BBM 444 – Programming Assignment X: Single-Image HDR Reconstruction

Due date: Monday, 11/05/2026, 11:59 PM.

High Dynamic Range (HDR) imaging captures the wide luminance range of real-world scenes that standard Low Dynamic Range (LDR) cameras compress and discard. In this assignment, you will build a deep learning pipeline that reconstructs an HDR image from a **single LDR input image**, conditioned on camera exposure metadata.

This is an ill-posed problem: information lost in saturated or underexposed regions must be *hallucinated* from context. Your model will learn to do this by leveraging both the pixel content of the LDR image and the physical imaging parameters encoded in the EXIF data.

Deliverables:

(i) Data preparation & processing pipeline.

(ii) Model architecture implementation (U-Net with EXIF conditioning).

(iii) Training with ablation study over conditioning strategies.

(iv) Quantitative evaluation (PSNR-T, SSIM).

(v) Qualitative visualisations and failure case analysis.

(vi) Discussion and conclusion.

Background & Motivation

Digital cameras compress the wide luminance range of real-world scenes into a limited set of intensity levels, resulting in a standard low dynamic range (LDR) image. When a scene contains both very bright and very dark regions, highlights clip to white and shadows crush to black — information that is permanently lost and cannot be recovered by simply brightening or darkening the image.

Single-image HDR reconstruction is particularly challenging because saturated and underexposed regions carry no recoverable pixel information; the network must instead rely on context and learned scene priors to hallucinate plausible content. This is in contrast to multi-exposure fusion, which sidesteps the problem by merging a bracketed stack where each image exposes a different portion of the dynamic range.

A key observation is that not all LDR images lose the same information. The fraction of the dynamic range that is clipped — and *where* it is clipped — depends directly on the exposure settings at capture time. The exposure value (EV) encodes exactly this context, and feeding it explicitly into the reconstruction network allows the model to adapt its hallucination strategy to the specific exposure regime of the input image.

In this assignment, you will learn and experiment with deep learning-based single-image HDR reconstruction. Starting from a single LDR photograph and its associated exposure metadata, you will design and train a modified U-Net model conditioned on EV parameters to predict the missing dynamic range.

Dataset

You will use the dataset from Kalantari et al. (2017), *"Robust Patch-Based HDR Reconstruction of Dynamic Scenes"* [1], which contains **74** and **15** scenes. Each scene provides three LDR exposures (short, medium, long) in `.tif` format with exposure values $\{0, 2, 4\}$ stored in `exposure.txt`, and a ground-truth HDR image fused from all three.

Your setup: Use **all three LDR exposures** as separate training samples, each paired with its EV value and the same ground-truth HDR target. This triples the effective training set to **222 training samples** and ensures that the EV varies across inputs, making the conditioning meaningful.

Download link: `https://cseweb.ucsd.edu/~viscomp/projects/SIG17HDR/`

Figure 1: Sample images under different exposure values from the Kalantari et al. dataset. From left to right: short exposure (EV = 0), medium exposure (EV = 2), long exposure (EV = 4), and the fused ground-truth HDR (tone-mapped for display).

Data Preparation & Processing (15 pts)

Input Preparation

(a) **Load the LDR images** for each scene as a `float32` tensor, normalised to $[0, 1]$.

(b) **Extract exposure metadata.** Read the EV value for each image from the corresponding `exposure.txt` file. Normalise EV values to zero mean and unit variance using training-set statistics only.

(c) **Compute the saturation mask.** For each pixel, compute a binary mask indicating clipped regions:

$$M = \mathbf{1}\big[\max(I_r, I_g, I_b) > \tau\big], \qquad \tau = 0.95.$$

You may experiment with a soft version: $M = \exp\big(-\alpha(1 - \max(I_r, I_g, I_b))\big)$ for $\alpha > 0$. The saturation mask is concatenated to the RGB image as a fourth channel, giving a **4-channel input tensor**.

(d) **Resize and crop** all images to 256×256 using centre-cropping. Apply random horizontal/vertical flips and 90ř rotations as training augmentation.

Label Preparation

(a) Load the ground-truth HDR image (provided as a `.hdr` file).

(b) **Tone-map for training:** Convert the HDR target to the μ-law log domain:

$$H_\mu = \frac{\log(1 + \mu \cdot H)}{\log(1 + \mu)}, \qquad \mu = 5000.$$

Training in this compressed space is important since HDR values span several orders of magnitude. The result lies in $[0, 1]$.

(c) Resize and crop to 256×256 matching the input pipeline.

