Neural Exposure Fusion for High-Dynamic Range Object Detection (Supplemental Document)

Emmanuel Onzon[1] Maximilian Bömer[1] Fahim Mannan[1] Felix Heide[1, 2]
[1]Torc Robotics [2]Princeton University

In this supplemental document, we provide additional details on our differentiable image signal processor (ISP), the training procedure and further quantitative and qualitative evaluations and ablation experiments. In Section 1, we provide details of the different stages of the ISP and list their trainable and non-trainable parameters. In Section 2, we discuss the training procedure of the proposed method in detail. In Section 3, we discuss alternative fusion strategies in detail. More description of the neural exposure control module is provided in Section 4. In Section 6, we present additional quantitative results for a *additional unseen test dataset*, conduct additional ablation studies, and show qualitative performance. Finally, we provide two videos illustrating the operation of the proposed pipeline on image sequences.

Contents

1. Differentiable ISP

In the following, we describe the differentiable ISP pipeline used in the proposed method. This ISP consists of a sequence of multiple operations as illustrated in Figure 1. The first ISP block is a contrast stretcher applied to the RAW image. This contrast stretcher performs a pixel-wise affine mapping based on the lower and upper percentile of all RAW values. The second step of the ISP is a differentiable variant of bilinear demosaicing, creating a three channel color image out of the contrast stretched intensities. This is followed by a resize operation of the image to a shape with height 600 pixels and width 960 pixels. The fourth step is a pixel-wise power transform $x \mapsto x^\gamma$ with $\gamma = 0.8$ where γ is not learned for this step. The fifth ISP block is the application of color correction matrix, *i.e.*, for each pixel, the (r, g, b) vector, of the red, green and blue values, is mapped linearly with a 3×3 matrix which is learned during training. The matrix is initialized to the identity mapping. The sixth step is a color space transform to the color space YCbCr, followed by a low-frequency denoiser. More precisely, it is a denoiser based on a difference of Gaussian (DoG) filters. To this end, we extract a detail image as

$$I_{\text{detail}} = K_1 * I_{\text{input}} - K_2 * I_{\text{input}}, \tag{1}$$

where $*$ is the convolution operator and K_1 and K_2 are Gaussian kernels with standard deviations σ_1 and σ_2 respectively, which are learned such that $\sigma_1 < \sigma_2$. The output of the DoG denoiser is

$$I_{\text{output}} = I_{\text{input}} - g \cdot I_{\text{detail}} \cdot \mathbf{1}_{|I_{\text{detail}}| \leqslant t}, \tag{2}$$

where the parameters g and t are learned. After that, a color conversion back to the previous RGB color space is applied. The ninth step is a thresholded unsharp mask filter where the standard deviation of the Gaussian filter, the magnitude, and the threshold are learned. The tenth step is a pixel-wise

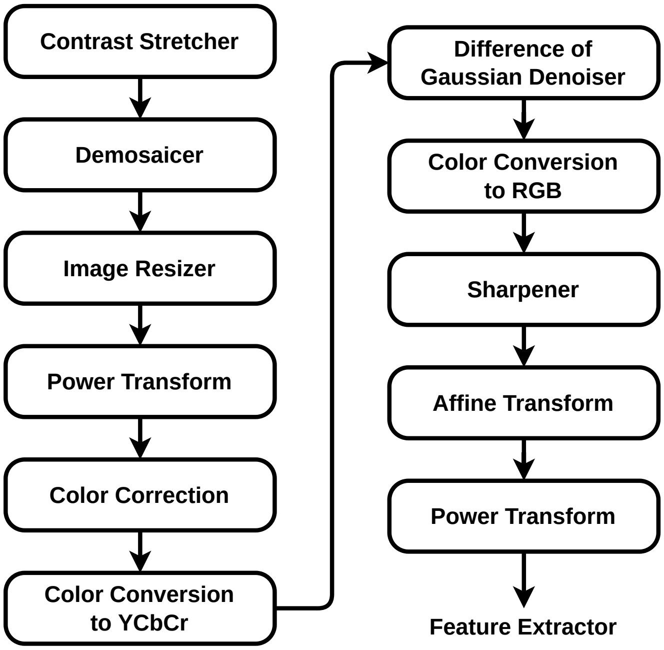

Figure 1. Block diagram of the differentiable ISP used for the experiments with all tested methods, see text for details.

affine transform with learned parameters. Finally, the last step is a learned gamma correction step.

2. Additional Training Details

In this section, we provide further details on the training procedure for the proposed model.

Pretraining The feature extractor was pretrained on ImageNet 1K. The object detector was pretrained jointly with the ISP with several public and proprietary datasets. The public datasets that were used for pretraining are MS-COCO [7], Kitti [3], Cityscapes [2], and BDD [11]. The resulting pretrained ISP and object detector pipeline is used as a starting point for the training of *all* the experiments reported in the paper.

Optimizer and Hyperparameters We train our model using stochastic gradient descent with momentum of value 0.9. We use a learning rate with exponential decay after an initial stage of constant learning rate for the first 10,000 iterations. In the initial stage, the learning rate is kept constant at 10^{-4}. Thereafter, the learning rate is multiplied by $0.7 * 10^{-4}$ at each training iteration, such that the learning rate is shrunk by a factor 0.7 every 10,000 iterations. We train for 160,000 iterations with a batch size of one training example.

Multi-Exposure Training Pipeline In our training pipeline for multi-exposure object detection, we simulate $n = 3$ LDR captures of the same scene (I_{lower}, I_{middle}, I_{upper}), the captures with the lower, middle and upper exposure. The middle exposure capture I_{middle} is simulated as in [8], except that instead of sampling the logarithm of the exposure shift in the interval $[\log 0.1, \log 10]$, we sample in the interval $[-15 \log 2, 15 \log 2]$. We empirically found this interval to be better suited to evaluate object detection performances under the challenges of high dynamic range conditions. The other captures (I_{lower} and I_{upper}) are simulated in the same way, except that in addition to the exposure shift an extra constant factor is applied (d_{lower}, d_{upper}), respectively. In our experiments we choose $d_{\text{lower}} = 16^{-1}$ and $d_{\text{upper}} = 16$.

3. Alternative Fusion Strategies

Next, we provide detailed descriptions of alternative fusion approaches we investigate in our work.

