Neural Auto-Exposure for High-Dynamic Range Object Detection
Supplemental Material

Emmanuel Onzon
Algolux

Fahim Mannan
Algolux

Felix Heide
Princeton University, Algolux

In this supplemental document, we provide additional detail on the image formation model, network architecture of the proposed exposure control method, and detailed training instructions to learn the proposed method. We also provide additional supplementary results at the end of this document.

1. Additional Details on the Image Formation Model

In the following, we provide detailed derivations of the noise model employed for the training in simulation and synthetic assessment, and we review multi-capture HDR image reconstruction.

1.1. Noise Model

We model the number of photoelectrons $y_p(\phi \cdot t)$ and dark currents electrons $y_d(\mu_d)$ for a given pixel with Poisson distributions, that is

$$y_p(\phi \cdot t) \sim \mathcal{P}(\phi \cdot t), \quad y_d(\mu_d) \sim \mathcal{P}(\mu_d).$$

The average number of electrons in the absence of light μ_d grows linearly with the exposure time

$$\mu_d = \mu_0 + \mu_I \cdot t.$$

We ignore the effect of temperature on μ_d.

Due to the properties of the Poisson distribution the variance equals the mean value *i.e.*, the standard deviations are as follows.

$$\sigma(y_p(\phi \cdot t)) = \sqrt{\phi \cdot t}, \quad \sigma(y_d(\mu_d)) = \sqrt{\mu_0 + \mu_I \cdot t}.$$

The pre- and post-amplification noises, are modeled as zero-mean gaussian variables.

$$n_{\text{pre}} \sim \mathcal{N}(0, \sigma^2_{\text{pre}}), \quad n_{\text{post}} \sim \mathcal{N}(0, \sigma^2_{\text{post}}).$$

We note that the constants μ_d, σ_{pre} and σ_{post} need to be calibrated.

The above sensor noise and the ADC's quantization noise affect the overall signal-to-noise ratio (SNR) and the dynamic range (DR) of the captured image.

1.2. SNR and Dynamic Range

Noise Variance. The total variance of the noise for unsaturated pixels of a single exposure can be derived from the model above. The unsaturated pixel value can be written as

$$I_{\text{unsat}} = g \cdot (y_p(\phi \cdot t) + y_d(\mu_d) + n_{\text{pre}}) + n_{\text{post}}$$

and its variance

$$V_{\text{unsat}} = g^2 \cdot \left((\phi + \mu_I) \cdot t + \mu_0 + \sigma^2_{\text{pre}}\right) + \sigma^2_{\text{post}} + \sigma^2_q. \tag{1}$$

The square error σ^2_q accounts here for the quantization error. We set it to the variance of the uniform probability distribution on [0, 1], *i.e.* $\sigma^2_q = 1/12$.

Signal-to-Noise Ratio. The squared signal-to-noise ratio (SNR) for a pixel receiving the radiant power ϕ can be derived as follows.

$$SNR(\phi)^2 = \frac{\phi^2 \cdot t^2 \cdot \delta_{I_{\text{sensor}} < M_{\text{white}}}}{(\phi + \mu_I) \cdot t + \mu_0 + \sigma^2_{\text{pre}} + (\sigma^2_{\text{post}} + \sigma^2_q)/g^2}.$$

The term $\delta_{I_{\text{sensor}} < M_{\text{white}}}$ is equal to 1 whenever the pixel value is below the maximum possible value and 0 otherwise. This term expresses the fact that the information is lost when a pixel is saturated at maximum value. For most sensors, the following is true for all ISO settings

$$M_{\text{white}} < g \cdot M_{\text{well}},$$

making M_{white} the deciding quantity for saturation. We could argue (as in [6]) that this loss of information may happen at lower values too, because of saturation at M_{well} followed by a negative noise n_{post}. We ignore this possibility here.

Dynamic Range. The dynamic range DR expressed in dB, is limited by the saturation at the higher end and by noise at the lower end. Here we consider the image sensor noise and ignore the optical noise which is acceptable for an LDR single-shot camera. Let ϕ_{sat} be the irradiance such that, on average, the pixel value just reaches M_{white} *i.e.*,

$$g \cdot (\phi_{\text{sat}} \cdot t + \mu_0 + \mu_I \cdot t) = M_{\text{white}},$$

and let ϕ_{min} be the irradiance such that the SNR equals 1. Solving for ϕ in the squared SNR expression we get:

$$\phi_{\text{min}} = \left(1 + \sqrt{1 + 4t \cdot v}\right)/(2t),$$

with

$$v = \mu_I + (\mu_0 + \sigma^2_{\text{pre}} + \left(\sigma^2_{\text{post}} + \sigma^2_q\right) \cdot g^{-2})/t.$$

The dynamic range DR expressed in dB is defined as

$$DR = 20 \cdot \log_{10}\left(\frac{\phi_{\text{sat}}}{\phi_{\text{min}}}\right).$$

1.3. Multi-Capture HDR Imaging

Next, we review multi-capture HDR image reconstruction. We note that multi-capture HDR imaging fails in dynamic scenes, introducing ghosting or SNR drops in the HDR transition regions [12]. In our work, we explore *single-shot* imaging with a learned adaptive exposure for dynamic scenes, departing from multi-capture methods that are fundamentally limited in dynamic scenes. A large body of HDR imaging methods [9, 1, 12, 4, 10, 7] has been devised in order to overcome the limitation of low dynamic range image sensors. Existing methods exceed the dynamic range of a single capture by acquiring multiple measurements using spatial or temporal multiplexing strategies. The most common approach is temporal multiplexing using exposure bracketing. In this widely popular approach, an LDR image sensor is employed to take several captures of the same scene with different exposures and then combine them to make an HDR image. Each of the LDR images covers a subset of the dynamic range of the scene, such that taken together the set of LDR images covers the full dynamic range of the scene. Numerous methods have been devised for merging the set of LDR images in order to produce an HDR image [11]. Existing methods [1, 12] typically express the reconstruction as a weighted average of the LDR images.

