing network parameters to 8-bit integers and the activations to 16-bit integers. We compare both schemes and present the results in Table 2 using the same network with the exception of the IN blocks which have been substituted by Batch Normalization blocks due to incompatibilities in the inference framework. The results show that there is a substantial loss in accuracy when the *int*-16 scheme is used. In the case of latency, we observe a substantial increase in latency, probably due to an issue of a lack of suitability of such quantization schemes with the deployment platform or the lack of suitable implementations in the deployment suite. All in all, aside from latency measurements, analyses on accuracy of models show the benefits of our MQN in contrast to the aforementioned scheme which has 8-bit integer parameters and 16-bit integer activations.

4.2.2 Analyses of Loss Functions

In addition, the effect of training models using the perceptual loss on predictions can be observed in Figure 5. As the results show, training models using perceptual loss helps learning feature representations of color coherency as well as color details, further enhancing the quality of the image. Examples are the structural coherency in the color-checker (row 1) or the sky color coherency (row 4).

To provide additional information about the training process, in Figure 6 we show the training and validation loss evolution of all combination of losses found in Table 1 in Main paper.

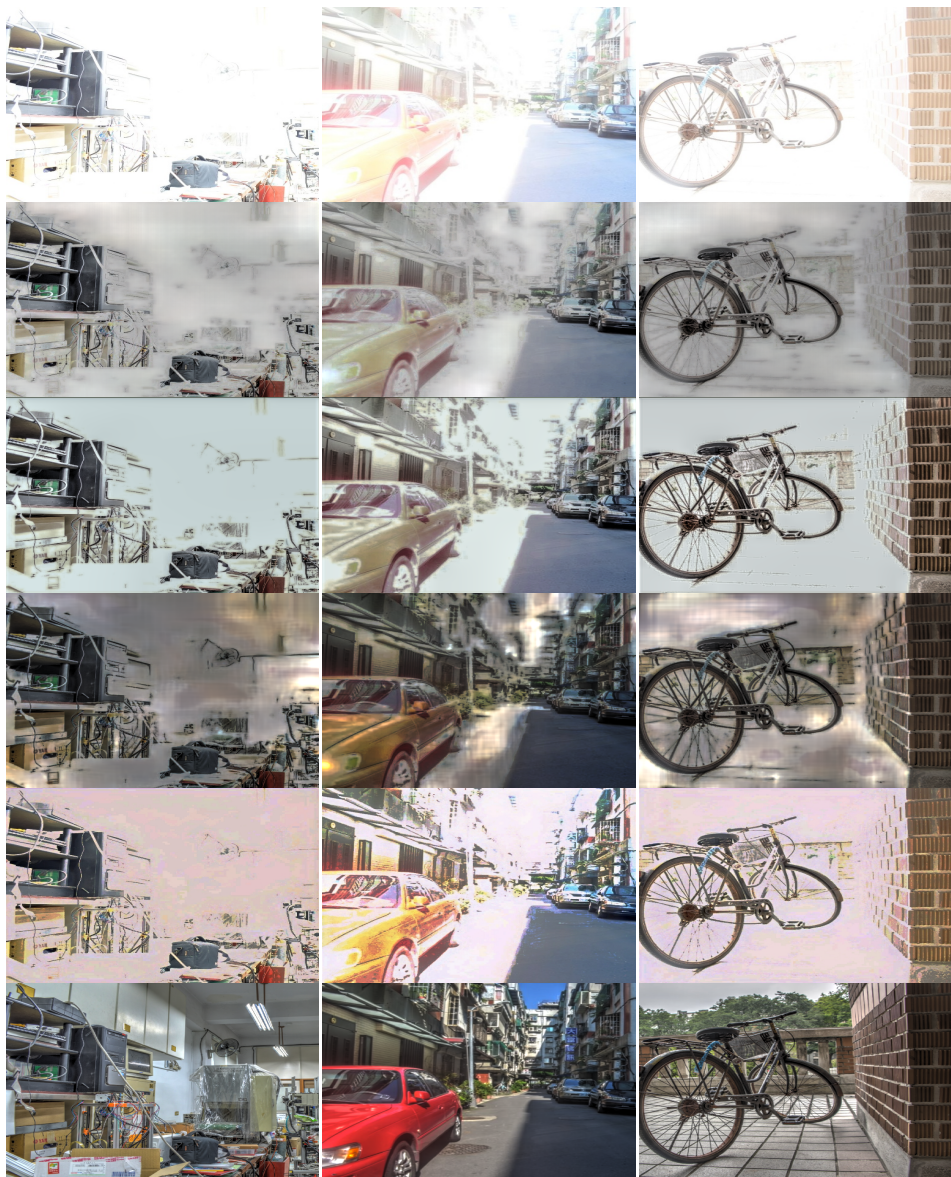


Figure 4: Three examples of extreme cases of over exposure in HDR Real dataset. Order from top to bottom is: input, HDRCNN [10], ExpandNet [11], SingleHDR [12], our method and ground truth.

