UHASSELT
KNOWLEDGE IN ACTION

Maastricht University

Faculteit Wetenschappen
School voor Informatietechnologie

master in de informatica

Masterthesis

Deep Learning Approaches for Blind Image Deconvolution, with application to astronomical images

Devon Vanaenrode

Scriptie ingediend tot het behalen van de graad van master in de informatica

PROMOTOR :

Prof. dr. Jori LIESENBORGS

2024
2025

De transnationale Universiteit Limburg is een uniek samenwerkingsverband van twee universiteiten in twee landen: de Universiteit Hasselt en Maastricht University.

UHASSELT
KNOWLEDGE IN ACTION

www.uhasselt.be
Universiteit Hasselt
Campus Hasselt:
Martelarenlaan 42 | 3500 Hasselt
Campus Diepenbeek:
Agoralaan Gebouw D | 3590 Diepenbeek

Maastricht University

Faculteit Wetenschappen
School voor Informatietechnologie

master in de informatica

Masterthesis

Deep Learning Approaches for Blind Image Deconvolution, with application to astronomical images

Devon Vanaenrode

Scriptie ingediend tot het behalen van de graad van master in de informatica

PROMOTOR :

Prof. dr. Jori LIESENBORGS

Universiteit Hasselt

Masterproef voorgedragen tot het behalen van de graad van master in de informatica

Deep Learning Approaches for Blind Image Deconvolution, with application to astronomical images

Auteur:
Devon Vanaenrode

Promotor:
Prof. dr. Jori Liesenborgs

Academiejaar 2024–2025

Acknowledgment

I gratefully thank my promotor, Professor Liesenborgs, for his steady guidance and insightful perspectives. In our many discussions he helped me maintain a clear view of our objectives and offered fresh angles whenever I felt uncertain. I have learned a great deal under his supervision, and I deeply appreciate his support throughout this work.

I also wish to thank my friends and fellow students for their camaraderie, and my girlfriend for her support throughout this year.

Abstract

Blind image deconvolution asks to unmix two unknowns, the latent sharp scene and the blur kernel (point-spread function, PSF), from a single, already degraded photograph. The problem is famously ill-posed: many image–kernel pairs can reproduce the same observation, and naïve optimisation often collapses to trivial answers [1, 2]. This thesis addresses the challenge from a deep-learning perspective tailored to astronomy, where diffraction and atmospheric turbulence routinely veil faint structures and an explicit PSF is rarely measured.

Two complementary neural routes are investigated. First, SelfDeblur, a zero-shot method pairing an encoder–decoder "image generator" with a fully connected "kernel generator", is employed. Both networks are trained solely on the test image, with architectural biases acting as deep priors that steer optimisation away from the delta-kernel pitfall while enforcing physical constraints via soft-max normalisation of the kernel.

Second, a supervised Convolutional Neural Network (CNN) pipeline is developed to regress PSFs and blur kernels directly, with the explicit goal of supplying accurate kernel estimates for subsequent non-blind deconvolution and recovering the latent image, typically the primary focus of blind deconvolution systems. Therefore, a modular data engine augments high-resolution natural photographs, convolves them with a diverse library of synthetic and telescope-derived PSFs (Gaussian, Moffat, Airy-disk, and real telescope kernels), and injects controlled noise to mimic acquisition conditions. Two network variants, a lightweight convolution-attention encoder and a fully connected regressor, enable systematic evaluation of how inductive bias, depth, and a two-stage loss schedule (MSE warm-up followed by SSIM refinement) impact stability and the identifiability challenge.

These investigations reveal several insights. First, deep-prior methods such as SelfDeblur seem capable of recovering realistic PSFs without external training data, albeit with considerable computational overhead. Second, feed-forward regression offers orders-of-magnitude speedups but risks mode collapse, frequently reverting to an average blur when strong structural cues are absent. Finally, the results demonstrate that the interplay between network architecture and the staged loss schedule is crucial, and only specific combinations enable effective latent-image recovery.

By clarifying these trade-offs and pinpointing the factors that alleviate the ill-posedness of blind deconvolution, this work offers valuable insights for designing more robust deblurring pipelines and could guide future efforts to recover sharper details from heavily blurred images.

Table of Contents

Chapter 1

Introduction

1.1 Problem statement and research questions

In modern imaging systems, a variety of factors such as optical aberrations, atmospheric turbulence, finite-aperture effects (which induce *diffraction artefacts*, namely the ring-like ripples and attendant loss of resolution caused by wave interference at the aperture edge), and detector imperfections, inevitably degrade the sharpness and fine-scale detail of a captured scene.

Mathematically, one often expresses the resulting observation y as

$$y = k \otimes x + n,$$

where x is the latent sharp image, k is the so-called Point Spread Function (PSF) that encodes how a single point of light is spread by the system, $\otimes$ denotes two-dimensional convolution, and n models additive noise. Restoring x from y therefore requires simultaneously "undoing" the blur imparted by k and suppressing the noise n. This process is typically known as deconvolution.

A fundamental difficulty arises from the fact that the PSF k is usually unknown. This uncertainty transforms the straightforward non-blind deconvolution problem into blind deconvolution, where both the latent image and the blur kernel must be estimated from the same observation. Without accurate knowledge of k, inversion techniques struggle to recover fine detail, and naive approaches often amplify noise or produce artifacts. Consequently, blind deconvolution remains a notoriously ill-posed problem, demanding more robust and adaptive strategies to reliably restore high-quality images.

Traditional methods like Wiener and Richardson–Lucy deconvolution have long served as benchmarks in the field. Wiener deconvolution [3], based on the Wiener filter, operates efficiently in the frequency domain but requires exact knowledge of both the PSF and the noise power spectrum. Richardson–Lucy deconvolution, independently introduced in two papers [4, 5], is an iterative maximum-likelihood algorithm that enforces non-negativity and is somewhat more tolerant of mild PSF mismatches (e.g., via early stopping or simple regularization), which explains its popularity in astronomical imaging.

However, both Wiener and unmodified Richardson–Lucy deconvolution fundamentally presuppose a known PSF; they do not estimate the blur kernel and thus cannot directly address blind deconvolution, where k must be recovered alongside the latent image x. In the setup of this thesis, where the PSF is entirely unknown, these classical techniques suffer from severe ill-posedness, ringing artifacts, and convergence to trivial solutions, motivating the development of more robust, data-driven approaches.

Furthermore, recent advances in deep learning ([6], [7]), particularly through CNNs, have opened new avenues for addressing these challenges. By learning complex blur representations

directly from data, CNN-based approaches offer a promising alternative to traditional iterative deconvolution methods. In numerous practical settings, a single dataset may encompass multiple distinct point-spread functions, none of which can be directly observed. Estimating such a set of kernels purely from blurred images is notoriously ill-posed because any image–kernel pair that shares the same convolution can masquerade as a solution. This thesis introduces a flexible experimental pipeline for training and evaluating multiple models, with the primary objective of estimating unknown blur kernels rather than directly restoring latent images alongside the blur kernels. This pipeline, together with selected model architectures, is evaluated for its ability to estimate this unknown blur kernel. Additionally, SelfDeblur [8] is examined as a representative blind-deconvolution technique. Subsequent chapters present the pipeline's overall architecture, detail the implementation of chosen model variants, explore the inner workings of SelfDeblur, and assess their potential to enhance image restoration in practical applications.

Drawing on the foregoing motivation and rationale, this thesis addresses three principal research questions:

- Can a CNN be designed to accurately learn and estimate multiple convolution kernels solely from blurred images, despite the intrinsic ambiguities of the convolution process?
- What architectural modifications or training strategies (e.g., regularization techniques, loss functions) are required to mitigate the identifiability problem[1] in blind deconvolution?
- Can the SelfDeblur algorithm be adapted (through changes in kernel parameterization, training data augmentation, or regularization) to recover a wider range of blur kernels than the predominantly motion-blur kernels studied in the original work, and can it do so effectively on astronomical imaging data?

Lastly, the structure of this thesis is organized as follows: this chapter continues with a brief summary of the research questions and key concepts. Chapter 2 is dedicated exclusively to SelfDeblur, the only blind-deconvolution method reviewed in this thesis. Chapter 3 then outlines the proposed methodology, detailing the underlying framework, datasets, and experimental pipeline. Chapter 4 presents and discusses the experimental results. Finally, Chapter 5 concludes the thesis by summarizing the main findings, noting limitations, and suggesting directions for future research.

1.2 Mathematical background

To ground the reader in fundamentals, the next section provides the mathematical background for convolution, deconvolution, the delta kernel and blind deconvolution.

Convolution is a mathematical operation that combines two functions to produce a third function describing how the shape of one is modified by the other. In signal and image processing [9], it models the action of a kernel (or impulse response) that is "slid" across the input, with their pointwise products summed at each position. Mathematically, the continuous convolution of functions f and g is defined as

$$(f * g)(t) = \int_{-\infty}^{\infty} f(\tau)\, g(t-\tau)\, d\tau.$$

In imaging applications, this continuous form is discretised over pixels. If $x[i,j]$ is the true scene and $k[m,n]$ the point-spread function (PSF), then

$$(k * x)[i,j] = \sum_{m,n} k[m,n]\, x[i-m, j-n],$$

[1] The identifiability problem refers to the fact that multiple (x, k) pairs can generate the same blurred observation $y = k * x$, making the true image–kernel pair non-unique.

which precisely captures how each scene point is "smeared" into its neighbourhood by the optics and sensor.

A degenerate case of blur is given by the **delta kernel**, defined as

$$\delta[m,n] = \begin{cases} 1, & (m,n) = (0,0), \\ 0, & \text{otherwise.} \end{cases}$$

Convolving any image x with δ leaves it unchanged, i.e. $(\delta * x)[i,j] = x[i,j]$. Blind-deconvolution methods must guard against this trivial solution, which mathematically corresponds to a kernel that explains no blur.

Given an observed image corrupted by blur and noise,

$$y = k \otimes x + n,$$

classical (non-blind) deconvolution assumes that k is known and recovers x by solving

$$\hat{x} = \arg\min_x \|k * x - y\|_2^2 + \lambda\, \Phi(x),$$

where $\Phi(x)$ is a regulariser (e.g. Tikhonov or total variation) and $\lambda > 0$ balances data fidelity against the prior. In contrast, **blind deconvolution** estimates both image and kernel via

$$(\hat{x}, \hat{k}) = \arg\min_{x,k} \|k * x - y\|_2^2 + \lambda\, \Phi(x) + \gamma\, \Psi(k),$$

where $\Psi(k)$ encodes kernel priors (e.g. non-negativity, unit sum, sparsity). Because many (x,k) pairs can explain the same y, this joint problem is severely ill-posed, and specialised algorithms are required.

1.3 Kernels and their archetypes

Kernels are typically implemented as small, two-dimensional *arrays* of real numbers when applied to a sampled image, but the coefficients themselves can be *generated by sampling a continuous function of two variables* (often separable, e.g. a Gaussian) rather than specified one-by-one [9, Sec. 3.4.4, Example 2]. The size of this kernel (e.g. 3×3 or 5×5) defines the neighbourhood of pixels around each target pixel that will contribute to the new, filtered value. Each coefficient in the kernel acts as a weight that determines how strongly the corresponding neighbour in the input image influences the output pixel. A simple example of how this works can be seen in Figure 1.1, which depicts a 3×3 filter kernel centred on a pixel: each of the nine kernel weights $w(r,c)$ is multiplied by the corresponding neighborhood pixel $f(x+r, y+c)$; the products are summed, and this single value replaces the central pixel as the kernel slides across the entire image.

In more general terms, when applying a kernel to an image through a linear spatial filtering process, one effectively performs a sum-of-products operation. Conceptually, the kernel is "positioned" over a region of the image so that its centre aligns with the pixel being updated. Each kernel coefficient is then multiplied by the corresponding image pixel intensity within the kernel's coverage. These products are summed, and the resulting value is assigned to the output image at that central pixel location. By sliding the kernel systematically over every valid position in the image, a new, filtered image is produced. This procedure can emphasize or suppress specific image features (such as edges, noise, or smooth regions) depending on the chosen kernel values.

Mathematically, if f denotes the original image, w the kernel, and g the filtered image, the operation at a pixel (x,y) is typically expressed as:

$$g(x,y) = \sum_{r=-1}^{1} \sum_{c=-1}^{1} w(r,c)\, f(x-r,\, y-c).$$

assuming a 3×3 kernel. This equation is the *strict* convolution of a 3×3 kernel with image f: the minus signs (or, equivalently, the pre-flipped kernel indices) implement the required 180° rotation of the kernel before the sum-of-products step. If the minus signs are dropped, the operation becomes *correlation*. In practice many texts use the term "convolution" for both, because the two coincide whenever the kernel is spatially symmetric.

Because each kernel coefficient can be chosen to highlight or suppress specific frequency components of the image, kernels are key to tasks such as **smoothing** (where coefficients might all be positive and sum to 1), **sharpening** (where negative coefficients highlight transitions), and **edge detection** (where the kernel emphasizes intensity gradients). The concept of "sliding" the kernel across the entire image, systematically multiplying and summing local neighborhoods, underpins the vast majority of linear filtering techniques in digital image processing.

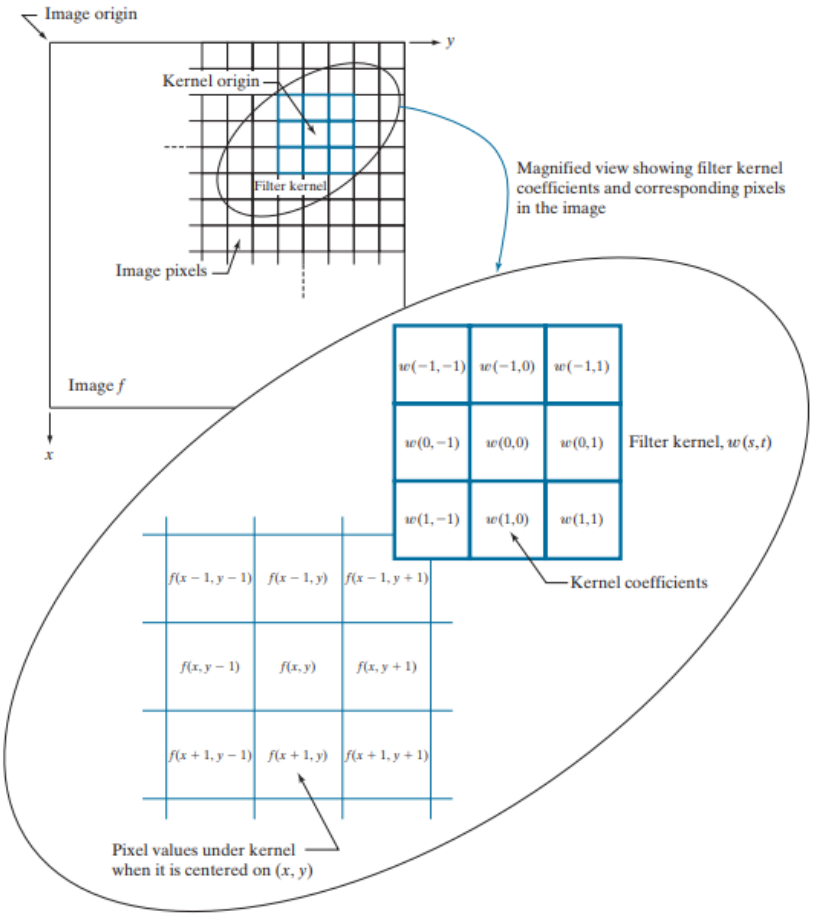
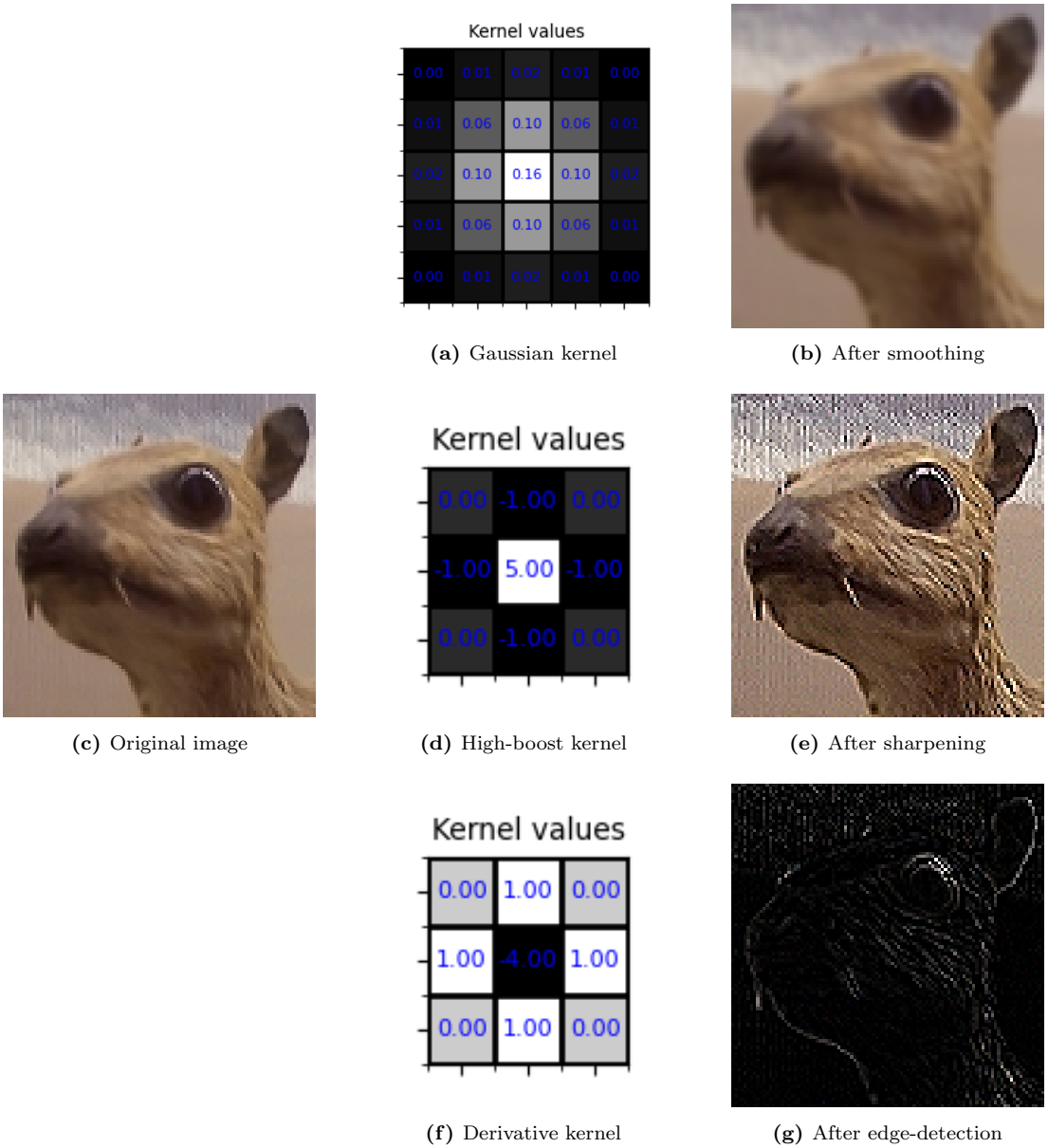


Figure 1.1: Illustration of linear spatial filtering with a 3×3 kernel. The image coordinate origin is at the top left, while the kernel is drawn with its origin at the centre for convenience; this choice is common but not required, and kernels need *not* be spatially symmetric. Pixels are rendered as square tiles solely to make the neighbourhood covered by the kernel easy to see. This image and caption were adapted from the aforementioned book: 'Digital Image Processing, Global Edition' [9]

Kernel archetypes encompass a wide array of filters; however, academic literature ([9], [10]) typically highlights three primary types: smoothing (low-pass) filters, sharpening (high-pass) filters, and edge detection (derivative) filters.

Smoothing kernels, sometimes called **low-pass filters**, are designed to reduce high-frequency variations such as noise or sudden changes in intensity. By blending or "averaging" neighboring

values, they provide a gradual transition between pixel intensities and effectively remove fine details. Common examples include the Gaussian kernel, which applies a weight based on the distance from the center pixel to achieve a softly blurred appearance, and simpler kernels like box or averaging filters, which distribute equal weighting among all neighbors in a specified neighborhood. Figure 1.2 displays an example of using a Gaussian Kernel to smooth out an image in the first row.



(a) Gaussian kernel **(b)** After smoothing

(c) Original image **(d)** High-boost kernel **(e)** After sharpening

(f) Derivative kernel **(g)** After edge-detection

Figure 1.2: Examples of three common kernel archetypes applied to the same base image. **Top row:** Gaussian smoothing. Subfigure (a) shows the Gaussian kernel and (b) the smoothed output. **Middle row:** High-boost sharpening. Subfigure (c) shows the original input image, (d) the sharpening kernel, and (e) the sharpened output. **Bottom row:** First-derivative edge detection. Subfigure (f) shows the derivative kernel and (g) the edge-highlighted result.

