

# Benchmark Study of Lensless Imaging Reconstruction Algorithms Towards Real-Time Optimization

ZIJIE CAI<sup>1</sup>, ANH NHU<sup>1</sup>, EHAAB BASIL<sup>1, 2</sup> Department of Computer Science, University of Maryland, College Park, MD 20742

Lensless cameras offer a promising alternative to conventional cameras for real-life applications requiring device miniaturization, such as medical imaging and surveillance. However, due to the modification, reconstruction algorithms are required to recover the sensor measurements, often with many computational challenges. In this paper, we benchmark various lensless imaging reconstruction algorithms on the DiffuserCam dataset, which includes 25,000 image pairs of lensed and lensless measurements. These algorithms include variants of two common optimization algorithms: Gradient Descent (GD) and Alternating Direction Method of Multipliers (ADMM), from iterative computational imaging approaches to deep learning approaches. By exploring the trade-offs between the speed and quality of these algorithms, we fine-tune a set of optimized hyperparameters for the evaluated algorithms with the goal of enabling real-time on-device reconstruction with minimal computational resources. In our study, we find that Le-ADMM with U-Nets achieves the best overall performance quantitatively and qualitatively while minimizing the computational resources needed. Our results provide useful insights for balancing computational efficiency and reconstruction fidelity for real-time lensless imaging applications.

CCS Concepts: • Computing methodologies → Image reconstruction.

Additional Key Words and Phrases: Lensless imaging, computational imaging, image reconstruction, deep learning, real-time optimization

## ACM Reference Format:

Zijie Cai<sup>1</sup>, Anh Nhu<sup>1</sup>, Ehaab Basil<sup>1</sup>. 2025. Benchmark Study of Lensless Imaging Reconstruction Algorithms Towards Real-Time Optimization. *ACM Trans. Graph.* 1, 1 (February 2025), 7 pages. <https://doi.org/10.1145/nnnnnnnn>

## 1 INTRODUCTION

Lensless cameras provide an exciting alternative to conventional cameras by replacing traditional bulky lenses with more compact light-modulating materials and reconstruction algorithms. This enables the development of compact and cost-effective imaging systems for applications such as tiny robotics, medical imaging, and surveillance, where size, weight, and cost constraints are critical. Instead of capturing direct optical images, lensless imaging systems record encoded light patterns spread across the pixels onto the sensor, which must then be reconstructed into focused images.

Lensless imaging reconstruction algorithms often pose several challenges due to the complexity of encoded measurements and the

---

Author's address: Zijie Cai<sup>1</sup>, Anh Nhu<sup>1</sup>, Ehaab Basil<sup>1</sup>, zai28@umd.edu, anhu@umd.edu, ehaab@umd.edu, <sup>1</sup> Department of Computer Science, University of Maryland, College Park, MD 20742.

---

Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. Copyrights for components of this work owned by others than the author(s) must be honored. Abstracting with credit is permitted. To copy otherwise, or republish, to post on servers or to redistribute to lists, requires prior specific permission and/or a fee. Request permissions from [permissions@acm.org](mailto:permissions@acm.org).

© 2025 Copyright held by the owner/author(s). Publication rights licensed to ACM. ACM 0730-0301/2025/2-ART  
<https://doi.org/10.1145/nnnnnnnn.nnnnnnn>

computational resources required for high-quality reconstruction. When designing reconstruction algorithms, it is critical to balance reconstruction quality and computational efficiency—a trade-off that varies depending on the algorithm used. Traditional optimization methods, such as Gradient Descent (GD) and Alternating Direction Method of Multipliers (ADMM), offer reliable solutions but are often computationally intensive. Deep learning-based approaches provide an alternative by introducing learned priors, which can enhance speed and accuracy, though they demand substantial training and computational resources.

This paper benchmarks popular reconstruction algorithms on the DiffuserCam dataset, which includes 25,000 paired lensless and lensed measurements. We evaluate the performance of iterative optimization methods alongside hybrid techniques utilizing deep learning priors, using metrics such as Peak Signal-to-Noise Ratio (PSNR), Learned Perceptual Image Patch Similarity (LPIPS), and processing time per file (in seconds).

Our analysis focuses on optimizing algorithmic performance to achieve acceptable real-time on-device reconstruction with minimal computational resources. The findings provide useful insights into the trade-offs between reconstruction quality and efficiency, advancing the development of practical lensless imaging systems for real-world applications.

## 2 RELATED WORK

Lensless cameras have become a popular ongoing research field with many recent advancements due to their potential for device miniaturization. Early work by Antipa et al. [Antipa et al. 2018] introduced the DiffuserCam, a lensless imaging system that uses a diffuser mask to encode light patterns for computational reconstruction. In this case, the authors used scotch tape, which produced a swimming pool surface-like pattern for the forward model of the physics. This work demonstrated the feasibility of single-exposure lensless imaging for 3D scene capture, laying the foundation for subsequent advances in both hardware and reconstruction algorithms.