3.1. Local Cross Attention RPN Fusion

We investigate a variant of the local cross attention fusion. Here, the region proposals are computed independently for each exposure. The union set of all proposals is used to crop from the aggregated n feature maps f_{agg} produced by the feature extractor. We call this variant *Local Cross Attention RPN Fusion*.

We treat the different exposure pipelines separate until the Region Proposal Network (RPN). The network predicts M first-stage proposals for each stream j, which results in $n \cdot M$ proposals in total. Based on them, the RoI pooling layer crops out of the aggregated feature map f_{agg}. f_{agg} refers to the n feature maps $f^{(1)}_{\text{fm},(r,c,k)}, \dots, f^{(n)}_{\text{fm},(r,c,k)}$, *i.e.*, for $k \in \{1, \dots, nd\}$ concatenated along the last axis,

$$f_{\text{agg},(r,c,k)} = f^{(\lfloor (k-1)/d \rfloor + 1)}_{\text{fm},(r,c,k\%d)}, \tag{3}$$

where $\%$ is the modulo operator, and each fused feature map $f^{(j)}_{\text{fm},(r,c,k)}$, with $j \in \{1, \dots, n\}$, is computed as

$$f^{(j)}_{\text{fm},(r,c,k)} = \sum_{j'=1}^{n} \alpha^{(j)}_{j',r,c} \cdot y_{j',r,c,k} \tag{4}$$

and,

$$\alpha^{(j)}_{\cdot,r,c} = \text{Attention}(Q^{(j)}, y_{\cdot,r,c,\cdot}), \tag{5}$$

similar to Equations (6) and (7) in the main paper. A single second stage box classifier, which is applied on the full list of cropped feature maps yield the second stage proposals, that is

$$f_{\text{ROI},i,j} = \text{NoC}(\text{RoiPool}(f_{\text{agg}}, \text{RPN}(\text{FE}(\text{ISP}(R_j)), i)). \tag{6}$$

We employ the loss from [9] without modifications for this fusion strategy.

3.2. Late Fusion Strategies with Modified Losses

We next provide details on the Late Fusion method of the main paper and we compare it to two enhanced variants. The method called Late Fusion in the main paper is called Late Fusion Standard Loss in this document, in order to better distinguish it from the two variants, which we dub Late Fusion Keep Best and Late Fusion NMS. These three late fusion strategies behave the same at inference time and only differ at training time. The late fusion strategies treat features of the individual exposures independently until the end of the second stage of the object detector. All the refined detection results produced from the n exposures are gathered in a single global set of detections. Per-class NMS is performed on this global set of detections, producing a refined and non-maxima suppressed set of detections pertaining to the n LDR exposures. In the main paper, we have evaluated the late fusion strategy where we use the standard object detection loss from [9].

Here, we further experiment with several alternative losses to improve the late fusion process. Similar to the other strategies, we train all blocks of the computer vision pipeline jointly purely using the object detection loss, which is a sum of the first stage loss L_{RPN} and second stage loss L_{2ndStage} like in [4, 9].

$$L_{\text{Total}} = L_{\text{RPN}} + L_{\text{2ndStage}}. \tag{7}$$

For the two proposed enhanced late fusion variants, L_{RPN} is computed as the sum of the lowest objectness L_{Obj} and localization losses L_{Loc} over all n exposure pipelines computed per anchor $a \in A$. The set of available anchors A is identical in each stream. The model is encouraged to have high diversity in predictions between different streams and is not punished if instances are missed that are recovered by other streams. The RPN loss which we investigate can be formalized as

$$\begin{aligned} L_{\text{RPN, prop.}} = \sum_a \min_{j \in \{1,\dots,n\}} \Bigg(& \frac{1}{N_{\text{Obj}}} L_{\text{Obj}}(p_{j,a}, p^*_a) \\ & + \frac{\lambda}{N_{\text{Loc}}} p^*_a L_{\text{Loc}}(t_{j,a}, t^*_a) \Bigg), \end{aligned} \tag{8}$$

while the standard RPN loss is,

$$\begin{aligned} L_{\text{RPN, std.}} = \sum_a \Bigg(& \frac{1}{N_{\text{Obj}}} L_{\text{Obj}}(p_{j,a}, p^*_a) \\ & + \frac{\lambda}{N_{\text{Loc}}} p^*_a L_{\text{Loc}}(t_{j,a}, t^*_a) \Bigg) \end{aligned} \tag{9}$$

We compute masked versions of the second stage loss used in [4, 9]. The mask coefficients α^i_j differ depending on the chosen late fusion strategy,

$$\begin{aligned} L_{2^{\text{nd}}\text{St., prop.}} = \sum_{j=1}^{n} \sum_i \alpha^i_j \Bigg(& \frac{1}{N_{\text{Cls}}} L_{\text{Cls}}(p^i_j, c^{*i}_j) \\ & + \frac{\lambda}{N_{\text{Loc}}} \mathbf{1}_{c^{*i}_j \geqslant 1} L_{\text{Loc}}(t^i_j, t^{*i}_j) \Bigg), \end{aligned} \tag{10}$$

c^{*i}_j and t^{*i}_j are the GT class and box assigned to the predicted box t^i_j. $\mathbf{1}_{c^{*i}_j \geqslant 1}$ is equal to 1 when the GT is an object and 0 when it is background. The coefficients α^i_j are the masks, each of them is set to 0 or 1. For comparison, we recall below the standard second-stage loss,

$$\begin{aligned} L_{2^{\text{nd}}\text{St., std.}} = \sum_{j=1}^{n} \sum_i \Bigg(& \frac{1}{N_{\text{Cls}}} L_{\text{Cls}}(p^i_j, c^{*i}_j) \\ & + \frac{\lambda}{N_{\text{Loc}}} \mathbf{1}_{c^{*i}_j \geqslant 1} L_{\text{Loc}}(t^i_j, t^{*i}_j) \Bigg). \end{aligned} \tag{11}$$

By pruning the less relevant loss components with the introduced masks, the resulting loss is specialized to well-exposed regions in the image for a given exposure pipeline. At the same time, it avoids false negatives in sub-optimal exposures, as these cannot be filtered out in the final NMS step.

Two alternative methods to define the masks are detailed below.

Strategy I, *Keep Best Loss*: For each ground truth object the loss components corresponding to the pipeline that performs best are kept and prunes the others.

Strategy II, *NMS Loss*: Prunes the loss components based on the same NMS step as performed at inference time.