Sharpening kernels, known as **high-pass filters**, work in the opposite way. Their goal is to refine sudden intensity changes and fine details in an image by responding to high-frequency information. These filters typically perform some form of subtraction of a blurred version of

the image from the original, which emphasizes edges while leaving uniform areas relatively unchanged. The result is a visually heightened contrast around boundaries, making features such as object edges more prominent. In practice, sharpening kernels are often used to restore clarity to slightly blurred images or to enhance texture in photographic and scientific applications. These kinds of kernels are commonly referred to as Sharpen Kernel, High-Boost Kernel or Laplacian Filter [9] (when it is centred around a Laplacian-like pattern, e.g. 4- or 8-connected neighbours summing to zero). Figure 1.2 shows an example of this sharpening effect caused by a kernel of this category in the second row.

Edge detection (or derivative kernels) seek to isolate the structural transitions within an image by highlighting regions of rapid intensity change. The most common examples include Sobel, Prewitt, and Laplacian filters. Sobel and Prewitt kernels compute approximate gradients in the horizontal and vertical directions, thereby revealing edges with particular orientations. Laplacian filters, in contrast, capture second-order changes in intensity, making them sensitive to both vertical and horizontal transitions simultaneously. These derivative-based filters form the basis of many higher-level vision tasks, such as object recognition and feature extraction, because they help to delineate important shapes and contours. This effect is demonstrated in the third row of Fig. 1.2.

In essence, smoothing filters soften data by reducing spatial variation, sharpening filters emphasize detail by highlighting transitions, and edge detection filters extract and underscore boundaries. The choice of a kernel type depends on the processing goal; whether the intention is to remove noise and irregularities, emphasise subtle features, or zero in on the object outlines that define an image's structure.

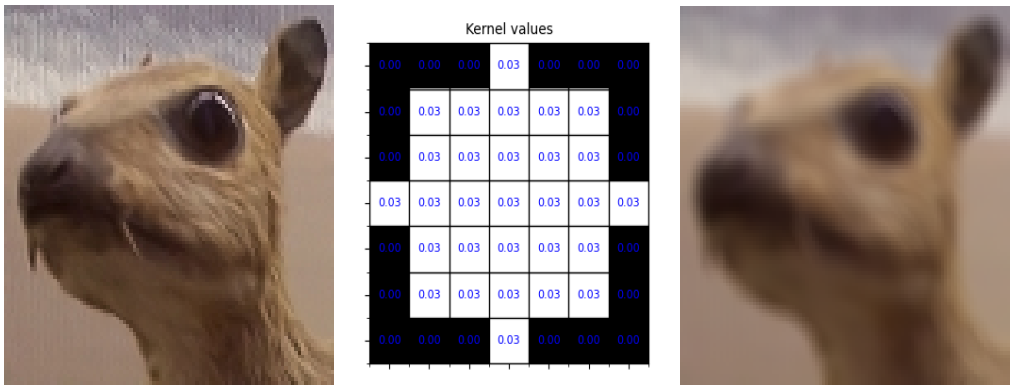
1.4 Key Concepts and Definitions

Blind deconvolution [1] addresses the problem of recovering both the original (sharp) image and the blur kernel when the latter is unknown. This inverse problem is 'blind' because the blur kernel (or filter) is not available *a priori*. Owing to the fact that different combinations of kernels and inputs can produce the same output, blind convolution is an *ill-posed* inverse problem.

Point Spread Function (PSF) describes how an imaging system responds to an idealised point source of light [11]. In practical terms, it characterises the blurring introduced by factors such as optical aberrations, diffraction, and motion during the acquisition process. In deconvolution problems, and throughout this thesis, the PSF plays the role of the convolution or blur kernel, and its accurate estimation is critical for reconstructing the true underlying scene from the observed (blurred) data. In astronomical imaging, the presence of a PSF is dictated by the fundamental physics of light propagation and the limitations of optical systems [12]. For example, in ground-based telescopes, atmospheric turbulence introduces rapid, random fluctuations in the refractive index along the light path, causing light from a point source to spread out rather than concentrate at a single pixel.

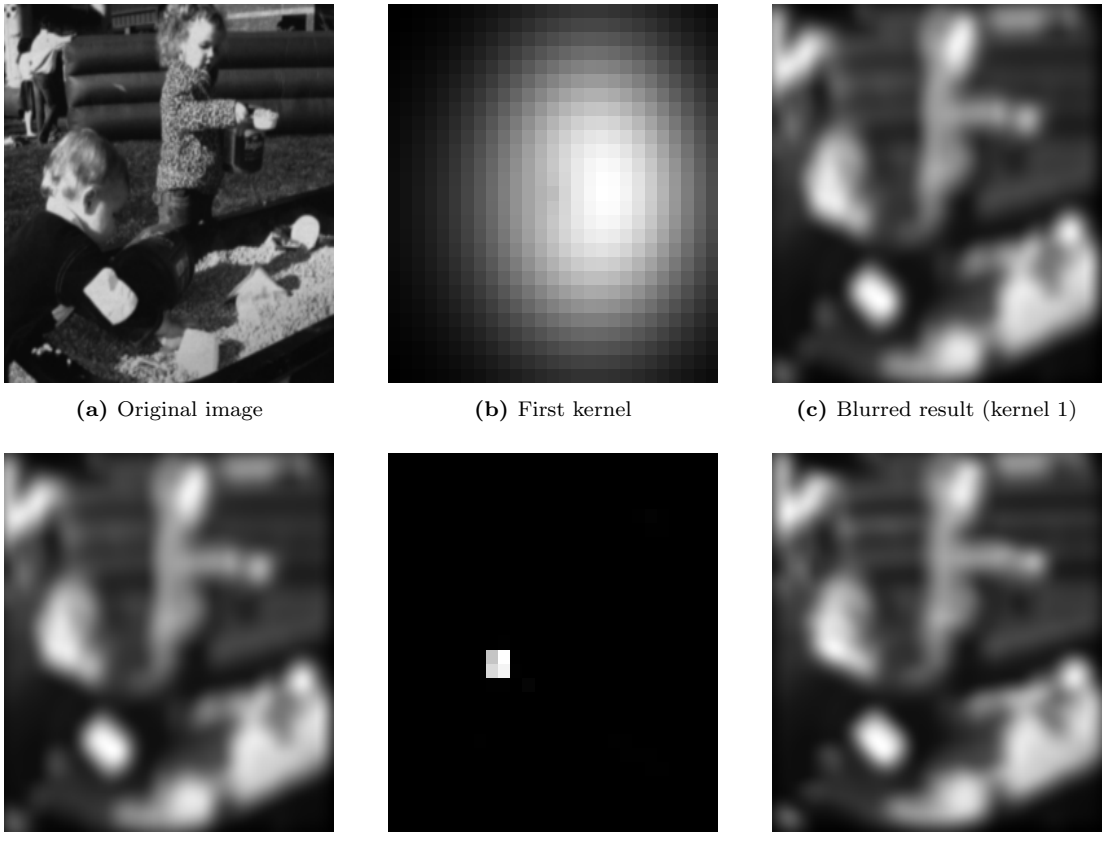
When viewed through the lens of Kernel Archetypes, the PSF aligns with the *smoothing* (or *low-pass*) class of filters. Because a PSF merely *redistributes* light, its coefficients are non-negative and normalised so that they sum to unity, preserving the total incoming flux while spreading it over several pixels. In effect, the PSF reduces abrupt intensity transitions in much the same way as common smoothing kernels (e.g., Gaussian or box filters). Physically, this behaviour stems from fundamental optics and sensor limitations in telescopes or cameras, which suppress high-frequency components and let lower-frequency features dominate. Figure 1.3 displays this effect, highlighting the similarity between a PSF kernel and a smoothing kernel. Crucially, a PSF is *normalised* (its non-negative coefficients sum to one) because it represents how a single point source of light is merely redistributed, not amplified or attenuated. This normalisation, grounded in the energy-conservation physics of optical systems, distinguishes a genuine PSF from a generic (and not necessarily normalised) smoothing mask. The reader can observe the

close resemblance to a Gaussian blur in Fig. 1.2.



(a) Original image without PSF blurring. **(b)** Illustration of the point spread function (PSF) kernel. **(c)** Resulting image after applying the PSF kernel.

Figure 1.3: Example of how a point spread function (PSF) acts as a smoothing kernel. Subfigure (a) shows the original input, (b) visualizes the PSF, and (c) demonstrates the blurred output image.



(a) Original image **(b)** First kernel **(c)** Blurred result (kernel 1)

(d) Blurred result (kernel 1) for comparison **(e)** Second (identity-variant) kernel **(f)** Blurred result (kernel 2)

Figure 1.4: Illustration of the "identity" problem in blind deconvolution. **Top row:** Original image (a), blur kernel 1 (b), and its blurred result (c). **Bottom row:** Same blurred image (d), blur kernel 2 (e), and its result (f). Despite distinct kernels, outputs are nearly identical.

Identifiability problem refers to the fact that the forward model mentioned previously

$$y = k \otimes x + n,$$

is *many-to-one*: distinct *pairs* (x, k) can produce exactly the same observation y. Typical ambiguities include (i) *scale* ($\alpha k * x/\alpha = y$); (ii) *spatial shift* (a shifted kernel convolved with an oppositely shifted image yields the same result); and (iii) *symmetry* (e.g. flipping both image and kernel). Because the mapping is not injective, a unique solution cannot be recovered without imposing extra assumptions, constraints or regularisers (sparsity, smoothness, non-negativity, unit-sum, etc.).

Figure 1.4 illustrates this non-uniqueness: a single base image is convolved with two different, mis-aligned kernels, yet the two blurred outputs are visually (and numerically) almost identical. Only additional prior knowledge could reveal which (x, k) pair is the most likely to have generated the observed blur.

Chapter 2

Related Work

Although the literature on blind deconvolution spans many classical and learning-based methods, this section (and consequently this thesis) primarily focuses on SelfDeblur [8][2] as a textbook example of zero-shot neural blind deconvolution. Examining its joint optimization of deep image and kernel generators, its Softmax-constrained priors, and its implicit handling of the highly ill-posed nature of the problem provides a clear understanding of the core workflows and inductive biases typical of this technique, knowledge that directly informs the proposed extensions.

2.1 Introduction

SelfDeblur is a neural blind-deconvolution method introduced to address key limitations of both classical optimization-based and learning-based approaches in image restoration. Traditional maximum-a-posteriori (MAP) [13, 14] frameworks approach blind deconvolution by seeking the latent image x and blur kernel k that maximise their joint posterior probability given the observed blurry image y:

$$(\hat{x}, \hat{k}) = \arg\max_{x,k} \; p(x, k \mid y),$$

Applying Bayes' rule and assuming that x and k are conditionally independent with respect to y yields

$$p(x, k \mid y) \;\propto\; p(y \mid x, k)\, p(x)\, p(k),$$

where $p(y \mid x, k)$ enforces data fidelity, while $p(x)$ and $p(k)$ encode prior beliefs about natural images and physically plausible kernels. Some works instead write the optimization as

$$(\hat{x}, \hat{k}) = \arg\max_{x,k} \; p(x \mid y)\, p(k \mid y),$$

which is algebraically equivalent once the same likelihood–prior factorization is substituted for each posterior term. In every formulation the problem is ill-posed because both x and k are unknown; without informative priors it collapses to trivial solutions such as the delta (identity) kernel. Handcrafted regularisers (gradient sparsity, patch self-similarity, low-rank constraints, and the like) can prevent this collapse but struggle to represent the diversity of real-world imagery. This shortcoming motivates data-driven approaches such as SelfDeblur, which learn image and kernel priors implicitly from deep-network architectures instead of relying solely on expert-designed regularisation.

[2]This summary is primarily based on the original SelfDeblur paper by Ren et al. [8], with emphasis on the aspects most relevant to this work.

Simultaneously, deep-learning methods trained on external datasets can struggle to generalise when confronted with very large or highly diverse blur kernels, particularly if the blur is spatially non-uniform or the noise statistics differ from those seen during training. In real-world scenes, spatially non-uniform blur arises when different image regions undergo distinct blurring (due to depth variations, complex camera motion or lens aberrations) rather than a single, consistent point-spread function across the frame. Modelling and inverting such locally varying kernels introduces a further layer of complexity that neither the original SelfDeblur paper nor the present work tackles. Addressing spatially non-uniform blur would require region-wise kernel estimation (or depth-aware modelling) and substantially greater computational effort. Therefore, to keep the scope tractable, this thesis assumes a spatially invariant blur kernel in all experiments and leaves non-uniform deconvolution to future investigation.

Building on these limitations, SelfDeblur reframes blind deconvolution by exploiting *deep priors* (the implicit structural biases of convolutional network architectures) rather than relying on large external datasets or hand-crafted regularization terms. In practice, SelfDeblur optimizes two generators directly on a single blurred image in a self-supervised manner, jointly recovering both the latent image and the blur kernel from that sole input. Crucially, this single-image optimization alleviates the ill-posedness of blind deconvolution by parameterizing and constraining both outputs through neural networks: the asymmetric autoencoder for the image inherently favours natural-image statistics[3], while the fully-connected kernel network employs a softmax output layer to enforce non-negativity and unit-sum constraints, making collapse to the trivial delta kernel unlikely (though not impossible) under gradient descent. This data-efficient strategy removes the need for external training data or extensive hyperparameter tuning while still guiding recovery toward plausible, data-consistent solutions.

Next, this thesis delves into the SelfDeblur methodology, examining how its design addresses these limitations.

2.2 Method

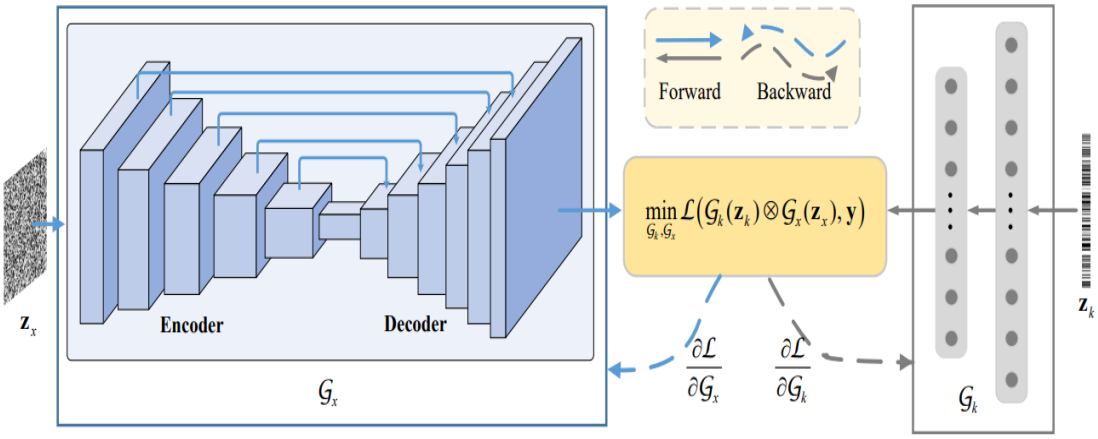


Figure 2.1: A global overview of the proposed method in 'Neural Blind Deconvolution Using Deep Priors', also known as 'SelfDeblur'[8].

From a theoretical standpoint, *SelfDeblur* frames blind deconvolution as unconstrained neural optimization, where two deep generators (one for the latent image and one for the blur kernel) are jointly learned using only the observed blurred input. In Figure 2.1, the left-hand branch, G_x, is an asymmetric encoder–decoder network with skip connections. Random noise z_x is fed into this network, which outputs a candidate latent image $G_x(z_x)$. Because its architecture

[3]Empirical regularities observed in photographs, such as sparse, heavy-tailed gradient distributions, local patch self-similarity, and structured textures typical of real-world scenes.

favours natural image statistics [15, 16], G_x tends to generate images resembling plausible clean scenes, even though it does not rely on external training data. These design choices (convolutional layers, skip connections, and multi-scale encoders) bias the network toward generating structured, coherent images rather than random noise.

In particular, at each decoding stage ℓ, the previous decoder output $D^{(\ell+1)}$ is upsampled (e.g. by transposed convolution or bilinear interpolation) to produce

$$\tilde{D}^{(\ell)} = \mathrm{Up}\big(D^{(\ell+1)}\big),$$

matching the spatial dimensions of the encoder activation $E^{(\ell)}$. These two feature maps are then concatenated along the channel dimension,

$$\big[\tilde{D}^{(\ell)} \,\|\, E^{(\ell)}\big],$$

where $\|$ denotes channel-wise concatenation. Finally, the combined tensor is processed by the decoder block $\mathrm{Dec}^{(\ell)}$ (typically two 3×3 convolutions plus a nonlinearity) to yield

$$D^{(\ell)} = \mathrm{Dec}^{(\ell)}\big([\tilde{D}^{(\ell)} \,\|\, E^{(\ell)}]\big).$$

By fusing coarse, context-rich features with fine spatial details at every scale, G_x reconstructs both global structure and local textures, guiding it toward visually plausible image-like outputs.

On the right-hand side, G_k is a simple, fully connected network that also begins with random noise, z_k. However, to ensure valid kernel constraints, such as those required for a PSF, SelfDeblur applies a Softmax activation at the final layer of G_k. In doing so, each forward pass of $G_k(z_k)$ yields a blur kernel (or PSF) that remains valid throughout the optimization.

These two outputs are convolved ($\otimes$) and compared to the blurred input y via a data-fidelity term, typically the squared ℓ_2 error:

$$\|G_k(z_k) \otimes G_x(z_x) - y\|^2.$$

Depending on the noise level or other considerations, SelfDeblur may include an additional total variation (TV) regulariser [17] on $G_x(z_x)$ to help suppress amplified noise. The TV penalty is defined as

$$\mathrm{TV}(u) \;=\; \sum_{i,j} \sqrt{\big(u_{i+1,j} - u_{i,j}\big)^2 + \big(u_{i,j+1} - u_{i,j}\big)^2},$$

which penalises rapid intensity fluctuations and thereby discouraging spurious high-frequency noise while still allowing true image edges to remain sharp. Incorporating this term yields the composite objective

$$\min_{G_x,\, G_k} \big\| G_k(z_k) \otimes G_x(z_x) - y \big\|_2^2 \;+\; \lambda\, \mathrm{TV}\big(G_x(z_x)\big),$$

where the first term enforces data fidelity (matching the blurred output to the observation) and the TV term imposes smoothness to prevent overfitting to noise. The scalar $\lambda > 0$ balances these two goals and is typically chosen in proportion to the estimated noise level (e.g. $\lambda = 0.1\sigma$), ensuring stable, plausible reconstructions.

Crucially, optimization proceeds by gradient descent on the parameters of both G_k and G_x. At each iteration, the forward pass predicts a blur kernel and a latent image, and the backward pass determines the gradients of the loss $\mathcal{L}$ with respect to the networks' parameters. This step updates G_x so that the generated image better explains the content of y under the current blur, and updates G_k so that the kernel moves closer to correctly modelling the blurring process. Since G_k is enforced to output non-negative, normalised values via Softmax, no separate constraints or projection steps are needed; as described earlier, the model naturally adheres to the usual physical assumptions about blur kernels.

This "zero-shot" training (where the only training sample is the blurred image itself) hinges on the idea that these neural generators impose deep priors on both the image and the kernel. The skip connections, convolutional structure, and learned parameters in G_x steer it toward outputs that look like clean images, while G_k, which is a fully connected layer with a Softmax prior, yields plausible kernels. As optimization proceeds, both networks progressively refine their estimates. Through iterative updates, one obtains the recovered latent image $G_x(z_x)$ and the estimated kernel $G_k(z_k)$ without ever resorting to a hand-crafted prior or an external training set.

2.3 Loss Metrics for training the networks

SelfDeblur starts by minimizing the pixel-wise Mean Squared Error (MSE)[4] for the initial iterations (for instance, the first 1,000 out of 5,000). MSE is defined as

$$\mathcal{L}_{\text{init}}(G_x, G_k) \;=\; \frac{1}{N}\sum_{i=1}^{N}\big[\,(G_k(z_k)\,\otimes\,G_x(z_x))_i \;-\; y_i\big]^2.$$

where $\big(G_k(z_k)\otimes G_x(z_x)\big)_i$ is the i-th pixel of the model's blurred reconstruction, y_i is the corresponding pixel of the observed image, and N is the total number of pixels. By penalising large per-pixel deviations, this loss refines the coarse alignment between the generators' outputs and the blurred input, encouraging rapid convergence towards a plausible baseline solution.