Traditional reconstruction methods for lensless imaging often rely on iterative optimization techniques. Methods such as Gradient Descent (GD) and Alternating Direction Method of Multipliers (ADMM) have been widely studied due to their mathematical simplicity and stability, as they do not require training data. However, these methods are computationally intensive and struggle to achieve the speed required for real-time applications. They may also introduce unwanted artifacts due to model mismatches or inaccuracies in the physical system.

More recent research has focused on incorporating deep learning into reconstruction pipelines to overcome these limitations. Khan et al. [Khan et al. 2020] proposed FlatNet, which leverages learned priors to improve reconstruction quality and achieve photorealistic

results. Similarly, Monakhova et al. [Monakhova et al. 2019] demonstrated that neural networks can significantly enhance the performance of traditional optimization algorithms, such as Le-ADMM, achieving a more balanced trade-off between traditional methods and deep learning-based approaches by incorporating pre-trained models to guide the reconstruction process.

In addition to advancements in reconstruction algorithms, light-weight hardware platforms for lensless cameras have also been developed. Bezzam et al. [Bezzam et al. 2023] introduced LenslessPi-Cam, a hardware-software platform based on the Raspberry Pi, which provides researchers with an accessible solution for real-time computational imaging research. It offers tools to build, evaluate, and test lensless cameras, as well as simulate and fabricate light-modulating mask patterns. The platform also includes PyTorch-accelerated reconstruction algorithms for faster processing times. These advancements in both hardware and algorithms highlight the growing potential for lensless imaging in practical applications.

While many of these works provide promising directions for improving reconstruction quality or designing new hardware for better efficiency, achieving real-time, on-device reconstruction remains a challenge. Our study builds on previous work by benchmarking existing popular reconstruction algorithms for lensless imaging on the DiffuserCam dataset. We study the trade-offs between computational efficiency and reconstruction fidelity with the objective of optimizing a pipeline on the Raspberry Pi for real-time, on-device reconstruction, thereby opening up possibilities for real-world applications.

### 3 METHODOLOGY

We benchmark reconstruction algorithms across multiple dimensions, including reconstruction quality, processing time, and suitability for real-time applications.

#### 3.1 Dataset

The DiffuserCam dataset [Antipa et al. 2018] is a dataset for lensless imaging research collected with a lensless diffuser camera. The dataset consists of 25,000 pairs of sensor measurements from lensless camera and ground truth images. Traditionally, the camera uses stack of lenses to map each point in the scene to a particular sensor pixel. In contrast, lensless camera like DiffuserCam maps each point in the scene to different pixels in some particular patterns, which can be described and visualized by a Point Spread Function (PSF). The PSF of the DiffuserCam camera can be shown in Figure 1.

Since a point in the scene is mapped to many sensor pixels, as shown in Figure 1, the resulting sensor measurements of a lensless camera do not show the true image in the scene. Some visualizations of raw DiffuserCam’s measurements and true images can be shown in Figure 2. Therefore, we need a reconstruction algorithm to accurately and efficiently reconstruct the true image from raw sensor measurements. Specifically, a reconstruction algorithm takes the blurry sensor measurements and reconstructs the true image from such blurry measurements. The reconstruction pipeline diagram is illustrated in Figure 3. In our study, we utilize this dataset for training and evaluation of different reconstruction algorithms described in Section 3.2.

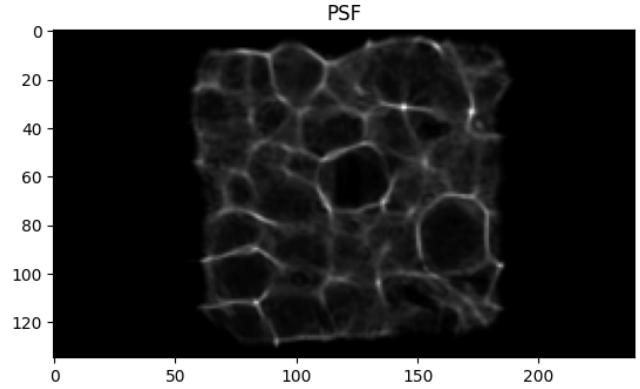


Fig. 1. Point Spread Function (PSF) of DiffuserCam describing how a point in the scene mapped to the sensor’s pixels. Unlike traditional cameras, the lensless cameras’ PSF patterns are complex and non-Gaussian.