While Strategy I more precisely prunes the loss across exposure pipelines, Strategy II is conceptually simpler, which makes it an interesting alternative to test. We review both strategies in detail below. For a quantitative comparison, see Section 6 and Table 2.

3.2.1 Strategy I: "Keep Best Loss"

In the second stage of the object detector, the refined bounding boxes of the different exposure pipelines are merged into a single set of predicted bounding boxes. The second stage loss is computed by assigning each box to a single ground truth (GT) object, see also [4]. If the ground truth (GT) object is positive (foreground), we first identify the exposure stream j that predicted the bounding box, which received the lowest aggregated loss $L^i_{\text{Agg},j} = L^i_{\text{Cls},j} + L^i_{\text{Loc},j}$ for this GT object. Afterward, we only backpropagate the losses for the bounding boxes assigned to those GT objects, which were predicted by the same pipeline j. As an exception, the losses of all of the bounding boxes that are associated with negative GT (background) are backpropagated, regardless of which exposure stream predicted them. With the notation from Eq. (11), this is

$$\alpha^i_j = \begin{cases} 1, & \text{if } c^{*i}_j \geqslant 1 \text{ and } \exists i' \text{ such that } \mathrm{GT}(i,j) = \mathrm{GT}(i',j), \\ & L^{i'}_{\text{Agg},j} \text{ minimal among all predictions for GT,} \\ 1, & \text{if } c^{*i}_j = 0, \\ 0, & \text{otherwise.} \end{cases} \tag{12}$$

3.2.2 Strategy II: "NMS Loss"

Like in strategy I, here, we get the final detection results after class-wise NMS on the combined set of all predictions. The non-suppressed proposals are the only ones for which the second stage loss gets backpropagated; that is

$$\alpha^i_j = \begin{cases} 1, & \text{if not filtered by NMS,} \\ 0, & \text{otherwise.} \end{cases} \tag{13}$$

4. Neural Exposure Control

Next, we further describe how we predict exposures for the separate HDR sub-frames. Specifically, we design an exposure control network similar to [8] to determine the exposure value of each of the LDR captures for the next time step. We generalize this module to work for multiple exposures. Let e_t be the exposure value produced by the network for time step t and $e^{(j)}_t$ the exposure value for time step t for capture $j \in \{1, \ldots, n\}$. Then $e^{(j)}_t$ is computed as,

$$e^{(j)}_t = e_t \cdot \delta^{j - \frac{n+1}{2}}, \tag{14}$$

where δ is a hyperparameter. We choose $\delta = 16$ and $n = 3$ in our experiments.

5. Noise simulation

5.1. Image formation

We simulate image captures from images that have been recorded with a camera equipped with the Sony IMX490 image sensor. These images already include some level of noise. We refer to this image sensor as the *source* image sensor. Based on a dataset of raw images collected with the source image sensor, our goal is to simulate raw images as if they would have been captured with another image sensor that we refer to as the *target* image sensor (ON Semi AR0231AT in our case). We add some amount of noise to the images collected with the source image sensor such that the total amount of resulting noise equals the amount of noise that would have been produced by the target image sensor. The method can be used for other source and target image sensors, as long as the source image sensor does not produce more noise than the target image sensor.

For the purpose of noise simulation, we rewrite the image formation model more precisely as follows. We consider the raw image pixel value y for some pixel in the image. This quantity is expressed in DN (digital numbers), a dimensionless unit used for clarity of the exposition. The value y can be expressed in terms of the following quantities, the number of photons N_p entering the pixels area during the exposure time t, the quantum efficiency η (expressed in e-/γ, i.e., electrons per photon), the camera conversion gain g (expressed in e-/DN), the camera gain setting K (a multiplier such that $K = 1$ for ISO 100), the number of electrons N_d that accumulate as dark current during exposure, the thermal noise $n_{\text{v,out}}$ which is added to the voltage at readout, the conversion factor g_{ADC} from voltage to digital numbers, and M_{white} the white level, i.e., the maximum sensor value that can be recorded. With these notations we can write

$$y = \min(N_\text{p} \cdot \eta \cdot g \cdot K + N_\text{d} \cdot g \cdot K + n_{\text{v,out}} \cdot g_{\text{ADC}} \cdot K,\, M_{\text{white}}) \tag{15}$$

The thermal noise can be conveniently expressed in terms of the equivalent number of electrons as

$$n_{\text{e,out}} = n_{\text{v,out}} \cdot \frac{g_{\text{ADC}}}{g} \tag{16}$$

Using the number of photo-induced electrons $N_\text{e} = N_\text{p} \cdot \eta$, we can expressed y as follows.

$$y = \min((N_\text{e} + N_\text{d} + n_{\text{e,out}}) \cdot g \cdot K,\, M_{\text{white}}) \tag{17}$$

The number of photo-induced electrons N_e is a Poisson random variable of parameter μ_e (such that μ_e is both the expectation and the variance of N_e). The number of electrons accumulated due to dark current N_d also follows a Poisson probability distribution. We make the usual approximation with a gaussian random variable for both N_e and N_d. This allows to combine the two signal independent noise terms into a single gaussian random variable $n = N_\mathrm{d} + n_\mathrm{e,out}$, such that we consider n as a gaussian random variable with expectation m and variance σ^2.

$$n \sim \mathcal{N}(m, \sigma^2) \tag{18}$$

This simplifies the expression of y,

$$y = \min((N_\mathrm{e} + n) \cdot g \cdot K,\ M_\mathrm{white}) \tag{19}$$

The correspondence with Equation (2) of the main paper is as follows. The pre-amplification noise is $n_\mathrm{pre} = N_\mathrm{e} + n - \phi_\mathrm{scene} \cdot t$, and the post-amplification noise is negligible in our application, so that we consider it to be zero.