$$\mathbf{I}_{\text{hdr}} \propto \sum_{i=1}^{n} \beta_i \cdot \mathbf{I}_i,$$

where $\mathbf{I}_i$, $i = 1, \ldots, n$ is a set of linear light LDR images, and β_i are corresponding weighting and scaling factors. In particular $\beta_i = 0$ whenever the corresponding pixel value $\mathbf{I}_i$ is saturated. All existing multi-shot HDR methods have in common that they *fail for dynamic scenes*, resulting in ghosting artefacts or SNR drops in the HDR transition regions. While recent methods have aimed to employ computational hallucination methods [2], relying on a single exposure, or de-ghosting approaches [3], the compute complexity prohibits such methods for real-time robotic applications, *e.g.*, autonomous vehicles, that require real-time processing.

2. Neural Exposure Control Architecture Details

In this section, we provide additional details of the proposed neural exposure control method. We refer to the main draft for an overview of our exposure control approach.

Table 1: Global Image Feature Branch Architecture

Layer	Operation	Number of filters	Kernel size	Stride	Output shape
0	Input tensor	-	-	-	[256, 59]
1	1D Convolution	128	4	4	[64, 128]
2	1D Convolution	256	4	4	[16, 256]
3	1D Convolution	512	4	4	[4, 512]
4	Dense layer	1024	-	-	[1024]
5	Dense layer	16	-	-	[16]
6	Dense layer	1	-	-	[1]

2.1. Global Image Feature Branch

The input to the global image feature branch is a tensor of shape [256, 59] that represents 59 histograms, each with 256 bins, stacked together. These histograms are computed at three different scales as described in Section 4.1 of the main paper. The coarsest scale is the whole image which yields one histogram. At the intermediate scale, the image is divided up into 3 by 3 sub-images, yielding 9 histograms. At the finest scale, the image is divided up into 7 by 7 sub-images, yielding 49 histograms.

After computation and stacking of the histograms, the global image feature branch starts with a one-dimensional CNN. The first 3 layers are 1D convolutions where the convolution operates along the histograms. The width of the layers increases by doubling every layer, starting at 128. The convolution kernel size and the stride are equal to 4. Three dense layers follow, with a decreasing number of units, 1024 units for Layer 4, 16 units for Layer 5 and a single unit for Layer 6 which is the last layer. The activation function is ReLU for each layer except the last one that as the custom activation function (4) given in the main paper page 4 in Section 4.1. Table 1 describes the linear architecture of the global image feature branch and describes the hyperparameters of each layer.

2.2. Semantic Feature Branch

The input of the semantic feature branch is the activation of the last layer of the block conv2 of ResNet. Here we reuse an intermediate layer of the feature extractor used by the object detector (Faster-RCNN [13, 14]) at the current frame. More precisely, we use a variant of ResNet (ResNet 18) [8] as a feature extractor. This variant allows us to run the whole pipeline at 30 frames per second on a Nvidia GTX 1070 GPU.

At the beginning of the semantic feature branch, the ResNet conv2 feature map is first cropped. The first 120 rows only are kept. This makes for a convenient shape when pooling at different scales later. We also note that no important information is lost in the process given that the bottom of the image is mostly occupied by the hood of the car. After that cropping, the feature map undergoes a channel compression from 64 to 26 by using a 1x1 convolution, producing the compressed feature map (CFM). The channels of the CFM are pooled at 3 different scales. The first two channels are max pooled with a stride of 10 along rows and 20 along columns, which amounts to dividing up the feature map along rows and columns into a 12 by 12 array of sub tensors and computing the maximum of each of them channel wise. The next 8 channels of the CFM are max pooled with a stride of 20 along rows and 40 along columns, which amounts to dividing up the feature map into a 6 by 6 array of sub tensors and computing the maximum of each of them channel wise. The last 16 channels of the CFM are average pooled with a stride of 40 along rows and 80 along columns, which amounts to dividing up the feature map into a 3 by 3 array of sub tensors and computing the average of each of them channel wise. A fourth pooling is performed image wide on the cropped (64-channel) feature map, *i.e.* each of the 64 channels is averaged along the two spatial dimensions. Each of the tensors resulting from those 4 pooling operations are flattened, yielding vectors of lengths 288, 288, 144, and 64 respectively. They are concatenated together to a 784-long vector. These 784 units are then densely connected to a 16-unit layer which is the output of the semantic feature branch. Table 2 details the directed acyclic graph architecture of the semantic feature branch by specifying the input of each layer, as well as their hyperparameters.