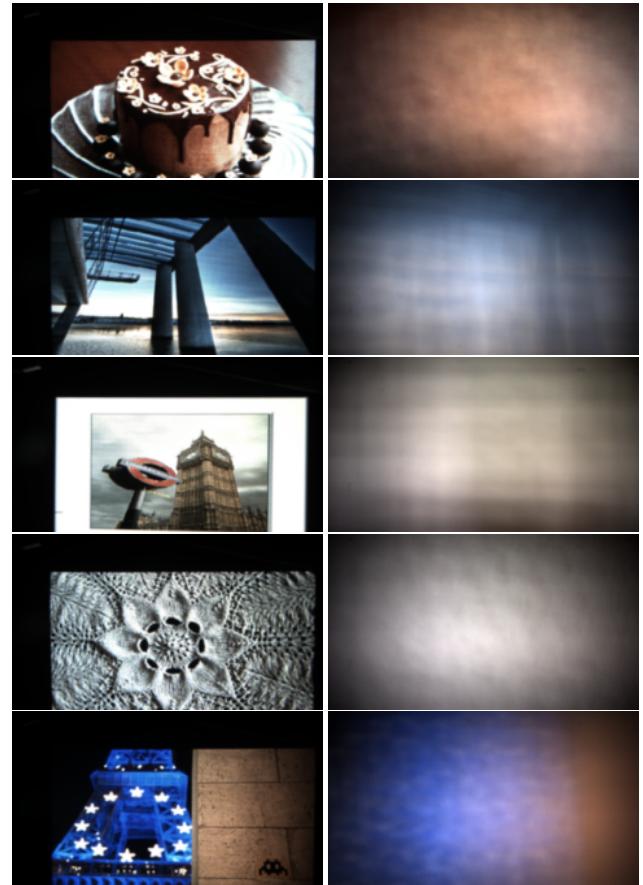


Fig. 2. Visualizations of True Images (left; targets) and lensless Camera’s Sensor Measurements (right; inputs) of different images. The lensless measurements are highly blurry without any important information, necessitating a reconstruction algorithm to recover the true image.

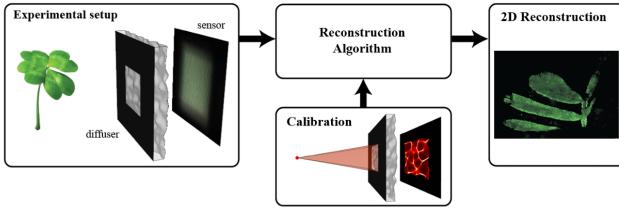


Fig. 3. Diagram of reconstruction algorithm from the raw lensless measurements. The diagram is taken from [Antipa et al. 2018].

### 3.2 Reconstruction Algorithms

We evaluate the following reconstruction methods:

- **Gradient Descent (GD):** Includes Vanilla GD, Nesterov GD, and FISTA for improved convergence.
- **ADMM Variants:** Standard ADMM, Plug-and-Play ADMM with deep priors (DRUNET), and Learned ADMM (Le-ADMM) unrolled as neural networks.

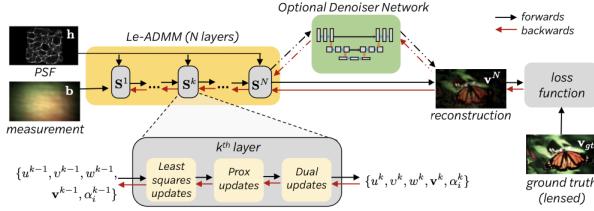


Fig. 4. Model-based reconstruction architecture, illustrating the iterative optimization process used in Gradient Descent and related algorithms [Antipa et al. 2018].

#### Color Notation:

**Red** is used to highlight learnable parameters or hyperparameters (e.g., learning rates, momentum coefficients, penalty terms).

**Blue** is used to indicate pre-trained modules or external components (e.g., denoisers like  $D(\cdot)$  or U-Net regularizers).

- (1) **Vanilla Gradient Descent (GD):** Gradient Descent (GD) iteratively minimizes the reconstruction loss:

$$L(x) = \frac{1}{2} \|b - Hx\|_2^2, \quad (1)$$

where  $b$  is the observed sensor measurement,  $H$  is the forward model, and  $x$  is the reconstructed image. The update rule for GD is:

$$x^{k+1} = x^k + \eta H^T (b - Hx^k), \quad (2)$$

where  $\eta$  is the learning rate.

- (2) **GD-FISTA (Fast Iterative Shrinkage-Thresholding Algorithm):** GD-FISTA accelerates GD by introducing a momentum term to improve convergence:

$$y^k = x^k + \frac{t_{k-1} - 1}{t_k} (x^k - x^{k-1}), \quad (3)$$

$$x^{k+1} = \text{prox}_{\lambda R}(y^k - \eta \nabla L(y^k)), \quad (4)$$

where  $t_k$  controls momentum, prox is the proximal operator, and  $R(x)$  is a regularizer.

- (3) **Nesterov Accelerated Gradient Descent (GD-Nesterov):** Nesterov GD improves convergence stability using a look-ahead step:

$$v^{k+1} = \beta v^k - \eta \nabla L(x^k + \beta v^k), \quad (5)$$

$$x^{k+1} = x^k + v^{k+1}, \quad (6)$$

where  $\beta$  is the momentum coefficient and  $v^k$  is the velocity term.

