A Appendices

A.1 Theoretical framework

Let $(X, Y) : \Omega \to \mathbb{R}^{H,W} \times \mathcal{Y}$ be the raw sensor data generating random variable on some probability space $(\Omega, \mathcal{F}, \mathbb{P})$, where $W, H \in \mathbb{N}$ correspond to width and height of a raw sensor image. For classification, set $\mathcal{Y} = \{0,1\}^K$ and $\mathcal{Y} = \{0,1\}^{H,W}$ for segmentation. We aim to learn a task model $\boldsymbol{\Phi}_{\text{Task}} : \mathcal{X} \to \mathcal{Y}$ within a fixed class of task models $\mathcal{H}$ that minimizes the expected loss wrt. the loss function $\mathcal{L} : \mathcal{X} \times \mathcal{Y} \to [0, \infty)$, i.e. we wish to find $\boldsymbol{\Phi}^\star_{\text{Task}}$ such that

$$\inf_{\boldsymbol{\Phi}_{\text{Task}} \in \mathcal{H}} \mathbb{E}[\mathcal{L}(\boldsymbol{\Phi}_{\text{Task}}(X), Y)]$$

is attained. Towards that goal, we determine during training $\hat{\boldsymbol{\Phi}}^\star_{\text{Task}}$ such that the empirical error

$$\frac{1}{N} \sum_{n=1}^{N} \mathcal{L}(\hat{\boldsymbol{\Phi}}^\star_{\text{Task}}(\boldsymbol{x}_n), y_n)$$

is minimized over the sample $\mathcal{S} = ((\boldsymbol{x}_1, y_1), ..., (\boldsymbol{x}_N, y_N))$. This procedure is justified, if the observations in the sample are realisations of independent and identical distributed (*i.i.d.*) copies of the raw data generating random variable. The inputs that are given to the task model $\boldsymbol{\Phi}_{\text{Task}}$ are the outputs of image signal processing (ISP). Within ISP, several transformations are applied to the raw sensor image $\boldsymbol{x} \in \mathbb{R}^{H,W}$ resulting in the processed image $\boldsymbol{v} \in \mathbb{R}^{C,H,W}$. We distinguish between the raw sensor image $\boldsymbol{x}$ and a *view* $\boldsymbol{\Phi}_{\text{Proc}}(\boldsymbol{x})$ of this image, where

$$\boldsymbol{\Phi}_{\text{Proc}} : \mathbb{R}^{H,W} \to \mathbb{R}^{C,H,W}$$

models the ISP. The classical setting directly starts with $(\boldsymbol{\Phi}_{\text{Proc}}(X), Y)$ as the data generating random variable, leading to the same target distribution $\mathcal{D}_t = \mathbb{P} \circ (\boldsymbol{\Phi}_{\text{Proc}}(X), Y)^{-1}$ as in our setting. To contrast the two approaches, let $\tilde{\boldsymbol{\Phi}}_{\text{Proc}}$ model a different ISP, generating a different view of the same underlying raw sensor data, with distribution

$$\mathcal{D}_s = \mathbb{P} \circ (\tilde{\boldsymbol{\Phi}}_{\text{Proc}}(X), Y)^{-1} \neq \mathcal{D}_t.$$

Assume further, that during inference wrt. observation drawn from $\mathcal{D}_s$ the performance of the task model unexpectedly decreases. In the classical setting, it is tempting to explain this decrease by a distribution shift wrt. the underlying raw data generating random variable, whereas it is truly caused by a different ISP. To go beyond this theoretical nuance, an explicit model of the ISP is needed. We provide two such models: a static model Φ^{stat}_{Proc} and a parametrized model Φ^{θ}_{Proc}. For a precise description of the two models see A.2.

A.2 Description of the processing models

In the following we denote by $\boldsymbol{x} \in [0,1]^{H,W}$ the normalized raw image, that is a grey scale image with a Bayer filter pattern normalized by $2^{16} - 1$, i.e.

$$x = \begin{bmatrix} A_{1,1} & . & . & . & A_{1,\frac{W}{2}} \\ . & . & & & . \\ . & & . & & . \\ . & & & . & . \\ A_{\frac{H}{2},1} & . & . & . & A_{\frac{H}{2},\frac{W}{2}} \end{bmatrix}, \quad \textit{with} \quad A_{i,j} = \begin{bmatrix} R_{2i+1,2j+1} & G_{2i+1,2j} \\ G_{2i,2j+1} & B_{2i,2j} \end{bmatrix},$$

where the values $R_{2i+1,2j+1}, G_{2i+1,2j}, G_{2i,2j+1}, B_{2i,2j}$ correspond to the values measured through the different sensors and normalized by $2^{16} - 1$. We provide a here a precise description of the transformations that we consider in our static model, followed by a description how we this converted this static model into a differentiable model.