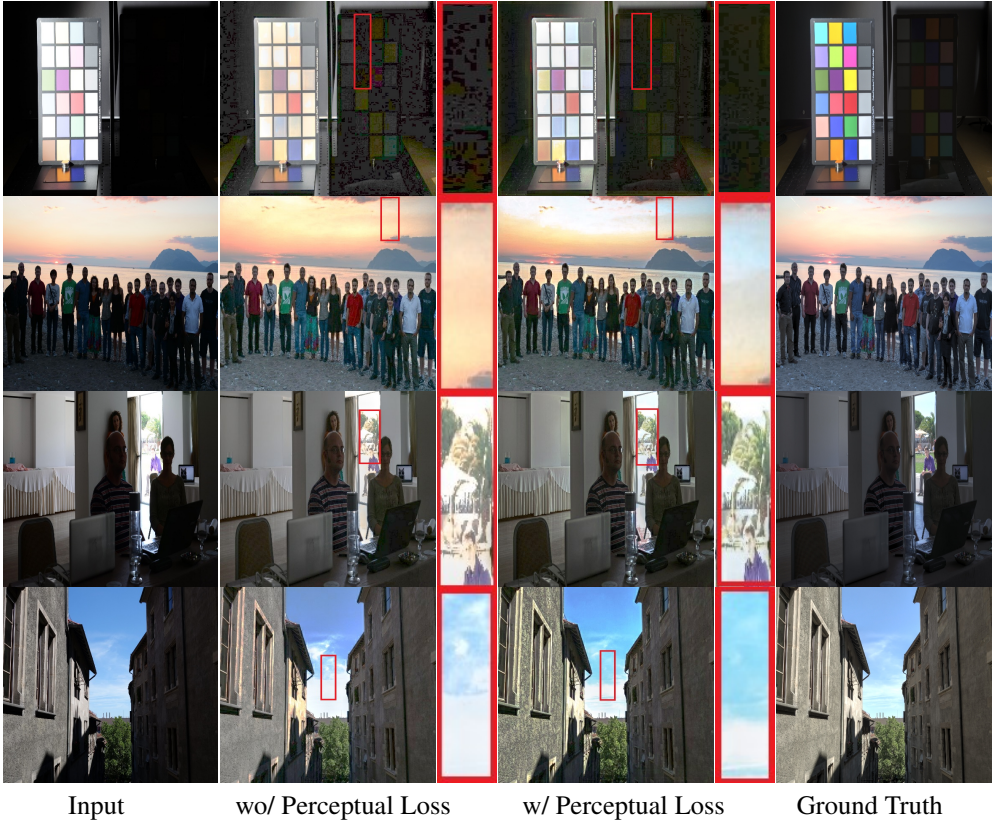


Figure 5: Visual comparison of the effect on predictions by including perceptual loss as a training component. It helps models learn representations of structural color coherency and improve color details.

5 Additional Implementation Details

5.1 Attention Mechanisms

As stated in Section 3.1 of the main paper, we analyzed accuracy of the proposed MQN using three types of gated attention mechanisms. First, at the end of the first IRLB block in the decoder, we add Spatial Attention (SA) [23] gated blocks. We define SA blocks by

$$\mathbf{O}_f = ((\sigma_1 \circ C_1)(I_f)) \odot I_f \quad (1)$$

where O_f and I_f are the input and output respectively with f channels. C_1 denotes a convolution with a filter and kernel of size 1, σ_1 is a sigmoid activation, and \odot indicates function composition.