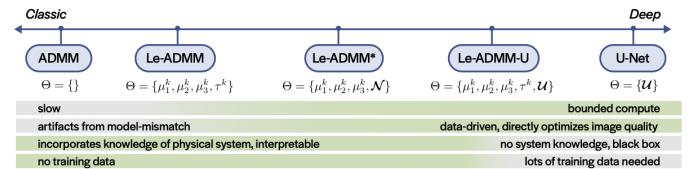


Fig. 5. Reconstruction methods on a scale from classical optimization to deep learning-based approaches [Antipa et al. 2018].

- (4) **Alternating Direction Method of Multipliers (ADMM):** ADMM solves the following optimization problem:

$$\min_{x, v} \frac{1}{2} \|b - Hv\|_2^2 + \lambda R(x), \quad \text{s.t. } v = x. \quad (7)$$

The iterative updates are:

$$x^{k+1} = \arg \min_x \lambda R(x) + \frac{\rho}{2} \|x - v^k + u^k\|_2^2, \quad (8)$$

$$v^{k+1} = \arg \min_v \frac{1}{2} \|b - Hv\|_2^2 + \frac{\rho}{2} \|x^{k+1} - v + u^k\|_2^2, \quad (9)$$

$$u^{k+1} = u^k + x^{k+1} - v^{k+1}. \quad (10)$$

- (5) **Plug-and-Play ADMM (PnP-ADMM):** PnP-ADMM replaces the regularization step with a pre-trained denoiser  $D(\cdot)$ :

$$x^{k+1} = D(v^k - u^k), \quad (11)$$

$$v^{k+1} = \arg \min_v \frac{1}{2} \|b - Hv\|_2^2 + \frac{\rho}{2} \|x^{k+1} - v + u^k\|_2^2, \quad (12)$$

$$u^{k+1} = u^k + x^{k+1} - v^{k+1}. \quad (13)$$

- (6) **Learned ADMM (Le-ADMM):** Le-ADMM unrolls ADMM into a neural network with learned parameters:

$$x^{k+1} = \text{prox}_{\lambda_k}(x^k - \eta_k \nabla L(x^k)), \quad (14)$$

$$v^{k+1} = \arg \min_v \frac{1}{2} \|b - Hv\|_2^2 + \frac{\rho_k}{2} \|x^{k+1} - v + u^k\|_2^2, \quad (15)$$

$$u^{k+1} = u^k + x^{k+1} - v^{k+1}. \quad (16)$$

Le-ADMM-U extends this by integrating a U-Net denoiser for further refinement.

### 3.3 Hyperparameter Tuning

Hyperparameter tuning plays a critical role in achieving optimal performance for each reconstruction method. In this section, we describe the hyperparameters we tuned, their effects on the algorithms, and the selected default values for our experiments.

**Color Notation:** Red is used to highlight the default hyperparameter values chosen for our experiments.

#### (1) Gradient Descent (GD):

- **Iterations** (`n_iter`): 100, 300, 500 Controls how long the optimization runs. A larger number of iterations allows the gradient updates to converge more accurately but increases computational time.

#### (2) Nesterov Accelerated Gradient Descent (GD-Nesterov):

- **Momentum** ( $\mu$ ): 0.5, 0.9, 0.99 The momentum parameter incorporates the history of gradients to accelerate convergence in directions with consistent gradients while dampening oscillations in noisy directions.
  - **Higher  $\mu$ :** Faster convergence but risks instability when combined with a large learning rate.
  - **Lower  $\mu$ :** Slower but more stable updates.

#### (3) GD-FISTA (Fast Iterative Shrinkage-Thresholding Algorithm):

- **Momentum** ( $t_k$ ): 1 The momentum parameter  $t_k$  accelerates convergence in FISTA.
  - **Smaller  $t_k$ :** Slower early acceleration but more stable, suitable for poorly conditioned problems.
  - **Larger  $t_k$ :** Faster early convergence but may lead to instability for non-smooth problems.

#### (4) Alternating Direction Method of Multipliers (ADMM):

- **Iterations** (`n_iter`): 5, 10, 20 Controls how many times ADMM alternates between primal and dual updates. Similar to GD, more iterations improve accuracy but increase runtime.
- **Step Sizes** ( $\mu_1, \mu_2, \mu_3$ ):  $\mu_1: 1e-7, 1e-6, 1e-5; \mu_2: 1e-5; \mu_3: 4e-5$  Step sizes balance data fidelity, regularization, and constraint enforcement.
  - **Small  $\mu$ :** Slower convergence but more stable updates.
  - **Large  $\mu$ :** Faster convergence but risks instability.
- **TV Regularization Strength** ( $\tau$ ): **1e-4**, 1e-3, 1e-2  $\tau$  controls the strength of Total Variation (TV) regularization for sparsity:
  - **Small  $\tau$ :** Weak regularization, retaining texture details but allowing noise.
  - **Large  $\tau$ :** Strong regularization, suppressing noise but potentially over-smoothing fine details.