5.2. Mean number of photo-induced electrons

The parameter μ_e is expressed in electrons and can be written as $\mu_e = \eta \cdot \mu_p$ where μ_p is the average number of photons expected to enter the pixel area during the exposure time t. The average number of photons can be expressed as

$$\mu_p = \frac{\phi \cdot t}{h \cdot c / \lambda} \tag{20}$$

where ϕ is the radiant power (in W) on the pixel surface, h is Planck constant, c is the speed of light in vacuum and λ the wavelength of the light that illuminates the pixel. The radiant power can be written $\phi = E \cdot A$ where E is the irradiance (in W/m^2) and A is the area of the pixel. We can then write

$$\mu_e = \frac{\eta(\lambda) \cdot E \cdot A \cdot t}{h \cdot c / \lambda} \tag{21}$$

5.3. Source and target image sensors

In the following we consider the case where the probability that the number of accumulated electrons reaches the full-well capacity is low. In such a case we can make the following approximation.

$$y = (N_\mathrm{e} + n) \cdot g \cdot K \tag{22}$$

We now consider the corresponding quantities when we illuminate the source image sensor and the target image sensor respectively. We use the subscript "src" for the quantities corresponding to the source image sensor and the subscript "tgt" for the quantities corresponding to the target image sensor.

$$y_\mathrm{src} = (N_\mathrm{src} + n_\mathrm{src}) \cdot g_\mathrm{src} \cdot K_\mathrm{src} \tag{23}$$

where

$$N_\mathrm{src} \sim \mathcal{N}(\mu_\mathrm{src}, \mu_\mathrm{src}) \tag{24}$$

$$n_\mathrm{src} \sim \mathcal{N}(m_\mathrm{src}, \sigma^2_\mathrm{src}) \tag{25}$$

therefore y_src is gaussian with the following expectation and variance,

$$\mathrm{E}(y_\mathrm{src}) = (\mu_\mathrm{src} + m_\mathrm{src}) \cdot g_\mathrm{src} \cdot K_\mathrm{src}, \tag{26}$$

$$\mathrm{Var}(y_\mathrm{src}) = \left(\mu_\mathrm{src} + \sigma^2_\mathrm{src}\right) \cdot g^2_\mathrm{src} \cdot K^2_\mathrm{src}. \tag{27}$$

Similarly,

$$y_\mathrm{tgt} = (N_\mathrm{tgt} + n_\mathrm{tgt}) \cdot g_\mathrm{tgt} \cdot K_\mathrm{tgt} \tag{28}$$

where

$$N_\mathrm{tgt} \sim \mathcal{N}(\mu_\mathrm{tgt}, \mu_\mathrm{tgt}) \tag{29}$$

$$n_\mathrm{tgt} \sim \mathcal{N}(m_\mathrm{tgt}, \sigma^2_\mathrm{tgt}) \tag{30}$$

therefore y_tgt is gaussian with the following expectation and variance,

$$\mathrm{E}(y_\mathrm{tgt}) = (\mu_\mathrm{tgt} + m_\mathrm{tgt}) \cdot g_\mathrm{tgt} \cdot K_\mathrm{tgt}, \tag{31}$$

$$\mathrm{Var}(y_\mathrm{tgt}) = \left(\mu_\mathrm{tgt} + \sigma^2_\mathrm{tgt}\right) \cdot g^2_\mathrm{tgt} \cdot K^2_\mathrm{tgt}. \tag{32}$$

5.4. Simulating the target pixel value based on the source image

We simulate the pixel value $\tilde{y}_\mathrm{tgt}$ of an image that would have been captured with the target image sensor, based on the pixel value y_src of the corresponding image that has been captured with the source image sensor. We introduce y^*_src,

$$y^*_\mathrm{src} = (y_\mathrm{src} - m_\mathrm{src} \cdot g_\mathrm{src} \cdot K_\mathrm{src}) \cdot \frac{\mu_\mathrm{tgt} \cdot g_\mathrm{tgt} \cdot K_\mathrm{tgt}}{\mu_\mathrm{src} \cdot g_\mathrm{src} \cdot K_\mathrm{src}}, \tag{33}$$

α and β,

$$\alpha = \left(1 - \frac{\mu_\mathrm{tgt}}{\mu_\mathrm{src}}\right) \cdot g_\mathrm{tgt} \cdot K_\mathrm{tgt}, \tag{34}$$

$$\beta = \left(\sigma^2_\mathrm{tgt} - \sigma^2_\mathrm{src} \cdot \frac{\mu^2_\mathrm{tgt}}{\mu^2_\mathrm{src}}\right) \cdot g^2_\mathrm{tgt} \cdot K^2_\mathrm{tgt}, \tag{35}$$

and assume $\mu_\mathrm{tgt} \leq \mu_\mathrm{src}$ and $\sigma_\mathrm{src} \leq \sigma_\mathrm{tgt}$ such that $\alpha \geq 0$ and $\beta \geq 0$. We compute $\tilde{y}_\mathrm{tgt}$ as follows,

$$\begin{aligned} \tilde{y}_\mathrm{tgt} = y^*_\mathrm{src} &+ \sqrt{\alpha \cdot \max\left(y^*_\mathrm{src},\, 0\right)} \cdot U_1 \\ &+ \sqrt{\beta} \cdot U_2 + m_\mathrm{tgt} \cdot g_\mathrm{tgt} \cdot K_\mathrm{tgt}, \end{aligned} \tag{36}$$

where $U_1 \sim \mathcal{N}(0,1)$ and $U_2 \sim \mathcal{N}(0,1)$ are two independent standard gaussian variables.

Then, the expectations of $\tilde{y}_\mathrm{tgt}$ and y_tgt are equal,

$$\mathrm{E}(\tilde{y}_\mathrm{tgt}) = \mathrm{E}(y_\mathrm{tgt}), \tag{37}$$

and, in the case where the probability $\mathrm{P}\left(y^*_\mathrm{src} < 0\right)$ is small, their variances are also approximately equal,

$$\mathrm{Var}(\tilde{y}_\mathrm{tgt}) \approx \mathrm{Var}(y_\mathrm{tgt}), \tag{38}$$

and $\tilde{y}_{\text{tgt}}$ is approximately gaussian.

Equation (36) can effectively be used to simulate a pixel value that would have been produced by the target image sensor because it is based on known quantities. The random variables U_1 and U_2 are sampled using a random number generator. The quantities g_{src}, σ_{src}, m_{src}, g_{tgt}, σ_{tgt}, m_{tgt} are calibrated using standard procedures (see [1]). The camera gain settings K_{tgt} and K_{src} are known camera settings and y_{src} is part of the dataset captured with the source image sensor. Finally the ratio $\mu_{\text{tgt}}/\mu_{\text{src}}$ is

$$\frac{\mu_{\text{tgt}}}{\mu_{\text{src}}} = \frac{\eta_{\text{tgt}} \cdot A_{\text{tgt}} \cdot t_{\text{tgt}}}{\eta_{\text{src}} \cdot A_{\text{src}} \cdot t_{\text{src}}}, \tag{39}$$

where the exposure times t_{tgt} and t_{src} are known camera settings, and the quantum efficiencies η_{tgt} and η_{src} and the pixel areas A_{tgt} and A_{src} are technical data given by the image sensor manufacturer.