To improve the previous attention mechanism, we add channel information through a depthwise convolution in parallel to the SA mechanism, which results in the channel spatial attention (CSA) mechanism defined by

$$\mathbf{O}_f = ((\sigma_1 \circ D_f)(I_f)) \odot (((\sigma_1 \circ C_1)(I_f)) \otimes I_f) \quad (2)$$

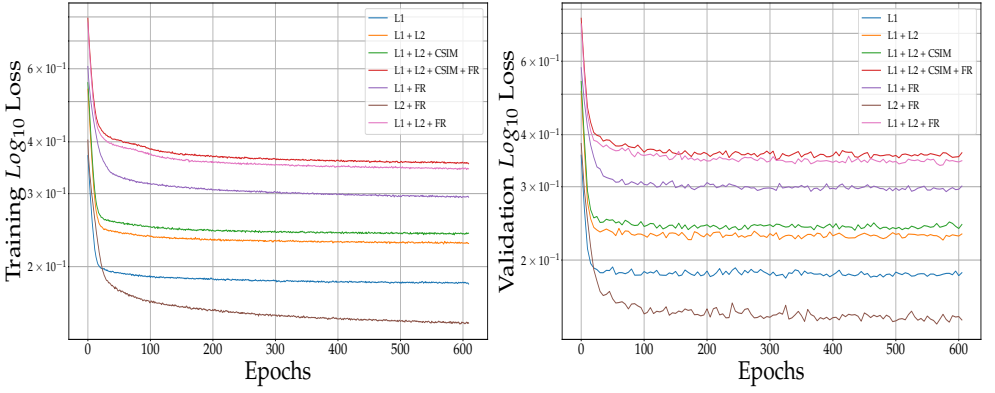


Figure 6: Training (left) and validation (right) loss plots for each loss combination established in Table 1 (Right) in Main Paper.

where D_f is the depthwise convolution with f filters. Finally, although with a higher computational cost due to pooling mechanisms, we also test channel attention (CA) blocks [24]. This operation, inspired by both residual layers and gated attention is defined by

$$\mathbf{O}_f = ((\sigma_1 \circ C_f \circ \sigma_2 \circ C_{f'} \circ GP)(\mathbf{I}_f)) \odot \mathbf{I}_f \quad (3)$$

where C_f denotes convolution with f filters and kernels of size 1, where $f' = f \cdot r$ and r is the reduction ratio of the attention mechanism. Finally, σ_2 is the ReLU function and GP denotes global pooling. All three attention mechanisms are depicted in Figure 7.

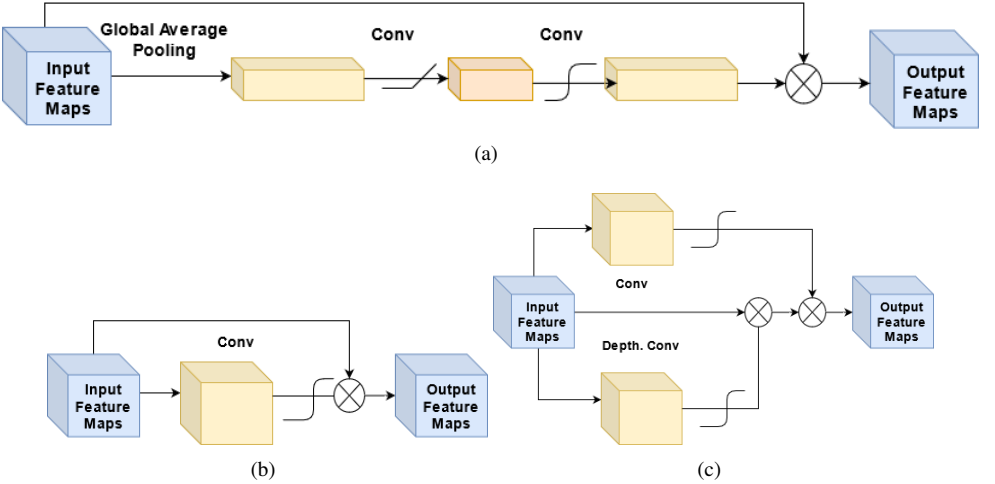


Figure 7: Depiction of (a) Channel Attention (CA) block, (b) Spatial Attention (SA) and Channel Spatial Attention (SCA) block. *Conv* refers to a standard convolution, a soft sigmoid form denotes the sigmoid activation, the rectilinear symbol denotes ReLU activation and the \otimes denotes element wise product.

5.2 Mixed Precision vs Mixed Quantization

Mixed precision¹ [25] is a methodology that uses floating point data types with different bit width (16 and 32, among other specifications) to train faster and with better convergence. Each activation function and operation has a different range, and each weight takes values from different ranges. Hence, it is enough to use 16-bit for some of them, while others need higher resolution.

However, an MQN model does not use a mixed precision training scheme. Instead, MQN uses two different types of quantization schemes: full post-training static 8-bit integer quantization² [26] and dynamic quantization³. First, we train the model in full float precision. Once we have trained the model, we quantize the Base Architecture (BA) to 8-bit integer post-training quantization. Hence, all the BA model has 8-bit weights and 8-bit activations.