2.3. Hybrid Model

The hybrid model combines both the global image feature branch and the semantic feature branch. The output of Layer 5 of the global image feature branch is summed to the output of the second fully connected layer (FC 2) of the semantic feature branch, after a rescaling. That is, the output of FC 2 is rescaled by a constant factor that we set to 0.01. This rescaling allows

Table 2: Semantic Feature Branch Architecture

Layer	Input	Operation	Number of filters	Kernel size	Stride	Output shape
ResNet conv2	-	Input tensor	-	-	-	[150, 240, 64]
Cropped feature map	ResNet conv2	Crop rows [0:120]	-	-	-	[120, 240, 64]
Compressed feature map (CFM)	Cropped feature map	Convolution	26	1	1	[120, 240, 26]
Max pool 1	CFM channels [0:2]	Max pool	-	10 x 20	[10, 20]	[12, 12, 2]
Max pool 2	CFM channels [2:10]	Max pool	-	20 x 40	[20, 40]	[6, 6, 8]
Avg pool 1	CFM channels [10:26]	Average pool	-	40 x 80	[40, 80]	[3, 3, 16]
Avg pool 2	ResNet conv2 cropped	Average pool	-	-	-	[1, 1, 64]
Pool out	Max pool 1 Max pool 2 Avg pool 1 Avg pool 2	Flatten + concatenate	-	-	-	[784]
FC 1	Pool out	Dense layer	1024	-	-	[1024]
FC 2	FC 1	Dense layer	16	-	-	[16]

the signal coming from both branches to be on the same order of magnitude.

2.4. Image Signal Processing (ISP) Pipeline

The raw image acquired by the camera is processed by a software *signal image processor* (ISP). For convenient end-to-end training it is entirely written with differentiable TensorFlow operations. Our ISP consists of a linear pipeline comprising processing blocks all commonly found in commercial ISPs. Specifically, we employ a demosaicer, a downsampler, a contrast enhancer, a low frequency denoiser, a sharpener and gamma correction. Some of these processing blocks use several parameters that are trained jointly with the other trainable variables (auto exposure, feature extractor and object detector). We note that the proposed method is orthogonal to the ISP employed, and, indeed, supports arbitrary image processing pipelines, as long as those are differentiable.

3. Training

3.1. HDR Training Dataset

The dataset that we use for training and evaluation has been captured with a camera equipped with the Sony IMX490 HDR sensor with fixed exposure settings. The whole dataset comprises 500 sequences of 5 successive HDR frames for a total of 2500 HDR frames. We select 400 sequences for the training data and use the 100 sequences left for test. From each of the 400 sequences we make 4 pairs of sucessive frames for a total of 1600 pairs, each of them being a training examples. We do the same for the test set which results in a total of 400 test examples.

3.2. LDR Image Capture Simulation

Noise Parameterization for Calibration and Capture Simulation. For the purpose of calibration and simulation we combine μ_I, μ_0 and σ_{pre} to a single term σ_d^2, which we call the variance of the dark noise, as follows:

$$\sigma_d^2 = \mu_I \cdot t + \mu_0 + \sigma_{\text{pre}}^2.$$

We do this for two reasons. The first reason is that we consider the exposure time as being fixed in the training pipeline, *i.e.* that the AE only adjusts the gain. This is an approximation which ignores that the camera gain setting K is bounded from below by 1. This approximation overestimates the standard deviation of the noise in the case where $K < 1$ is simulated. However, in the case of our target camera, the error induced by that approximation is bounded from above by $0.54 \cdot M_{\text{white}}$, such that we deem this approximation as acceptable in practice. The second reason for grouping these noise terms under σ_d^2 is that we do the common approximation of replacing the Poisson distribution of dark currents electrons $y_d(\mu_d)$ by a Gaussian distribution, which allows to simulate all the dark noise created before amplification as a single Gaussian random variable with a variance σ_d^2 which is the sum of σ_{pre}^2 and of the variance of $y_d(\mu_d)$.

For our target sensor (Sony IMX249) we also need to consider a row-wise amplifier noise that takes the form of horizontal lines on the images. This allows us to break down the variance of the dark noise σ_d^2 into two terms: $\sigma_d^2 = \sigma_{d,\text{ pix}}^2 + \sigma_{d,\text{ line}}^2$, where $\sigma_{d,\text{ line}}^2$ is the variance of the component of the dark noise that shows up as horizontal lines and $\sigma_{d,\text{ pix}}^2$ the variance of the component of the dark noise that is spatially uncorrelated.

Noise Calibration. Following the parametrization introduced in the paragraph above and in Section 1.1, we need to calibrate the following noise parameters: $\sigma_{d,\text{ pix}}$, $\sigma_{d,\text{ line}}$, σ_{post} and g_1. The parameter g_1 is not a standard deviation but it characterizes the camera shot noise. We recall that g_1 is the gain from electrons to DN (digital numbers) at ISO 100 (*i.e.*, when $K = 1$), such that, in the general case, the gain g can be written $g = g_1 \cdot K$.

The signal independent noise can be calibrated from a set of dark frame captures (raw images) taken at various gains. The variance of that noise can be written as $K^2 \cdot g_1^2 \cdot \sigma_d^2 + \sigma_{\text{post}}^2$, such that a regression against K^2 allows to estimate $g_1^2 \cdot \sigma_d^2$ and σ_{post}^2. In the case of our target camera we find out that σ_{post}^2 is negligible.

Then $g_1^2 \cdot \sigma_{d,\text{ line}}^2$ is estimated using the dark frames averaged along the rows. From $g_1^2 \cdot \sigma_d^2$ and $g_1^2 \cdot \sigma_{d,\text{ line}}^2$ we deduce $g_1^2 \cdot \sigma_{d,\text{ pix}}^2$.

Once $g_1^2 \cdot \sigma_{d,\text{ pix}}^2$ and $g_1^2 \cdot \sigma_{d,\text{ line}}^2$ have been calibrated, the gain g_1 is estimated from raw images of a set of pictures of a color checker chart, taken at various gains under a roughly uniform illumination. The temperature of the illuminant is not of importance in this process. The mean value of each patch pixel is estimated using a local polynomial estimator within the pixel's patch.