5.5. Derivation of the expectation and variance of the simulated pixel value

The expectation of y^*_{src} is

$$\mathrm{E}(y^*_{\text{src}}) = \mu_{\text{tgt}} \cdot g_{\text{tgt}} \cdot K_{\text{tgt}}, \tag{40}$$

Since y^*_{src} and U_1 are independent random variables,

$$\begin{aligned}\mathrm{E}&\left(\sqrt{\alpha \cdot \max\left(y^*_{\text{src}}, 0\right)} \cdot U_1\right) \\ &= \mathrm{E}\left(\sqrt{\alpha \cdot \max\left(y^*_{\text{src}}, 0\right)}\right) \cdot \mathrm{E}(U_1) = 0,\end{aligned} \tag{41}$$

because $\mathrm{E}(U_1) = 0$. Since $\mathrm{E}(U_2) = 0$, we deduce

$$\begin{aligned}\mathrm{E}(\tilde{y}_{\text{tgt}}) &= \mu_{\text{tgt}} \cdot g_{\text{tgt}} \cdot K_{\text{tgt}} + m_{\text{tgt}} \cdot g_{\text{tgt}} \cdot K_{\text{tgt}} \\ &= \mathrm{E}(y_{\text{tgt}})\end{aligned} \tag{42}$$

The variance of y^*_{src} is

$$\mathrm{Var}(y^*_{\text{src}}) = \left(\mu_{\text{src}} + \sigma^2_{\text{src}}\right) \cdot \frac{\mu^2_{\text{tgt}}}{\mu^2_{\text{src}}} \cdot g^2_{\text{tgt}} \cdot K^2_{\text{tgt}} \tag{43}$$

Assuming $\mathrm{P}\left(y^*_{\text{src}} < 0\right)$ is negligible, we can write

$$\begin{aligned}\mathrm{Var}&\left(y^*_{\text{src}} + \sqrt{\alpha \cdot \max\left(y^*_{\text{src}}, 0\right)} \cdot U_1\right) \\ &= \mathrm{Var}\left(y^*_{\text{src}} + \sqrt{\alpha \cdot y^*_{\text{src}}} \cdot U_1\right)\end{aligned} \tag{44}$$

and,

$$\begin{aligned}\mathrm{Var}&\left(y^*_{\text{src}} + \sqrt{\alpha \cdot y^*_{\text{src}}} \cdot U_1\right) \\ &= \mathrm{Var}(y^*_{\text{src}}) + \mathrm{Var}\left(\sqrt{\alpha \cdot y^*_{\text{src}}} \cdot U_1\right) + \\ &\qquad 2 \cdot \mathrm{Cov}\left(y^*_{\text{src}}, \sqrt{\alpha \cdot y^*_{\text{src}}} \cdot U_1\right).\end{aligned} \tag{45}$$

The covariance term is zero,

$$\begin{aligned}\mathrm{Cov}&\left(y^*_{\text{src}}, \sqrt{\alpha \cdot y^*_{\text{src}}} \cdot U_1\right) \\ &= \mathrm{E}\left(\left(y^*_{\text{src}} - \mathrm{E}(y^*_{\text{src}})\right) \cdot \sqrt{\alpha \cdot y^*_{\text{src}}} \cdot U_1\right) \\ &= \mathrm{E}\left(\left(y^*_{\text{src}} - \mathrm{E}(y^*_{\text{src}})\right) \cdot \sqrt{\alpha \cdot y^*_{\text{src}}}\right) \cdot \mathrm{E}(U_1) \\ &= 0\end{aligned} \tag{46}$$

because $\mathrm{E}(U_1) = 0$. The second term of the right-hand side of Equation (45) is

$$\begin{aligned}\mathrm{Var}&\left(\sqrt{\alpha \cdot y^*_{\text{src}}} \cdot U_1\right) \\ &= \mathrm{E}\left(\alpha \cdot y^*_{\text{src}} \cdot U_1^2\right) - \left(\mathrm{E}\left(\sqrt{\alpha \cdot y^*_{\text{src}}} \cdot U_1\right)\right)^2,\end{aligned} \tag{47}$$

where

$$\mathrm{E}\left(\sqrt{\alpha \cdot y^*_{\text{src}}} \cdot U_1\right) = \mathrm{E}\left(\sqrt{\alpha \cdot y^*_{\text{src}}}\right) \cdot \mathrm{E}(U_1) = 0. \tag{48}$$

Thus,

$$\begin{aligned}\mathrm{Var}\left(\sqrt{\alpha \cdot y^*_{\text{src}}} \cdot U_1\right) &= \mathrm{E}\left(\alpha \cdot y^*_{\text{src}} \cdot U_1^2\right) \\ &= \alpha \cdot \mathrm{E}(y^*_{\text{src}}),\end{aligned} \tag{49}$$

since $\mathrm{E}\left(U_1^2\right) = 1$. We can deduce

$$\begin{aligned}\mathrm{Var}&\left(y^*_{\text{src}} + \sqrt{\alpha \cdot y^*_{\text{src}}} \cdot U_1\right) \\ &= \mathrm{Var}(y^*_{\text{src}}) + \alpha \cdot \mathrm{E}(y^*_{\text{src}}) \\ &= \left(\mu_{\text{src}} + \sigma^2_{\text{src}}\right) \cdot \frac{\mu^2_{\text{tgt}}}{\mu^2_{\text{src}}} \cdot g^2_{\text{tgt}} \cdot K^2_{\text{tgt}} \\ &\quad + \left(1 - \frac{\mu_{\text{tgt}}}{\mu_{\text{src}}}\right) \cdot \mu_{\text{tgt}} \cdot g^2_{\text{tgt}} \cdot K^2_{\text{tgt}} \\ &= \left(\mu_{\text{tgt}} + \frac{\mu^2_{\text{tgt}}}{\mu^2_{\text{src}}} \cdot \sigma^2_{\text{src}}\right) \cdot g^2_{\text{tgt}} \cdot K^2_{\text{tgt}}\end{aligned} \tag{50}$$

Now,

$$\mathrm{Var}\left(\sqrt{\beta} \cdot U_2\right) = \beta, \tag{51}$$

so that we can conclude

$$\begin{aligned}\mathrm{Var}(\tilde{y}_{\text{tgt}}) &= \left(\mu_{\text{tgt}} + \frac{\mu^2_{\text{tgt}}}{\mu^2_{\text{src}}} \cdot \sigma^2_{\text{src}}\right) \cdot g^2_{\text{tgt}} \cdot K^2_{\text{tgt}} + \beta \\ &= \left(\mu_{\text{tgt}} + \sigma^2_{\text{tgt}}\right) \cdot g^2_{\text{tgt}} \cdot K^2_{\text{tgt}}. \\ &= \mathrm{Var}(y_{\text{tgt}})\end{aligned} \tag{52}$$

6. Additional Evaluations

This section reports additional qualitative and quantitative evaluations along with additional ablation experiments.