After this initial phase, the optimisation switches to the Structural Similarity Index Measure (SSIM)[5] by defining the adapted SSIM loss as

$$\mathcal{L}_{\text{ssim}}(G_x, G_k) \;=\; 1 \;-\; \text{SSIM}\big(G_k(z_k)\otimes G_x(z_x),\, y\big).$$

This "one minus SSIM" loss emphasises perceptual similarity in luminance, contrast and structure, guiding the network to restore features critical for visual perception. By focusing on luminance, contrast, and structural consistency, SSIM more closely mimics human visual perception than simple pixel-wise errors. To illustrate this, Figure 2.2 demonstrates SSIM's sensitivity to image degradation by applying increasing levels of Gaussian blur to a single base image. The leftmost panel shows the original image. Under a mild blur ($\sigma = 2.0$, centre), SSIM between the blurred and original images is 0.830 while MSE remains low at 0.0022, indicating that most structural content is preserved. In contrast, an extreme blur ($\sigma = 8.0$, right) reduces SSIM to 0.462 and increases MSE to 0.0142, reflecting a substantial loss of edges and texture. The concurrent decline in SSIM and rise in MSE underscore SSIM's closer alignment with perceptual similarity and structural coherence as image quality deteriorates. These loss functions improve training, but they do not fully resolve the core ill-posedness of blind deconvolution, a challenge the ensuing discussion tackles via Softmax constraints.

[4]Here the standard MSE, $\frac{1}{N}\sum_{i=1}^{N}(x_i-\hat{x}_i)^2$, is adapted by treating the model's blurred output $\big(G_k(z_k)\otimes G_x(z_x)\big)$ as the reconstruction $\hat{x}$ and the observed image y as the reference x. In the case of SelfDeblur, the values x_i are fixed (they always come from the same observed image), so the reference does not change during optimization.

[5]The structural similarity index (SSIM) between two images a and b is defined as

$$\text{SSIM}(a,b) \;=\; \frac{(2\mu_a\mu_b + C_1)\,(2\sigma_{ab} + C_2)}{(\mu_a^2+\mu_b^2+C_1)\,(\sigma_a^2+\sigma_b^2+C_2)},$$

where μ_a, μ_b are the mean intensities, σ_a^2, σ_b^2 their variances, σ_{ab} their covariance, and C_1, C_2 small constants for numerical stability.

Figure 2.2: Demonstration of SSIM sensitivity: as Gaussian blur increases (σ rising from left to right), SSIM declines while MSE grows, reflecting the metric's response to image degradation.

2.4 Ill-posedness and Softmax Constraints

Blind deconvolution is fundamentally ill-posed: infinitely many (x, k) pairs can generate the same blurry image y. *SelfDeblur* mitigates this by parameterising the blur kernel via a Softmax layer in the kernel generator G_k. Concretely, let f_k be the network that produces unnormalised logits $\alpha = f_k(z_k)$. Then the kernel entries are given directly by

$$\big[G_k(z_k)\big]_i = \mathrm{Softmax}\big(f_k(z_k)\big)_i = \frac{\exp\big(f_k(z_k)_i\big)}{\sum_j \exp\big(f_k(z_k)_j\big)},$$

automatically enforcing $\big[G_k(z_k)\big]_i \geq 0$ and $\sum_i \big[G_k(z_k)\big]_i = 1$. Although a delta kernel (all mass at a single pixel) remains mathematically possible, it will only minimise the data-fidelity term when the blurred input y itself is nearly identical to the latent sharp image (i.e. almost no blur). In most realistic scenarios, an identity kernel simply transmits noise and artefacts instead of explaining them through convolution, leading to a large residual and high loss.

Transitioning from a nearly uniform 25×25 kernel (where each entry is initially $\approx 1/625$) to a true delta kernel requires exponentiating one logit far above hundreds of others. Under gradient descent on an MSE or SSIM objective, such an extreme reconfiguration is very unlikely: small gradient updates tend instead to shape the kernel toward the true blur pattern present in the data.

Thus, the Softmax parameterisation, together with the deep-prior architecture and the fidelity loss, steers optimisation toward physically plausible kernels rather than trivial identity solutions. An interesting avenue for investigation is to determine the specific conditions under which SelfDeblur may nonetheless converge to a delta kernel, and to evaluate the consistency and robustness of its optimisation; the experiments presented in this thesis include such tests to characterise this behaviour.

Having detailed SelfDeblur's mechanics, this work now considers its suitability for the specific context and its integration into the thesis.

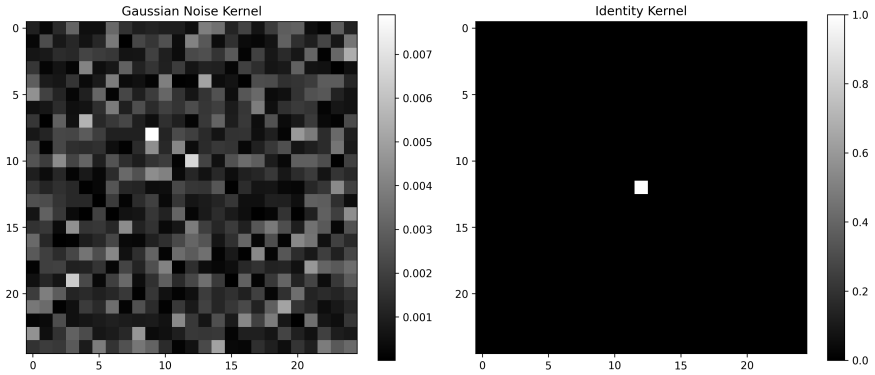


Figure 2.3: Comparison of a randomly initialised Softmax output ("Gaussian Noise Kernel", left) and the delta kernel ("Identity Kernel", right). Moving from a diffuse distribution (values $\approx 1/625$) to a single-spike distribution (value 1.0) requires a substantial shift in the Softmax logits, which gradient descent will avoid unless the data fidelity term strongly favours the identity solution.

2.5 SelfDeblur and the topic of this thesis

In the context of PSF estimation for astronomical data, SelfDeblur's reliance on Gaussian-like kernels (a natural match for the diffraction-limited blurs often encountered in telescopes) suggests it could perform well in restoring fine details of faint celestial objects. Because the model treats the kernel as learnable parameters passed through a Softmax layer (thus ensuring physical validity), it can adapt to wide ranges of blurring patterns without requiring additional handcrafted constraints. Many astrophysical imaging scenarios involve approximately Gaussian kernels (e.g., from telescope diffraction or atmospheric turbulence), and SelfDeblur's capacity for large, diffuse kernels means it can capture the outer wings of a PSF more reliably than simpler, smaller-window methods. Consequently, when integrated into this thesis, SelfDeblur appears well-suited to produce robust reconstructions and kernel estimates, potentially recovering subtle structures that are crucial for scientific interpretation. Section 4 presents the experiments and results of this approach. However, before those empirical findings are detailed in Chapter 4, Chapter 3 outlines the supervised methodology and corresponding processing pipeline developed for this study.

Chapter 3

Approach

3.1 Dataset, Augmentation and Batches

This chapter details the proposed supervised method, contrasting with the methodology of SelfDeblur and serving as an intermediate step towards deconvolution. It begins by describing the dataset and data-augmentation strategy (Section 3.1), followed by the overall pipeline design (Section 3.2), the types of blur kernels used (Section 3.3), the model architecture (Section 3.4) and, finally, the hyperparameter-tuning strategy (Section 3.5).

A dataset of high-resolution natural images (all images share a size of 512×512 pixels) was used as diverse source material for subsequent blur synthesis.[6] The landmark sub-directory offers a wide range of real-world scenes (urban façades, landscapes, and architectural details) ensuring that a broad spectrum of synthetic blur kernels can be evaluated across heterogeneous visual content. Representative examples are shown in Figure 3.1, illustrating the typical scenes under consideration.

(a) Dataset Example 1. **(b)** Dataset Example 2. **(c)** Dataset Example 3.

Figure 3.1: Dataset Examples

Although the complete "130 K" landmark subset supplies substantial diversity, loading the entire collection onto a Graphics Processing Unit (GPU) in a single pass is infeasible. Accordingly, a two-tier sampling strategy is employed. First, a deterministic 60 % / 20 % / 20 % split partitions the data into training, validation and held-out test sets. At the start of each training epoch, a modest proportion of the training images (typically 5–10 %) is randomly selected to form a sequence of mini-batches that fits comfortably within memory and runtime limits.

[6] Kaggle landmark subset, "130 K Images 512 × 512"

Because the sampling relies on a fixed random seed, the exact pools are reproducible, while successive epochs still expose the network to fresh images drawn from the broader training set. This batching scheme retains the diversity of the full dataset yet keeps each run tractable; by contrast, SelfDeblur processes only a single image and therefore does not encounter the same data-handling constraint.

After sampling, each image is augmented on-the-fly using a deliberately minimal set of transformations. To avoid interpolation artefacts, rotations are limited to exact multiples of 90° (90°, 180°, 270°). Random horizontal and vertical flips are applied, and brightness is scaled within ±20 %. Crucially, Gaussian-noise injection and other pixel-level manipulations are excluded, as such noise could interact unpredictably with the diverse blur kernels applied later and confound the learning signal. All operations preserve the native 512 × 512 resolution, ensuring that augmented images remain free of resampling artefacts. With this policy (roughly a 3× expansion in unique appearances) the resulting variation seems adequate to mitigate over-fitting while maintaining clean, blur-only degradations for the regression task. Augmentation is disabled on validation and test splits so that evaluation always reflects unaltered photographs.

Training proceeds with shuffled mini-batches of 16–32 images, each paired with its own randomly drawn blur kernel. This size was chosen empirically: smaller batches introduced high-variance gradients, whereas batches much larger than 32 images exceeded GPU memory. Within every batch, images, augmentation parameters, and kernel types are sampled independently, ensuring that the network sees a heterogeneous mixture of scenes and blur patterns at every optimisation step. Combined with gradient accumulation and an adaptive learning-rate schedule, this design yields stable convergence while remaining within a single-GPU memory footprint. These considerations (sub-sampling for throughput, rich augmentation for robustness, and carefully balanced batch sizes) form the backbone of the data pipeline that feeds the kernel-regression network. After preparing a diverse, manageable training set through careful batching and augmentation, the following subsection elaborates on this batch-based pipeline setup and then outlines the design of the training pipeline that uses these data.

3.2 Pipeline Design

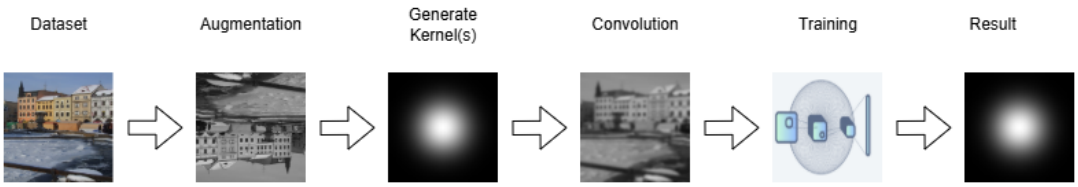


Figure 3.2: Schematic overview of the blur-kernel estimation pipeline. Starting from high-resolution raw images, data augmentation (rotation, flips, brightness/contrast jitter) is applied, then randomized blur kernels are generated. Each augmented image is convolved with its kernel to produce the blurred input, which is fed into a CNN-based regression model. The network should learn to predict the original blur kernel, yielding the recovered kernel as the final output.

Following the sampling and augmentation stages described in the previous section, the processing pipeline (illustrated in a flowchart in Figure 3.2) encompasses all steps from raw images to predicted blur kernels. The chart illustrates the path of one image throughout the pipeline. First, the input images are passed through the augmentation module, which randomly applies the transformations (rotation, flip, brightness/contrast jitter) described above. This yields an augmented batch of images with increased variability. Next, each augmented image is convolved with a randomly generated blur kernel, producing the final blurred image received by the network. A library of two kernel types (detailed in Section 3.3) is maintained, and for each image one kernel is sampled at random, thereby simulating a seemingly wide range of blur effects. The

convolution operation is implemented in the spatial domain, treating each kernel as a blur kernel (for example, the point-spread function) applied to a 2-D image signal. This step ensures that the model's input data consist of realistically blurred images with known ground-truth kernels. After convolution, each blurred image is paired with its kernel (the kernel serves as the regression target for that sample). These image–kernel pairs are then fed into the supervised learning model. The CNN-based model takes the blurred 512×512 image as input and predicts the corresponding kernel (e.g. a 32×32 grid of weights) as output. Thus, the pipeline converts raw images into training pairs and, through different kinds of models, produces a predicted kernel, emphasising a regression approach in which continuous kernel values are estimated directly rather than classified, since the goal is to estimate the exact blur filter.

This end-to-end pipeline allows the model to learn the mapping from image blur patterns to the underlying kernel and the modular design also facilitates experimentation; for example, new kernel types may be swapped in or augmentation adjusted without altering the core model training, simply by modifying the parameters used to execute the pipeline. The pipeline is executed initially in a straightforward serial manner, and can be parallelized across batches for efficiency once validated. Structuring the workflow in this sequential yet well-defined manner ensures reproducibility and a clear separation of concerns between data preparation, convolution and learning.

A strictly sequential, modular layout was selected because it improves debuggability, reproducibility, and tunability. Each module can be executed in isolation, allowing intermediate artefacts (such as augmented images, blurred outputs, or predicted kernels) to be inspected whenever unexpected behaviour arises. The linear structure also supports automatic checkpoints; for example, the blurred images produced by the convolution stage can be cached and reused in subsequent experiments, eliminating the need to recompute them and thereby accelerating development cycles. Finally, all critical hyper-parameters (including sample fraction, augmentation rate, kernel library, batch size, learning-rate schedule, and loss-term weights) are surfaced through a single configuration file. Consequently, the pipeline follows a *plug-and-play* strategy: individual stages can be parallelised or swapped without touching surrounding code (for instance, replacing the backbone with a deeper network, substituting a lightweight variant, or introducing an alternative kernel generator) thereby shortening testing cycles substantially. The result is a pipeline that is straightforward to maintain and extend yet remains efficient enough to scale from single-GPU prototypes to larger multi-GPU deployments. Accordingly, the following section introduces the blur kernel types incorporated into the pipeline.

3.3 Kernel Generation and Types

3.3.1 Empirical PSFs (E-PSF)

Two main families of blur kernels are incorporated: empirical PSFs (E-PSF) extracted from measured instrument data and analytic PSFs (A-PSF) generated via the Photutils interface. E-PSF kernels are pre-computed point-spread functions derived from astronomical instruments such as the *Spitzer* and *Herschel* telescopes. Following the framework of Aniano et al. (2011)[18], disparate instrument PSFs are matched to a common resolution by Fourier-domain deconvolution. A Wiener-filter regularisation step suppresses high-frequency noise, and each kernel is subsequently normalised to conserve total flux, ensuring that convolution preserves integrated intensity. The resulting `convkern` files, publicly released through NASA/IPAC, span a wide variety of instrument PSFs and can be loaded directly as FITS[7] or NumPy arrays for spatial-domain convolution.

[7]The Flexible Image Transport System (FITS) is the de-facto standard format for astronomical images and associated metadata.

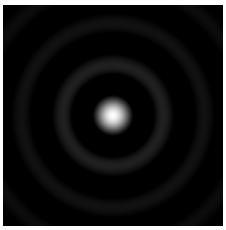
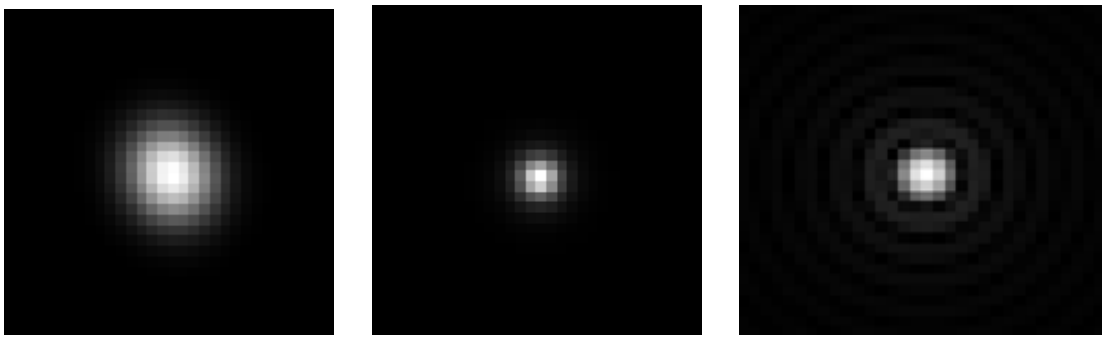


Figure 3.3: Example of an empirical PSF (E-PSF) kernel, illustrating the characteristic ripple pattern typical of this family.

3.3.2 Analytic PSFs (A-PSF)

The PSF subpackage of Photutils supplies a suite of analytic two-dimensional PSF models, each implemented as a subclass of Astropy's `Fittable2DModel` and normalised so that the analytical integral over the entire plane equals the specified total flux. Although the toolbox can generate many distinct PSF kernels, only three model types (Gaussian2D, Moffat2D, and AiryDisk2D) are displayed here to give the reader an idea of what these kernels can look like. Further information on Photutils is available online[8].



(a) Gaussian2D kernel. **(b)** Moffat2D kernel. **(c)** AiryDisk2D kernel.

Figure 3.4: Examples of analytic A-PSF kernels.

The Gaussian2D (Figure 3.4a) model represents a symmetric bell-shaped curve, defined by its total flux, centre coordinates (x_0, y_0), full width at half maximum along the x and y axes, and an optional rotation angle θ. When sampled over the grid, it produces a smooth, isotropic or anisotropic blur without diffraction artefacts.

The Moffat2D model (Figure 3.4b) captures PSFs with a sharp core and extended wings, which are characteristic of atmospheric seeing in ground-based telescopes. It is parameterised by amplitude (or flux), centre position, γ (controlling the core width), and α (governing how rapidly the wings fall off). This profile often provides a more realistic fit to real-world stellar images affected by turbulence.

The AiryDisk2D model (Figure 3.4c) implements the diffraction-limited PSF of a circular aperture. It is defined by its amplitude, centre, and disk radius, and yields the familiar pattern of a bright central peak surrounded by concentric faint rings due to diffraction. This model allows the optical blurring seen in space-based and high-quality ground-based imaging systems to be simulated. The reader will notice the similarities between the AiryDisk2D model and the kernel shown in the E-PSF section.

Within the pipeline, each of these models is sampled across a range of plausible parameters to generate a diverse set of A-PSFs. Each kernel array is evaluated on the chosen grid size,

[8] `https://photutils.readthedocs.io/en/latest/user_guide/psf.html`

clipped to non-negative values, and renormalised to unit sum, ensuring energy conservation and compatibility with the regression framework.

3.4 Model Architecture

Two distinct model variants were explored for kernel regression. The first begins with a shallow convolutional encoder: two 3×3 Conv2D layers (with 32 and 64 filters, respectively, and Rectified Linear Unit (ReLU) activations) preserve spatial dimensions via "same" padding and extract local blur features. A 1×1 Conv2D attention layer then produces a spatial mask that is multiplied element-wise with the feature maps, helping the network emphasise regions most indicative of the underlying kernel. Global average pooling collapses the attended feature maps into a 64-dimensional vector, which is passed through a 256-unit dense bottleneck before a final fully-connected decoder reshapes the output into a 32×32 kernel via a `tanh` activation. This design leverages convolutional layers for local pattern extraction while retaining the flexibility of a dense network to regress arbitrary continuous kernel values.

Figure 3.5: Global overview of the CNN architecture integrated into the pipeline. Only the principal layers are shown; intermediate and auxiliary layers (e.g. the attention layer) have been omitted to keep the diagram readable.

As a contrast, a "pure NN" variant was implemented in which all convolutional layers are removed and the 512×512 input is flattened into a single vector. This vector is fed directly into one or more dense layers before producing the 32×32 kernel output. While empirical evaluation remains pending, theoretical expectations suggest that this all-dense architecture, because it forgoes translation invariance and local receptive fields, would require substantially more parameters, train more slowly, and be more susceptible to over-fitting on limited data. Nevertheless, exploring a pure NN baseline remains instructive: it offers a clear point of comparison for assessing the gains afforded by convolutional structure, is trivial to implement and adapt to different input resolutions, and may reveal scenarios where global, non-local feature combinations suffice. In addition, the simplicity of this design makes it an attractive starting point for rapid prototyping, and any observed performance gaps can directly quantify the value of the inductive biases present in more sophisticated models.

3.5 Hyperparameter Tuning Strategy

The pipeline exposes a concise set of hyper-parameters that govern three successive stages: data preparation, model optimisation, and loss balancing. This section enumerates several of these variables, indicates the workflow component affected, and summarises the qualitative effect of increasing or decreasing each value. These descriptions frame the search space; concrete choices and empirical results are presented in Chapter 4. The list is not exhaustive; rather, it serves as a representative sample illustrating how the tuning procedure prioritises, balances, and integrates the most influential hyper-parameters.