#### (5) Plug-and-Play ADMM (PnP-ADMM):

- **Denoiser:** DRUNET (pre-trained) The primary hyperparameter for PnP-ADMM is the choice of the denoiser. We use DRUNET, a deep pre-trained denoiser, to enforce realistic image priors during the ADMM updates.

#### (6) Learned ADMM (Le-ADMM):

- **Unrolled Iterations:** 10 Le-ADMM employs unrolling, where each iteration corresponds to a fixed layer in a neural network. The number of unrolled iterations balances reconstruction accuracy and computational cost.
- **Regularizer Network:** U-Net (UNet-ADMM10-UNet) The regularizer  $N$  is a small U-Net consisting of a single encoding and decoding step. Increasing the U-Net size improves the reconstruction quality but requires more computational resources and training data.

## 4 EVALUATION RESULTS

### 4.1 Evaluation Metrics

To quantitatively compare the performance of different reconstruction models, we primarily focus on 3 evaluation metrics:

- (1) **PSNR:** Peak Signal-to-Noise Ratio
- (2) **LPIPS:** Learned Perceptual Image Patch Similarity
- (3) **Processing Time:** inference time in seconds per image

PSNR metric is formally defined as:  $PSNR = 10 \log_{10} \left( \frac{R^2}{MSE} \right)$ , where  $R = 255$  is the maximum pixel value possible in the image, and  $MSE$  is the mean-squared-error between the reconstructed image  $\hat{I}_{recon}$  and the ground-truth image  $I_{gt}$ :

$$MSE = \frac{1}{M \times N} \sum_{i=1}^M \sum_{j=1}^N (I_{gt}(i, j) - \hat{I}_{recon}(i, j))^2$$

On the other hand, the LPIPS metric measures the perceptual difference between two images by considering the feature extracted from a pre-trained CNN in latent space instead of direct raw pixel-wise differences. The formal definition of LPIPS is:

$$LPIPS = MSE(f_{CNN}(I_{gt}), f_{CNN}(\hat{I}_{recon}))$$

where  $f_{CNN}$  denotes the CNN's feature extractor, such as an AlexNet [Krizhevsky et al. 2012], ResNet [He et al. 2016] or VGG [Simonyan and Zisserman 2015] model. In our evaluation, the CNN feature extractor is pre-trained AlexNet, which is also the default option.

Finally, the processing time is computed by averaging the total time required for each model to reconstruct each image over 900 test images, measured in seconds per image.

### 4.2 Quantitative Results

In this section, we present the quantitative evaluation of different reconstruction algorithms in Table 1. All our experiments during the inference process were executed on *CPU-only machines without GPUs* to mimic the computational resources available in a camera. In the table, \* denotes the best performance across all models.

Experimental results show that the Le-ADMM model significantly outperforms all other methods on both evaluation metrics. Le-ADMM achieves the highest PSNR of 26.1, which is **nearly 2× higher** than PnP-ADMM's second-highest PSNR of 14.3, and the lowest LPIPS of 0.077, which is **more than 5× lower** than ADMM's second-best LPIPS of 0.405. Despite having significantly better performance over other methods, Le-ADMM also has efficient

