AI

Review

Fractional Calculus Meets Neural Networks for Computer Vision: A Survey

Cecília Coelho [1,*], M. Fernanda P. Costa [1] and Luís L. Ferrás [1,2]

[1] Centre of Mathematics (CMAT), University of Minho, 4710-057 Braga, Portugal; mfc@math.uminho.pt (M.F.P.C.); lferras@fe.up.pt (L.L.F.)
[2] Department of Mechanical Engineering (Section of Mathematics) and CEFT—Centro de Estudos de Fenómenos de Transporte—FEUP, University of Porto, 4200-465 Porto, Portugal
* Correspondence: cmartins@cmat.uminho.pt

Abstract: Traditional computer vision techniques aim to extract meaningful information from images but often depend on manual feature engineering, making it difficult to handle complex real-world scenarios. Fractional calculus (FC), which extends derivatives to non-integer orders, provides a flexible way to model systems with memory effects and long-term dependencies, making it a powerful tool for capturing fractional rates of variation. Recently, neural networks (NNs) have demonstrated remarkable capabilities in learning complex patterns directly from raw data, automating computer vision tasks and enhancing performance. Therefore, the use of fractional calculus in neural network-based computer vision is a powerful method to address existing challenges by effectively capturing complex spatial and temporal relationships in images and videos. This paper presents a survey of fractional calculus neural network-based (FC NN-based) computer vision techniques for denoising, enhancement, object detection, segmentation, restoration, and NN compression. This survey compiles existing FFC NN-based approaches, elucidates underlying concepts, and identifies open questions and research directions. By leveraging FC's properties, FC NN-based approaches offer a novel way to improve the robustness and efficiency of computer vision systems.

Keywords: computer vision; neural networks; fractional calculus; object detection; denoising; segmentation; image generation

check for updates

Citation: Coelho, C.; Costa, M.F.P.; Ferrás, L.L. Fractional Calculus Meets Neural Networks for Computer Vision: A Survey. *AI* **2024**, *5*, 1391–1426. https://doi.org/10.3390/ai5030067

Academic Editor: Arslan Munir

Received: 3 July 2024
Revised: 18 July 2024
Accepted: 25 July 2024
Published: 7 August 2024

1. Introduction

Computer vision has become a transformative field with significant impact across various industries. Using computers to interpret visual information has been crucial for enhancing surveillance systems and aiding medical imaging for disease diagnosis.

Traditional computer vision involves many algorithms and methods that are used to extract meaningful information from images. These techniques typically include steps like image preprocessing, feature extraction, and classification or inference [1].

One of the fundamental tasks in computer vision is image segmentation, where traditional techniques such as thresholding, edge detection, and region-based methods are commonly employed to partition images into semantically meaningful regions or objects. Object detection, another important task, was traditionally addressed using techniques such as Haar cascades, Histogram of Oriented Gradients (HOG), and feature-based classifiers. These methods rely on manually designed features and classifiers to detect objects within images, often requiring the careful tuning of parameters and heuristics to achieve optimal performance. Similarly, image denoising, restoration, and enhancement have been tackled using conventional filtering techniques, such as median filtering, Gaussian blurring, and wavelet transforms, aimed at removing noise, restoring lost details, and improving overall image quality [2,3].

Fractional calculus (FC) is a powerful mathematical framework that extends integer-order derivatives and integrals to non-integer orders. This generalisation allows for more degrees of freedom and provides a more nuanced and flexible way to model systems with memory effects, long-range dependencies, and anomalous behaviours, which are prevalent in real-world phenomena [4]. In recent years, FC has gained attention across various fields, including physics [5], engineering [6], biology [7], finance [8], and medicine [9], due to its ability to capture complex dynamics that cannot be adequately described by classical integer-order derivatives.

Due to proven advantages, several FC-based approaches have been introduced in the literature to improve performance in computer vision tasks [10–13]. Although the traditional FC-based approaches have been instrumental in enabling various applications, they often struggle with complex, real-world scenarios and lack the ability to generalise well across different domains. These approaches rely heavily on manual feature engineering and human knowledge or assumptions on the ground-truth image (prior). Furthermore, FC-based approaches often face challenges in handling variability in lighting conditions, viewpoint changes, and occlusions, which are prevalent in real-world images [2].

FC has also been attracting increasing attention from researchers in the field of deep learning algorithms. The inherent ability to capture complex dynamics and temporal relationships is closely aligned with the objectives of ML, particularly in tasks that involve sequential data analysis, such as time-series forecasting, natural language processing, and sequential decision making [14–16].