Data Split

Each of the 74 training scenes contributes 3 samples (one per exposure), giving 222 training samples in total. Split these into **90% train** and **10% validation** at the *scene level* (not sample level) to avoid data leakage. The 15 test scenes are held out. Compute mean and standard deviation from the training split only for EV normalisation.

Visualisation (required): Plot three sample pairs showing the LDR input, the saturation mask, and the tone-mapped HDR target side by side.

Model Architecture & Training Setup (40 pts)

Baseline Architecture

Implement a **U-Net** encoder-decoder with the following structure:

- **Input:** $4 \times H \times W$ tensor — RGB (3) + saturation mask (1).
- **Encoder:** repeated 3×3 conv + ReLU + 2×2 max-pool, doubling channels at each level. Starting channels: $F = 8$ or more.
- **Bottleneck:** $16F$ channels.
- **Decoder:** bilinear upsampling, skip-concat, 3×3 conv + ReLU, halving channels at each level.
- **Output:** 1×1 conv $\rightarrow$ 3 channels, followed by sigmoid.

EXIF Conditioning

You will implement and compare conditioning strategies as ablations (see Experiments section):

(A) **No conditioning (baseline):** Plain U-Net with 4-channel input and no EV information. This is your M1 baseline.

(B) **Saturation Mask only:** Already included in the 4-channel input. Gives the network spatial awareness of lost information regions with no additional parameters.

(C) **FiLM Conditioning:** Use Feature-wise Linear Modulation [2] to condition all decoder feature maps on the EV embedding:

$$\mathbf{F}' = \gamma(\mathbf{e}) \odot \mathbf{F} + \beta(\mathbf{e}),$$

where $\gamma(\cdot)$ and $\beta(\cdot)$ are learned linear projections of the EV embedding $\mathbf{e}$, applied channel-wise to each spatial feature map $\mathbf{F}$. The embedding $\mathbf{e}$ is obtained by passing the normalised EV scalar through a small MLP. Separate γ and β projection layers are used at each decoder level. As shown in Figure 2, the bottleneck feature map may optionally be flattened to a vector of dimension N before being passed to the decoder; the value of N is a design choice left to you :).

See Figure 2 for an illustration of the model architecture. The base channel count F should be 8 or more.