A.2.1 The static pipeline

Black level correction (BL) In this transformation, thermal noise and readout noise generated from the camera sensor are removed by $\boldsymbol{bl} \in [0,1]^4$. The transformation is given by

$$\Phi_{BL} : [0,1]^{H,W} \to [0,1]^{H,W}, \boldsymbol{x} \mapsto \boldsymbol{v}_{BL},$$

with

$$\begin{aligned}(v_{BL})_{2h+1,2w+1} &= x_{2h+1,2w+1} - bl_1\\ (v_{BL})_{2h,2w+1} &= x_{2h,2w+1} - bl_2\\ (v_{BL})_{2h+1,2w} &= x_{2h+1,2w} - bl_3\\ (v_{BL})_{2h,2w} &= x_{2h,2w} - bl_4,\end{aligned}$$

for all $h \leq H$ and all $w \leq W$. By construction, $\|\boldsymbol{b}\|_{max} \leq \min_{h,w}\{x_{h,w}\}$, such that $\boldsymbol{v}_{BL}$ is again an element of $[0,1]^{H,W}$.

Demosaicing (DM) The transformation

$$\Phi_{DM} : [0,1]^{H,W} \to [0,1]^{3,H,W}, \boldsymbol{v} \mapsto \boldsymbol{v}_{DM}$$

is applied to reconstruct the full RGB color image, by applying a certain interpolation rule. We use one out of the three demosaicing algorithms from the color-demosaicing

- BayerBilinear (Φ_{DM}^{Bil})
- Menon2007 (Φ_{DM}^{Men})
- Malvar2004 (Φ_{DM}^{Mal}).

For the used interpolation formulars see [5], [29] and [26].

White balance (WB) To obtain a neutrally illuminated image, intensities are adjusted by $\boldsymbol{wb} \in [0,1]^3$. The corresponding map is given by

$$\Phi_{WB} : [0,1]^{3,H,W} \to [0,1]^{3,H,W}, \boldsymbol{v} \mapsto \boldsymbol{v}_{WB},$$

where

$$(v_{WB})_{c,h,w} = wb_c \cdot (v_{DM})_{c,h,w} \quad \textit{for} \quad c \leq 3 \quad \textit{and all} \quad h \leq H, w \leq W.$$

Color correction (CC) By considering color dependencies, this transformation is applied to balance hue and saturation of the image. Let $\boldsymbol{M} \in \mathbb{R}^{3,3}$ be color matrix. The transformation is defined by

$$\Phi_{CC} : [0,1]^{3,H,W} \to \mathbb{R}^{3,H,W}, \boldsymbol{v} \mapsto \boldsymbol{v}_{CC} := \Phi_{CC}(\boldsymbol{v}_{WB}),$$

where

$$\begin{bmatrix}(v_{CC})_{1,h,w}\\(v_{CC})_{2,h,w}\\(v_{CC})_{3,h,w}\end{bmatrix} = \boldsymbol{M}\begin{bmatrix}(v_{WB})_{1,h,w}\\(v_{WB})_{2,h,w}\\(v_{WB})_{3,h,w}\end{bmatrix} \quad \textit{for all} \quad h \leq H, w \leq W.$$

The entries of the resulting $\boldsymbol{v}_{CC}$ are no longer restricted to $[0,1]$.

Sharpening (SH) To reduce image blurriness, sharpening is applied. We use one out of the two methods:

- sharpening filter
- unsharp masking.