Recently, neural networks (NNs) have demonstrated remarkable capabilities in learning complex patterns and features directly from raw data. In the context of computer vision, NNs have enabled breakthroughs in image denoising, enhancement, segmentation, restoration, and object detection. Moreover, constant efforts by the research community have also lead to reducing the computational cost of these tasks. By leveraging large datasets and powerful computational resources, NNs have surpassed traditional computer vision approaches, achieving state-of-the-art performances across a wide range of tasks. Furthermore, the ability of NNs to automatically learn and extract features from data has eliminated the need for manual feature engineering and enabled systems to adapt and generalise to diverse and complex visual environments [2].

More recently, FC has emerged in NN-based computer vision, offering a compelling approach to enhance the performance of NN architectures. Through incorporating non-integer orders derivatives, FC makes it possible to account for the intricate spatial and temporal relationships, combining local and global features inherent in images and videos [17]. For instance, NNs have been used to leverage the power of fractional-order differential mask operators, enabling the discovery of optimal mask orders to improve their performance in achieving specific image goals, such as denoising and enhancement [18]. Additionally, the introduction of fractional-order convolutional kernels has facilitated the compression of NN architectures, resulting in significant reductions in the number of trainable parameters [19]. These advancements underscore the versatility and efficacy of fractional derivatives in not only enhancing the performance of computer vision systems but also streamlining computational processes, thereby propelling the field toward more efficient and scalable solutions.

Based on the above, this work aimed to conduct a survey of the FC NN-based computer vision techniques presented in the literature, focusing on the tasks of denoising, enhancement, object detection, segmentation, restoration, and NN compression. The aim was to compile various approaches and provide a concise yet intuitive explanation of the underlying concepts. Additionally, we aimed to identify open questions and research directions that stem from each paper covered in this survey.

This paper is organised as follows. In Section 2, we provide a brief background to understand the basics of FC. Section 3 starts with a concise overview of computer vision and the various tasks it encompasses, namely denoising, enhancement, segmentation, object detection, restoration, and the compression of NN architectures. Subsequently, for each task, we present the methods in the literature that combine FC and NN-based computer vision, concluding with a brief discussion of open questions and potential research directions arising from each method. Finally, the paper ends in Section 4 with a summary of the findings and conclusions drawn. Additionally, Appendix A provides a summary table of all the methods discussed in this work, briefly detailing their advantages, experimental setups, and results as reported in the original papers.

2. Fractional Calculus

Contrary to popular belief, fractional differential calculus is not a recent subject. For example, the symbol $d^n y/dx^n$ was first proposed by Leibniz, and, in 1695, L'Hôpital asked Leibniz about the meaning of $d^{1/2}y/dx^{1/2}$, effectively asking, "What if n is fractional?". Leibniz responded,

> "Although infinite series and geometry are distant relations, infinite series admits only the use of exponents that are positive and negative integers and does not, as yet, know the use of fractional exponents".

He continued, "This is an apparent paradox from which, one day, useful consequences will be drawn".

This correspondence can be seen as the beginning of fractional differential calculus. The term "fractional" comes from L'Hôpital's question about the fraction 1/2, although the order of differentiation can be any real or complex number. From this letter, we learn that classical and fractional differential calculus were conceived almost simultaneously [4,20–22]. In 1716, Leibniz died, but the interest in understanding derivatives of fractional (non-integer) order increased. Several other authors devoted their time to this subject, and the well-known Leonard Euler also contributed to the understanding and generalisation of fractional differential calculus. He extended the notion of the factorial, $n!$, to non-integer values. This extension was later named the Gamma function, $\Gamma(.)$, by Adrien-Marie Legendre around 1811:

$$\Gamma(z) = \int_0^\infty t^{z-1}e^{-t}\,\mathrm{d}t \tag{1}$$

where $\Re(z) > 0$. The Gamma function plays a crucial role in defining generalised derivatives, as shown below.

The idea of generalising derivative and integral operators to an order n seems simple because one only needs to obtain operators that can be defined for non-integer n values and that match the classical operators when n is an integer. Consequently, several definitions have been proposed in the literature by different authors. Since this work is not a survey on fractional calculus, only three definitions of fractional derivatives will be considered here: the Riemann–Liouville and Caputo definitions (often used in applications in physics and engineering), and the Grünwald-Letnikov definition, which seems to be preferred in NN-based computer vision.

The Riemann–Liouville and Caputo fractional derivatives will now be introduced. However, before defining these derivatives, we will first provide the definition of a fractional integral. For that, recall the Fundamental Theorem of Calculus.

Theorem 1. ***Fundamental Theorem of Calculus:*** *Let $f : [a,b] \to \mathbb{R}$ be a continuous function, and let $F : [a,b] \to \mathbb{R}$ be defined by*

$$F(x) = \int\limits_a^x f(t)dt. \tag{2}$$

Then, F is differentiable, and

$$F' = \frac{dF}{dt} = f. \tag{3}$$

In order to write in a more compact way some of the results to come, we now define the following differential and integral operators.