Noise Adaptation. We propose to train our model with images that contain noise distributed as the noise created by the target camera (the one which our trained model will be used with). Our training dataset is composed of images taken with the Sony IMX490. As such, they already contain noise produced by that sensor. We perform *noise adaptation* during training from the source camera sensor (Sony IMX490) to the target camera sensor (Sony IMX249). This consists in adding just the right amount of noise to the image such that after noise adaptation the noise contained in the image matches the distribution of the noise of the target camera.

To do so we need to calibrate the noise distributions of both the source and target camera. We use the approach described above for those calibration, even though the induced noise model is only an approximation here.

First we note that the images of the training set have been rescaled to match the resolution of the target camera. For a given pixel in an (HDR) image of the training set, there is a mean number of photo-induced electrons $\mu_{\text{p, source}}$. Suppose the exact same scene was taken with the target camera from the exact same point of view. Then for the corresponding pixel in the resulting raw image, there is a mean number of photo-induced electrons $\mu_{\text{p, target}}$. We assume that $\mu_{\text{p, source}} = \mu_{\text{p, target}}$, when the camera gain settings $K = 1$ for the target camera. This can be realized in practice by adjusting the aperture and exposure time of the target camera given that the images of the training set have all been taken with the same fixed exposure settings. Those adjustments are based on the aperture and exposure time of the source camera, as well as the pixel sizes and the quantum efficiencies of both the source and target sensors. The assumption $\mu_{\text{p, source}} = \mu_{\text{p, target}}$ implies that to simulate a raw image for the target camera from the source camera we need to multiply the raw pixel value of the source camera by $g_{\text{source}}^{-1} \cdot g_{\text{target}}$. Here g_{source} and g_{target} are the quantities corresponding to the gain g introduced in Section 3 of the main paper, for the source and target cameras respectively. However the resulting simulated raw image still does not include noise adaptation for the dark noise.

To complete noise adaptation we need to match the dark noise of the target camera. Assuming $\sigma_{d,\text{ source}}^2$ and $\sigma_{d,\text{ target}}^2$ are the variances of the dark noise for the source and the target cameras, we add a to the pixel values a Gaussian noise of variance

$$\sigma_{\text{sim}}^2 = g_{\text{target}}^2 \cdot (\sigma_{d,\text{ target}}^2 - \sigma_{d,\text{ source}}^2).$$

This is only possible if $\sigma_{d,\text{ source}} < \sigma_{d,\text{ target}}$, which is the case for our chosen source and target sensors. For the special case of a target sensor that includes an horizontal line noise as described above, we add both spatially uncorrelated and horizontal line noises with corresponding variances computed as follows

$$\sigma_{\text{sim, pix}}^2 = g_{\text{target}}^2 \cdot (\sigma_{d,\text{ target}}^2 - \sigma_{d,\text{ pix}}^2), \quad \sigma_{\text{sim, line}}^2 = \sigma_{d,\text{ line}}^2.$$

Noise Augmentation. For the purpose of data augmentation we depart slightly from the way noise adaptation is outlined above. We randomly vary the strength of the simulated dark noise around the strength targeted by noise adaptation. More precisely, we compute σ_{sim}^2 as

$$\sigma_{\text{sim}}^2 = \max(0, g_{\text{target}}^2 \cdot (\sigma_{d,\text{ target}}^2 \cdot k_{\text{aug}} - \sigma_{d,\text{ source}}^2)),$$

where $\log(k_{\text{aug}})$ is sampled uniformly in $[\log(0.25), \log(4)]$ and set to the same value for all the pixels of a given image pair example. In the case of a target sensor that includes an horizontal line noise, the noise augmentation is applied as follows.

$$\sigma^2_{\text{sim, pix}} = g^2_{\text{target}} \cdot (\sigma^2_{d,\text{ target}} \cdot k_{\text{aug}} - \sigma^2_{d,\text{ pix}}), \quad \sigma^2_{\text{sim, line}} = \sigma^2_{d,\text{ line}} \cdot k_{\text{aug}}.$$

3.3. Network Training

In this section, we provide the values of the training hyperparameters and the learning rate schedule, as well as additional pretraining details.

Pretraining. The feature extractor has first been pretrained on ImageNet (ILSVRC2012). Then the object detector has been pretrained jointly with the ISP on several public and proprietary automotive datasets. This trained joint model (ISP + object detector) is reused as a starting point for the training of the two baselines and the two proposed models discussed in the main document.

Learning Rate Schedule. For each of the two baselines and the two proposed models, the learning rate schedule is the same. We train for 20,000 steps with a learning rate 0.0003, then an additional 20,000 steps with a learning rate 0.0001 and finally 20,000 more steps with a learning rate 0.00003.

Training Hyperparameters. We use a batch size of 1. The localization and objectness loss weights of the RPN are 4 and 3, the localization and classification loss weights of the second stage are 4 and 2. The number of proposals from the RPN is 300. We use L2 regularization for the weights of the autoexposure neural network only, with weight 0.001. We clip the gradient when the norm is above 10 for the global image feature branch and for the hybrid model, and when the norm is above 5 for the semantic image feature branch alone.

Two stage training for the hybrid model. The hybrid model is trained in two stages. We first train the semantic feature branch alone. Next, we add the global image feature branch to the network to make the full hybrid model and we repeat the training, following the same training procedure, including the same learning rate schedule.

