Adaptive Stain Normalization for Cross-Domain Medical Histology

Tianyue Xu, Yanlin Wu, Abhai K. Tripathi, Matthew M. Ippolito, Benjamin D. Haeffele ✉
haeffele@upenn.edu

Background

Digital pathology is critical for disease diagnosis, but stain color variability creates major domain shifts.

Causes:
- Dye chemistry and reaction time differences
- Slide/sample preparation variability
- Scanner and imaging hardware differences

Impact:
- Pathologists adapt easily → AI models fail
- Reduced generalization → limits clinical use

Limitations of existing methods:
- Template-based (Reinhard, Macenko, Vahadane) – sensitive & unstable, template-dependent
- GAN-based– hallucinate structures, unreliable
- Generic DL approaches – not physics-informed

Our Method: BeerLaNet

Core Idea: BeerLaNet embeds the physics of histological staining, Beer–Lambert law, into a trainable network by unrolling matrix factorization, adaptively separating stain spectra and density without a template.

$\mathbf{x_0} \in \mathbb{R}^c$: Background color. $\mathbf{X} \in \mathbb{R}^{c \times p}$: Observed image.
$\mathbf{S} \in \mathbb{R}^{c \times r}$: Color spectra. $\mathbf{D} \in \mathbb{R}^{p \times r}$: Optical density.
$r \in \mathbb{R}$: Rank of factorization (number of stains).
$\gamma, \lambda \in \mathbb{R}$: Regularization parameters.

Beer–Lambert physics: $\bar{\mathbf{X}} = (\bar{x}_0 \mathbf{1}^\top) \odot e^{-\mathbf{S}\mathbf{D}^\top}$

$$\min_{x_0,\ \mathbf{S},\mathbf{D}} \frac{1}{2} \left|\left|x_0 \mathbf{1}^\top - \mathbf{X} - \mathbf{S}\mathbf{D}^\top\right|\right|_F^2 + \lambda \sum_{i=1}^{r} \left|\left|s_i\right|\right|_2 \left(\gamma \left|\left|d_i\right|\right|_1 + \left|\left|d_i\right|\right|_2\right), s.t. \mathbf{S}, \mathbf{D} \geq \mathbf{0}$$

Key Contributions:
- Adaptive stain separation → generalizes beyond H&E
- Algorithmic unrolling of NMF → interpretable, trainable
- Physics-informed (Beer–Lambert law) → models stain absorption
- Plug-and-play → works with YOLO, ResNet, etc, template-free

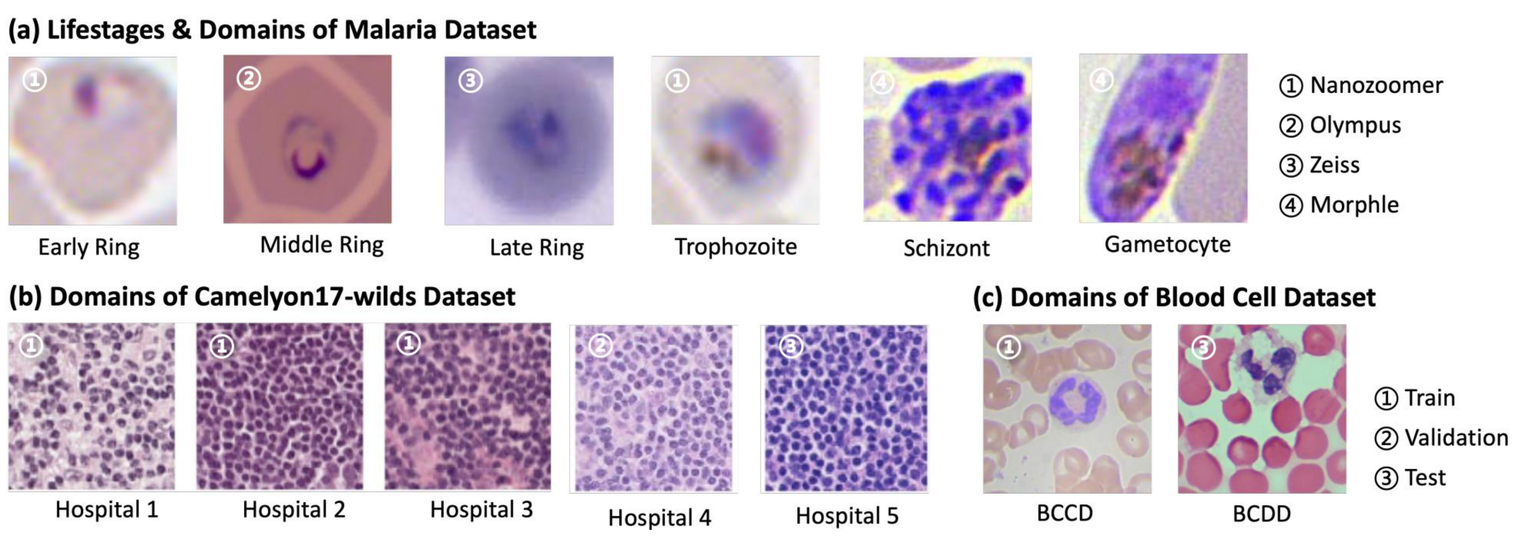
Dataset

Malaria blood smears (Detection):
- MGG-stained thin smears → RBC, WBC, platelets, parasites