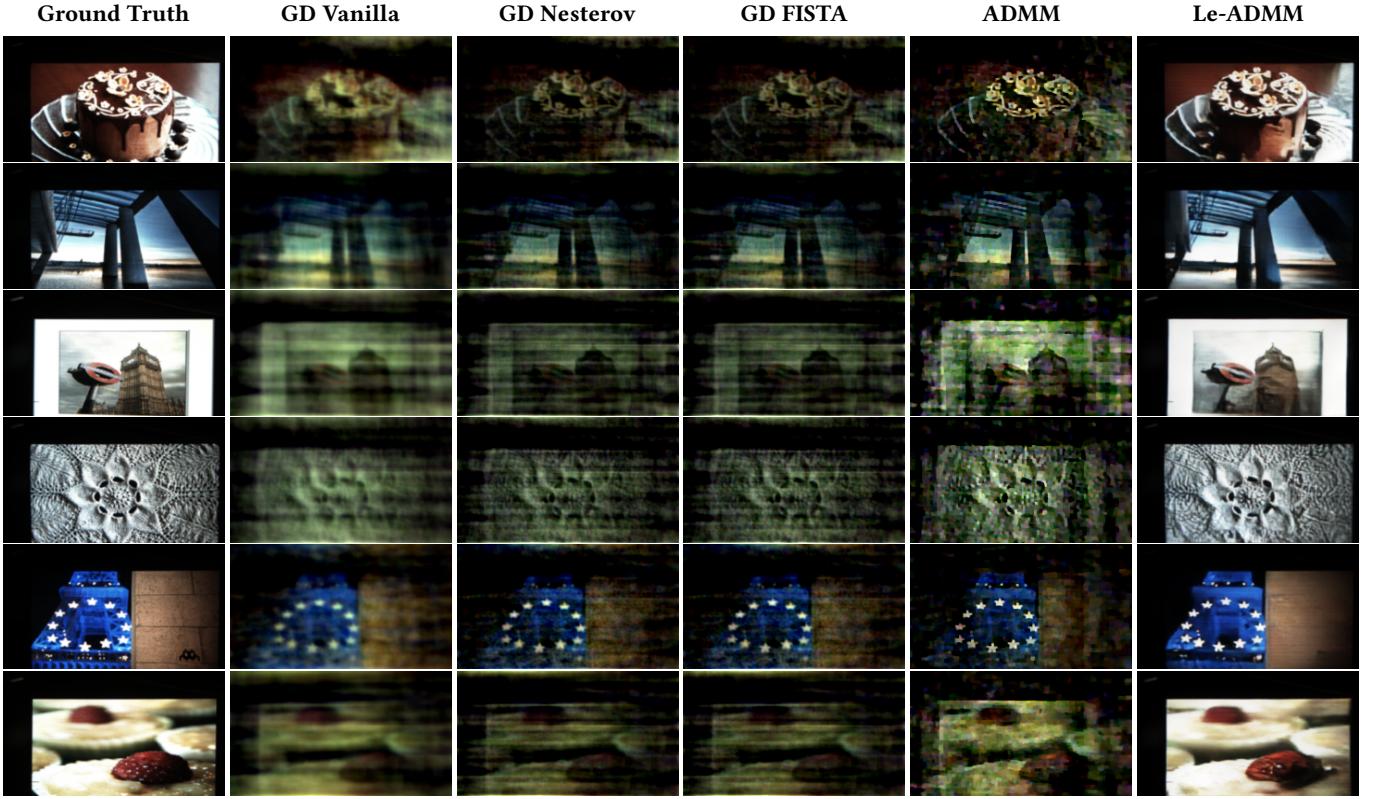


Fig. 6. Comparison of reconstructions between different reconstruction algorithms.

Table 1. Quantitative Results for Different Reconstruction Algorithms

Algorithm	PSNR ↑	LPIPS ↓	Time (sec/image)
Vanilla GD	13.2	0.499	2.079
Nesterov GD	12.2	0.457	1.408
GD-FISTA	12.0	0.451	0.904*
ADMM	13.3	0.405	1.05
PnP-ADMM	14.3	0.470	86.12
Le-ADMM	26.1*	0.077*	1.72

inference runtime, needing only 1.72 seconds/image compared to PnP-ADMM’s 86.12 seconds/image, which is **50× more computationally efficient**.

#### 4.3 Qualitative Results

To provide visual comparisons of the reconstruction quality between different reconstruction algorithms, we conducted qualitative comparisons between different models in Figure 6. We observe that most algorithms (Vanilla GD, GD Nesterov, GD FISTA, ADMM, PnP-ADMM) successfully manage to reconstruct the overall structures and high-level shapes in the ground-truth images but fail to produce the high-frequency details (blurry small details) and true color space distributions (greenish, dark tone). Furthermore, their reconstructed images often include high levels of noise such as horizontal stripes,

which significantly degrade the image reconstruction quality. Most notably, for some images, such as the flower pattern image ( $4^{th}$  row) and the berry cake  $6^{th}$  row, most algorithms completely generated high levels of noise that make the quality of reconstructed images extremely poor. However, when using UNets as the pre-and-post-processing models, Le-ADMM is able to reconstruct images with impressive high quality in terms of overall structures, perceptual quality, high-frequency details, and color space distributions. Our qualitative evaluations show that the Le-ADMM method has almost identical visual quality to the ground-truth images, proving it to be a most promising algorithm for image reconstruction in lensless camera imaging.

## 5 DISCUSSION

Our experiments and evaluations highlight the trade-offs between speed and reconstruction quality between different reconstruction models. While traditional methods like GD (Vanilla, Nesterov, FISTA) are generally computationally inexpensive, their performances significantly lag behind more advanced techniques like Le-ADMM. The integration of deep learning priors further enhances image quality, as demonstrated by PnP-ADMM. We observe that while Le-ADMM has impressive performance both quantitatively and qualitatively compared to other algorithms, they are also highly