4. Evaluation

4.1. Annotation Process

General requirements For training and evaluation street objects are grouped into 6 categories, namely, Car/Van/SUV, Bus/Truck/Tram, Bike, Person, Traffic Sign, Traffic Lights. The Car/Van/SUV category is mainly for light to medium sized vehicles, while Bus/Truck/Tram includes medium to heavy duty vehicles, such as, construction vehicles. The Bike category includes bicycles, motorcycles and any other light transportation that have similar shape to a bicycle or motorcycle. Person category includes pedestrians, cyclists and their full extent is annotated. For groups of people, every individual is annotated separately. Traffic sign includes all standard traffic sign categories including electronic signs, and Traffic lights include lights for vehicles, public transports, pedestrians and cyclists.

For all annotations only the visible extent of the objects are annotated as tightly as possible. Objects smaller than 5x5 pixels are ignored.

Annotation requirements for live evaluation data For live evaluation, captures were obtained by running two different auto exposure algorithms on a stereo pair. The main challenge while annotating these LDR images is that some of the regions can be either underexposed or overexposed. However, because we use two different algorithms, one of the two exposures are likely to have these regions properly exposed.

To annotate these live evaluation data, we used a sequence of exposure pairs for annotation. The annotations for over and underexposed images were done by first trying to adjust the brightness and contrast of the images to maximize object visibility. If they are still not visible, the annotators chose the corresponding well exposed image and transferred the annotation to the badly exposed image while making sure that the annotations are spatially and temporally consistent. Each annotated sequence was checked for correctness by a quality controller and the annotations were adjusted as needed.

Table 3: Synthetic comparison between a conventional HDR pipeline and LDR images auto-exposed with our proposed method. The reported scores are the average precision at IoU 0.5 for each of the 6 classes and the mean across classes. See text for additional details.

Method	**Classes**						
	All Classes	Bike	Bus & Truck	Car & Van	Person	Traffic Light	Traffic Sign
CONVENTIONAL HDR DETECTION	10.6	3.4	12.9	29.9	8.8	2.1	6.4
PROPOSED LDR HYBRID NN *(ours)*	**25.0**	**19.7**	**22.0**	**47.0**	**24.2**	**13.6**	**23.5**

4.2. Synthetic Assessment

Comparison with Conventional HDR Detection Pipelines. In Table 3 we provide results of a synthetic comparison between object detection on the output of an HDR ISP, the ARM Mali C71 which ingests an HDR RAW image, and the proposed method using an LDR image exposed using the proposed neural exposure control. We run the commercial ARM Mali C71 HDR ISP on the HDR raw images and run the pretrained object detector mentioned in Section 3.3 on the output of that ISP. The detector was finetuned on the post-ISP images from this HDR ISP. For comparison, we simulate a LDR capture from the previous frame HDR raw image and compute an exposure adjustment for the test frame (HDR raw image), from which we simulate a LDR capture that we process with our trained pipeline (ISP + object detector). For this experiment, we do not apply noise adaptation nor noise augmentation as the goal is to compare the use of HDR images with the use of LDR images auto-exposed with our method, but not to validate the method for a specific target camera. It can be seen from Table 3 that the use of our joint model (trained AE + ISP + detector) outperforms the traditional pipeline consisting of an HDR sensor followed by a conventional HDR ISP and an object detector trained on ISP-processed RGB images.

4.3. Experimental Assessment

Prototype Vehicle Setup. We provide additional information on the prototype vehicle in this section. Each of the two cameras is free-running and takes input image streams from separate imagers mounted side-by-side on the windshield of a vehicle, see Figure 1. Images are recorded with the object detector and each AE algorithm running live. All compared AE methods and inference pipelines run in real-time on two separate machines, each equipped with a Nvidia GTX 1070 GPU.

Qualitative Validation. We provide additional qualitative results from the experimental side-by-side comparison experiment. Those could not be included in the main paper because of space limitation and appear in Fig. 2 of this supplemental document. Again it can be seen that in many instances, the proposed method is capable of carefully balancing the exposure between dark and bright objects even in rapidly changing conditions. We also refer the reader to the companion videos included as supplemental material.

Comparison with HDR exposure selection methods We implemented the method from [5]. For a fair comparison, we use the same camera as with our live experiment. We captured data with their method and show an example in Fig. 3. We see ghosting artifacts on vehicles and lane marking, validating the effectiveness of the proposed approach over conventional HDR acquisition methods with exposure selection.

References

[1] Paul E. Debevec and Jitendra Malik. Recovering high dynamic range radiance maps from photographs. In *SIGGRAPH '08*, 1997. 2

[2] Gabriel Eilertsen, Joel Kronander, Gyorgy Denes, Rafał K Mantiuk, and Jonas Unger. Hdr image reconstruction from a single exposure using deep cnns. *ACM Transactions on Graphics (TOG)*, 36(6):178, 2017. 2

[3] Orazio Gallo, Natasha Gelfandz, Wei-Chao Chen, Marius Tico, and Kari Pulli. Artifact-free high dynamic range imaging. *2009 IEEE International Conference on Computational Photography (ICCP)*, pages 1–7, 2009. 2

[4] Michael D. Grossberg and Shree K. Nayar. High dynamic range from multiple images: Which exposures to combine? 2003. 2

[5] Mohit Gupta, Daisuke Iso, and Shree K Nayar. Fibonacci exposure bracketing for high dynamic range imaging. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 1473–1480, 2013. 7, 11