Hyperparameter choices rarely act in isolation: adjusting one often shifts the optimal setting for another. For example, a higher augmentation rate, which broadens the data distribution, may necessitate a modest increase in learning rate to maintain convergence speed, whereas very small batch sizes tend to benefit from gradient-accumulation steps to stabilize updates.

Furthermore, adopting conservative, standard defaults from image-regression practice (initial learning rate $\eta \approx 5 \times 10^{-4}$, batch size $= 16$, loss-term weights $\alpha : \beta : \gamma = 0.3 : 0.2 : 0.5$), hyperparameter adjustments are made only when training instability arises (as indicated by

diverging loss curves). In such cases, the learning rate is reduced or the effective batch size is increased via gradient accumulation before any architectural changes are considered. To balance numerical accuracy against structural fidelity, the SSIM weight is raised if overall intensity is well-reproduced but fine ring patterns are missed, whereas the MAE weight is increased if global amplitude drifts. Finally, batch size, augmentation rate, and epoch budget are always chosen to conform to available GPU memory and wall-time constraints, providing a principled framework for systematic hyperparameter search.

Stage	**Hyperparameter**	**Role in the pipeline**	**Qualitative impact when tuned**
Data prep.	Sample fraction	Portion of the landmark corpus drawn each epoch	Higher values enlarge scene diversity but lengthen epochs; lower values shorten epochs at the risk of over-fitting.
	Augmentation rate	Number of augmented copies per raw image	Increasing boosts appearance variability (higher GPU cost); decreasing yields faster I/O but less variation.
	Kernel-library size	Distinct PSFs available during convolution	A larger library exposes the model to more blur shapes; a smaller one biases training toward a limited subset.
Optimisation	Model depth & size	Number of hidden layers and units per layer	Increasing depth (more layers) and width (more neurons per layer) expands representational capacity, helping the network learn input-specific mappings and avoid collapse;
	Weight-decay (L_2 penalty λ)	Strength of the ℓ_2 regulariser on all trainable weights	Higher λ shrinks weights toward zero, spreading representational capacity and avoiding trivial constant mapping; lower λ allows large weights to dominate, which can exacerbate collapse.
	Adam β_1 (momentum)	Exponential decay rate for the first-moment estimate in Adam	Lower β_1 (e.g. 0.8→0.9) makes updates more responsive to recent gradients, helping escape flat constant-output basins; higher β_1 (e.g. 0.95→0.99) yields smoother updates but may trap in local constant modes.
Loss balance	MAE weight α	Emphasis on pixel-wise accuracy	Higher α tightens numeric match; lower α lets structural terms dominate.
	Cosine weight β	Emphasis on global alignment	Increasing β enforces overall shape; decreasing β relaxes that constraint.
	SSIM weight γ	Emphasis on perceptual structure	Higher γ preserves ring patterns and anisotropy; lower γ shifts focus to numeric losses.

Table 3.1: Principal parameters, their roles, and the qualitative effects of tuning each.

In summary, the methodology and models have been defined; Chapter 4 will evaluate their performance through a series of experiments.

Chapter 4

Experiments and Results

As previously mentioned in Chapter 2, SelfDeblur introduces a zero-shot neural optimisation framework in which coupled deep-image and kernel generators implicitly capture image and PSF priors. In Chapter 3, a modular supervised pipeline was designed to augment high-resolution photographs, synthetically blur them via PSF convolution, and employ CNN-based regressors to predict the underlying kernels. Therefore, this chapter brings those two threads together by first subjecting SelfDeblur to a series of controlled studies: beginning with a loss-function ablation, then probing stability under different initialisations, noise levels and custom benchmark kernels (including the Levin dataset), and finally demonstrating performance on real astronomical PSFs. Thereafter, the thesis turns to the pipeline experiments, where a lightweight convolution–attention encoder is compared against a fully-connected regressor across the same diverse blur scenarios. In doing so, the deep-prior methodology of SelfDeblur is validated and the generalisation and practical applicability of the supervised pipeline are assessed.

4.1 SelfDeblur Experiments

4.1.1 Loss-function ablation: Pure MSE, Pure SSIM and Hybrid

The three optimisation variants differ only in how the loss function is composed during training; the network architecture and all other hyper-parameters remain identical. Pure MSE keeps the objective strictly pixel-wise throughout the whole run, forcing the model to minimise

$$\mathcal{L}_{\text{MSE}} = \text{mse}(\text{out\_y}, y)$$

for every iteration. Pure SSIM replaces that objective by the perceptual surrogate

$$\mathcal{L}_{\text{SSIM}} = 1 - \text{ssim}(\text{out\_y}, y),$$

so that the gradient is driven entirely by structural similarity at every step. The hybrid variant starts with the numerically stable MSE to obtain a coarse alignment, then switches to the structure-aware SSIM once the solution is close enough; in practice the switch occurs after 1000 iterations, implemented as

```
if step < 1000:
    total_loss = mse(out_y, y)      % pixel-wise phase
else:
    total_loss = 1 - ssim(out_y, y) % perceptual phase
```

Thus the hybrid schedule combines the fast, stable convergence of MSE in the early stage with the perceptual sharpening encouraged by SSIM in later iterations, whereas the pure variants rely on a single error signal from start to finish.

To assess SelfDeblur's two outputs, complementary criteria are assigned to each. For the recovered image $\hat{x}$ the emphasis on pixel-level fidelity is conveyed by the peak-signal-to-noise ratio (PSNR),

$$\mathrm{PSNR}(x, \hat{x}) = 10 \log_{10}\Big(\frac{255^2}{\mathrm{MSE}(x, \hat{x})}\Big),$$

while perceptual structure is captured by SSIM, which is sensitive to local contrast and edge coherence but largely indifferent to harmless global shifts in intensity. PSNR is preferred over raw MSE because its decibel scale yields interpretable figures and prevents a few high-energy pixels from dominating the score. The recovered kernel $\hat{k}$ is compared with its ground-truth counterpart k^{gt} through the MSE (reflecting the importance of matching absolute amplitudes) and through the Maximum of Normalised Convolution (MNC) [19],

$$\mathrm{MNC}(\hat{k}, k^{\mathrm{gt}}) = \max\Big(\frac{\hat{k} \otimes k^{\mathrm{gt}}}{\|\hat{k}\|_2 \, \|k^{\mathrm{gt}}\|_2}\Big),$$

which peaks at unity only when the two kernels coincide up to scale and translation, thereby gauging their structural overlap. Together PSNR and SSIM provide a balanced view of numerical and perceptual accuracy for images, whereas MSE and MNC offer the same dual perspective for kernels.

Before discussing the outcomes, the test data used in these experiments is described. The experiments were executed on the following inputs: Figure 4.1 shows the base image, and Figure 4.2 shows the used kernels which are 8 in total. The base image has a resolution of 255x255 and the kernels have varying resolutions, in between the ranges 13x13 and 23x23.

Figure 4.1: Levin benchmark test image (255 × 255 px): a grayscale cut-out of children playing in a sandbox. Used as the base observation for all SelfDeblur experiments.

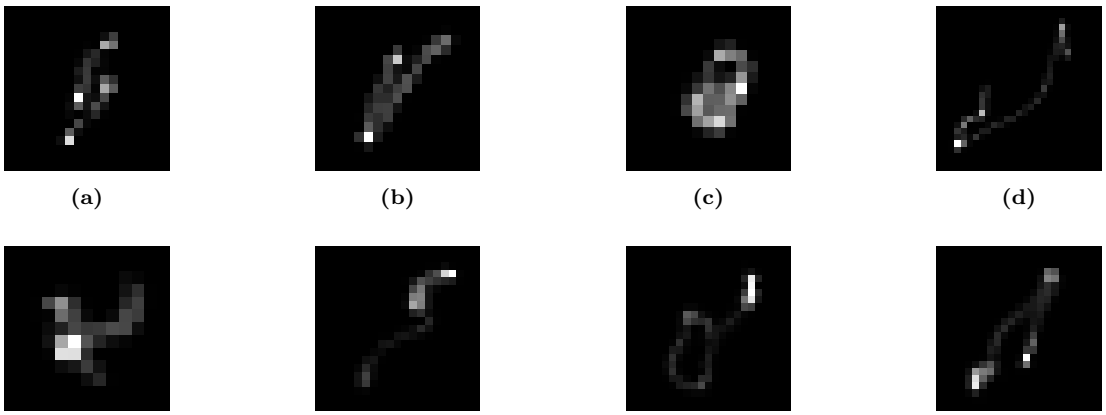


Figure 4.2: Eight linear motion–blur kernels from the Levin et al. (2009) blind-deconvolution benchmark dataset [1].

Table 4.1: Performance metrics for recovered images and kernels based on the pure MSE method.

Kernel	PSNR (dB)	SSIM	MSE	MNC
Kernel 1	31.17	0.9087	13.4100	0.7488
Kernel 2	20.88	0.6299	23.4637	0.5888
Kernel 3	24.58	0.8084	23.6800	0.6279
Kernel 4	18.49	0.4561	18.3086	0.8767
Kernel 5	26.44	0.8513	26.5799	0.6539
Kernel 6	20.21	0.6316	16.3696	0.4953
Kernel 7	19.63	0.5916	14.6503	0.7572
Kernel 8	20.18	0.6124	17.9414	0.9421
Averages	22.70	0.6863	19.30	0.7113

Prior to examining the results, some clarification of the tables in this subsection is required. PSNR and SSIM quantify the fidelity of the recovered images relative to the base image shown in Figure 4.1; conversely, MSE and MNC assess the accuracy of the estimated kernels against the ground-truth kernels. Each row denotes the kernel used for convolution (for example, "Kernel 4" corresponds to kernel (d) in Figure 4.2). Finally, the coloured formatting does not identify column-wise extremes but rather highlights metric pairs: one colour for PSNR & SSIM and another for MSE & MNC, with a marginal emphasis on SSIM and MNC owing to their greater sensitivity to structural similarity.

Table 4.1 summarises the outcome when SelfDeblur is optimised with a pure MSE objective. Averaged over the eight blur instances, the reconstruction attains 22.7 dB in PSNR and 0.69 in SSIM, signalling an overall reduction in pixel error (when compared to the other tables in this subsection) but only moderate perceptual quality. The per-kernel scores, however, reveal pronounced variability. The small, nearly isotropic *Kernel 1* is handled very well (31.17 dB, SSIM 0.91), whereas the broader, strongly anisotropic *Kernel 4* yields a mere 18.49 dB and an SSIM of 0.46. A similar spread is seen in the kernel domain: although the mean kernel MSE stands at 19.3, the MNC still averages 0.71, and for *Kernel 8* the structural overlap almost reaches the ideal value (MNC 0.94) despite a sizeable amplitude error (MSE 17.9).

These contrasts hint at a clear pattern. A loss that is strictly pixel-based copes well with compact kernels but deteriorates as soon as the blur occupies a larger footprint or introduces strong directional structure. In other words, the performance gap between kernels hints at a marked sensitivity of SelfDeblur to the specific blur kernel involved: the method converges to numerically sound yet perceptually over-smoothed images for wide or anisotropic kernels, while delivering sharper results when the underlying point-spread function is narrow and symmetric.

Table 4.2: Performance metrics for the eight kernels based on the pure SSIM method.

Kernel	PSNR (dB)	SSIM	MSE	MNC
Kernel 1	15.60	0.3593	23.3906	0.7341
Kernel 2	19.07	0.4803	27.4048	0.7595
Kernel 3	22.49	0.5998	24.7289	0.7211
Kernel 4	16.27	0.3653	20.7641	0.9170
Kernel 5	21.38	0.6147	30.7811	0.5376
Kernel 6	15.63	0.2579	20.7279	0.5607
Kernel 7	15.36	0.2412	25.2382	0.6634
Kernel 8	16.55	0.4130	21.9452	0.8144
Average	17.79	0.4164	24.37	0.7135

As a further illustration of loss-schedule effects, Table 4.2 reports the results when SelfDeblur is driven solely by an SSIM objective from the first iteration. The reconstructed images exhibit

relatively low fidelity, with a mean PSNR of 17.79 dB and an SSIM of 0.416. Kernel estimates also suffer, yielding an average MSE of 24.37 and an MNC of 0.7135, although Kernel 4 still achieves a high structural match (MNC 0.9170).

Moreover, the degradation is systematic rather than anecdotal. All kernels that benefited under the pixel-based loss now suffer substantial drops (Kernel 1 loses over 15.5 dB in PSNR, Kernel 7 around 4.3 dB) and no kernel exhibits an unequivocal gain. These observations suggest that an initial MSE phase plays a stabilising role: by supplying strong, spatially distributed gradients it guides the network towards a coarse alignment of image and kernel before the more discriminative but less informative SSIM term refines perceptual detail. Omitting this phase leaves the optimisation dominated by SSIM's highly non-convex landscape, where early gradients are weak and frequently lead to sub-optimal basins.

Because the pure-SSIM configuration seems to underperform across nearly every metric and every blur instance, it likely will not offer any clear advantages for the forthcoming astronomical experiments. Therefore, the tests done in Subsection 4.1.5 concentrate on the pure-MSE and hybrid objectives, which together span the practical trade-off between pixel fidelity and structural robustness.

Table 4.3: Performance metrics for the eight kernels based on the Hybrid method.

Kernel	PSNR (dB)	SSIM	MSE	MNC
Kernel 1	28.05	0.8626	15.80	0.6585
Kernel 2	20.08	0.5744	19.84	0.6362
Kernel 3	23.78	0.7659	24.78	0.5972
Kernel 4	19.66	0.5249	18.11	0.9324
Kernel 5	23.57	0.7684	30.95	0.6834
Kernel 6	17.61	0.4435	15.61	0.6304
Kernel 7	19.60	0.5526	12.83	0.7549
Kernel 8	19.28	0.5415	17.47	0.9168
Average	21.45	0.6292	19.42	0.7262

Finally, the hybrid schedule (optimising with MSE for an initial coarse-alignment phase and switching to SSIM once the estimate has stabilised) produces a metric profile that is subtly different, rather than unequivocally better or worse, than the pure-MSE counterpart (Table 4.3). Average PSNR and SSIM fall slightly (22.70 dB $\rightarrow$ 21.45 dB and 0.686 $\rightarrow$ 0.629), reflecting a modest relaxation of pixel-wise accuracy, but this loss is offset by a small rise in the average kernel-overlap score (MNC 0.711 $\rightarrow$ 0.726) and, more importantly, by a clear narrowing of the spread across individual blur cases. Kernels that previously posed difficulties (particularly the wide, directional fourth kernel) benefit from the later SSIM phase, gaining structural coherence in both image and kernel space, while kernels that were already handled well under pure MSE surrender only a few decibels of PSNR.

In practice, the two training regimes (one optimized with MSE loss and the other with the hybrid loss function) offer complementary strengths. Pure MSE seems to preserve the highest pixel fidelity when the blur kernel is compact and isotropic, whereas the hybrid objective may offer greater robustness once the point-spread function becomes broader or more anisotropic. However, in real astronomical imaging the ground truth is unavailable, so one cannot directly compare PSNR or SSIM to the ground truth. Instead, selection must rely on blind-deconvolution diagnostics to infer which reconstruction is most plausible. Nonetheless, the experimental results indicate that tailoring the loss schedule to PSF morphology can enhance reconstruction reliability under realistic, ground-truth-free conditions.

4.1.2 Robustness and stability tests (ill-posedness)

To probe how SelfDeblur handles the inherent ill-posedness of blind deconvolution, ten independent optimisation runs are launched on *exactly the same* blurred observation. Each run receives the identical input pair: the 255×255 scene shown in Figure 4.1 convolved with the 19×19 point-spread function of Figure 4.2 (a), but starts from a fresh random initialisation of the generator weights and input noise. Because the method is stochastic and the underlying optimisation landscape is known to contain many local minima, repeating the experiment under identical external conditions offers a direct test of *robustness and stability*: if the algorithm consistently converges to comparable images and kernels, the implicit deep priors[9] guide it toward a well-defined basin; large dispersion, by contrast, would imply sensitivity to initialisation and thus limited practical reliability. In the analysis that follows, dispersion across the ten outputs is quantified in the kernel space (MSE and MNC).

Table 4.4: MSE and MNC over ten independent runs (highlighted: best in green, worst in red). PSNR and SSIM (metrics normally used to assess reconstructed-image fidelity) are omitted here, as the focus is exclusively on kernel reconstruction.

Run	MSE	MNC
Run 1	11.5596	0.7123
Run 2	12.8781	0.7158
Run 3	12.6704	0.7284
Run 4	21.8144	0.7775
Run 5	9.0499	0.7236
Run 6	16.7895	0.6360
Run 7	13.1247	0.7531
Run 8	22.1135	0.7618
Run 9	18.2964	0.7473
Run 10	16.8200	0.7470
Average	15.5117	0.7303

From Table 4.4 it is evident that the ten runs converge to broadly similar kernel estimates. Numerical pixel accuracy (MSE) spans a modest band (from 9.0 to 22.1 with a mean of 15.51) while structural overlap (MNC) remains tightly grouped around 0.73, never dropping below 0.64 and peaking at 0.78. The worst-amplitude case (Run 8) still aligns structurally almost as well as the best-overlap run, and conversely the lowest-overlap run (Run 6) shows only a mid-range MSE. Such clustering suggests that random initialisation rarely drives the optimiser into qualitatively different basins: most runs recover essentially the same kernel geometry, with residual spread confined chiefly to scale and fine-detail amplitudes.

To illustrate this variability, Figure 4.3 visualises the dispersion already hinted at in Table 4.4. The left-hand boxplot shows that the kernel-wise MSE varies more broadly across runs, with the inter-quartile range spanning nearly 6 units and the whiskers extending from just under 9 to slightly above 22. In other words, the exact pixel amplitudes of the estimated kernels are sensitive to the stochastic elements of the algorithm, such as random weight initialisation and the initial noise inputs. By contrast, the right-hand plot reveals that the MNC scores cluster tightly around 0.74: aside from a single mild outlier, every run lies within a window of ± 0.03. Such consistency indicates that SelfDeblur almost always locks onto the correct *structure* of the point-spread function even when the fine-scale amplitudes fluctuate. Because a faithful geometric match is the principal requirement for accurate non-blind deconvolution, this stability suggests that the method will, in practice, converge to a reconstruction that remains reasonably close to the unknown ground-truth image.

[9] Implicit deep priors denote the structural biases inherent to the deep-image and kernel generators (e.g. convolutional autoencoders and fully-connected kernel networks), which tend to produce natural-image statistics and physically plausible PSFs without explicit regularisation; see Chapter 2, where SelfDeblur's use of deep priors is first introduced.

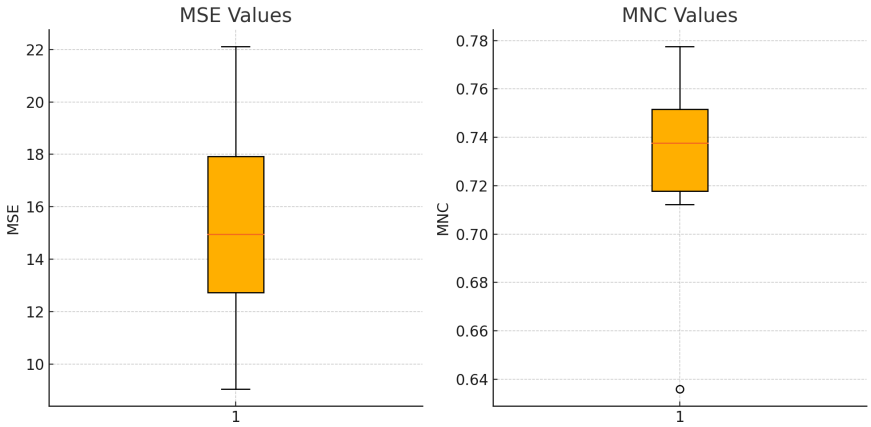
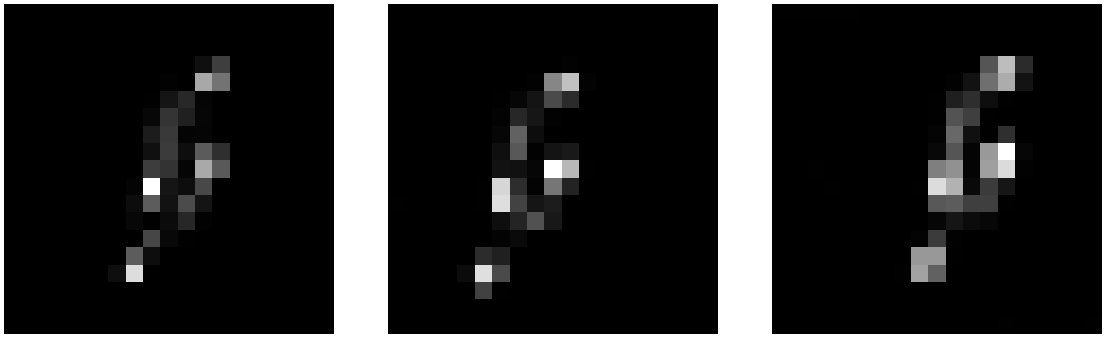


Figure 4.3: Box-and-whisker plots of kernel error (MSE, left) and structural overlap (MNC, right) over ten independent SelfDeblur runs. The narrow spread in MNC contrasts with the wider spread in MSE, illustrating that the algorithm reliably recovers the kernel's overall shape but exhibits run-to-run variation in pixel-exact amplitudes.