Since we aim to recover images with a higher precision for ITM with fast inference, we use dynamic quantization for the high precision (HP) head: it quantizes the weights to 8 bits but activations have 32-bit floating point precision. For this purpose, we employ two different types of post-training quantization to be able to obtain fast inference but still have high precision output.

5.3 Description of the Supplementary Video

As an addition to the complementary material, we apply our method to a video sequence from Unvanquished [27] gameplay and a cityscape at night. In the first video inference is presented in the form of a moving split screen with the original input LDR on the left side and the tonemapped HDR prediction provided by MQN on the right side. In the second video, the original LDR image is in the upper part while the HDR prediction is in the lower part. As applied to all images presented in the article, the frames (images) of the video have been tone mapped to adapt them to commodity non-HDR-capable displays.

5.4 Scaled Images

In this section, we show the images that appear in the main paper, but in a higher scale for a more comfortable visualization.

¹Tensorflow Mixed Precision API

²Tensorflow Post Training 8-bit Integer Quantization

³Tensorflow Post Training Dynamic Range Quantization



Figure 8: **Mobile HDR reconstruction from a single LDR image.** Mixed quantization and efficient blocks help reduce the computational requirements of HDR reconstruction and enable its deployment to mobile platforms, achieving a latency of ≈ 21 ms on a Samsung Note 20 Exynos 990.

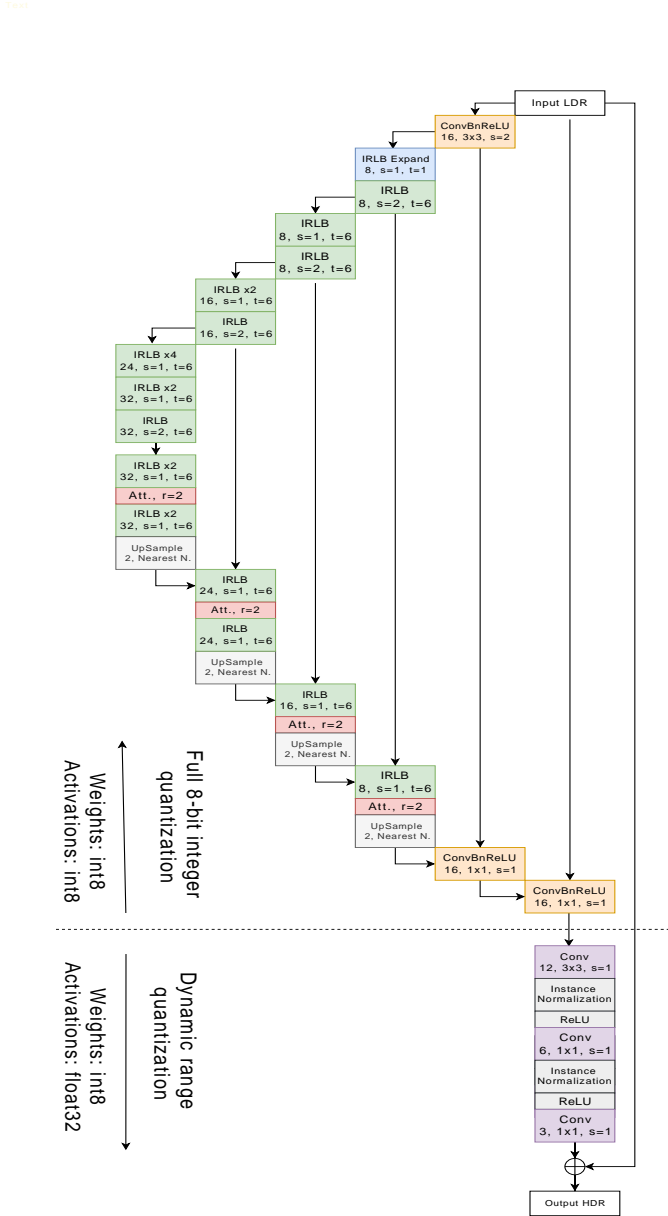


Figure 9: Illustration of the base encoder-decoder architecture and high precision head that conform the MQN. The first number indicates filters, s stands for stride, t denotes the expansion ratio in the Inverted Residual Linear Bottleneck (IRLB) blocks, r is the ratio in the attention blocks. Kernel size is indicated when appropriate by $k \times k$. The dotted line indicates the separation between the fully quantized architecture and the dynamically quantized head.