[6] Samuel W Hasinoff, Frédo Durand, and William T Freeman. Noise-optimal capture for high dynamic range photography. In *2010 IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, pages 553–560. IEEE, 2010. 2

[7] Samuel W. Hasinoff, Frédo Durand, and William T. Freeman. Noise-optimal capture for high dynamic range photography. *2010 IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, pages 553–560, 2010. 2

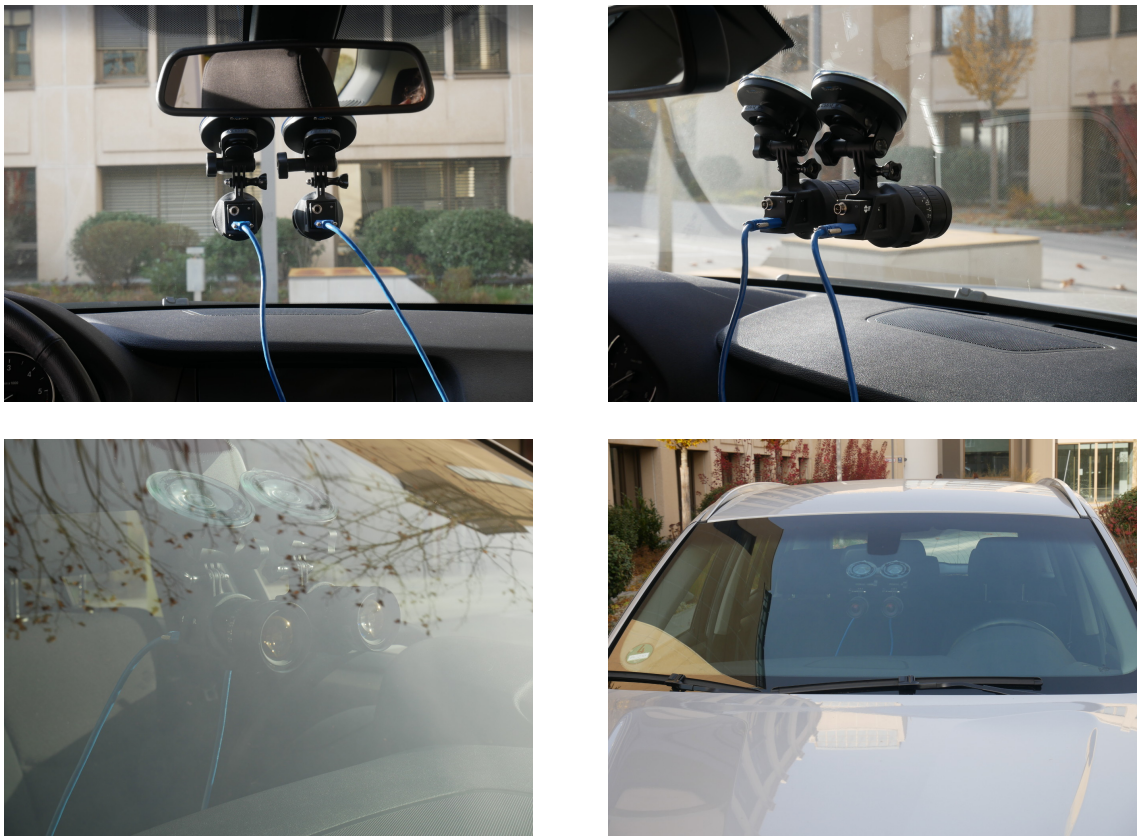


Figure 1: Vehicle Prototype. We mount two cameras with free-running auto-exposure side-by-side on the windshield of a testing vehicle. The detections of the two separate camera systems are independently annotated for a fair comparison and evaluation.

[8] 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. 3

[9] Steve Mann and Rosalind W. Picard. Being 'undigital' with digital cameras: extending dynamic range by combining differently exposed pictures. 1994. 2

[10] Tom Mertens, Jan Kautz, and Frank Van Reeth. Exposure fusion: A simple and practical alternative to high dynamic range photography. *Comput. Graph. Forum*, 28:161–171, 2009. 2

[11] Erik Reinhard, Wolfgang Heidrich, Paul Debevec, Sumanta Pattanaik, Greg Ward, and Karol Myszkowski. *High dynamic range imaging: acquisition, display, and image-based lighting*. Morgan Kaufmann, 2010. 2

[12] Erik Reinhard, Greg Ward, Summant Pattanaik, Paul E. Debevec, Wolfgang Heidrich, and Karol Myszkowski. High dynamic range imaging: Acquisition, display, and image-based lighting. 2010. 2

[13] 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

[14] Shaoqing Ren, Kaiming He, Ross Girshick, Xiangyu Zhang, and Jian Sun. Object detection networks on convolutional feature maps. *IEEE transactions on pattern analysis and machine intelligence*, 39(7):1476–1481, 2016. 3

[15] 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.