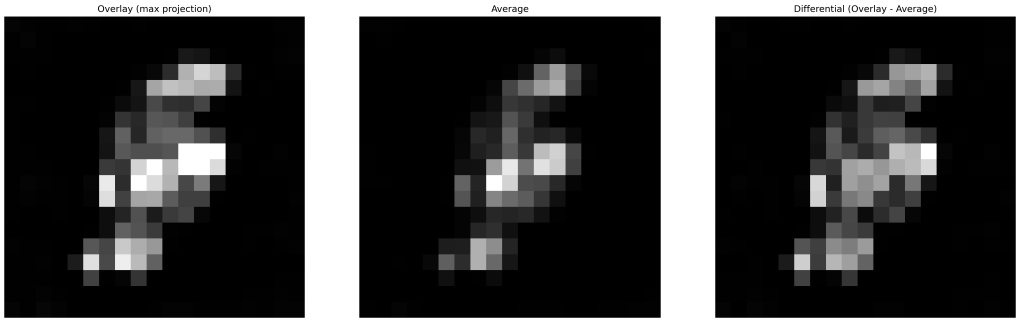


(a) Base **(b)** Run 4 **(c)** Run 6

Figure 4.4: Comparison of the ground-truth kernel (a) with the reconstruction from the best run (b, Run 4) and the worst outlier (c, Run 6), as identified in Figure 4.3's box-and-whisker plots; the pronounced pixel-level discrepancies in (c) illustrate the extent of variability under stochastic initialization.

As illustrated in Figure 4.4, these statistical trends correspond to clear visual differences: the reconstruction from the best run (Run 4) closely reproduces both the shape and amplitudes of the ground-truth kernel, whereas the worst outlier (Run 6) preserves overall geometry but exhibits pronounced pixel-level deviations. Consequently, the examples in Figure 4.4 corroborate the quantitative variability observed in the box-and-whisker plots of Figure 4.3.

As a further visual complement to the box-plot analysis in Figure 4.3, Figure 4.5 presents three views of the ten recovered kernels: (a) the max-projection overlay, (b) the pixel-wise average, and (c) the absolute difference between overlay and mean. Panel (a) demonstrates that even when all ten kernels are overlaid, no estimate exhibits a radically different structure or spurious sub-structure, confirming the absence of alternative convergence modes. Panel (b) then reveals a sharply defined diagonal streak that matches the ground-truth PSF in Figure 4.2(a), with no stray lobes or secondary maxima. Finally, panel (c) highlights the pixel-wise amplitude variability, isolating areas where individual optimisations deviate most from the mean. Together, these triplets confirm that SelfDeblur reproducibly converges to the same kernel geometry, with run-to-run randomness affecting only moderate intensity variations.



(a) Overlay of all ten kernels (max projection).

(b) Pixel-wise average.

(c) Difference between overlay and mean.

Figure 4.5: Three views of kernel reconstructions across ten independent runs (cf. Figure 4.3). Panel (a) shows the max-projection overlay, emphasising regions of consistent vs. variable energy; panel (b) plots the mean estimate, which highlights the stable kernel structure; and panel (c) displays the pixel-wise difference between overlay and mean, marking areas where stochastic initialisation produces the largest deviations. A common greyscale colourbar (right) ensures that intensity levels are directly comparable across panels.

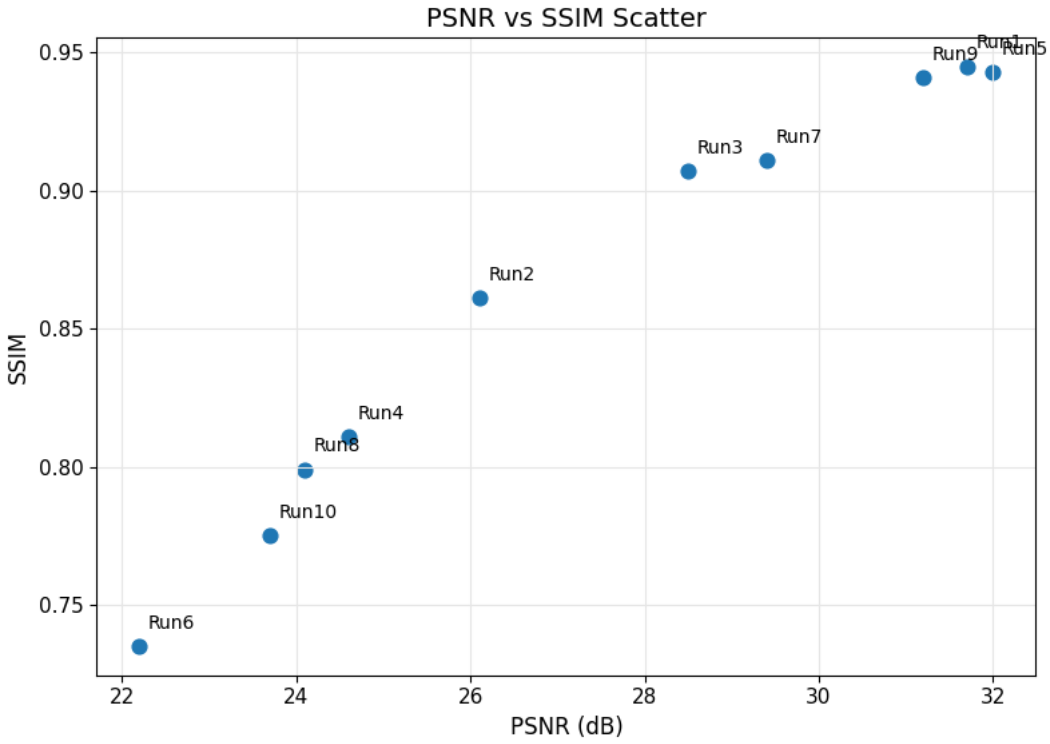


Figure 4.6: Forward-model consistency across ten independent runs, plotted as PSNR (horizontal axis) versus SSIM (vertical axis) between each synthetic observation $\tilde{y}_i$ and the fixed input y. Once more, Run 6 emerges as an outlier, whereas the remaining runs achieve relatively high and consistent reconstruction metrics.

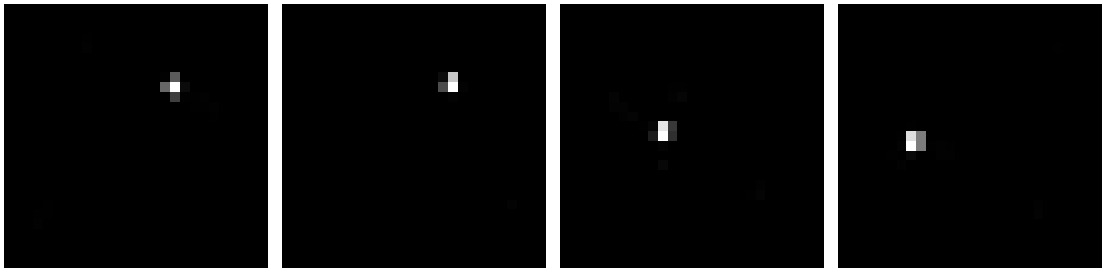
Furthermore, Figure 4.6 closes the loop on the forward model by plotting PSNR against SSIM for each reconvolution $\tilde{y}_i = \hat{k}_i \otimes \hat{x}_i$. Had the optimisation settled into qualitatively different minima, the points would scatter broadly; instead, they form a compact cluster between approximately 25–32 dB PSNR and 0.86–0.95 SSIM, with only Run 6 appearing as a clear outlier

at 22 dB/0.74 SSIM. Thus, despite stochastic initialisation and the theoretical existence of infinitely many (x, k) pairs that explain the data, SelfDeblur consistently yields forward-model outputs of nearly indistinguishable fidelity. This reinforces the earlier conclusions from the kernel overlay and boxplots: the algorithm robustly recovers the essential blur structure and delivers reconstructions that are, for practical purposes, interchangeable.

(a) Groundtruth image y. **(b)** Recovered latent image $\hat{x}_4$. **(c)** Recovered kernel $\hat{k}_4$.

Figure 4.7: Input–output identity test for run 4 under near–noise-free conditions: (a) the blurred observation y, (b) the reconstructed latent image $\hat{x}_4$, and (c) the corresponding kernel $\hat{k}_4$, which collapses to four central pixels.

Finally, Figure 4.7 presents a stress-test for the classic blind-deconvolution failure mode: when the observation is virtually noise-free, the data fidelity term alone admits the trivial solution in which the recovered image matches the blurred input and the kernel collapses to an identity-like spike. Under otherwise identical conditions but with additive noise reduced to a negligible level, SelfDeblur indeed drifts toward this outcome. Panel (c) shows the recovered $\hat{k}_4$ with structure confined to four central pixels rather than a true delta impulse, while panel (b) confirms that $\hat{x}_4$ is almost indistinguishable from the input in panel (a). The presence of four bright pixels hints at a minute residual blur or premature termination of optimisation. Therefore, although the deep-prior framework remains stable under typical noise levels encountered in practice, explicit regularisation (such as the TV penalty introduced in Section 2.2) or at least an accurate noise estimate to prevent kernel collapse is essential whenever measurements approach the noise-free regime.



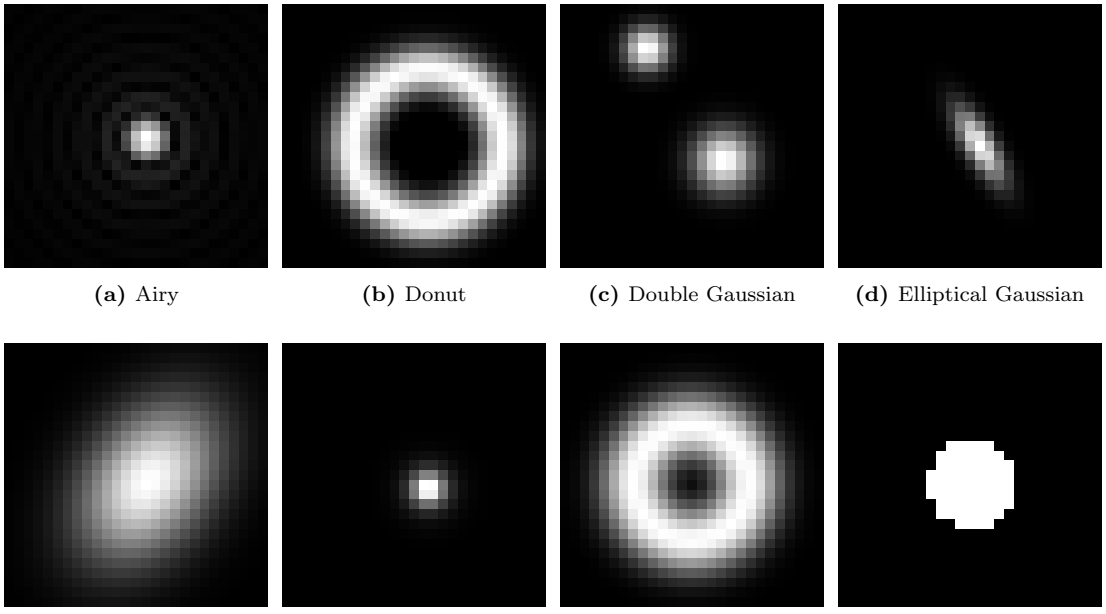
(a) Kernel found for the first run. **(b)** Kernel found for the fourth run. **(c)** Kernel found for the seventh run. **(d)** Kernel found for the eighth run.

Figure 4.8: Four example "identity" kernels retrieved by SelfDeblur when applied to an already-blurred input. Each panel shows the kernel found in an independent run (with random initialisation), rendered at full scale. Despite stochastic starts, all runs converge to a near–delta spike, confirming that the algorithm can collapse to the identity solution when no nontrivial blur is present.

As shown in Figure 4.8(a–d), ten independent runs on the same near–noise-free base image consistently recover an identity-like kernel. Subtle differences in individual pixel amplitudes, apparent as slight brightness variations across the panels, stem from sub-pixel misalignments

and scaling. This confirms that, even when exact per-pixel values vary, SelfDeblur robustly captures or approximates the true kernel structure. The reader can find further argumentation around this topic in the next section.

4.1.3 Levin dataset with custom kernels



(a) Airy **(b)** Donut **(c)** Double Gaussian **(d)** Elliptical Gaussian

(e) Gaussian **(f)** Moffat **(g)** Ring **(h)** Top Hat

Figure 4.9: The eight custom kernels used in the experiments.

The next experiment again employs the Levin sandbox image (255×255 px; Figure 4.1) and convolves it with eight purpose-built kernels chosen simply to provide a broad spread of blur morphologies (Figure 4.9). By retaining the same base image as in the original Levin benchmarks, the set-up isolates the consequences of replacing the canonical linear-motion kernels with a spectrum of more varied blur kernel shapes. Each kernel is first resampled to 25×25 px and normalised to unit flux, ensuring that all simulated observations share identical noise statistics. The collection spans, for example, a centrally peaked Airy pattern, a hollow annulus, a pair of displaced Gaussians, and an elongated elliptical Gaussian, together with a classical circular Gaussian, a wing-dominated Moffat profile, a broad ring, and a sharp-edged top-hat aperture. While the assortment is not intended to mimic any single instrument, it supplies a representative cross-section of possible blur behaviours; in particular, kernels (a) and (e) illustrate how diffraction-limited and seeing-dominated PSF shapes, respectively, imprint themselves on an image through convolution. Consequently, convolving the Levin image with each kernel in turn allows us to assess how SelfDeblur copes with blurs that range from compact and isotropic to highly directional or annular.

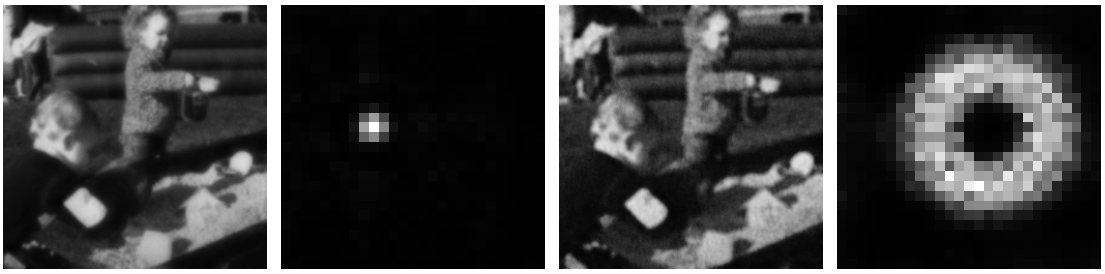
Table 4.5 confirms that, while SelfDeblur seems to recover the blur kernels with moderate fidelity (average MNC = 0.918), the quality of the restored images is appreciably more variable (average PSNR = 20.08 dB, SSIM = 0.576). For instance, the Donut and Gaussian PSFs achieve comparable kernel overlaps (MNC $\approx$ 0.896 and 0.962, respectively), yet the Donut blur is deblurred more cleanly, yielding a PSNR that is 6.1 dB higher and an SSIM that is 0.31 larger than for the Gaussian case. Conversely, the Ring kernel attains the best kernel match in the set (MNC = 0.970) but still exhibits residual artefacts, so that its PSNR (23.04 dB) only marginally surpasses the ensemble mean.

This behaviour is best understood by recalling the two–stage nature of blind deconvolution. After estimating the PSF, the algorithm must still invert the blur; when that blur is governed

Table 4.5: Performance metrics for the eight custom kernels. The colours here are used analogously to those in the previous section.

Kernel	PSNR (dB)	SSIM	MSE	MNC
Airy	20.91	0.6253	30.54	0.8700
Donut	23.61	0.7465	65.58	0.8958
Double Gaussian	20.77	0.6336	32.82	0.9556
Elliptical Gaussian	16.53	0.3791	12.60	0.9108
Gaussian	17.48	0.4382	77.06	0.9620
Moffat	16.85	0.3976	9.50	0.9447
Ring	23.04	0.7090	75.85	0.9703
Top Hat	21.43	0.6816	12.52	0.8369
Average	20.08	0.5764	39.56	0.9183

by a broad, smooth kernel (such as the Gaussian or Ring) the inversion step inevitably amplifies noise and ringing, leading to lower PSNR and SSIM. Conversely, compact or sparse kernels (e.g. Donut, Double Gaussian) impose a gentler inversion and therefore yield sharper reconstructions, even when their kernel overlaps are slightly smaller. Figure 4.10 makes the distinction clear. A side-by-side inspection with the reference kernels in Figure 4.9 shows that the Ring kernel is reproduced faithfully, whereas the faint concentric rings of the Airy pattern are largely lost. Therefore, high MNC scores do not guarantee that every fine feature has been captured. Nevertheless, the restored sandbox images in panels (a) and (c) remain visually close to the original photograph (Figure 4.1), indicating that any residual kernel errors translate into only minor differences in the final images.



(a) Restored sandbox image when the underlying blur is an Airy PSF

(b) SelfDeblur's corresponding estimate of the Airy kernel

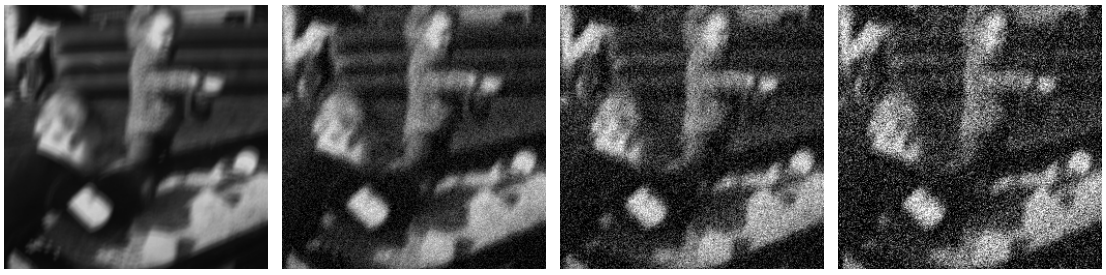
(c) Restored sandbox image for the broad ring PSF

(d) Recovered ring kernel

Figure 4.10: Qualitative results for two representative blur scenarios. Together these panels illustrate that the algorithm successfully retrieves both the scene detail and the distinctive structure of sharply peaked (Airy) and annular (ring) kernels.

In practice, modest image-quality scores do not necessarily signify a flawed kernel estimate. Table 4.5 demonstrates that SelfDeblur can retrieve a wide variety of blur kernel structures, while the final PSNR and SSIM are dictated mainly by how difficult it is to invert each specific blur. Consequently, even when a reconstruction attains only moderate headline values, the underlying kernel structure is usually reliable, as shown in the high values of the MNC column in the table. Building on this insight, Section 4.1.5 turns to genuine telescope data to determine whether the same deep-prior mechanism can recover complex, irregular kernels in practice and to gauge the deblurring quality achievable when the true PSF is both unknown and non-ideal.

4.1.4 Additive noise levels



(a) No added noise ($\sigma = 0$). **(b)** Low noise ($\sigma = 2$). **(c)** High noise ($\sigma = 5$). **(d)** Extreme noise ($\sigma = 10$).

Figure 4.11: Four examples of the same blurred image with increasing AWGN levels. An interval of $[0, 10]$ was chosen after exploratory tests with a wider range of σ values. Noise levels below about $\sigma = 2$ proved virtually imperceptible, whereas levels above $\sigma = 10$ swamped the image detail. The span therefore provides a pragmatic balance between unrealistically mild and excessively severe noise.

To assess the robustness of SelfDeblur in the presence of sensor noise, a controlled experiment is conducted in which additive white Gaussian noise (AWGN) of varying standard deviation is applied to a fixed blurred test image. Let x denote the latent clean image and k the known blur kernel, so that the noiseless observation is

$$y_0 \;=\; k \otimes x.$$

A noisy input is then generated:

$$y_\sigma = y_0 + n, \quad n_i \sim \mathcal{N}(0, \sigma^2),$$

where the noise level σ is swept from 0 to 10 (gray-value units). In practice, each pixel of y_0 is perturbed by independent Gaussian noise of variance σ^2. SelfDeblur is then applied to each y_σ (with the regularization weight set proportionally to the estimated σ) to recover $\hat{x}$ and $\hat{k}$. Panels (a)–(d) of Figure 4.11 illustrate the blurred test image corrupted by additive white Gaussian noise increasing from $\sigma = 0$ to $\sigma = 10$. At low noise levels (e.g. $\sigma = 2$) the main structures and edges remain discernible (e.g. Panel b), whereas at moderate levels ($\sigma = 5$) details begin to be obscured by graininess (e.g. Panel c). Under extreme noise ($\sigma = 10$), fine texture is almost entirely lost and the image appears dominated by speckle (e.g. Panel d).