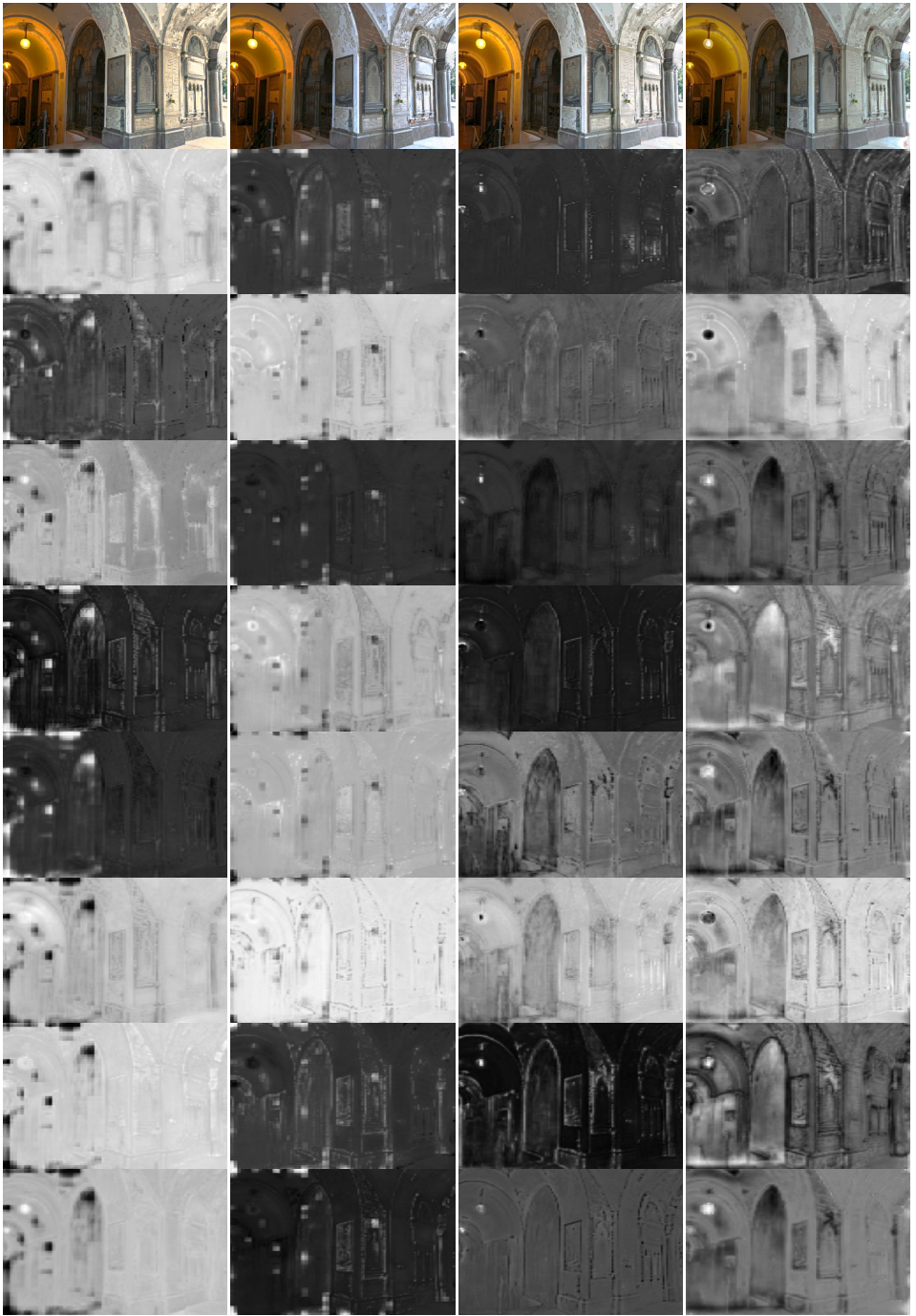


Figure 10: Illustration of the feature maps learned after the last attention mechanism. Columns in order correspond to: no attention mechanism, SA, CSA and CA. Rows correspond to prediction and feature maps respectively for the first and the rest.