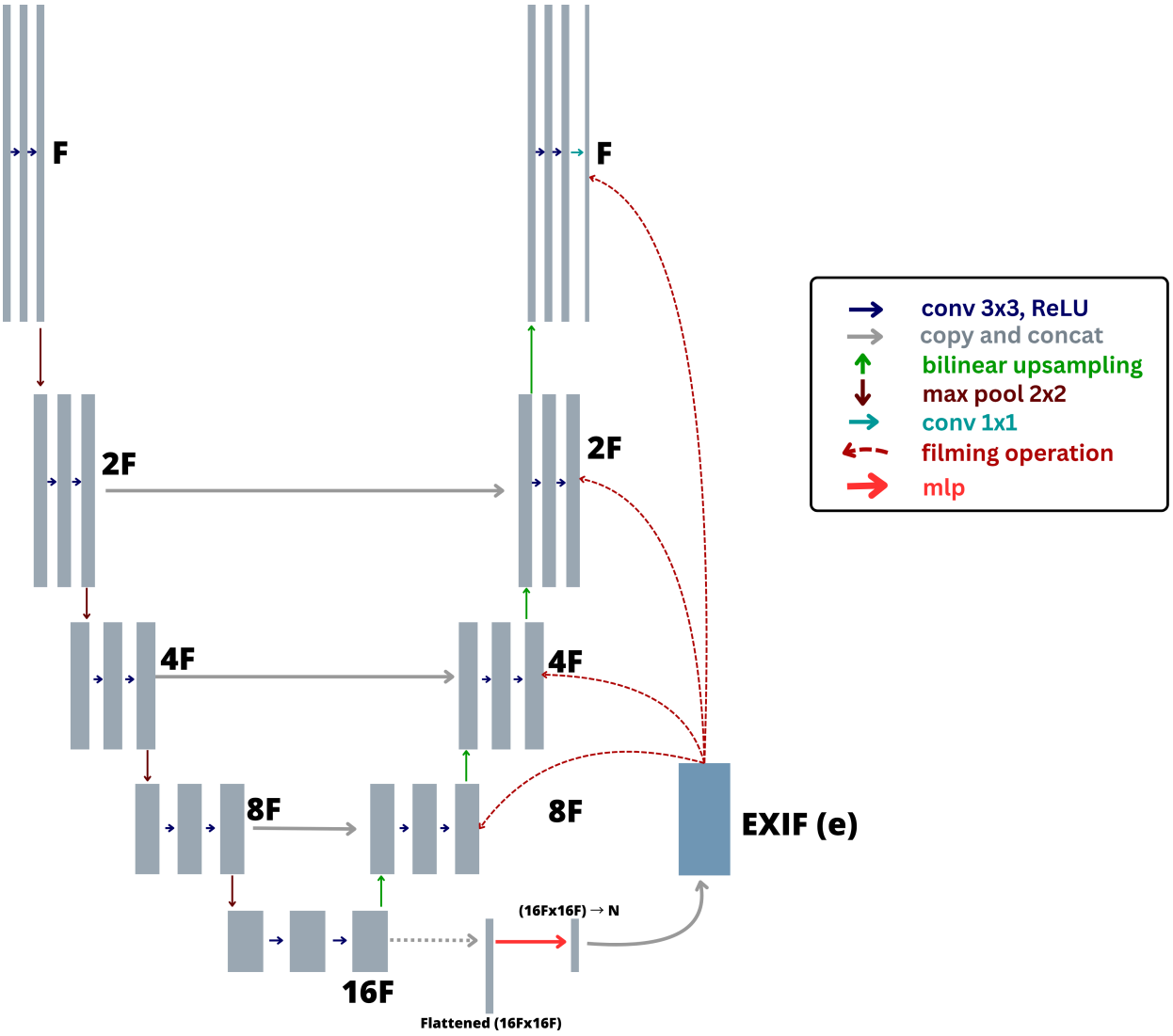

Figure 2: Modified U-Net architecture for single-image LDR → HDR reconstruction with FiLM-based EXIF conditioning. The encoder (left) applies repeated `conv 3×3 + ReLU` blocks followed by `max-pool 2×2` downsampling, doubling the number of feature channels at each level. The decoder (right) mirrors this structure using bilinear upsampling, with skip connections (gray arrows) transferring encoder feature maps to the corresponding decoder level via `conv 3×3 + ReLU`. F denotes the base channel count; channels grow as $F \to 2F \to 4F \to 8F \to 16F$ toward the bottleneck. The blue block represents the EXIF embedding vector $\mathbf{e}$, obtained by passing the normalised EV scalar through a small MLP. Red dashed arrows indicate FiLM conditioning operations, where $\mathbf{e}$ is projected to per-channel scale γ and shift β at each decoder level, modulating the feature maps as $\gamma(\mathbf{e}) \odot \mathbf{F} + \beta(\mathbf{e})$.

Loss Function

Train with a combined reconstruction loss:

$$\mathcal{L} = \lambda_1 \, \|\hat{H} - H\|_1 + \lambda_2 (1 - \mathrm{SSIM}(\hat{H}, H)),$$

with $\lambda_1 = 0.8$, $\lambda_2 = 0.2$. All losses are computed in the μ-law tone-mapped domain.

Training Recipe

- **Optimizer:** Adam, cosine annealing over 100 epochs.
- **Augmentation:** random flips, 90ř rotations.
- **Best model:** select checkpoint with lowest validation loss.

Pseudo-code:

```
best_val_loss = inf
for epoch in 1..100:
    for (ldr, mask, ev, hdr_target) in train_loader:
        input = concat(ldr, mask)          # [B, 4, H, W]
        pred  = model(input, ev)           # [B, 3, H, W]
        loss  = 0.8*L1(pred, hdr_target)
              + 0.2*(1 - SSIM(pred, hdr_target))
        backprop & step
    val_loss = evaluate(val_loader)
    if val_loss < best_val_loss:
        best_val_loss = val_loss
        save_checkpoint(model)
```

Experiments (20 pts)

Conduct a systematic ablation study by training the following model variants. Keep all hyperparameters identical across runs; vary only the conditioning mechanism.

Model	Saturation Mask	FiLM (EV)
M1: Baseline	–	–
M2: + Mask	✓	–
M3: + FiLM	–	✓
M4: + Mask + FiLM	✓	✓

M1 is the base model with no saturation mask or FiLM conditioning.

For each trained model, report PSNR-T and SSIM on the test split. Summarise results in a table and plot training/validation loss curves for each model. Discuss the effect of each component and explain why certain models succeed or fail.

Results & Visualisation (10 pts)

Quantitative Metrics

Evaluate all trained models on the **test split**. All predictions must be tone-mapped before metric computation.