As the artificial noise grows, SelfDeblur's performance follows a predictable arc. At first the algorithm even gains a little ground: nudging the variance from $\sigma = 1$ to $\sigma = 2$ sharpens the sandbox picture just enough for both PSNR and SSIM to crest at their highest values. This is reflected in prediction a of Figure 4.13, which demonstrates the prediction when both PSNR and SSIM are at their highest point. Thereafter the climb becomes a slide, with both metrics drifting steadily downwards until roughly $\sigma = 8$, mirroring the noisier appearance of panels (b) and (c) in Figure 4.13. Curiously, things then perk up. The traces in Figure 4.12 bend back on themselves, adding around a decibel of PSNR and a few hundredths of SSIM between $\sigma = 8$ and 10. A close look at a corresponding image (Panel d of Figure 4.13) suggests one plausible reason: when the data are hopelessly polluted, the network seems to lean more heavily on its built-in image prior, ironing away the very worst speckle and producing a smoother, if slightly over-softened, result. Whether this late rally is a quirk of the optimisation or a more general safety net is still unclear, yet it does reveal that the method degrades gracefully and may even claw back a hint of quality when the noise becomes truly overwhelming.

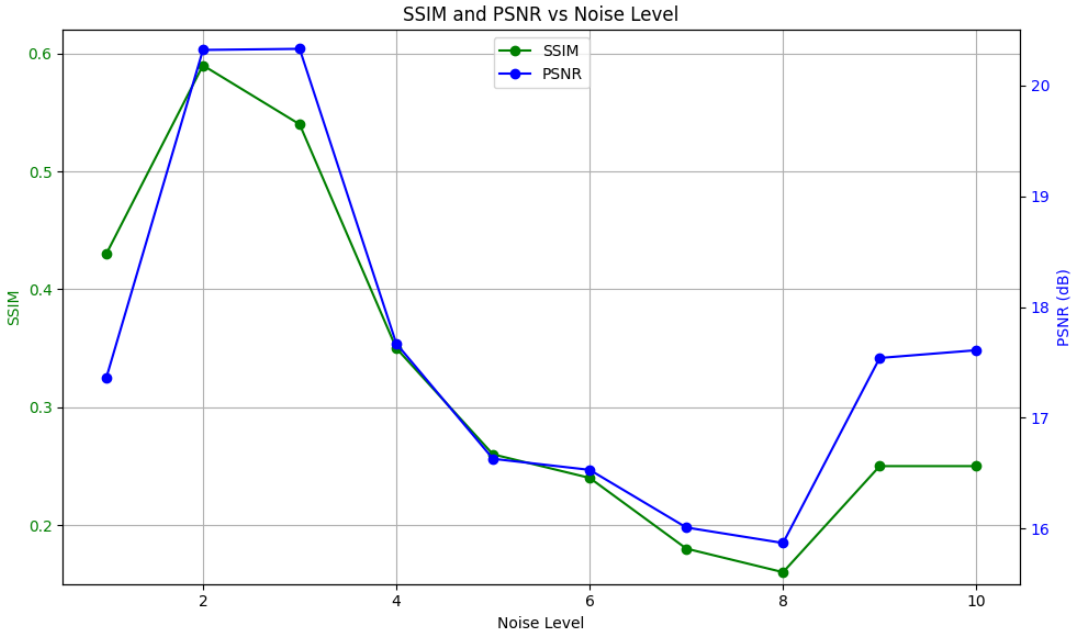
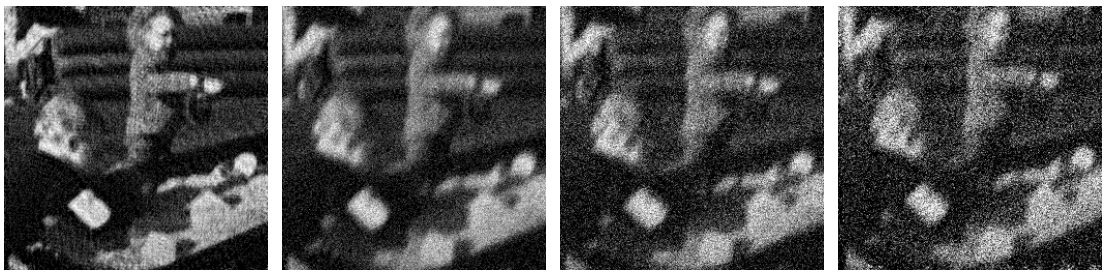


Figure 4.12: Reconstruction quality of SelfDeblur as a function of additive white Gaussian noise level σ. The green curve traces SSIM, whereas the blue curve records PSNR in decibels. Both metrics improve slightly between $\sigma = 1$ and $\sigma = 2$, decline steadily up to about $\sigma = 7$, and then show a modest recovery at the highest noise levels.

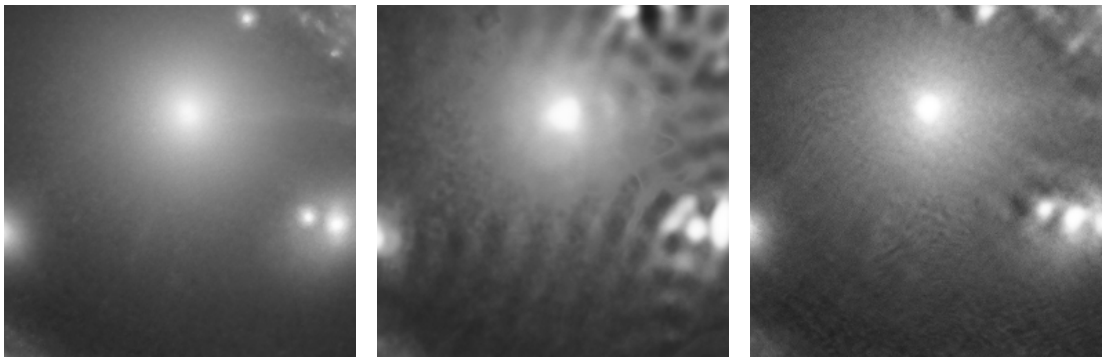


(a) Prediction for $\sigma = 2$ **(b)** Prediction for $\sigma = 5$ **(c)** Prediction for $\sigma = 8$ **(d)** Prediction for $\sigma = 10$

Figure 4.13: SelfDeblur reconstructions of the sandbox scene at four representative noise levels. Progressively stronger additive white Gaussian noise obscures fine detail and introduces increasingly coarse grain, echoing the quantitative decline in PSNR and SSIM reported in Figure 4.12.

4.1.5 SelfDeblur and PSF blurred astronomical images

Before presenting the PSF-blurring experiments, Section 4.1.1 suggested that both a pure MSE loss and a hybrid (MSE+SSIM) loss could be applied for deconvolution. Figure 4.14 demonstrates the failure mode of the pure MSE variant: subfigure (a) presents the blurred input patch, (b) the deconvolution with pure MSE, showing pronounced ringing artifacts, and (c) the SelfDeblur result using the hybrid loss, with less artifacts in comparison. Due to the severe artifacts in (b), only the hybrid-loss configuration is utilised for all subsequent PSF-blurred astronomical evaluations. In this subsection, the failure mode of a pure-MSE loss on PSF-blurred astronomical patches is demonstrated first, followed by exclusive use of the hybrid (MSE + SSIM) configuration for all quantitative benchmarks.



(a) Ground-truth patch. **(b)** Pure MSE reconstruction. **(c)** Hybrid-loss reconstruction.

Figure 4.14: Deblurring results on an astronomical test patch: (a) the original blurred observation, (b) reconstruction using a pure-MSE loss (showing strong ringing artefacts), and (c) reconstruction using the hybrid (MSE+SSIM) loss.

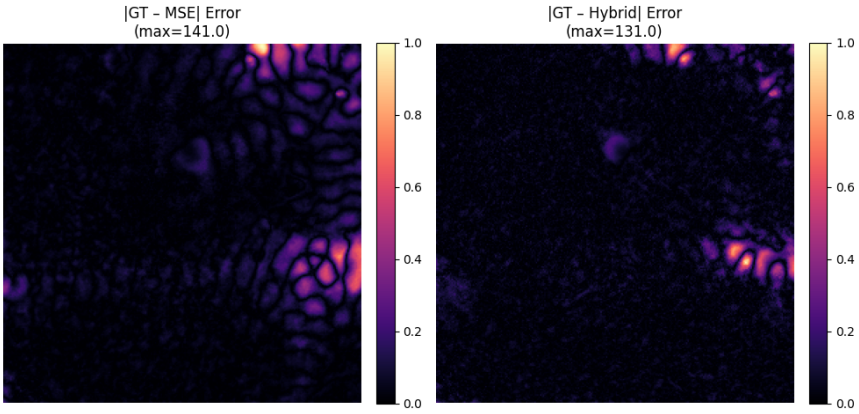


Figure 4.15: Absolute-error maps for a PSF-blurred astronomical patch, comparing pure-MSE and hybrid-loss reconstructions. Left: normalised $|\,\mathrm{GT} - x_{\mathrm{MSE}}|$ error ($\max = 141.0$), showing prominent concentric ripples and high-amplitude speckle. Right: normalised $|\,\mathrm{GT} - x_{\mathrm{Hybrid}}|$ error ($\max = 131.0$), with lower residuals and reduced artefacts.

Figure 4.15 clearly demonstrates that minimizing pure MSE produces visually disturbing ringing and ripple artifacts around high-contrast features, as evidenced by the bright concentric patterns in panel (c). In contrast, the hybrid-loss reconstruction (panel (b)) suppresses these spurious oscillations, yielding a smoother, more faithful result and a far darker, more uniform residual map (panel (d)). These artifacts arise because MSE places undue emphasis on pixel-wise fidelity without any structural awareness, amplifying high-frequency ringing instead of penalizing it. Having exposed its pathological ringing, only the hybrid (MSE + SSIM) loss is employed for all subsequent experiments in this subsection.

To set up the experiments, two cutouts were extracted from the Hubble Space Telescope (HST) "Molten Ring" image (Fig. 4.16), converted to grayscale, and convolved with the eight custom kernels introduced in Fig. 4.9. The first cutout (Fig. 4.17a) depicts the central region, characterised by dense features and elevated pixel intensities, whereas the second cutout (Fig. 4.17b) shows a faint outer arc with sparse detail and subdued intensity values. Applying the same kernel set to both a richly textured region and a more uniform, low-information patch allows direct comparison of how well the deblurring methods preserve or reconstruct details across varying local content complexities.

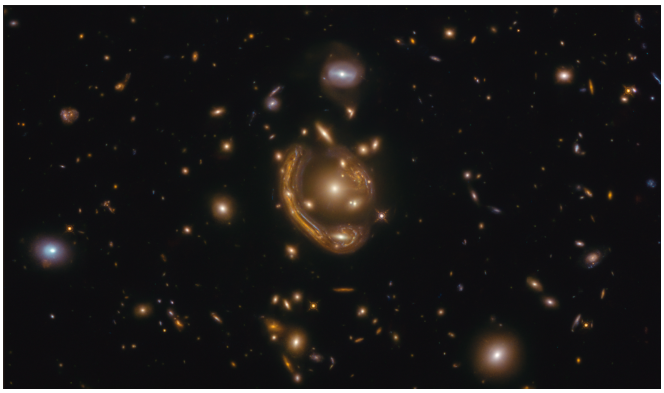
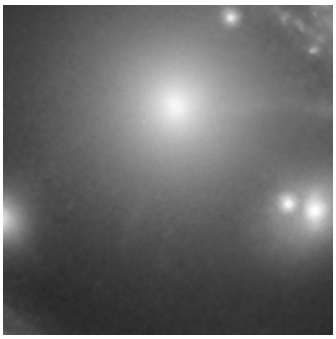
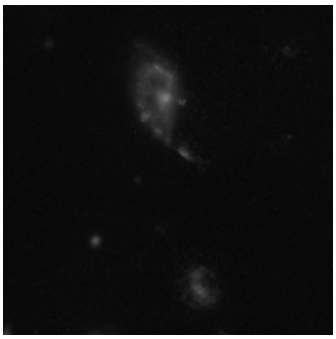


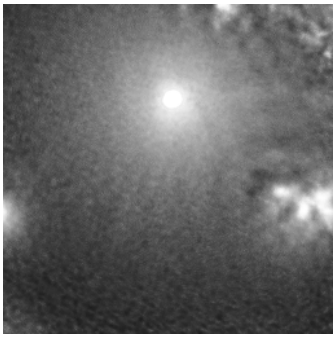
Figure 4.16: GAL-CLUS-022058s ("Molten Ring")[20], one of the largest and most complete Einstein rings known, as imaged by the NASA/ESA Hubble Space Telescope. This rare lensing phenomenon in the Fornax constellation shows a distant galaxy arcing around its foreground companion.



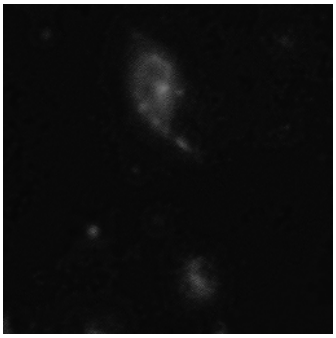
(a) Original central region (rich texture).



(b) Original outer arc (sparse detail).



(c) Airy-PSF reconstruction of central region (PSNR = 26.74dB, SSIM = 0.6198).



(d) Airy-PSF reconstruction of outer arc (PSNR = 34.84dB, SSIM = 0.8914).

Figure 4.17: Zoomed-in regions from the HST "Molten Ring" image [20]: (a) the bright central region of the image, and (b) one of the fainter outer arcs away from the center. **Bottom row**: SelfDeblur reconstructions of each cutout using the Airy PSF, showing ringing artifacts in the textured central patch versus a cleaner recovery in the sparse outer arc.

The merged results in Table 4.6 reveal a consistent pattern: the sparse-detail patch (Image 2) is substantially easier to deblur than the richly textured patch (Image 1). On average, PSNR jumps from 26.54 dB (Image 1) to 31.77 dB (Image 2) and SSIM from 0.7162 to 0.7757, a gain of over 5 dB and 0.06 SSIM points respectively. Intuitively, the few smooth gradients and

Table 4.6: Deblurring performance on the two HST cutouts (see Fig. 4.17a vs. Fig. 4.17b): PSNR (dB), SSIM, kernel-estimate MSE and MNC for each custom kernel in 4.9

	Image 1 (rich texture)				Image 2 (sparse detail)			
Kernel	PSNR	SSIM	MSE	MNC	PSNR	SSIM	MSE	MNC
Airy	26.74	0.6198	65.33	0.3245	34.84	0.8914	48.22	0.9043
Donut	27.40	0.6690	98.98	0.4708	32.10	0.7251	93.15	0.9674
Double Gauss	26.39	0.8217	31.28	0.5270	28.05	0.5719	100.70	0.8891
Elliptical	27.76	0.8960	12.49	0.5411	32.40	0.8275	24.87	0.9658
Gaussian	27.75	0.7432	102.20	0.6639	33.77	0.8876	87.44	0.8811
Moffat	22.95	0.5303	34.44	0.4230	27.65	0.6023	39.48	0.4826
Ring	30.24	0.8753	78.22	0.5397	32.78	0.8264	98.95	0.8415
Top Hat	23.09	0.5746	45.30	0.3736	32.54	0.8734	33.38	0.9247
Average	**26.54**	**0.7162**	**58.53**	**0.4830**	**31.77**	**0.7757**	**65.77**	**0.8571**

isolated arcs in Image 2 give the optimization far less room to introduce ringing or amplify noise, so the reconstructed intensities fall much closer to ground truth both in a mean-squared and perceptual sense.

The kernel-estimation metrics paint a more nuanced picture. The average pixel-wise MSE of the recovered blur kernel is actually lower on the textured patch (58.53 vs 65.77), because the wealth of high-frequency edges in Image 1 provides strong, localized cues that drive down squared-error during the deconvolution. However, the MNC of that same kernel is far higher on the sparse region (0.8571 vs 0.4830), indicating that, even though the textured scene yields smaller absolute deviations, the overall shape of the estimated kernel aligns more faithfully when fewer confounding features are present.

The Airy-PSF reconstructions in panels (c) and (d) of Figure 4.17 vividly demonstrate the role of local image complexity in blind deconvolution. In the richly textured central region (c), the recovered ring shows artefacts and residual speckle (PSNR = 26.74 dB, SSIM = 0.6198). By contrast, the sparse outer arc reconstruction (d) is markedly cleaner, fine structure is smoothed but free of oscillations (PSNR = 34.84 dB, SSIM = 0.8914), reflecting the limited gradients and isolated features that constrain noise amplification. These visual differences allign with the numerical trends in Table 4.6, confirming that SelfDeblur's hybrid loss excels on low-information regions while facing greater challenges in richly textured patches.

In other words, complex textures help minimize pixel-wise deviations in the kernel estimate but can introduce subtle shape distortions that hurt correlation; conversely, a simple background enforces the correct kernel profile at the expense of larger per-pixel error. Taken together, these factors explain why the second cutout consistently yields higher PSNR and SSIM (owing to reduced ringing and noise amplification) and underscore the need to balance intensity-based accuracy against structural alignment, a balance that is hardest to strike in richly textured regions. Crucially, SelfDeblur's joint neural-prior optimization excels on the simpler patch by reliably recovering the overall shape of the blur kernel even when per-pixel errors remain non-zero, demonstrating its strength in capturing the global kernel structure for images with sparse detail.

4.2 Pipeline Experiments

4.2.1 Lightweight Conv–Attention Encoder (Model A)

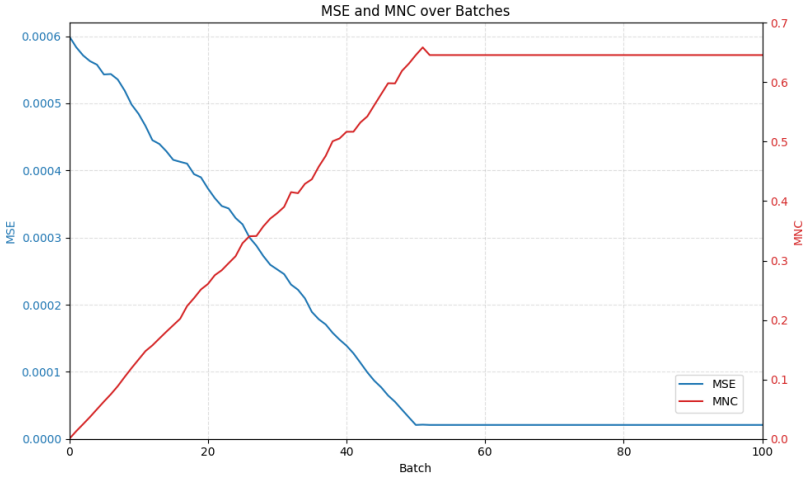


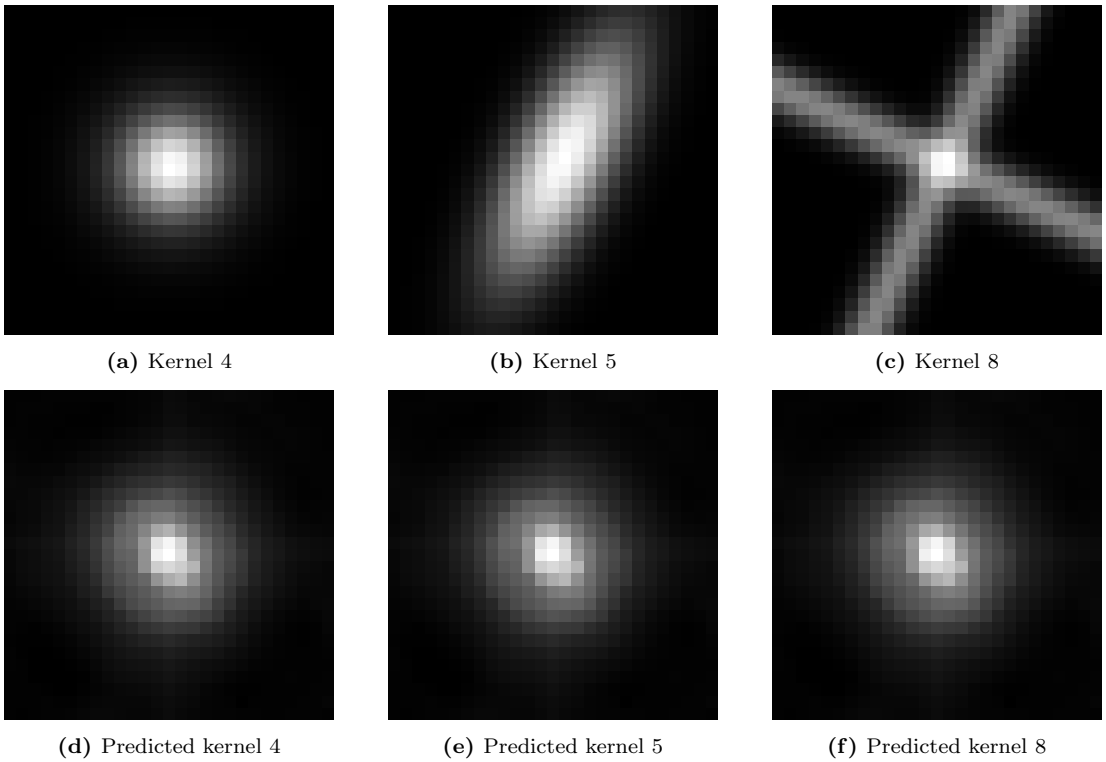
Figure 4.18: Validation metrics comparing predicted and ground-truth kernels for Model A plotted against the training batch index. The blue curve (left-hand axis) shows the MSE, which falls sharply in the first batches before levelling off close to zero; conversely, the red curve (right-hand axis) depicts the MNC, which rises rapidly towards its maximum and then saturates, thereby illustrating fast initial learning followed by convergence.