Definition 1. ***Derivative and Integral Operators:*** *We denote the differential operator that maps a function f into its derivative $Df(x) = f'(x)$ by D and the integral operator that maps a function $f(x)$ into its primitive (whenever the integration can be performed on the compact interval $[a,\ b]$) by J_a:*

$$J_a f(x) = \int_a^x f(t)dt, \ x \in [a, b] \tag{4}$$

These operators can be generalised to perform n-fold iterates:

$$D^n f(x) = \frac{d}{dt} \dots \frac{d}{dt}\frac{df}{dt} = D^1 D^{n-1} f(x), \tag{5}$$

$$J_a^n f(x) = \int_a^x \dots \int_a^x \int_a^x f(t)dt = J_a^1 J_a^{n-1} f(x). \tag{6}$$

The following Lemma introduces a way to write the *n-fold* integral using only one integral symbol.

Lemma 1. ***n-fold Integration:*** *Let f be Riemann integrable on $[a, b]$. Then, for $a \leq x \leq b$ and $n \in \mathbb{N}$, we have*

$$J_a^n f(x) = \frac{1}{(n-1)!} \int_a^x (x-t)^{n-1} f(t)dt. \tag{7}$$

To generalise the previous integral to non-integer orders, one simply needs to replace $(n-1)!$ with the Gamma function defined earlier, $\Gamma(n)$, taking into account that $\Gamma(n) = (n-1)!$, $n \in \mathbb{N}$.

Definition 2. ***Riemann–Liouville Fractional Integral:*** *Let $n \in \mathbb{R}_+$ and J_a^n be the operator defined on $L_1[a, b]$ by*

$$J_a^n f(x) = \frac{1}{\Gamma(n)} \int_a^x (x-t)^{n-1} f(t)dt, \ x \in [a, b]. \tag{8}$$

Then, J_a^n is called the Riemann–Liouville fractional integral operator of order n.

The fractional derivative is obtained by taking a derivative (of a certain integer order) of the fractional integral just defined. This implies that fractional derivatives may depend on integral operators.

Recall that in the classical case (integer orders), we have the following lemma.

Lemma 2. ***n-fold Integration:*** *Let $m, n \in \mathbb{N}$ with $m > n$, and let f be a function with a continuous n^{th} derivative on the interval $[a, b]$. Then,*

$$D^n f = D^m J_a^{n-n} f. \tag{9}$$

Therefore, a generalisation of this lemma leads to the definition of a Riemann–Liouville fractional derivative [21].

Definition 3. ***Riemann-Liouville Fractional Derivative:*** *Let $\alpha \in \mathbb{R}_+$ and $m = \lceil \alpha \rceil$. The Riemann–Liouville fractional derivative of order α (${}_a^R D_t^\alpha f$) is given by*

$${}_a^R D_x^\alpha f(x) = D^m J_a^{m-\alpha} f(x) = \frac{D^m}{\Gamma(m-\alpha)} \int_a^x (x-t)^{m-\alpha-1} f(t)dt. \tag{10}$$

For $n = 0$, we have ${}_a^R D_t^0 := I$.

This definition of the fractional derivative generalises the classical case of integer-order derivatives. However, it may lead to properties that could be seen as less appealing. For instance, the Riemann–Liouville derivative of a constant is not zero. Although, if we exchange the order of integration and differentiation, this *less appealing characteristic* can be easily supressed, and we obtain a new definition of a fractional derivative, proposed by M. Caputo [23].

Definition 4. ***Caputo Fractional Derivative:*** *Let $\alpha \in \mathbb{R}_+$, $m = \lceil \alpha \rceil$ and $D^m f(t) \in L_1([a, b])$. The Caputo fractional derivative of order α (${}_a^C D_t^\alpha f$) is given by*

$$ {}_a^C D_x^\alpha f(x) = J_a^{m-\alpha} D^m f(x) = \frac{1}{\Gamma(m-\alpha)} \int_a^x (x-t)^{m-\alpha-1} D^m f(t) dt. \tag{11}$$

Note the resemblance with the Riemann–Liouville fractional derivative.

These two definitions of fractional derivatives have expressions that depend on integrals. Therefore, besides the property of order generalisation, they are often used in modelling physical problems where memory is an important factor. At each instant, the integral computes the past history. The range of applications for generalised derivatives is immense. Consequently, many recent works have used these general operators.

There is another definition of a fractional derivative that has captured the attention of many researchers. This definition is based on classical differentiation and seems to be more intuitive. Grünwald (1867 [24]), Post (1930 [25]), and Letnikov (1872 [26]) presented the idea of the fractional derivative as the limit of a sum.