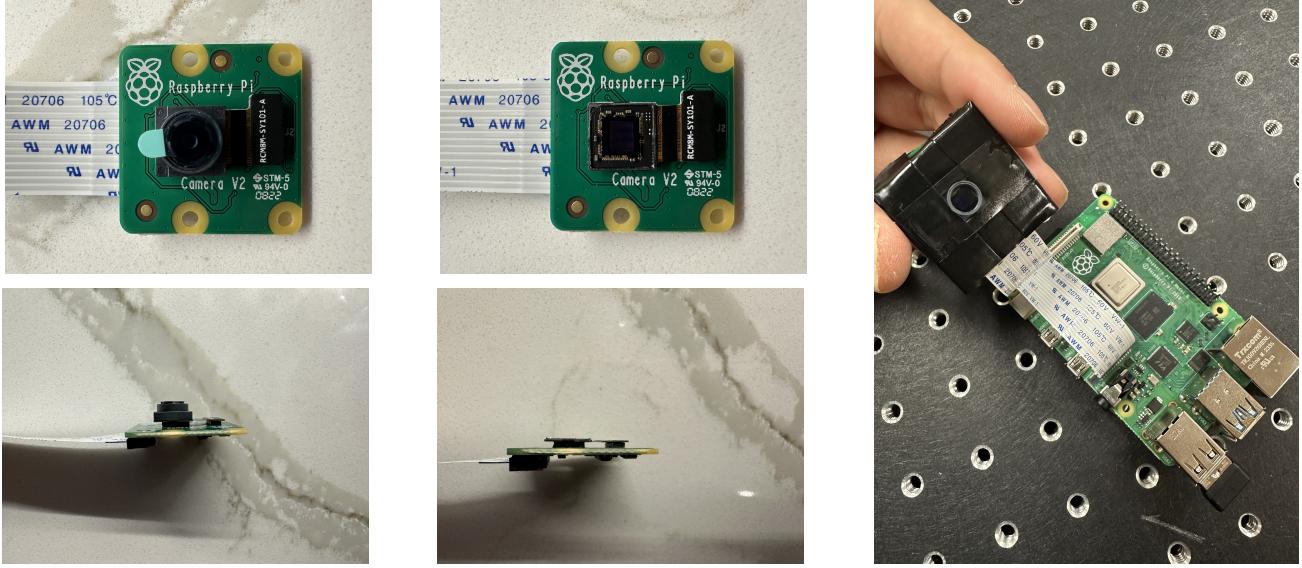


Fig. 7. Prototype development of a lensless camera based on the DiffuserCam design [Antipa et al. 2018] using a Raspberry Pi. Left: Original Raspberry Pi Camera Module v2 (front and side views). Middle: Camera with lens and mounting platform removed, exposing the imaging sensor (front and side views). Right: Current prototype with a Scotch tape diffuser mask over the imaging sensor and the mounting platform.

computationally efficient, only requiring marginally more processing time compared to GD methods. The high performance of Le-ADMM can be explained by its high expressive power that allows it to learn highly complex representations of the inverse mapping. Although the training process of Le-ADMM is computationally demanding and time-consuming, once it converges, the model can be applied efficiently to reconstruct the true image from raw lensless measurements, explaining its high inference efficiency. On the contrary, PnP-ADMM solves the optimization problem during the inference process by iteratively updating the primal and dual variables with some denoising priors. This on-the-fly optimization requires more iterations and is computationally expensive, which leads to slow inference time.

However, Le-ADMM does not come without drawbacks. Since Le-ADMM is a learning-based method, it requires datasets with a large amount of high-quality samples in order to learn accurate reconstruction functions. However, in many application domains such as medical imaging and underwater imaging, such high-quality datasets may not be publicly available and are expensive to collect. Furthermore, like any learning-based algorithms, Le-ADMM cannot generalize to reconstruct images captured by a lensless camera with different settings, designs, and PSF, limiting its usability to only the DiffuserCam.

## 6 CONCLUSION

In this paper, we benchmark reconstruction algorithms for lensless imaging, focusing on the trade-offs between computational efficiency and reconstruction quality. Using the DiffuserCam dataset, we evaluated traditional optimization techniques and deep learning-enhanced hybrid methods, identifying Learned ADMM as a decent approach for achieving real-time reconstruction with balanced speed

and quality. However, as a hybrid model, it requires some training data and pre-trained priors. The runner-ups, based on our observations, are fine-tuned GD-FISTA or ADMM, which come with trade-offs such as certain artifacts but offer acceptable reconstruction quality and fast inference speed. Our analysis highlights the potential for deploying lensless imaging systems in resource-constrained applications.

A limitation of our study is that all of our evaluations and results were based on the DiffuserCam dataset, which represents a specific lensless imaging setup. Therefore, the generalizability of our findings regarding the optimized algorithms to other lensless imaging systems or datasets remains uncertain. Future work will address this limitation by benchmarking across various lensless imaging systems and datasets to evaluate performance under diverse conditions.