To convert the view $\boldsymbol{v}_{VV}$ to YUV-color space and back to the RGB-color space we use the skimage.color functions rgb2yuv (Φ_{YUV}) and yuv2rgb (Φ_{YUV}^{-1}). The sharpening filter

$$\Phi_{SH}^{SF} : \mathbb{R}^{H,W} \to \mathbb{R}^{H,W},$$

is defined by a convolution

$$(\Phi_{SH}^{SF}(\boldsymbol{v}))_{h,w} := (\boldsymbol{v} \star \boldsymbol{k})_{h,w}, \quad \boldsymbol{k} := \begin{bmatrix}0 & -1 & 0\\ -1 & 5 & -1\\ 0 & -1 & 0\end{bmatrix}. \tag{2}$$

For unsharp masking we use the ski.filters function unsharp_mask (Φ_{SH}^{UM}). To formally define the sharpening we write

$$\Phi_{SH} : \mathbb{R}^{3,H,W} \to \mathbb{R}^{3,H,W}, \boldsymbol{v} \mapsto \boldsymbol{v}_{SH} := \Phi_{YUV}^{-1} \circ h \circ \Phi_{YUV}(\boldsymbol{v}_{CC}),$$

with $h \in \{\Phi_{SH}^{SF}, \Phi_{SH}^{UM}\}$.

Denoising (DN) Sharpening indroduces noise in the image. To reduce this noise we apply one out of the two methods

used methods	$(\Phi_{DM}^{Bil}, \Phi_{SH}^{SF}, \Phi_{DN}^{MD})$	$(\Phi_{DM}^{Bil}, \Phi_{SH}^{SF}, \Phi_{DN}^{GD})$	$(\Phi_{DM}^{Bil}, \Phi_{SH}^{UM}, \Phi_{DN}^{MD})$	$(\Phi_{DM}^{Bil}, \Phi_{SH}^{UM}, \Phi_{DN}^{GD})$
abbreviation	(bi,s,me)	(bi,s,ga)	(bi,u,me)	(bi,u,ga)
used methods	$(\Phi_{DM}^{Men}, \Phi_{SH}^{SF}, \Phi_{DN}^{MD})$	$(\Phi_{DM}^{Men}, \Phi_{SH}^{SF}, \Phi_{DN}^{GD})$	$(\Phi_{DM}^{Men}, \Phi_{SH}^{UM}, \Phi_{DN}^{MD})$	$(\Phi_{DM}^{Men}, \Phi_{SH}^{UM}, \Phi_{DN}^{GD})$
abbreviation	(me,s,me)	(me,s,ga)	(me,u,me)	(me,u,ga)
used methods	$(\Phi_{DM}^{Mal}, \Phi_{SH}^{SF}, \Phi_{DN}^{MD})$	$(\Phi_{DM}^{Mal}, \Phi_{SH}^{SF}, \Phi_{DN}^{GD})$	$(\Phi_{DM}^{Mal}, \Phi_{SH}^{UM}, \Phi_{DN}^{MD})$	$(\Phi_{DM}^{Mal}, \Phi_{SH}^{UM}, \Phi_{DN}^{GD})$
abbreviation	(ma,s,me)	(ma,s,ga)	(ma,u,me)	(ma,u,ga)

Table 1: Abbreviations for the used configurations of the static pipeline to create twelve different views of the same underlying raw sensor data.

- Gaussian denoising
- Median denoising.

Again, the view is converted to YUV-color space and back to the RGB-color space. For Gaussian denoising, we apply a Gaussian filter (Φ_{DN}^{GD}) with standard deviation of $\sigma = 0.5$ from the scipy.ndimage package. For median denoising we apply a median filter (Φ_{DN}^{MD}) of size three from the scipy.ndimage package. Formally, this reads as

$$\Phi_{DN} : \mathbb{R}^{3,H,W} \to \mathbb{R}^{3,H,W}, \boldsymbol{v} \mapsto \boldsymbol{v}_{SH} := \Phi_{YUV}^{-1} \circ h \circ \Phi_{YUV}(\boldsymbol{v}_{CC}),$$

with $h \in \{\Phi_{DN}^{GD}, \Phi_{DN}^{MD}\}$.

Gamma correction (GC) In the end, the overall brightness of the image is equilibrated. First, the entries of the view $\boldsymbol{v}_{DN}$ are clipped to $[0, 1]$ leading to

$$(v_{CP})_{c,h,w} := (v_{DN})_{c,h,w} \mathbb{1}_{\{0 \leq (v_{DN})_{c,h,w} \leq 1\}} + \mathbb{1}_{\{(v_{DN})_{c,h,w} > 1\}}.$$

The transformation is than defined as

$$\Phi_{GC} : \mathbb{R}^{3,H,W} \to [0, 1]^{3,H,W}, \boldsymbol{v} \mapsto \boldsymbol{v}_{CC} := ((\boldsymbol{v}_{CP})_{c,h,w})^{\frac{1}{\gamma}},$$

for some $\gamma > 0$. Note that zero-clipping is necessary for $(v_{CC})_{c,h,w}$ to be well-defined.