- **PSNR-T** (tone-mapped PSNR): compute PSNR on μ-law tone-mapped outputs:

$$\text{PSNR-T}(\hat{H}, H) = 10 \log_{10}\left(\frac{1}{\text{MSE}(\hat{H}_\mu, H_\mu)}\right).$$

- **SSIM:** structural similarity on tone-mapped outputs.

```
from torchmetrics.functional import (
    structural_similarity_index_measure as ssim)
ssim_val = ssim(pred_tmo, gt_tmo, data_range=1.0)
```

Report average metrics over all 15 test scenes in a summary table.

Qualitative Visualisation

Select **three representative test scenes** and display the following for your best model (M4):

1. LDR input image (gamma-corrected for display).
2. Saturation mask overlaid on the LDR input.
3. Predicted HDR (tone-mapped with μ-law).
4. Ground-truth HDR (tone-mapped).
5. Per-pixel absolute error heat-map on the tone-mapped outputs.

Additionally, select one scene with a **heavily saturated region** and one with **deep shadows**. Show a close-up crop comparing M1 (baseline) vs. M4 (Mask + FiLM) to demonstrate the benefit of conditioning.

Discussion (15 pts)

Write a concise analysis addressing the following points:

- **Ablation findings:** What does each additional component (mask, FiLM conditioning) contribute? Support with quantitative evidence from your table. Discuss why one or more models you tried failed or succeeded.
- **Failure modes:** Identify at least two situations where your best model fails (e.g. large uniform saturated regions, specular highlights, extreme underexposure). Show visual examples.
- **FiLM vs. no conditioning:** Discuss the qualitative and quantitative difference. Why might FiLM be better at leveraging exposure information than simply concatenating a scalar?
- **Limitations of μ-law training:** What are the drawbacks of training in tone-mapped space? How might training in linear HDR space change results?
- **Future directions:** Suggest two concrete architectural or training improvements and reason about their expected benefit.

Conclusion

Write a brief conclusion summarising:

- What your model achieves and how it compares to the baseline.
- The most important design decision from your ablation study.
- One open problem in single-image HDR reconstruction you found during this assignment.

Hints & Troubleshooting

(1) **Starter Notebook:** The official assignment starter notebook is available at: `https://drive.google.com/file/d/155PY9l7L5-X1RqgSJjxFNyQUyE4NXD6P/view?usp=sharing`.

(2) **Log-domain instability:** If HDR values are near zero, $\log(0)$ causes `nan`. Always add a small epsilon: $\log(\epsilon + H)$.

(3) **FiLM implementation:** Broadcast $\gamma(\mathbf{e})$ and $\beta(\mathbf{e})$ along the spatial dimensions. Each should have shape $(B, C, 1, 1)$.

(4) **Scene-level split:** Make sure train/val split is done at the scene level, not the sample level. Mixing exposures from the same scene across train and val constitutes data leakage.

(5) **OOM:** Reduce batch size or enable mixed precision (`torch.cuda.amp`).

(6) **Limited hardware:** Use Google Colab (T4 GPU) and save checkpoints to Google Drive regularly. Debug your pipeline in CPU mode first.

Submission Checklist

`assignment_hdr.ipynb`	Code, training logs, plots, tables, written answers
`weights/best_model.pth`	Best model weights
`dataset/`	Leave empty

Rubric (100 pts)

- **Data Preparation & Processing** – 15 pts
- **Model Architecture & Training** – 40 pts
- **Experiments (ablation)** – 20 pts
- **Results & Visualisation** – 10 pts
- **Discussion & Conclusion** – 15 pts

References

[1] N. K. Kalantari and R. Ramamoorthi. Deep High Dynamic Range Imaging of Dynamic Scenes. *ACM Transactions on Graphics (SIGGRAPH)*, 36(4), 2017.

[2] E. Perez, F. Strub, H. de Vries, V. Dumoulin, and A. Courville. FiLM: Visual Reasoning with a General Conditioning Layer. In *AAAI*, 2018.

[3] W. Kim, G. Kim, J. Lee, et al. ParamISP: Learned Forward and Inverse ISPs Using Camera Parameters. In *CVPR*, 2024.

[4] G. Eilertsen, J. Kronander, G. Denes, R. K. Mantiuk, and J. Unger. HDR Image Reconstruction from a Single Exposure Using Deep CNNs. *ACM Transactions on Graphics*, 36(6), 2017.

[5] S. W. Zamir, A. Arora, S. Khan, M. Hayat, F. S. Khan, M. Yang, and L. Shao. CycleISP: Real Image Restoration via Improved Data Synthesis. In *CVPR*, 2020.