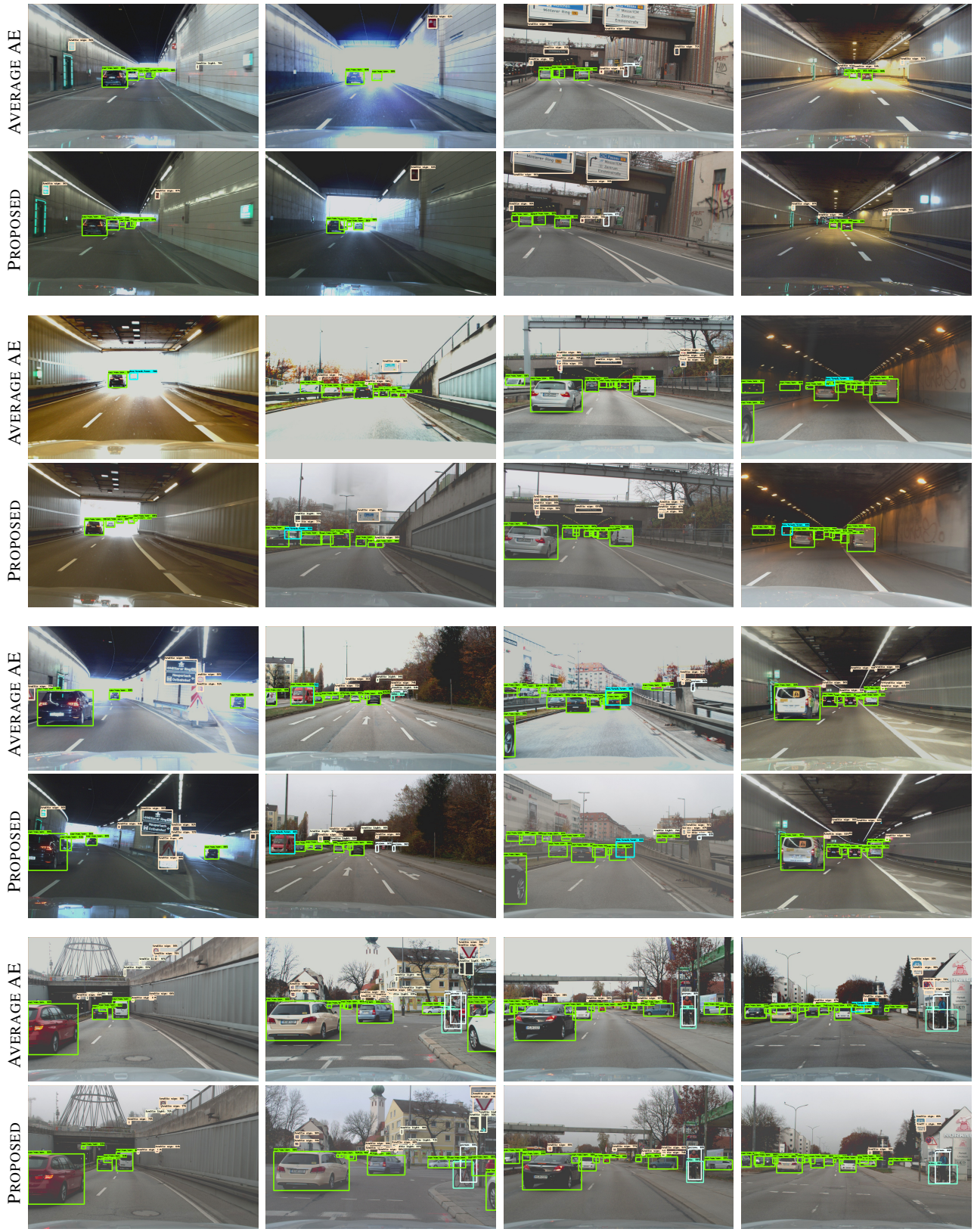


Figure 2: Experimental prototype results of the proposed neural AE compared to the Average AE baseline method using the real-time side-by-side prototype vehicle capture system shown in Fig. 1. The proposed method accurately balances exposure between objects and adapts itself robustly to changing conditions.