Table 1. HDR object detection evaluation for different neural exposure fusion strategies compared to conventional HDR imaging and object detection pipelines for an additional dataset of scenes of entrances and exits of tunnels.

Method	Point of Fusion	**Classes**						**mAP**
		Bike	Bus & Truck	Car & Van	Person	Traffic Light	Traffic Sign	
Shim et al. [10] (LDR)	N/A	5.8	6.7	28.5	14.6	9.3	13.4	13.1
Onzon et al. [8] (LDR)	N/A	13.1	22.5	74.2	40.7	25.0	39.8	35.9
Raw HDR	Pre-ISP	11.5	24.5	79.2	44.2	25.9	39.4	37.5
Deep HDR [6]	Post-ISP	12.8	23.2	79.1	39.8	25.4	37.0	36.2
Max Pooling Fusion *(ours)*	Conv4	13.6	26.9	79.6	43.6	26.5	41.5	38.6
Conv 1 x 1 Fusion *(ours)*	Conv4	12.0	20.8	76.6	33.9	21.8	37.1	33.7
Conv 3 x 3 Fusion *(ours)*	Conv4	**14.5**	20.6	76.7	30.7	20.9	36.8	33.4
Late Fusion *(ours)*	2nd Stage	11.8	21.9	**81.1**	43.1	25.6	40.7	37.4
Local Cross Attention *(ours)*	Conv4	14.0	**27.1**	80.2	**45.6**	**27.0**	**42.0**	**39.3**

Table 2. HDR object detection assessment for additional exposure fusion strategies evaluated on the test set used in the main paper. The results reported here complement those reported in Table 1 of the main paper.

Method	Point of Fusion	**Classes**						**mAP**
		Bike	Bus & Truck	Car & Van	Person	Traffic Light	Traffic Sign	
Late Fusion Standard Loss	2nd Stage	27.5	14.2	73.8	47.2	42.8	52.3	43.0
Late Fusion Keep Best Loss	2nd Stage	27.6	16.1	74.4	**48.4**	42.9	54.7	44.0
Late Fusion NMS Loss	2nd Stage	28.1	**16.5**	74.3	46.4	44.3	**55.9**	**44.3**
Local Cross Attention RPN	RPN	**28.2**	15.7	**74.7**	47.7	**44.9**	54.5	**44.3**

6.1. Additional Quantitative Evaluation

We report additional evaluation results for a *separate unseen dataset* in Table 1. This additional dataset is composed of challenging scenes of entrances and exits of tunnels. The dataset has been collected over three days of test driving. The data has been subsampled to 1Hz and challenging HDR scenarios with entrances and exits of tunnels have been manually selected, resulting in 418 test scenarios.

Table 1 further validates that our method Local Cross Attention Fusion (last row) performs best overall in terms of mAP. It also performs best for 4 out of 6 of the considered object classes. This also validates the results in Table 1 of the main paper. See Figure 2 for qualitative examples of the proposed methods on the additional test dataset.

We note that our method performs better than the method Deep HDR on this dataset of exits and entrances of tunnels, although Deep HDR was the best-performing method on the Tunnel subset of the main paper (see results reported in column 5 of Table 2 of the main paper). The discrepancy can be explained by the fact that the tunnel subset of the main paper not only contains challenging HDR scenes like entrances and exits of tunnels but also the inner regions which are fairly temporally consistent.

6.2. Additional Ablation Experiments

As an additional ablation experiment, we train and test networks with the alternative fusion strategies described in Section 3 on the same training set and test set as in the main paper. We report these findings in Table 2. We note that the method named “Late Fusion Standard Loss” in this table corresponds with the method named “Late Fusion” in Table 2 of the main paper. Results are repeated here to better compare with the two other late fusion strategies with modified training losses as described in Section 3.2. We find that these modifications are effective at improving the overall mAP by 1% and 1.3%. Moreover, the results reported in Table 2 show that these enhanced training losses also allow to improve the AP for most of the considered object classes. The last row of Table 2 reports the results for the method Local Cross Attention RPN Fusion. This finding demonstrates that the use of our local cross attention module proves effective across architectural variants.

Table 3. Ablation experiments with Local Cross Attention fusion at different stages of the 28-layer ResNet variant.

Method	Point of Fusion	**Classes**						**mAP**
		Bike	Bus & Truck	Car & Van	Person	Traffic Light	Traffic Sign	
Local Cross Attention *(ours)*	Conv1	26.8	**16.9**	74.1	45.6	43.0	53.6	43.3
Local Cross Attention *(ours)*	Conv2	26.4	16.7	74.2	46.7	44.0	55.6	43.9
Local Cross Attention *(ours)*	Conv3	**26.9**	16.7	**74.5**	**47.1**	44.1	55.2	44.1
Local Cross Attention *(ours)*	Conv4	26.8	16.6	74.3	47.0	**44.4**	**56.3**	**44.2**

As described in Section 6.3 of the main paper we also perform an ablation study, where we evaluate fusing different exposure features at varying stages of the feature extractor. Detailed evaluation results can be found in Table 3. We follow the terminology of [5], where Conv1 refers to the initial 7x7 convolution and Conv2/Conv3/Conv4 to the following three residual blocks of the feature extractor. Results validate that performance increases when fusing at later stages, with diminishing returns, though.

6.3. Additional Qualitative Results and Videos

We provide video sequences that show the demosaiced raw images for the first, second and third exposure as well as the overlayed detections of our Local Cross Attention Fusion model for challenging automotive HDR scenes. We removed the contrast stretcher from the ISP for the images shown in these videos to make the difference in exposure between the three exposures more pronounced in the visualization.