To better support the benchmark study, we initiated the development of our own lensless camera prototype based on DiffuserCam and Raspberry Pi. Figure 7 illustrates the timeline of this process. On the left is the original Raspberry Pi camera module v2 with its lens and side view. In the middle is the camera module with the lens and mounting platform removed, along with its side view, showing a significant decrease in size. On the right is the current build with a scotch tape diffuser mask, but without the mounting platform. Removing the mounting platform posed challenges, as it broke the circuits and caused the camera to stop functioning. Despite hardware malfunctions and resource limitations, this prototyping process provided valuable insights. Platforms like LenslessPiCam offered helpful guidelines and scripts for setting up and evaluating lensless imaging pipelines.

Future work will focus on continuing to extend the prototype and setting up an on-device reconstruction pipeline. The next step, based on our current prototype, is to measure and calibrate the

Point Spread Function (PSF), despite the limitation of the mounting platform preventing the diffuser mask from being placed closer to the imaging sensor. Once the PSF is accurately measured, we can capture our own custom DiffuserCam datasets. Specifically, we aim to focus on in-the-wild scene capture, challenging existing algorithms to handle diverse conditions. This effort seeks to advance the applicability of lensless imaging systems and bridge the gap between research prototypes and real-world deployment.

## 7 CONTRIBUTIONS

- **Zijie Cai:** Fine-tuned and collected experimental results for Gradient Descent (GD), GD-FISTA, GD-Nesterov, ADMM, and PnP-ADMM algorithms using the DiffuserCam dataset. Responsible for the initial project proposal and hardware setup, including sourcing and assembling components for the lensless camera prototype. Organized and managed project workflow. Wrote the Abstract, Introduction, Related Work, and Conclusion sections of the report.
- **Anh Nhu:** Implemented, fine-tuned, and collected experimental results for Le-ADMM using U-Nets as pre-and-post-processing modules. Downloaded and implemented implementations around the DiffuserCam dataset for training and evaluations of lensless imaging reconstruction models. Experimented with different settings and variants of the Le-ADMM algorithm to select the best-performing model for the report. I was responsible for writing the Experimental Results section (Quantitative and Qualitative Evaluations), detailed analysis, parts of Dataset Section and Discussion Section. Also proof-read the writing to fix minor issues and improve clarity of the manuscript.
- **Ehaab Basil:** Authored the Methodology section, including detailed explanations of the reconstruction algorithms (Gradient Descent, Nesterov GD, GD-FISTA, ADMM, PnP-ADMM, and Le-ADMM) with formal updates and descriptions. Authored the Hyperparameter Tuning section, ensuring clear presentation of default values and their impact on algorithm performance. Conducted hyperparameter tuning for evaluated algorithms, analyzing the effects of key hyperparameters such as iterations, momentum, step sizes, and regularization strengths. Assisted in enhancing the manuscript's clarity by improving visual presentation, and organizing equations for better readability.

## REFERENCES

- Nick Antipa, Grace Kuo, Reinhard Heckel, Ben Mildenhall, Emrah Bostan, Ren Ng, and Laura Waller. 2018. DiffuserCam: lensless single-exposure 3D imaging. *Optica* 5 (2018), 1–9.
- E. Bezzam, S. Kashani, M. Vetterli, and M. Simeoni. 2023. LenslessPiCam: A Hardware and Software Platform for Lensless Computational Imaging with a Raspberry Pi. *Journal of Open Source Software* 8, 86 (2023), 4747. <https://doi.org/10.21105/joss.04747>
- Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. 2016. Deep Residual Learning for Image Recognition. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*.
- S. Khan, V. Sundar, V. Boominathan, A. Veeraraghavan, and K. Mitra. 2020. FlatNet: Towards Photorealistic Scene Reconstruction from Lensless Measurements. *IEEE Transactions on Pattern Analysis & Machine Intelligence* (2020). <https://doi.org/10.1109/TPAMI.2020.3033882>
- Alex Krizhevsky, Ilya Sutskever, and Geoffrey E Hinton. 2012. ImageNet Classification with Deep Convolutional Neural Networks. In *Advances in Neural Information Processing Systems*, F. Pereira, C.J. Burges, L. Bottou, and K.Q. Weinberger (Eds.), Vol. 25. Curran Associates, Inc. [https://proceedings.neurips.cc/paper\\_files/paper/2012/file/c399862d3b9d6b76c8436e924a68c45b-Paper.pdf](https://proceedings.neurips.cc/paper_files/paper/2012/file/c399862d3b9d6b76c8436e924a68c45b-Paper.pdf)
- K. Monakhova, J. Yurtsever, G. Kuo, N. Antipa, K. Yanny, and L. Waller. 2019. Learned reconstructions for practical mask-based lensless imaging. *Optics Express* 27, 20 (2019), 28075–28090.
- K Simonyan and A Zisserman. 2015. Very deep convolutional networks for large-scale image recognition. *3rd International Conference on Learning Representations (ICLR 2015)*, 1–14.