We call the composition

$$\Phi_{Proc}^{stat} : [0, 1]^{H,W} \mapsto [0, 1]^{C,H,W},$$

with

$$\boldsymbol{x} \mapsto \Phi_{GC} \circ \Phi_{DN} \circ \Phi_{SH} \circ \Phi_{CC} \circ \Phi_{WB} \circ \Phi_{DM} \circ \Phi_{BL}(\boldsymbol{x}), \tag{3}$$

of the above steps the *static pipeline*.

To test the effect of different static pipelines on the performance of two task models, we fix the continuous features $\boldsymbol{bl}, \boldsymbol{wb}, \boldsymbol{M}$ and γ, but vary the demosaicing method, the sharpening method and the denoising method, resulting in twelve different views of the same underlying raw sensor data, generated by different choices for Φ_{Proc}^{stat}. In Fig. **??** are shown all pipelines used for the ABtesting experiment 5.1.

A.2.2 The parametrized pipeline

We aim to design a differentiable model

$$\Phi_{Proc}^{\theta} := \Phi_{Proc}(\cdot, \boldsymbol{\theta}),$$

such that

$$\Phi_{Proc}(\cdot, \boldsymbol{\theta}_0) = \Phi_{Proc}^{stat}$$

holds true for some choice of parameters $\boldsymbol{\theta}_0$.

Black level correction (BL) For fixed $\boldsymbol{x} \in [0,1]^{H,W}$ define the map

$$\Phi^{\theta^{BL}} : \mathbb{R}^4 \to \mathbb{R}^{H,W}, \boldsymbol{\theta}_1 \mapsto \Phi_{BL}(x)|_{\boldsymbol{bl}=\boldsymbol{\theta}_1}.$$

Demosaicing (DM) We restrict to the case, where BayerBilinear is used as the method for demosaicing. Fix $\boldsymbol{v}_{BL} \in [0,1]^{H,W}$. Let

$$\Phi^{\theta^{DM}} : \mathbb{R}^{3,3} \times \mathbb{R}^{3,3} \to \mathbb{R}^{3,H,W},$$

where we define for $\boldsymbol{\theta}_2 := (\boldsymbol{k}_1, \boldsymbol{k}_2)$ with $\boldsymbol{k}_1, \boldsymbol{k}_2 \in \mathbb{R}^{3,3}$

$$\begin{aligned}
(\Phi^{\theta^{DM}}(\boldsymbol{k}_1, \boldsymbol{k}_2))_{2h+1,2w+1} &:= (\boldsymbol{v}_{BL} \star \boldsymbol{k}_2)_{2h+1,2w+1} \\
(\Phi^{\theta^{DM}}(\boldsymbol{k}_1, \boldsymbol{k}_2))_{2h,2w+1} &:= (\boldsymbol{v}_{BL} \star \boldsymbol{k}_1)_{2h,2w+1} \\
(\Phi^{\theta^{DM}}(\boldsymbol{k}_1, \boldsymbol{k}_2))_{2h+1,2w} &:= (\boldsymbol{v}_{BL} \star \boldsymbol{k}_1)_{2h+1,2w} \\
(\Phi^{\theta^{DM}}((\boldsymbol{k}_2, \boldsymbol{k}_2))_{2h,2w} &:= (\boldsymbol{v}_{BL} \star \boldsymbol{k}_2)_{2h,2w}.
\end{aligned}$$

White balance (WB) Fix $\boldsymbol{v}_{DM} \in \mathbb{R}^{3,H,W}$ and define

$$\Phi^{\theta^{WB}} : \mathbb{R}^3 \to \mathbb{R}^{3,H,W}, \boldsymbol{\theta}_3 \mapsto \Phi_{WB}(\boldsymbol{v}_{DM})|_{\boldsymbol{wb}=\boldsymbol{\theta}_3}$$

Color correction (CC) Define

$$\Phi^{\theta^{CC}} : \mathbb{R}^{3,3} \to \mathbb{R}^{3,H,W}, \boldsymbol{\theta}_4 \mapsto \Phi_{CC}(v_{WB})|_{\boldsymbol{M}=\boldsymbol{\theta}_4}.$$