Figures 2 and 3 provide further qualitative results. The compared baselines (Raw HDR and Deep HDR) are outperformed by our proposed Local Cross Attention Fusion method. Examples where the baselines fail are False Negatives like the person in first row or the traffic sign in the second row as well as False Positives like the bike detections in the fourth row or falsely detected pedestrian in row five of Figure 3. The highest margins in improvement are achieved in scenes with large dynamic ranges, where conventional HDR pipelines fail to maintain details in the task-relevant image regions. Our approach differs from existing work as we fuse exposures in feature instead of image space.

HDR imaging pipelines (e.g. *Raw HDR* and *Deep HDR*, see main paper) are fusing the information of the different exposures in image space. For a large range of luminances in a given frame this can lead to under or overexposed regions. Moreover the HDR imaging pipelines have to compress the dynamic range, which inevitably entails a loss of contrast in at least some parts of the image. These effects combine together and result in sub-optimal local detection performances.

The proposed learned fusion approach avoids losing details during image fusion by moving it in feature space. Our approach outperforms single exposure systems in two ways: 1) Details that are not visible in one stream can be recovered by relying on features of streams that expose the observed image region better. 2) Streams can collaborate by fusing features with higher quality than each of them in isolation.

6.4. Conceptual Comparison of Evaluated Methods

Following the results of Table 2, we see that additional variations of the architecture and losses can lead to marginal improvements compared to Local Cross Attention Fusion, which we proposed in the main paper.

The proposed early fusion approach that aggregates feature maps from the different exposure streams using local cross attention is independent of the used downstream architecture and losses. This is not the case for the other presented variants, which exploit architectural properties of two-stage detectors (Local Cross Attention RPN) or require an adaption of the task specific losses (Keep Best Loss, NMS Loss). This allows our proposed method to be easily integrated into other downstream computer vision architectures apart from object detection (e.g., segmentation), avoiding the need of loss modifications.

Furthermore, our method is computationally efficient, as discussed in the Section on runtime and complexity in the main manuscript.

References

[1] European Machine Vision Association. Emva standard 1288, standard for characterization of image sensors and cameras, release 3.1. 2016. 6

[2] Marius Cordts, Mohamed Omran, Sebastian Ramos, Timo Rehfeld, Markus Enzweiler, Rodrigo Benenson, Uwe Franke, Stefan Roth, and Bernt Schiele. The cityscapes dataset for semantic urban scene understanding. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 3213–3223, 2016. 2

[3] Andreas Geiger, Philip Lenz, and Raquel Urtasun. Are we ready for autonomous driving? the kitti vision benchmark suite. In *2012 IEEE Conference on Computer Vision and Pattern Recognition*, pages 3354–3361. IEEE, 2012. 2

[4] Ross Girshick. Fast r-cnn. In *Proceedings of the IEEE international conference on computer vision*, pages 1440–1448, 2015. 3, 4

[5] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 770–778, 2016. 7

[6] Nima Khademi Kalantari and Ravi Ramamoorthi. Deep high dynamic range imaging of dynamic scenes. *ACM Trans. Graph.*, 36:144:1–144:12, 2017. 7, 9, 10

[7] Tsung-Yi Lin, Michael Maire, Serge Belongie, James Hays, Pietro Perona, Deva Ramanan, Piotr Dollár, and C Lawrence Zitnick. Microsoft coco: Common objects in context. In *European conference on computer vision*, pages 740–755. Springer, 2014. 2

[8] Emmanuel Onzon, Fahim Mannan, and Felix Heide. Neural auto-exposure for high-dynamic range object detection. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 7710–7720, 2021. 2, 4, 7

[9] Shaoqing Ren, Kaiming He, Ross Girshick, and Jian Sun. Faster r-cnn: Towards real-time object detection with region proposal networks. In *Advances in neural information processing systems*, pages 91–99, 2015. 3

[10] Inwook Shim, Tae-Hyun Oh, Joon-Young Lee, Jinwook Choi, Dong-Geol Choi, and In So Kweon. Gradient-based camera exposure control for outdoor mobile platforms. *IEEE Transactions on Circuits and Systems for Video Technology*, 29(6):1569–1583, 2018. 7

[11] Fisher Yu, Haofeng Chen, Xin Wang, Wenqi Xian, Yingying Chen, Fangchen Liu, Vashisht Madhavan, and Trevor Darrell. Bdd100k: A diverse driving dataset for heterogeneous multitask learning. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2020. 2