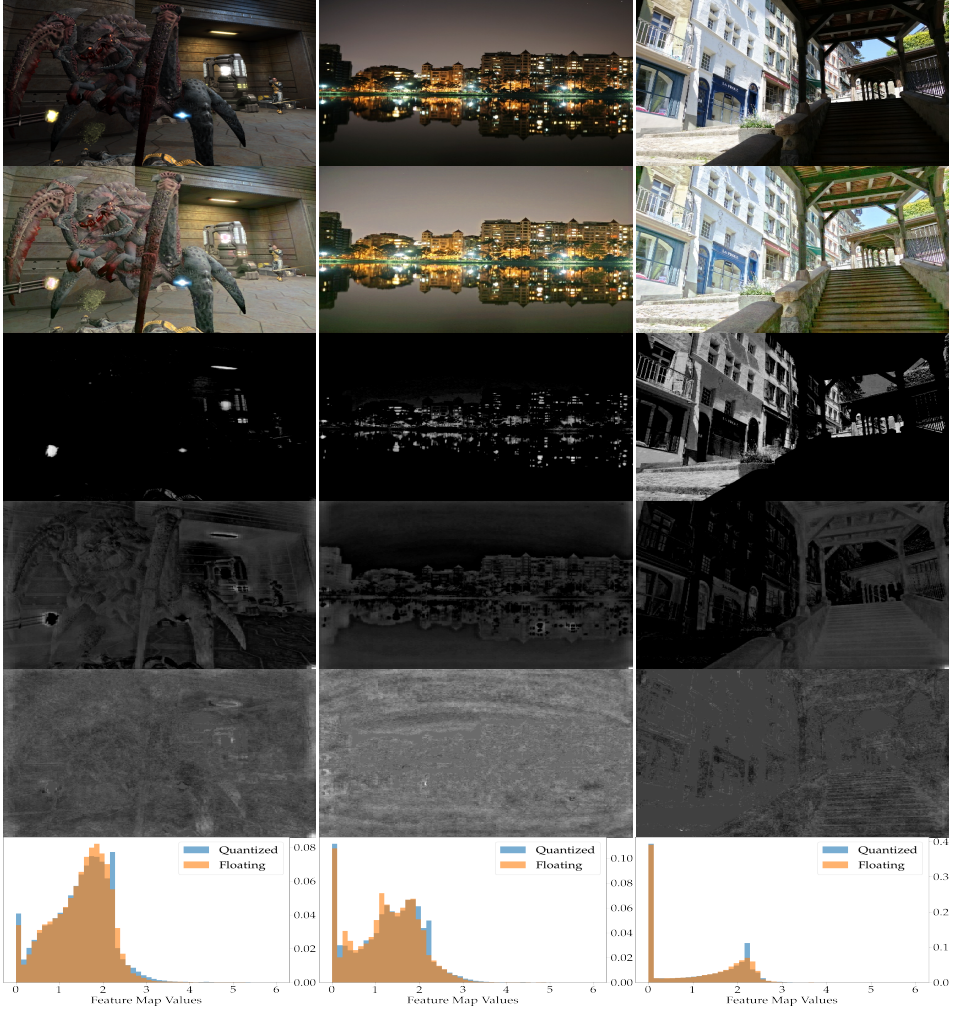


Figure 11: Analyses of the quantized $Q(\mathbf{f})$ and floating point $F(\mathbf{f})$ features \mathbf{f} learned by the MQN at the ConvBnReLU3 layer at Figure 5 using three sample input images (first row) with predictions \hat{H} (second row). Visualization of the first channel f_1 and the second channel f_2 of \mathbf{f} (third and fourth rows, respectively), and the difference map $\Delta f_2 = \|Q(f_2) - F(f_2)\|_1$ (fifth row). We show the probability mass function (PMF) of $Q(f_2)$ and $F(f_2)$ (last row).