Sharpening (SH) We restrict to the case where the sharpening filter method is applied. Fix $\boldsymbol{v}_{CC} \in \mathbb{R}^{C,H,W}$. The transformation from the RGB-color space to the YUV-color space can be written as a convolution of $\boldsymbol{v}_{CC}$ and a kernel $\boldsymbol{k}_{rgb2yuv} \in \mathbb{R}^{3,3}$ and the transformation from the YUV-color space to the RGB-color space can be written as well as a convolution of $\Phi_{SH}(\Phi_{YUV}(\boldsymbol{v}_{CC}))$ and a kernel $\boldsymbol{k}_{rgb2yuv} \in \mathbb{R}^{3,3}$. Define

$$h_1 : \mathbb{R}^{3,3} \to \mathbb{R}^{H,W},$$

where

$$h_1(\boldsymbol{\theta}_5)_{h,w} := (\Phi_{YUV}(\boldsymbol{v}_{CC}) \star \boldsymbol{\theta}_5)_{h,w}$$

and set

$$\Phi^{\theta^{SH}} : \mathbb{R}^{3,3} \to \mathbb{R}^{3,H,W}, \boldsymbol{\theta}_5 \mapsto \Phi_{YUV}^{-1} \circ h_1(\boldsymbol{\theta}_5) \circ \Phi_{YUV}(\boldsymbol{v}_{CC}).$$

Denoising (DN) We restrict to the case where the Gaussian denoising method is applied. Applying the Gaussian filter from scipy.ndimage with $\sigma = 0.5$ is equivalent to a convolution of $\Phi_{YUV}(\boldsymbol{v}_{SH})$ with a specific $\boldsymbol{k}_{gauss} \in \mathbb{R}^{5,5}$. Fix $\boldsymbol{v}_{SH} \in \mathbb{R}$ and define

$$h_2 : \mathbb{R}^{5,5} \to \mathbb{R}^{H,W},$$

where

$$h_2(\boldsymbol{\theta}_6)_{h,w} := (\Phi_{YUV}(\boldsymbol{v}_{SH}) \star \boldsymbol{\theta}_6)_{h,w}$$

and set

$$\Phi^{\theta^{DN}} : \mathbb{R}^{5,5} \to \mathbb{R}^{3,H,W}, \boldsymbol{\theta}_6 \mapsto \Phi_{YUV}^{-1} \circ h_2(\boldsymbol{\theta}_6) \circ \Phi_{YUV}(\boldsymbol{v}_{SH}).$$

Gamma correction (GC) The clipping is the only non-differentiable operation that is used. Assuming $\mathbb{P}(\boldsymbol{v}_{DN} \in \{0,1\}) = 0$ ensures an a.e.-differentiable map

$$\Phi^{\theta^{GC}} : \mathbb{R} \to [0,1], \boldsymbol{\theta}_7 \mapsto (v_{CP})^{\boldsymbol{\theta}_7}.$$

Using all the above steps, define the parameter space

$$\Theta := \mathbb{R}^4 \times \left(\mathbb{R}^{3,3} \times \mathbb{R}^{3,3}\right) \times \mathbb{R}^3 \times \mathbb{R}^{3,3} \times \mathbb{R}^{3,3} \times \mathbb{R}^{5,5} \times \mathbb{R}$$

and the processing model

$$\Phi_{Proc} : [0,1]^{C,H,W} \times \Theta \to [0,1]^{C,H,W}$$

with

$$(\boldsymbol{x}, \boldsymbol{\theta}_1, \ldots, \boldsymbol{\theta}_7) \mapsto \left(\Phi^{\theta^{GC}}(\theta_7) \circ \Phi^{\theta^{DN}}(\theta_6) \circ \Phi^{\theta^{SH}}(\theta_5) \circ \Phi^{\theta^{CC}}(\theta_4) \circ \Phi^{\theta^{WB}}(\theta_3) \circ \Phi^{\theta^{DM}}(\theta_2) \circ \Phi^{\theta^{BL}}(\theta_1)\right)(\boldsymbol{x}). \quad (4)$$

Finally, for fixed $\boldsymbol{x}$ we define by $\Phi^{\theta}_{Proc} := \Phi_{Proc}(\boldsymbol{x}, \cdot)$ the desired a.e.-differentiable model. For readability we further say that Φ^{θ}_{Proc} is differentiable, noting that this holds only $\mathbb{P} - a.e.$. We call $\Phi^{\theta_0}_{Proc}$ the *parametrized pipeline*.