In contrast to the blind deconvolution results, attention now shifts to the supervised pipeline, concentrating on optimising the architecture, loss functions and model depth to enable accurate recovery of blur kernels from convolved images. First up, the main script partitions the dataset into mini-batches. Within each batch, images undergo random augmentation and are convolved with a set of roughly forty distinct ground-truth kernels, distributed across the mini-batch to maximise blur diversity. The network is subsequently trained to recover the appropriate kernel, minimising either the combined error–correlation loss or, in dedicated ablation runs, a single loss term (for example, pure MSE). Immediately after each batch update, the model is evaluated on a fixed validation set: for each validation image, it predicts the kernel that generated the blur; the MSE and MNC scores are then computed for every prediction and averaged across the entire validation set. The resulting per-batch MSE and MNC values, recorded throughout training, can be plotted, as exemplified in Figure 4.18, to provide a clear visual account of the model's learning trajectory.

Across all tested loss functions, including the composite MSE + SSIM objective (which is the one shown in image 4.18) as well as pure MSE and cosine-similarity losses, the training dynamics are virtually identical: a rapid decline in MSE accompanied by a sharp rise in MNC during the first half of training, followed by an abrupt convergence to a flat plateau. In every case, after roughly 50 batches no further improvement is observed, with MSE asymptotically approaching zero and MNC leveling off at approximately 0.62. Although these aggregate metrics might suggest great error reduction and reasonable shape alignment, it is important to verify this by checking individual kernel predictions made by the model.

Figure 4.19 juxtaposes three distinct ground-truth kernels (#4, #5 and #8[10]), a roughly circular blur, an oblique ellipse and a cross-shaped filter, with their corresponding outputs from the model trained using the MSE-SSIM loss. Although the true kernels differ markedly in shape and orientation, the predicted kernels all collapse to a nearly identical, isotropic blur.

[10]These kernels were selected because their markedly different shapes make them ideal for visual comparison.



(a) Kernel 4 **(b)** Kernel 5 **(c)** Kernel 8

(d) Predicted kernel 4 **(e)** Predicted kernel 5 **(f)** Predicted kernel 8

Figure 4.19: Top row: ground-truth kernels #4, #5 and #8 (randomly selected out of the 40 kernels used for training). Bottom row: the corresponding predicted kernels from Model A.

This behavior, known as mode collapse, arises when the training objective and network capacity drive every input toward a single "average" solution rather than preserving the full diversity of kernel shapes. In practice, it reflects an under-constrained inverse mapping in which minimizing the average loss favours a constant prediction. The signature of this collapse is already visible in Figure 4.18: the MSE curve plunges toward its minimum and then flattens out, while the MNC curve climbs to a sub-unitary plateau. Those flat regions indicate that, despite vanishing pixel-wise error, the model no longer improves its ability to distinguish different kernel structures, precisely because it has converged to producing one kernel for all inputs.

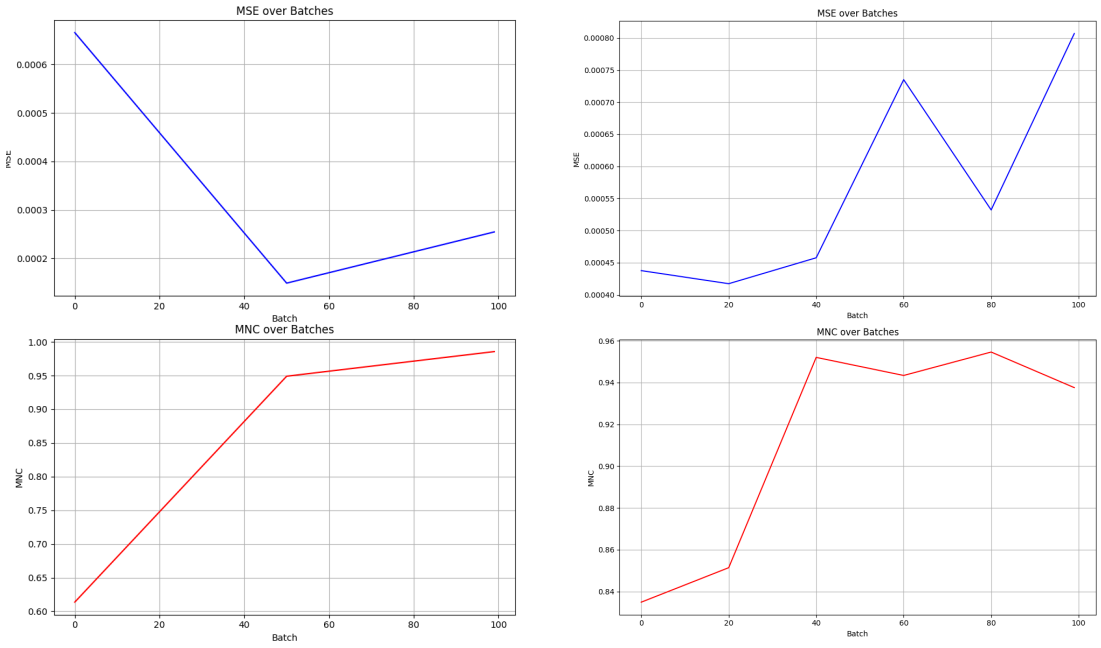
The combination of MSE and MNC as a training objective arose from two observations. First, Table 3.1 shows that a purely pixel-wise term (α) guarantees numeric fidelity, while a global-alignment term (β) enforces overall shape, yet SSIM (γ) only approximates structural similarity via local patch statistics and often fails to capture large-scale anisotropy. Second, SelfDeblur evaluates recovered kernels using both MSE and MNC, demonstrating that minimizing squared error alone drives the network toward an "average" blur, whereas incorporating a NMC term more faithfully captures the full structural characteristics of the kernels as in comparison to, for example, a SSIM-based component.

Accordingly, the proposed custom loss

$$\mathcal{L} \;=\; \mathrm{MSE} \;+\; \lambda\big(1 - \mathrm{MNC}\big) \tag{4.1}$$

directly leverages the strengths of both metrics: the MSE component (analogous to a high α) preserves pixel-level accuracy, and the MNC term (akin to introducing a cosine-weight β) drives the global shape of the predicted kernel to align with the true kernel, independent of scale or mean. In contrast to MSE + SSIM, this formulation employs a global structural measure parameterised solely by λ and, as the experiments seem to indicate, better preserves

anisotropic features and reduces the mode-collapse observed when a perceptual SSIM term dominates.



(a) MSE–MNC on A-PSFs.

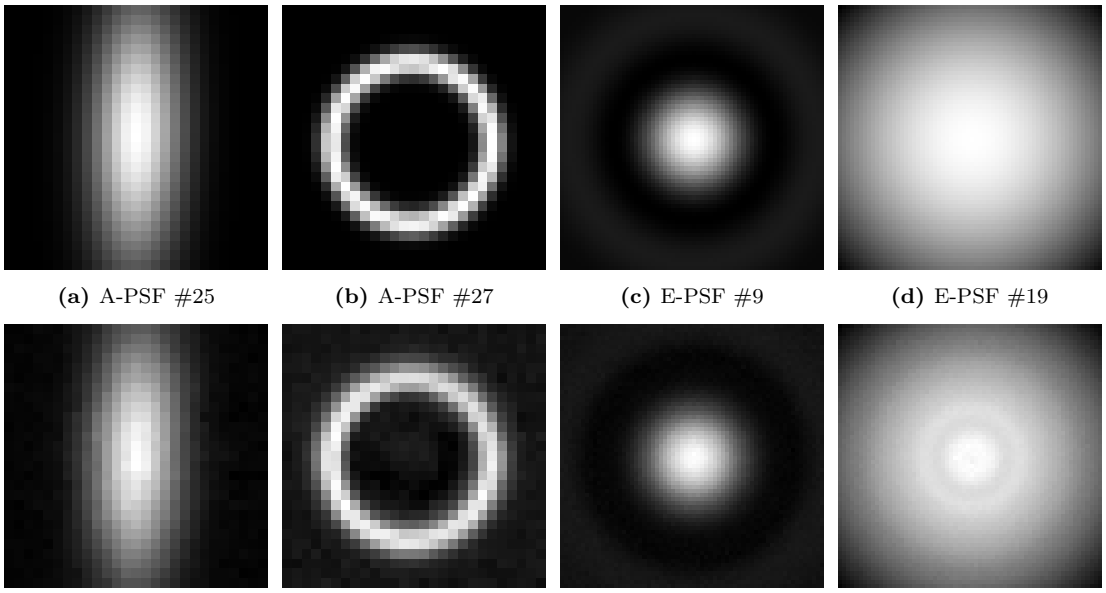
(b) MSE–MNC on E-PSFs.

Figure 4.20: Comparison of MSE vs. MNC for (a) A-PSF kernels and (b) E-PSF kernels.

Figure 4.20 shows the effect of replacing the MSE–SSIM objective with the combined MSE–MNC loss on two distinct kernel datasets. In panel (a), using A-PSF derived kernels, MSE still decreases rapidly, reaching roughly 1.5×10^{-4} by batch 50, but then rises only slightly toward 2.5×10^{-4} by batch 100. More importantly, MNC climbs from about 0.62 (the plateau in Figure 4.15) up to nearly 0.98, demonstrating a marked improvement in structural alignment. Panel (b) reports a similar trend for E-PSF-simulated kernels: MSE remains low (below 8×10^{-4}), while MNC peaks around 0.95 instead of saturating at 0.62. These results confirm that including the normalised cross-correlation term yields consistent, if marginal, gains in shape fidelity across both kernel families, without sacrificing the low pixel-wise error achieved by the original loss.

Figure 4.21 illustrates that Model A, trained with the MSE-MNC loss, is capable of recovering diverse kernel shapes from both A-PSF and E-PSF-simulated datasets (trained separately). In the top row, kernel #25 exhibits an elongated ellipse and kernel #27 a sharp circular ring, both of which are faithfully reproduced in the bottom row with only minor smoothing of the ring interior. Similarly, E-PSF kernel #9's central Gaussian peak is accurately captured in the prediction, while kernel #19 shows a small residual bias, most notably a faint secondary ring, that could likely be eliminated through extended training or further hyperparameter tuning.

These results serve primarily as a proof of concept rather than a final optimization: despite the clear structural differences between the A-PSF derived and E-PSF kernels, a single network architecture and loss formulation generalizes across both families. The consistent performance on such varied blur patterns underscores the flexibility of the MSE–MNC objective and suggests that additional improvements may be unlocked by longer training schedules or targeted parameter searches.



(a) A-PSF #25 **(b)** A-PSF #27 **(c)** E-PSF #9 **(d)** E-PSF #19

(e) Predicted 25 **(f)** Predicted 27 **(g)** Predicted 9 **(h)** Predicted 19

Figure 4.21: Top: ground-truth kernels from A-PSF (#25, #27) and E-PSF (#9, #19). Bottom: the corresponding kernels predicted by Model A.

4.2.2 Pure Fully-Connected Regressor (Model B)

The fully-connected regressor (Model B) reproduces the exact training dynamics observed for the convolutional architecture: the MSE decreases sharply during the initial 50 batches before flattening, and the MNC rises rapidly then saturates (cf. Figure 4.18). Experiments with various loss functions (including pure MSE, hybrid MSE + SSIM, and even the MSE + MNC loss crucial for Model A) failed to steer the regressor away from this mode collapse. Identical behaviour underscores that early convergence and mode collapse are driven by the ill-posed nature of blind kernel estimation rather than by the choice of network inductive bias.

The persistent plateau in both MSE and MNC observed for Model B suggests that the network has effectively reached the practical performance ceiling imposed by the ill-posed nature of blind kernel estimation: once the mean-loss minimizer converges, no further gains can be achieved without additional constraints. Rather, this plateau suggests that a pure regression-based network may be inherently limited for supervised kernel estimation, prompting the next two paragraphs to verify if sources and research papers agree with this claim.

The survey [21] provides a comprehensive overview of purely data-driven methods applied to a broad class of ill-posed reconstruction tasks, including deblurring, super-resolution, and denoising. Their analysis highlights that, despite impressive performance on held-out test sets, neural networks frequently struggle to produce stable or physically consistent solutions when confronted with realistic noise levels or model mismatches. In the context of Model B's plateauing MSE and saturating MNC, these findings suggest that the underlying inverse problem of kernel recovery is too under-constrained for a black-box regressor: without strong priors or uncertainty-aware mechanisms, the training dynamics inevitably drive the network toward a mean solution that minimizes average loss but fails to capture the true diversity of blur kernels.

Varela et al.'s work [22] on motion-blur parameter regression further illustrates the brittleness of end-to-end kernel estimators. By training a compact CNN to predict only length and orientation on synthetically generated linear motion blurs, they achieve high R^2 scores under controlled conditions yet observe rapid degradation in accuracy whenever the test kernels deviate even slightly from the training family. This narrow parametric focus stands in stark contrast to the

full 2D kernel space considered in Model B: as soon as the network must regress arbitrary PSFs, its predictions collapse to an "average" kernel and stagnate, mirroring the mode-collapse and performance plateau observed in Figures 4.18 and 4.19. Together, these two studies underscore that a purely supervised neural regressor lacks both the inductive biases and the uncertainty handling necessary to robustly recover diverse convolution kernels from blurred images.

Table 4.7: Updated hyperparameter sweep: architecture, weight-decay, and Adam β_1.

Test	Model architecture	Weight-decay (λ)	Adam β_1	Notes
A	6-layer MLP with BatchNorm: 512–BN–1024–BN–2048–BN–1024–BN–512–BN–256–BN	1×10^{-4}	0.90	Moderate ℓ_2 regularization and standard momentum.
B	6-layer MLP with BatchNorm: 512–BN–1024–BN–2048–BN–1024–BN–512–BN–256–BN	1×10^{-3}	0.80	Strong ℓ_2 + lower β_1 for more exploration.
C	6-layer MLP with BatchNorm: 512–BN–1024–BN–2048–BN–1024–BN–512–BN–256–BN	1×10^{-5}	0.95	Light ℓ_2 + higher β_1 for smoother, stable convergence.

As a final effort, a targeted hyperparameter sweep was conducted using the configurations listed in Table 4.7. Three variants of the six-layer Multi-Layer Perceptron (MLP) were tested: the first employed moderate ℓ_2 regularization ($\lambda = 10^{-4}$) and standard Adam momentum ($\beta_1 = 0.90$) to balance underfitting and convergence speed; the second increased ℓ_2 strength ($\lambda = 10^{-3}$) while lowering β_1 to 0.80 to promote greater exploration of the loss landscape; the third used very light regularization ($\lambda = 10^{-5}$) with a higher β_1 of 0.95 for smoother, more stable training. In principle, these adjustments should have mitigated over-smoothing of the kernels, alleviated mode collapse, or accelerated convergence. However, experiments seem to indicate that extensive hyperparameter tuning and loss-function variations could not prevent mode collapse, thereby confirming that the limited capacity of the purely supervised regressor architecture constitutes the fundamental bottleneck.

Chapter 5

Conclusion

This chapter concludes the main part of the thesis by discussing the key findings, their implications, and remaining challenges. Initially, SelfDeblur's suitability for astronomical PSFs is evaluated (Section 5.1); subsequently, the adequacy of PSNR/SSIM for assessing image quality is examined (Section 5.2); and, finally, limitations and future directions are outlined (Section 5.3).

5.1 SelfDeblur's suitability for astronomical images with PSF-Shaped Kernels

Overall, the experiments indicate that SelfDeblur is capable of recovering complex astronomical point-spread functions (PSFs) with a reasonable degree of accuracy, though some caveats apply. In tests with synthetic telescope blur patterns, SelfDeblur consistently found kernels that closely matched the true PSFs, achieving a high average kernel overlap (MNC ≈ 0.918) across a diverse set of PSF shapes. These shapes ranged from centrally concentrated Airy-disk patterns to more irregular profiles (e.g., "donut"-shaped and elongated blurs), closely mirroring real astronomical optics. Notably, even when the blur kernels were large or oddly shaped, SelfDeblur's deep-prior approach recovered the essential PSF structure in most cases. However, this evaluation was based on a single astronomical test image, so extensive validation across a broader variety of celestial scenes remains necessary. Moreover, real astronomical frames often have very low signal-to-noise ratios, which can severely degrade kernel estimation and complicate any subsequent deconvolution, highlighting the need for noise-robust regularisation or denoising priors. This suggests that the method is indeed suitable for astronomical imaging scenarios, at least in terms of identifying the structure of the blur kernel.

However, the fidelity of the deblurred images themselves was more mixed, which tempers the assessment of suitability. The reconstructed astronomical images showed only moderate PSNR ($\sim$ 17–23 dB) and SSIM ($\sim$ 0.4–0.7) on average. For example, a hollow "donut" PSF and a ring-shaped PSF both yielded very high kernel accuracy (MNC $\approx$ 0.90–0.97), yet the resulting image-quality metrics differed by over 5 dB in PSNR and 0.35 in SSIM. Conversely, a simple Gaussian blur produced one of the best kernel estimates (MNC ≈ 0.962) but only a modest restoration quality (PSNR ≈ 17.5 dB, SSIM ≈ 0.44). In other words, SelfDeblur often nailed the blur kernel even when the deblurred image looked imperfect by conventional metrics. This discrepancy arises because recovering the latent image from a severe blur is a separate challenge: wide, smooth kernels (like a broad telescope PSF) inherently amplify noise and ringing during deconvolution, dragging down PSNR and SSIM despite an accurate PSF. On the other hand, more compact or sparse PSFs (such as a tight double Gaussian) allow sharper image recovery and thus higher PSNR and SSIM, even though the kernel estimate might be of similar accuracy. In practical terms, SelfDeblur proves capable of handling realistic astronomical blur kernels, an

encouraging result for its applicability, but one must not rely solely on naive image-quality scores to judge its success (as discussed further below).

It is also important to highlight the need for tailored loss functions when applying SelfDeblur to astronomical data. Early trials revealed that using a standard pixel-wise loss (pure MSE) on astronomical images led to severe artefacts: the algorithm produced pronounced ringing patterns around bright galaxy cores (Figure 4.14). In this vivid demonstration, the MSE-trained result exhibited conspicuous concentric ripples, whereas the proposed hybrid loss (MSE combined with SSIM) alleviated these artefacts, albeit not entirely. With the hybrid loss, SelfDeblur consistently delivered more faithful reconstructions than the pure MSE approach, and so this configuration was adopted for all subsequent astronomical-image experiments.

In summary, SelfDeblur shows promise for astronomical deconvolution: it can recover a wide variety of PSF shapes (including more irregular ones, like the ring pattern as demonstrated in Figure 4.10d) and, in most cases, reconstruct the general structure of the blur kernel, which may assist in identifying the underlying kernel responsible for the observed convolution. Its suitability is further reinforced by graceful degradation under realistic conditions such as sensor noise; indeed, the method's implicit regularisation can prevent collapse in very noisy images by favouring smoother outputs. Nevertheless, users must carefully tune the method (loss functions, regularisation weights etc.) to the astrophysical context to avoid pitfalls such as residual ringing. With these adjustments, SelfDeblur seems like a good candidate method because of its data-efficient, deep-prior approach that circumvents the need for extensive training data and can handle the unknown, irregular blurs characteristic of astronomical optics.