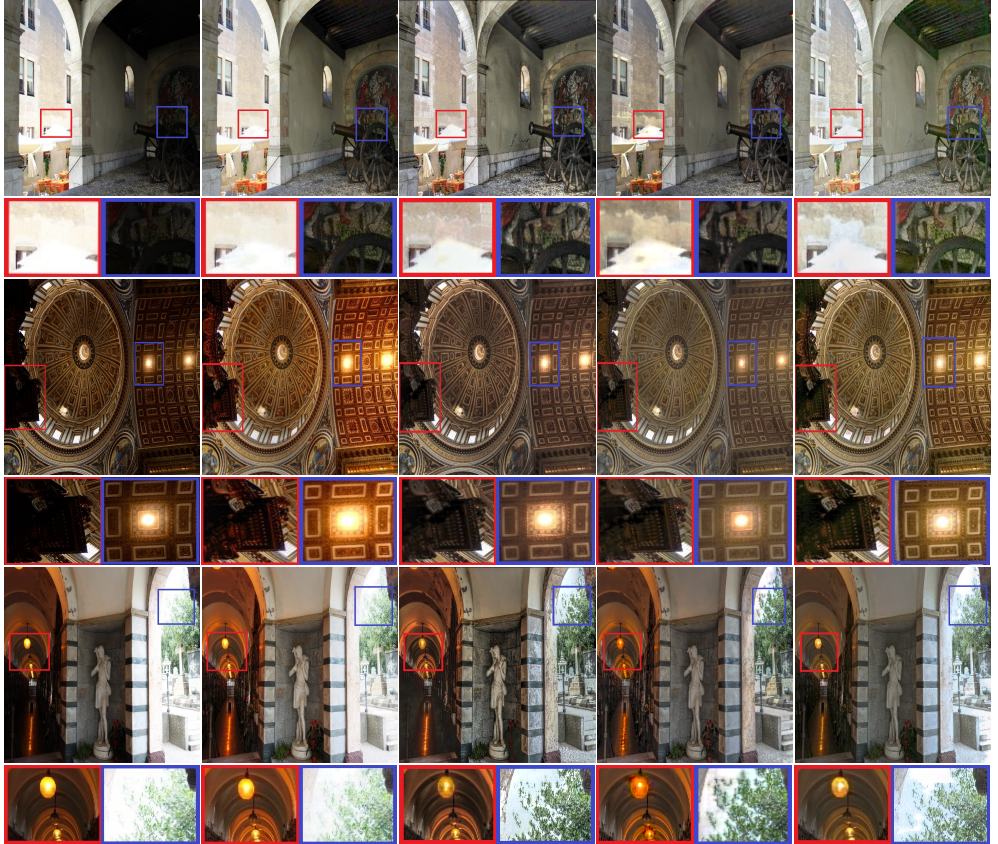


Figure 12: Visual comparison of our MQN and state-of-the-art ITM methods. From left to right: input LDR, HDRCNN [15], ExpandNet [16], SingleHDR [17] and our MQN.

References

- [1] Frédéric Drago, Karol Myszkowski, Thomas Annen, and Norishige Chiba. Adaptive logarithmic mapping for displaying high contrast scenes. In *Computer graphics forum*, volume 22, pages 419–426. Wiley Online Library, 2003. Issue: 3.
- [2] Rafał Mantiuk, Scott Daly, and Louis Kerofsky. Display adaptive tone mapping. *ACM Transactions on Graphics (TOG)*, 27(3):1–10, 2008. Publisher: ACM New York, NY, USA.
- [3] Erik Reinhard, Michael Stark, Peter Shirley, and James Ferwerda. Photographic tone reproduction for digital images. In *Proceedings of the 29th annual conference on Computer graphics and interactive techniques*, pages 267–276, 2002.
- [4] Open Source Vision Foundation . OpenCV, 2021. URL <https://opencv.org/>.
- [5] Greg Ward. High dynamic range image encodings, 2006. URL <http://www.anywhere.com/gward/hdrenc/pages/originals.html>. Publisher: Cite-seer.
- [6] Rafael Mantiuk . PFSTools. High Dynamic Range Images and Videos, 2015. URL http://pfstools.sourceforge.net/hdr_gallery.html.
- [7] Brian Funt and Lilong Shi. The effect of exposure on MaxRGB color constancy. In *Human Vision and Electronic Imaging XV*, volume 7527, page 75270Y. International Society for Optics and Photonics, 2010. URL https://www2.cs.sfu.ca/~colour/data/funt_hdr/.
- [8] Samuel W. Hasinoff, Dillon Sharlet, Ryan Geiss, Andrew Adams, Jonathan T. Barron, Florian Kainz, Jiawen Chen, and Marc Levoy. Burst photography for high dynamic range and low-light imaging on mobile cameras. *ACM Transactions on Graphics*, 35(6):192:1–192:12, November 2016. ISSN 0730-0301. doi: 10.1145/2980179.2980254. URL <https://doi.org/10.1145/2980179.2980254>.
- [9] Hiromi Nemoto, Pavel Korshunov, Philippe Hanhart, and Touradj Ebrahimi. Visual attention in LDR and HDR images. In *9th International Workshop on Video Processing and Quality Metrics for Consumer Electronics (VPQM)*, 2015. Issue: CONF.
- [10] Siyeong Lee, Gwon Hwan An, and Suk-Ju Kang. Deep recursive hdri: Inverse tone mapping using generative adversarial networks. In *Proceedings of the European Conference on Computer Vision (ECCV)*, pages 596–611, 2018.
- [11] Yu-Lun Liu, Wei-Sheng Lai, Yu-Sheng Chen, Yi-Lung Kao, Ming-Hsuan Yang, Yung-Yu Chuang, and Jia-Bin 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*, pages 1651–1660, 2020.
- [12] Duc-Tien Dang-Nguyen, Cecilia Pasquini, Valentina Conotter, and Giulia Boato. Raise: A raw images dataset for digital image forensics. In *Proceedings of the 6th ACM Multimedia Systems Conference*, pages 219–224, 2015.
- [13] HDRSoft Ltd. . Photo Editing Software for HDR & Real Estate Photography | Photomatrix, 2017. URL <https://www.hdrsoft.com/>.