Malaria parasite classification:
- Giemsa-stained single-cell images
- 2 different microscopes, 6 classes

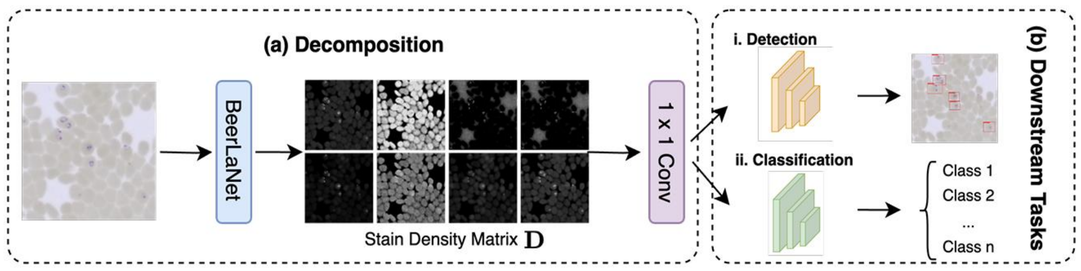
Camelyon17-WILDS:
- H&E-stained tumor/normal patches
- 5 hospitals

Whole blood cell detection:
RBC, WBC, platelets



Pipeline

Input pathology image → Unrolled layers (stain decomposition) → Stain-invariant representation → Backbone network



Algorithm (BeerLaNet)

Input: Image $\mathbf{X}$, iterations $\mathbf{K}$, stains r
Learnable parameters: $\gamma, \lambda, \mathbf{S}_{init}$
Output: Stain density map $\mathbf{D}$

1. Initialize $\mathbf{S} = \mathbf{S}_{init}$, $\mathbf{D}$ = 0, x_0 = 0
2. For $k = 1, ..., K$:
- Update background x_0
- Update $\mathbf{D}$ by proximal gradient descent
  - Enforces sparsity
  - Applies non-negativity constraint
- Update $\mathbf{S}$ by proximal gradient descent
  - Applies non-negativity constraint

3. Return $\mathbf{D}$

Results

Table 1. Comparison of Stain Normalization Techniques. The best and the second-best results are **boldfaced** or starred (*), respectively. (C17 denotes Camelyon17-WILDS)

Task	Detection (YOLOv8)					Classification (ResNet18)				
Dataset	Malaria		Whole Blood Cells			Malaria		$C17_{test}$	$C17_{val}$	
Metrics (%)	mAP_{50}	mAP_{50-95}	mAP_{50}	mAP_{50-95}	APU	Acc	RAcc	Acc	Acc	APU
Baseline	91.03	52.00	65.20	36.70	18.10	21.32	45.59	85.21	83.38	31.70
Reinhard	90.17	51.47	79.53	44.03	11.17	29.34	62.30	94.55	91.36*	18.36
Macenko	72.03	39.27	65.43	35.27	29.27	29.59	73.54	**95.92**	85.77	16.27
Vahadane	92.10*	55.87*	57.57	31.30	20.76	38.81*	81.04*	95.85*	86.43	9.31*
StainGAN	81.67	37.70	**89.60**	**53.97**	12.02	31.46	70.94	94.28	90.77	15.12
LStainNorm	91.80	53.67	85.83	50.00	5.25*	21.17	61.82	93.23	**92.56**	22.72
BeerLaNet	**95.07**	**57.10**	86.80*	51.33*	**2.00**	**48.66**	**90.33**	91.36	90.09	**1.86**

Consistency & Generalizability

- **APU (Average Percent Underperformance):** measures how far a method is from the best in each task, then averages across tasks.
- BeerLaNet achieves the **lowest APU**, meaning its performance is consistently close to the best across all benchmarks.

Acknowledgments & Codebase

Supported by:
- NIH NIAID grant R21AI16936
- Bloomberg Philanthropies through the Johns Hopkins Malaria Research Institute and Parasitology Core Laboratory
- Johns Hopkins Institute for Data-Intensive Engineering and Science

Github.com/xutianyue/BeerLaNet