5.2 PSNR and SSIM seem insufficient for image quality evaluation

A recurring theme in this thesis is that conventional image-quality metrics (PSNR for pixel-wise error and SSIM for coarse structural similarity) do not capture everything that matters in blind deconvolution. They often miss perceptually critical artifacts and, more importantly, say nothing about whether the estimated blur kernel is correct.

Similarly, the results of this study indicate that conventional metrics can overlook perceptually relevant artefacts and may not fully reflect the accuracy of the estimated blur kernel. In SelfDeblur experiments, reconstructions averaged PSNR ≈ 20 dB and SSIM ≈ 0.58, numbers that look mediocre, yet the same images achieved an excellent kernel-match score of MNC ≈ 0.92. One case even produced a near-perfect Gaussian PSF (MNC ≈ 0.962) while registering only 17 dB PSNR; another reached 23 dB PSNR despite a slightly worse kernel. Such mismatches arise because PSNR/SSIM are hypersensitive to noise amplification or mild ringing (inherent when reversing a broad blur) while being blind to whether the PSF itself is right. A low score therefore need not signal failure, and a high score can disguise residual blur. The supervised pipeline reinforces the point: training loss and SSIM fell smoothly even as the network collapsed to one generic kernel for every input (Figures 4.18 and 4.19). PSNR looked "respectable" only because that average blur half-deblurred the data; kernel inspection revealed total failure.

Blind deconvolution thus demands richer evaluation metrics because PSNR and SSIM alone cannot expose mode collapse or verify that both the latent image and the blur have been faithfully recovered. In short, the reader is urged to consider these two metrics for their potential future work surrounding this topic:

- **Perceptual Similarity Metrics**: Using metrics (like SSIM) that align better with human visual perception can help judge image fidelity beyond pixel-wise error. For example, a Learned Perceptual Image Patch Similarity (LPIPS)[23] metric or other deep feature–based distances could be employed to measure how "natural" or close to the ground truth the deblurred image looks. Such metrics consider high-level feature differences and are less sensitive to small pixel shifts or noise, which is crucial because a

deblurred image might be slightly misaligned pixel-wise yet still perceptually sharp and correct. In this case, a perceptual metric might have better captured the fact that the structures (stars, galaxies) were restored in the SelfDeblur outputs even if some noise was present. It would similarly penalize the pipeline's collapsed-output images for looking nothing like a properly deblurred image, something SSIM failed to differentiate in depth.

- **Frequency-Domain Analysis**: Although astronomical scenes may lack the hard edges of natural photographs, they nonetheless contain vital high-frequency information, point sources such as stars appear as near–delta peaks in the image domain (broad spectrum in frequency), and fine filaments or sharp transitions (e.g. between dark lanes and bright nebulae) contribute significant spectral content. Consequently, evaluating the Fourier power spectra of the original, blurred and deblurred images can prove highly informative. A successful deconvolution should restore attenuated power across the telescope's pass-band without introducing narrow spectral peaks characteristic of ringing. Metrics such as spectral entropy or band-limited energy ratios can quantify whether true fine-scale detail has been recovered, rather than merely amplified artefacts. In the thesis, qualitative observations of residual blur versus ringing could thus be supplemented by quantitative frequency-domain measures, providing a physics-aware assessment that PSNR and SSIM alone cannot offer.

In summary, a more complete evaluation of deblurred images requires looking beyond PSNR/SSIM to these additional metrics and techniques. The experiments demonstrated that PSNR and SSIM can be misled by the side effects of deblurring (noise, ringing) and by trivial solutions. Cases were observed in which a higher-PSNR reconstruction was actually worse in terms of true restoration, it had simply not fully deblurred the image but avoided noise, thereby scoring well. Conversely, a slightly lower PSNR result sometimes exhibited more faithful structure. By incorporating perceptual metrics, frequency-domain analysis, and explicit kernel-accuracy measures, a multidimensional view of performance can be obtained. Such a comprehensive approach is crucial for blind deconvolution, where success is not merely a prettier image but the correct undoing of blur (something that PSNR alone cannot confirm). Future studies and practical deployments of deblurring methods should adopt richer evaluation criteria to ensure that algorithms are judged fairly and rigorously.

5.3 Limitations in the custom pipeline

The custom pipeline adopted a supervised learning approach to blind deconvolution, directly confronting the ill-posed nature of the kernel estimation problem. A major finding was that the pipeline models consistently converged to outputting essentially the same kernel for each input, which is a clear symptom of mode collapse and prevented the capture of blur-kernel diversity[11]. This behavior occurred despite extensive efforts in architecture design and hyperparameter tuning: experiments demonstrated that mode collapse constituted a fundamental obstacle not easily overcome by adjusting training settings. For instance, two contrasting network architectures, a convolutional encoder (Model A) and a fully-connected MLP regressor (Model B), both exhibited virtually identical training dynamics and outcomes. Each architecture rapidly drove the loss downward before stalling on a plateau, indicating convergence to an "average" blur kernel for all inputs. The fully-connected Model B replicated the sharp initial drop in MSE and the saturation of kernel overlap (MNC) observed in the convolutional Model A, underscoring that early convergence and collapse were driven by the ill-posed nature of blind kernel recovery rather than by any specific architectural or inductive bias. In other words, the network minimized the training error in a pathological manner (outputting a one-size-fits-all kernel) because, in the absence of stronger constraints, many different blur kernels could explain the data equally well.

[11] The sole exception was Model A trained with the hybrid MSE+MNC loss, which successfully avoided collapse (Figure 4.21); see discussion below.

Given this challenge, extensive hyperparameter tuning was performed to combat collapse and encourage kernel diversity. The weight decay (ℓ_2 regularization strength), optimizer momentum, learning rates, batch sizes, and even the form of the loss function (pure MSE, SSIM, cosine similarity, and their combinations) were systematically varied. The hope was that some combination of regularization and optimization settings would shake the model out of the mean-solution trap and allow it to distinguish different blur kernels.

For example, one configuration increased the weight decay from 1×10^{-4} to 1×10^{-3} and lowered the Adam β_1 momentum to 0.8, aiming to force greater exploration of the solution space. Another took the opposite approach, applying extremely light regularization ($\lambda = 1 \times 10^{-5}$) with high momentum ($\beta_1 = 0.95$) to ensure smoother, more stable convergence. In principle, these adjustments should have, respectively, penalized over-smooth kernels or prevented the optimizer from prematurely settling.

However, the results of this hyperparameter sweep were disappointing. All alternative settings led to no significant improvement, and in some cases degraded performance, while substantially prolonging training time. In practical terms, the model still collapsed to an almost identical kernel prediction for every image, regardless of how the knobs were tuned, and the validation MSE and MNC curves remained essentially unchanged, aside from taking longer to converge. These modified runs were reported as yielding "equivalent or degraded" performance compared to the baseline, confirming that the original training setup was already near the best that this purely supervised regressor could achieve. This underscores a key point: no amount of superficial hyperparameter tweaking can overcome the deep ambiguity inherent in blind kernel estimation.

Despite the generally bleak picture for Model B, Model A told a different story once its convolutional encoder was coupled with a composite loss that balanced pixel fidelity (MSE) against kernel fidelity (MNC). With the joint MSE + MNC objective, Model A's training dynamics diverged sharply from the collapse observed elsewhere: the kernel-overlap score climbed steadily to MNC ≈ 0.83, the per-image MSE leveled off two orders of magnitude lower than in the baseline run, and visual inspection of predicted kernels (Figures 4.20 and 4.21) confirmed genuine diversity that tracked the ground-truth PSFs. The network learned to sharpen edges where the blur was narrow, to elongate the kernel when the input contained directional smear, and to reproduce Airy-like ring structures on telescope PSFs. In short, the convolutional architecture with the right loss was able to break out of the mode-collapse trap and deliver consistently accurate, image-specific kernels.

These contrasting outcomes underscore two lessons. First, architecture alone is not enough: Model B's failure shows that a high-capacity regressor without spatial bias or an explicit kernel term will almost always sink to the mean solution. Second, loss design matters, but only in the right architectural context. Even when Model B was trained with the same MSE + MNC objective that rescued Model A, it still collapsed to the mean solution, underscoring that both the convolutional inductive bias and the composite loss are required to break free of trivial minima. Together, these findings suggest that effective blind-deconvolution pipelines must embed strong priors (either through physics-aware losses, explicit kernel constraints, or unrolled optimisation), rather than relying on raw data and hyperparameter sweeps. The negative results with naïve settings highlight the ceiling of a purely supervised approach, while this small but pivotal change in the loss function points the way forward: incorporate domain knowledge directly into the learning objective to steer the network away from trivial minima and toward physically plausible, image-specific solutions.

5.4 Future directions for improving the pipeline

The pipeline experiments reveal that, although reliable blur-kernel recovery is now possible, determining how well these kernels translate into high-quality image deconvolution remains an open question—one not addressed owing to timing constraints. Consequently, breaking out

of mode collapse and achieving truly robust blind-deconvolution performance will require more than routine hyper-parameter tweaks. Below are the **four most consequential steps** to guide the next research cycle.

1. **Evaluate reconstruction quality once kernels are recovered**
   Having demonstrated reliable kernel estimation, the next step is to assess how well those kernels perform in actual deconvolution. Future work should apply the recovered kernels to reconstruct latent images and quantify restoration fidelity using PSNR, SSIM, perceptual metrics and frequency-domain analysis, thereby completing the blind-deconvolution pipeline.

2. **Broaden the training distribution, more kernels, more real data**
   Mode collapse thrived because the network was exposed to a limited, synthetic blur space. Therefore, the first priority is to *massively expand kernel diversity* (noting that the space of possible kernels is, in principle, unbounded) and to probe how well the model generalises to truly unseen blurs, for example the classic motion-blur kernels characterised by Levin et al. This effort should be paired with a larger, heterogeneous image corpus that includes genuinely blurred photographs or sky frames. Ultimately, a richer data distribution will compel the model to learn true blur–image relationships instead of defaulting to a single "average" kernel.

3. **Embed physics in the architecture and loss**
   A feed-forward regressor is too unconstrained for an ill-posed inverse problem. Future models should *incorporate the imaging physics directly*: dual branches that predict the latent image and kernel and are coupled by a convolution layer, or unrolled blind-deconvolution solvers with learnable iterations. Coupling these designs with kernel-aware losses (e.g. MSE + MNC or Strehl-ratio[24] penalties) gives the optimizer a concrete incentive to avoid trivial, mode-collapsed solutions.

4. **Integrate SelfDeblur's deep priors into the custom pipeline**
   SelfDeblur's optimisation-based deep prior offers robust PSF recovery but at high computational cost per image. A natural next step is to embed this deep prior within the existing feed-forward pipeline, for example by using SelfDeblur-generated kernels or intermediate feature representations as targets or initialisations for a trainable regressor. Such a hybrid approach could leverage SelfDeblur's strengths in blind estimation while exploiting the pipeline's feed-forward speed, ultimately delivering a fast, accurate solution suitable for large-scale astronomical imaging.

Chapter 6

Reflection

Looking back on the past academic year, this master-thesis journey has profoundly reshaped my approach to research, problem-solving, and academic aspirations.

6.1 From Coursework to Structured Practice

The Research Methods course, taken at Malmö University, served as the first nudge away from "just reading interesting papers" and towards actively weaving ideas together within a structured framework. Initially, a literature survey felt like a linear exercise: collect citations, summarise them, move on. However, the course taught me to construct a living conversation between sources: mapping each author's assumptions, identifying points of tension or complementarity, and allowing those tensions to inspire new experiments. Meanwhile, a fixed rhythm of weekly meetings with my supervisor proved invaluable in translating this theoretical mindset into concrete action. Each meeting transformed vague "next steps" into clear deliverables and tentative experiment plans, effectively turning our discussions into miniature design reviews in which hypotheses were tested, logs dissected, and plans refined.

This integrated practice carried directly into thesis work: every model failure became an opportunity to ask, "Which piece of theory does this contradict? Which paper might contain the missing clue?" Conversely, the habit of disciplined reflection ensured that experimental setbacks were not merely obstacles but integral components of a broader learning cycle. Together, rigorous literature engagement and structured supervision have fundamentally shaped my approach to research, and will remain essential as I pursue a PhD path.

6.2 Detour and Discovery

Halfway through the year, the custom supervised pipeline repeatedly collapsed each estimated kernel into the same uniform blurry kernel structure. Despite varying loss functions, adjusting dataset splits, reducing learning rates and even simplifying the architecture to a single dense block, the model failed to learn diverse PSFs. Parallel checks (validating the convolution implementation and scaling across GPUs on the VSC cluster) also failed to resolve the issue. Although frustrating, this period proved instructive: a more rigorous experimental protocol was adopted, in which one parameter at a time was altered and every outcome meticulously logged. Familiarity with the HPC environment deepened through writing SLURM scripts, monitoring GPU utilisation and managing Conda environments, while detailed experiment journals ensured that past setbacks could be revisited without retracing the same steps.

Revisiting SelfDeblur sparked the idea to integrate its kernel-comparison metrics (MSE and MNC) directly into the CNN's loss. Once both pixel-wise error and structural overlap were

combined in the objective, the pipeline immediately escaped mode collapse and began recovering distinct, plausible PSFs. This simple design tweak demonstrates how a single loss-function choice can transform training dynamics.

6.3 Looking ahead

- **Research is bricolage.** Real progress often comes from stitching together ideas from experimenting with different concepts from different sources.
- **Ill-posed problems demand strong priors.** Whether implicit (SelfDeblur) or explicit (kernel normalisation layers), modelling assumptions are not optional.
- **Logs beat memory.** Version control, experiment tracking, and a fixed meeting cadence saved me from circular rabbit holes.
- **Technical skills are as vital as theoretical insights.** Proficiency in coding and high-performance computing, writing SLURM scripts, managing interactive GPUs, scheduling resources, and MPI debugging, is as essential to advancing research as any conceptual framework or theoretical paper.
- **Stagnation is signal.** When nothing improves for weeks, the right move may be to pause, read broadly, or prototype something radically different.

This thesis has confirmed that open-ended questions, where method matters as much as results, are where I thrive. I intend to pursue doctoral research, ideally in computational imaging, to refine these skills and push my work toward publication. Areas for growth include deepening mathematical intuition (especially in inverse-problem theory) and honing scientific writing. Above all, I truly believe that disciplined experimentation coupled with creative idea-borrowing will guide my future breakthroughs and research.

If I could advise my earlier self: keep the notebooks tidy, embrace HPC quirks early, and trust that the breakthrough often hides in the paper you read to "take a break."

Bibliography

[1] Anat Levin et al. "Understanding and evaluating blind deconvolution algorithms". In: *2009 IEEE Conference on Computer Vision and Pattern Recognition*. 2009, pp. 1964–1971. DOI: 10.1109/CVPR.2009.5206815.

[2] Anat Levin et al. "Efficient marginal likelihood optimization in blind deconvolution". In: *CVPR 2011*. 2011, pp. 2657–2664. DOI: 10.1109/CVPR.2011.5995308.

[3] Norbert Wiener. *Extrapolation, Interpolation, and Smoothing of Stationary Time Series: With Engineering Applications*. The MIT Press, Aug. 1949. ISBN: 9780262257190. DOI: 10.7551/mitpress/2946.001.0001. eprint: https://direct.mit.edu/book-pdf/2313079/book_9780262257190.pdf. URL: https://doi.org/10.7551/mitpress/2946.001.0001.

[4] William Hadley Richardson. "Bayesian-Based Iterative Method of Image Restoration*". In: *J. Opt. Soc. Am.* 62.1 (Jan. 1972), pp. 55–59. DOI: 10.1364/JOSA.62.000055. URL: https://opg.optica.org/abstract.cfm?URI=josa-62-1-55.

[5] L. B. Lucy. "An iterative technique for the rectification of observed distributions". In: *The Astronomical Journal* 79 (June 1974), p. 745. DOI: 10.1086/111605.

[6] Seungjun Nah, Tae Kim, and Kyoung Mu Lee. "Deep Multi-scale Convolutional Neural Network for Dynamic Scene Deblurring". In: (Dec. 2016). DOI: 10.48550/arXiv.1612.02177.

[7] Orest Kupyn et al. "DeblurGAN: Blind Motion Deblurring Using Conditional Adversarial Networks". In: *2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition*. 2018, pp. 8183–8192. DOI: 10.1109/CVPR.2018.00854.

[8] Dongwei Ren et al. *Neural Blind Deconvolution Using Deep Priors*. Aug. 2019. DOI: 10.48550/arXiv.1908.02197.

[9] R.C. Gonzalez and R.E. Woods. *Digital Image Processing, Global Edition*. Pearson Education, 2018. ISBN: 9781292223070. URL: https://books.google.se/books?id=p74oEAAAQBAJ.

[10] Richard Szeliski. *Computer Vision: Algorithms and Applications*. 2nd ed. Texts in Computer Science. Hardcover ISBN: 978-3-030-34371-2 (Published: 05 January 2022), Softcover ISBN: 978-3-030-34374-3 (Published: 06 January 2023), eBook ISBN: 978-3-030-34372-9 (Published: 03 January 2022); XXII, 925 pages; 374 b/w illustrations, 144 in colour. Springer Cham, 2022. ISBN: 978-3-030-34372-9. DOI: 10.1007/978-3-030-34372-9.

[11] J.W. Goodman. *Introduction to Fourier Optics*. McGraw-Hill physical and quantum electronics series. W. H. Freeman, 2005. ISBN: 9780974707723. URL: https://books.google.se/books?id=ow5xs_Rtt9AC.

[12] F. Roddier. "The effects of atmospheric turbulence in optical astronomy". In: *Progess in Optics* 19 (Jan. 1981), pp. 281–376. DOI: 10.1016/S0079-6638(08)70204-X.

[13] Dilip Krishnan, Terence Tay, and Rob Fergus. "Blind deconvolution using a normalized sparsity measure". In: *CVPR 2011*. 2011, pp. 233–240. DOI: 10.1109/CVPR.2011.5995521.

[14] T.F. Chan and Chiu-Kwong Wong. "Total variation blind deconvolution". In: *IEEE Transactions on Image Processing* 7.3 (1998), pp. 370–375. DOI: 10.1109/83.661187.

[15] Dmitry Ulyanov, Andrea Vedaldi, and Victor Lempitsky. "Deep Image Prior". In: *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. June 2018.

[16] Yossi Gandelsman, Assaf Shocher, and Michal Irani. ""Double-DIP": Unsupervised Image Decomposition via Coupled Deep-Image-Priors". In: *CoRR* abs/1812.00467 (2018). arXiv: 1812.00467. URL: http://arxiv.org/abs/1812.00467.

[17] Leonid Rudin, Stanley Osher, and Emad Fatemi. "Nonlinear total variation based noise removal algorithms". In: *Physica D: Nonlinear Phenomena* 60 (Nov. 1992), pp. 259–268. DOI: 10.1016/0167-2789(92)90242-F.

[18] G. Aniano et al. "Common-Resolution Convolution Kernels for Space- and Ground-Based Telescopes". In: *Publications of the Astronomical Society of the Pacific* 123.908 (Oct. 2011), pp. 1218–1236. ISSN: 1538-3873. DOI: 10.1086/662219. URL: http://dx.doi.org/10.1086/662219.

[19] Zhe Hu and Ming-Hsuan Yang. "Good Regions to Deblur". In: *Computer Vision – ECCV 2012*. Ed. by Andrew Fitzgibbon et al. Berlin, Heidelberg: Springer Berlin Heidelberg, 2012, pp. 59–72. ISBN: 978-3-642-33715-4.

[20] ESA/Hubble & NASA, S. Jha. *Rings of Relativity*. https://esahubble.org/images/potw2050a/. ESA/Hubble Picture of the Week 2050a; acknowledgement: L. Shatz. Dec. 2020.

[21] Gregory Ongie et al. *Deep Learning Techniques for Inverse Problems in Imaging*. 2020. DOI: 10.48550/arXiv.2005.06001. arXiv: 2005.06001 [eess.IV]. URL: https://arxiv.org/abs/2005.06001.

[22] Luis G. Varela et al. *Estimation of motion blur kernel parameters using regression convolutional neural networks*. 2023. DOI: 10.48550/arXiv.2308.01381. arXiv: 2308.01381 [eess.IV]. URL: https://arxiv.org/abs/2308.01381.

[23] Richard Zhang et al. *The Unreasonable Effectiveness of Deep Features as a Perceptual Metric*. 2018. arXiv: 1801.03924 [cs.CV]. URL: https://arxiv.org/abs/1801.03924.

[24] G. Desiderà and M. Carbillet. "Strehl-constrained iterative blind deconvolution for post-adaptive-optics data". In: *A&A* 507.3 (2009), pp. 1759–1762. DOI: 10.1051/0004-6361/200912913. URL: https://doi.org/10.1051/0004-6361/200912913.