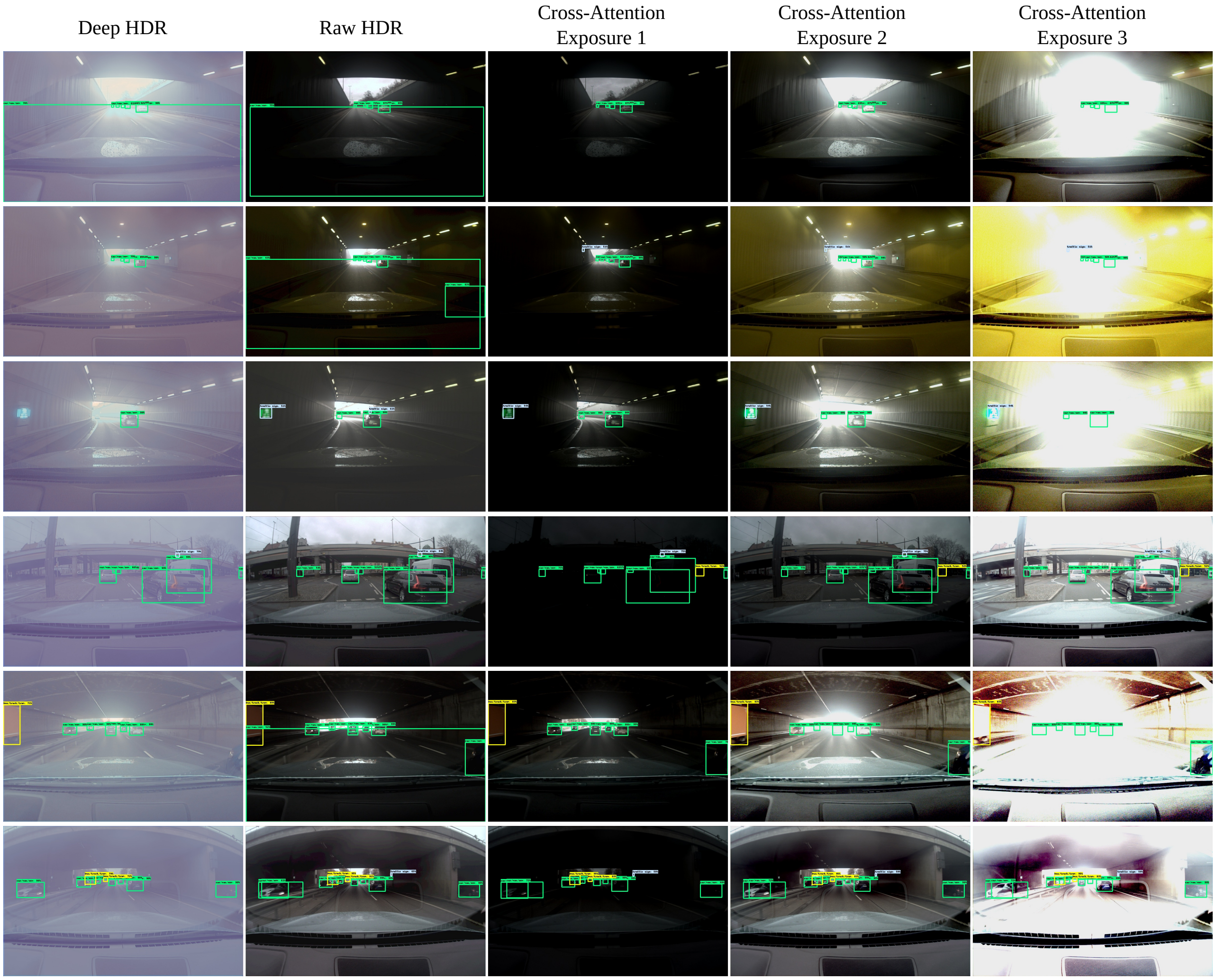
Deep HDR
Raw HDR
Cross-Attention Exposure 1
Cross-Attention Exposure 2
Cross-Attention Exposure 3

Figure 2. Qualitative comparison of the proposed *Local Cross-Attention Fusion* with the baseline methods *Raw HDR* and *Deep HDR* [6] on challenging scenes. Examples from the additional dataset of entrances and exits of tunnels, see supplemental text.

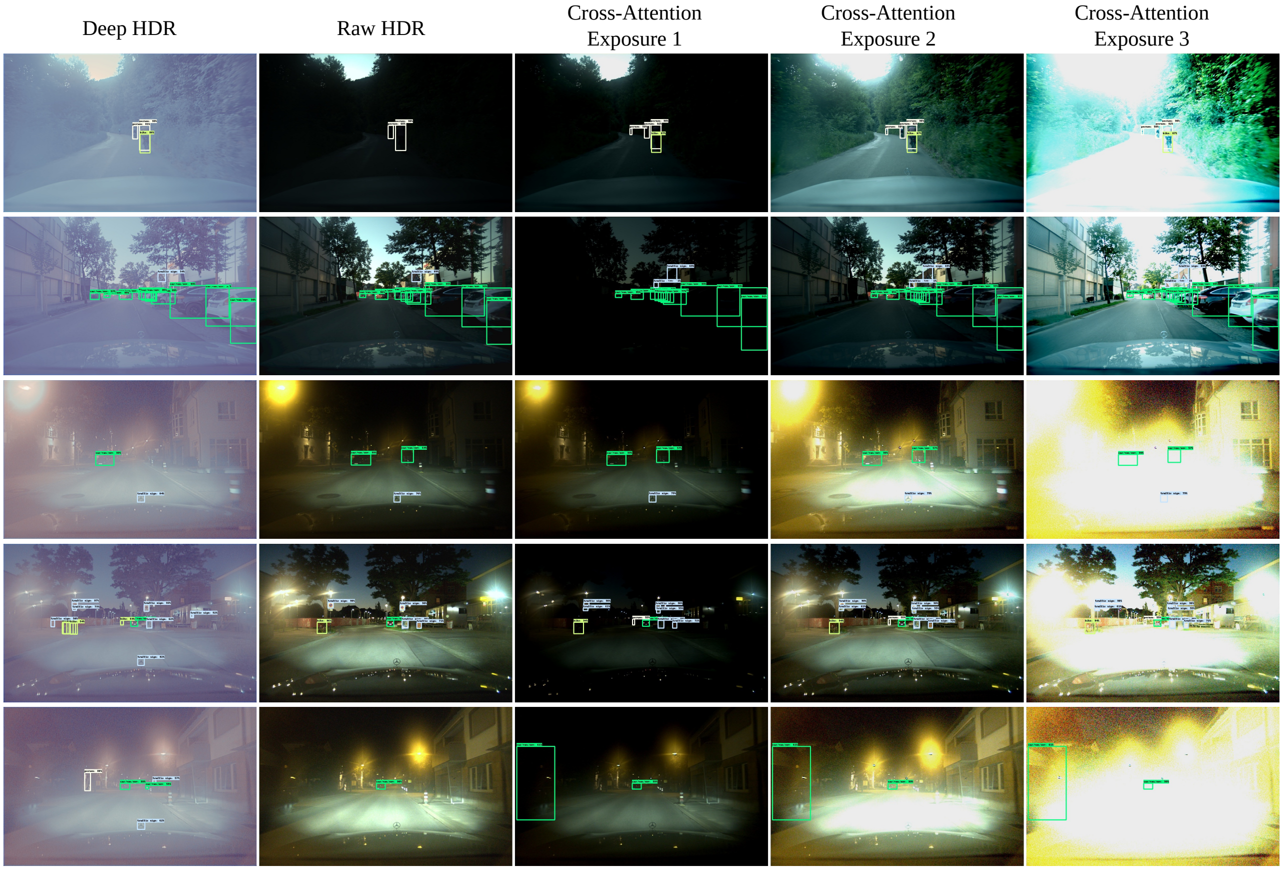
Deep HDR
Raw HDR
Cross-Attention Exposure 1
Cross-Attention Exposure 2
Cross-Attention Exposure 3

Figure 3. Qualitative comparison of the proposed *Local Cross-Attention Fusion* with the baseline methods *Raw HDR* and *Deep HDR* [6] on challenging scenes. Our neural fusion module recover features from separate exposure streams to support vision the downstream vision task.