- [14] Siyeong Lee, Gwon Hwan An, and Suk-Ju Kang. Deep chain hdri: Reconstructing a high dynamic range image from a single low dynamic range image. *IEEE Access*, 6: 49913–49924, 2018. Publisher: IEEE.
- [15] Gabriel Eilertsen, Joel Kronander, Gyorgy Denes, Rafal K. Mantiuk, and Jonas Unger. HDR image reconstruction from a single exposure using deep CNNs. *ACM transactions on graphics (TOG)*, 36(6):1–15, 2017. Publisher: ACM New York, NY, USA.
- [16] Tensorflow . Performance measurement | TensorFlow Lite, 2021. URL <https://www.tensorflow.org/lite/performance/measurement>.
- [17] Demetris Marnerides, Thomas Bashford-Rogers, Jonathan Hatchett, and Kurt Debatista. Expandnet: A deep convolutional neural network for high dynamic range expansion from low dynamic range content. In *Computer Graphics Forum*, volume 37, pages 37–49. Wiley Online Library, 2018. Issue: 2.
- [18] Zeeshan Khan, Mukul Khanna, and Shanmuganathan Raman. FHDR: HDR Image Reconstruction from a Single LDR Image using Feedback Network. *arXiv preprint arXiv:1912.11463*, 2019.
- [19] Marcel Santana Santos, Tsang Ing Ren, and Nima Khademi Kalantari. Single image HDR reconstruction using a CNN with masked features and perceptual loss. *arXiv preprint arXiv:2005.07335*, 2020.
- [20] S. M. A. Sharif, Rizwan Ali Naqvi, Mithun Biswas, and Kim Sungjun. A Two-stage Deep Network for High Dynamic Range Image Reconstruction. *arXiv preprint arXiv:2104.09386*, 2021.
- [21] Xiangyu Chen, Yihao Liu, Zhengwen Zhang, Yu Qiao, and Chao Dong. HDRUNet: Single Image HDR Reconstruction with Denoising and Dequantization. *arXiv:2105.13084*, 2021.
- [22] Post-training integer quantization with int16 activations, 2021. URL https://www.tensorflow.org/lite/performance/post_training_integer_quant_16x8.
- [23] Sanghyun Woo, Jongchan Park, Joon-Young Lee, and In So Kweon. Cbam: Convolutional block attention module. In *Proceedings of the European conference on computer vision (ECCV)*, pages 3–19, 2018.
- [24] Yulun Zhang, Kunpeng Li, Kai Li, Lichen Wang, Bineng Zhong, and Yun Fu. Image Super-Resolution Using Very Deep Residual Channel Attention Networks. In Vittorio Ferrari, Martial Hebert, Cristian Sminchisescu, and Yair Weiss, editors, *Computer Vision – ECCV 2018*, Lecture Notes in Computer Science, pages 294–310, Cham, 2018. Springer International Publishing. ISBN 978-3-030-01234-2. doi: 10.1007/978-3-030-01234-2_18.
- [25] Paulius Micikevicius, Sharan Narang, Jonah Alben, Gregory Diamos, Erich Elsen, David Garcia, Boris Ginsburg, Michael Houston, Oleksii Kuchaiev, and Ganesh Venkatesh. Mixed precision training. *arXiv preprint arXiv:1710.03740*, 2017.

- [26] Benoit Jacob, Skirmantas Kligys, Bo Chen, Menglong Zhu, Matthew Tang, Andrew Howard, Hartwig Adam, and Dmitry Kalenichenko. Quantization and training of neural networks for efficient integer-arithmetic-only inference. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 2704–2713, 2018.
- [27] Unvanquished/Unvanquished, April 2021. URL <https://github.com/Unvanquished/Unvanquished>. original-date: 2011-09-30T16:44:44Z.