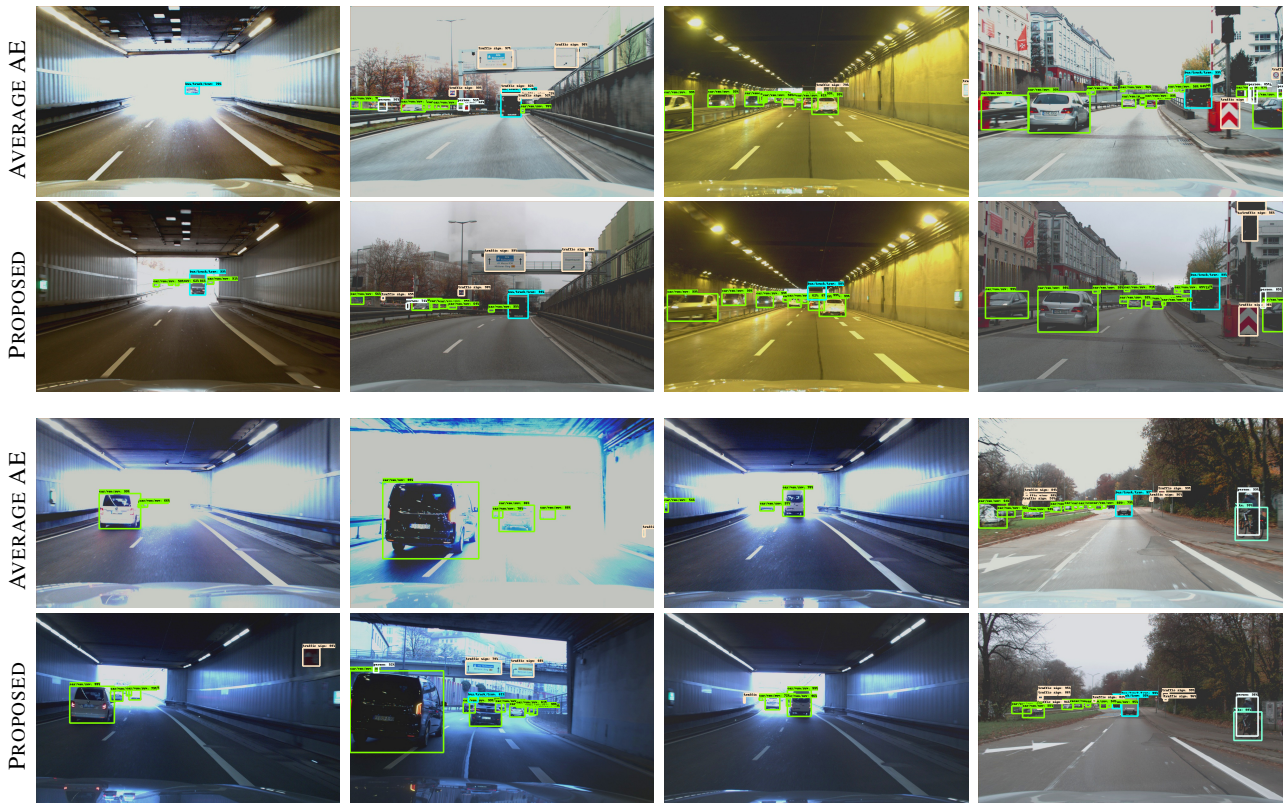
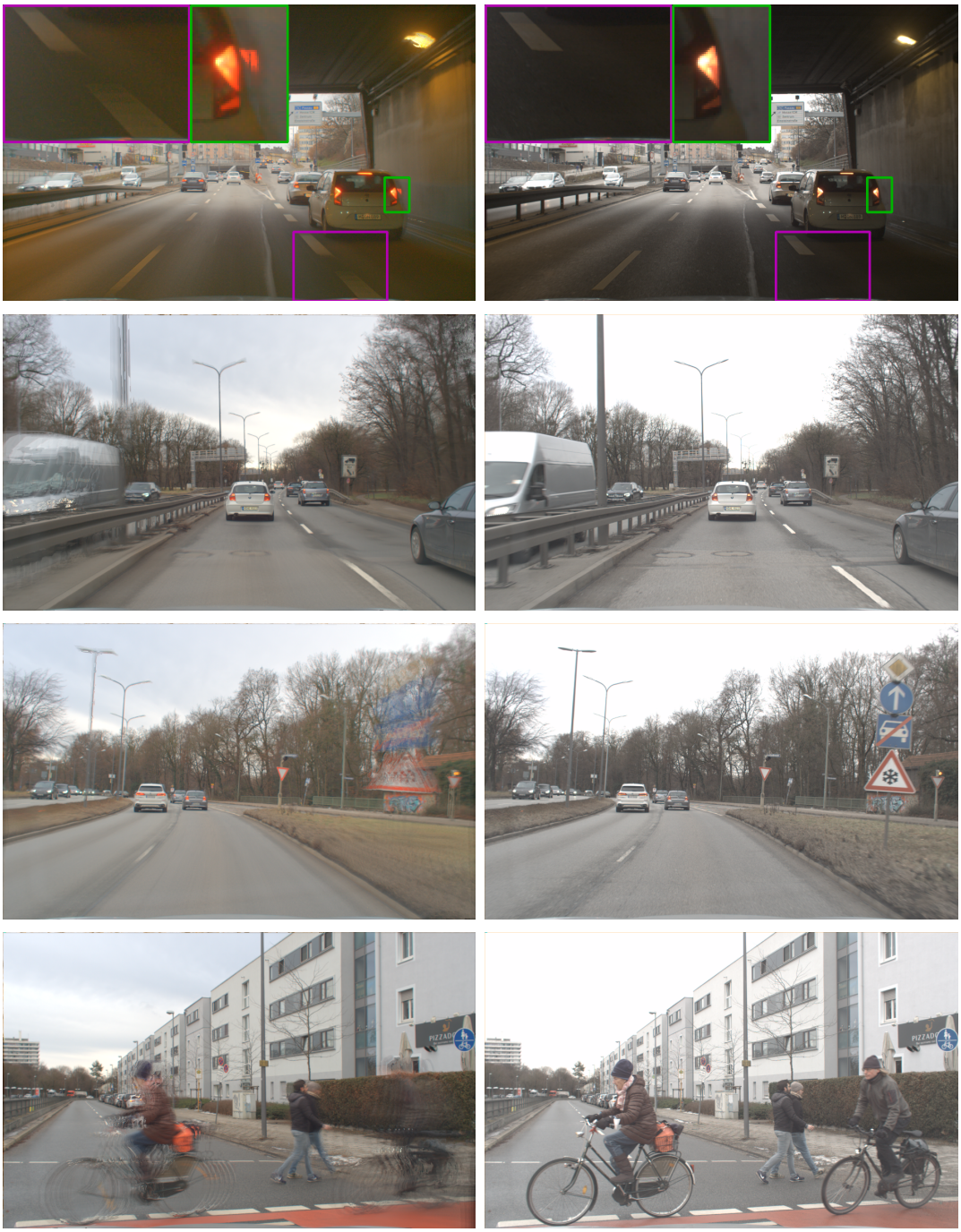


Figure 2: Experimental prototype results of the proposed neural AE compared to the Average AE baseline method using the real-time side-by-side prototype vehicle capture system shown in Fig. 1. The proposed method accurately balances exposure between objects and adapts itself robustly to changing conditions.



HDR fusion (Gupta *et al.* [5])

Best LDR exposure

Figure 3: HDR exposure fusion results in severe ghosting artifacts in scenes with a lot of motion. This makes it impractical for our automotive application.