Itde variáveis is well known that a classical derivative can be approximated as a limit of difference quotients. For example,

$$ f'(x) = D^1 f(x) = \lim_{h \to 0} \frac{f(x) - f(x-h)}{h} \tag{12}$$

We know that

$$\begin{aligned} \nabla_h^1 f(x) = &\ f(x) - f(x-h) \\ \nabla_h^2 f(x) = &\ \nabla_h^1 f(x) - \nabla_h^1 f(x-h) = f(x) - 2f(x-h) + f(x-2h) \\ \vdots \quad &\ \vdots \\ \nabla_h^n f(t) = &\ \textstyle\sum_{k=0}^n (-1)^k \begin{pmatrix} n \\ k \end{pmatrix} f(x-kh). \end{aligned} \tag{13}$$

where $\begin{pmatrix} n \\ k \end{pmatrix} = \dfrac{n(n-1)\cdots(n-k+1)}{k!}$ is the binomial coefficient. Therefore, we can state the following.

Theorem 2. *Let $n \in \mathbb{N}$, $f \in C^n([a, b])$ and $a < t \leq b$. Then,*

$$ D^n f(t) = \lim_{h \to 0} \frac{\nabla_h^n f(t)}{h^n} \tag{14}$$

For example, a second-order derivative can be written as

$$ D^2 f(x) = \lim_{h \to 0} \frac{f(x) - 2f(x-h) + f(x-2h)}{h^2}. \tag{15}$$

Grünwald and Letnikov performed a generalisation of this result to non-integer n values, leading to the following definition of a fractional derivative.

Definition 5. *Let $\alpha \in \mathbb{R}_+$, $f(t) \in C^{\lceil \alpha \rceil}([a, b])$ and $h_N = (t-a)/N$. The Grünwald–Letnikov fractional derivative of order α (${}_a^{GL} D_x^\alpha f$) is given by*

$$ {}^{GL}_{a}D^{\alpha}_{x}f(x) = \lim_{N\to\infty} \frac{\nabla^{\alpha}_{h_N} f(t)}{h^{\alpha}_{N}}. \tag{16}$$

with

$$\nabla^{\alpha}_{h} f(x) = \sum_{k=0}^{\infty}(-1)^k \begin{pmatrix} \alpha \\ k \end{pmatrix} f(x-kh). \tag{17}$$

Note that

$$\begin{pmatrix} \alpha \\ k \end{pmatrix} = \frac{\Gamma(\alpha+1)}{\Gamma(k+1)\Gamma(\alpha-k+1)} \tag{18}$$

is the fractional binomial coefficient, a generalisation of the classical binomial coefficient to non-integer values [20].

To simplify the notation, we will drop the *GL*, and the Grünwald–Letnikov fractional derivative will be simply represented by $D^{\alpha}f(x)$. It is also common to represent the derivative by making $h_N \to 0$, that is,

$$D^{\alpha}f(x) = \lim_{h_N\to 0} \frac{1}{h^{\alpha}_{N}} \sum_{k=0}^{\infty}(-1)^k \frac{\Gamma(\alpha+1)f(x-kh)}{\Gamma(k+1)\Gamma(\alpha-k+1)}. \tag{19}$$

This definition allows for the discretisation of fractional derivatives, enabling their computation using simple finite differences. Note that there are various definitions of fractional derivatives, each with its own advantages and disadvantages (the interested reader should consult [22]).

In computer vision, integer-order calculus plays a pivotal role. Techniques such as gradient-based edge detection use derivatives to identify abrupt changes in intensity, forming the foundation of edge detection algorithms. Furthermore, integral calculus finds applications in image processing tasks such as convolution and filtering, where convolution operations are analogous to computing the integral of a function over a given region. The mathematical principles of differential and integral calculus also form the foundation for numerous algorithms in feature extraction, object recognition, and image segmentation, allowing computers to effectively interpret and analyse visual information [27]. Therefore, in the following chapters, we present various works that use fractional calculus to enhance neural network-based computer vision. By fractional calculus, we mean works involving fractional derivatives or generalising integer-order operations (e.g., using the Gamma function instead of the factorial, or employing non-polynomials or polynomials of non-integer order instead of classical polynomials, etc). While some studies use fractional optimisation algorithms to optimise neural network parameters for computer vision tasks, this survey focuses exclusively on methods that leverage fractional calculus to modify or contribute to the architecture, such as in feature extraction or image enhancement.

3. Computer Vision

Computer vision encompasses a diverse range of tasks, ranging from denoising and enhancement to object detection, segmentation, and restoration [28], as shown in Figure 1.

Recent advancements in the literature have shown significant performance improvements in computer vision, and therefore, in this section, we introduce and elaborate on neural network architectures and techniques that use fractional calculus to enhance computer vision tasks. These innovative approaches not only improve the tasks' performance but also contribute to reducing the computational cost associated with NN-based computer vision systems, making them more practical and scalable for real-world applications.

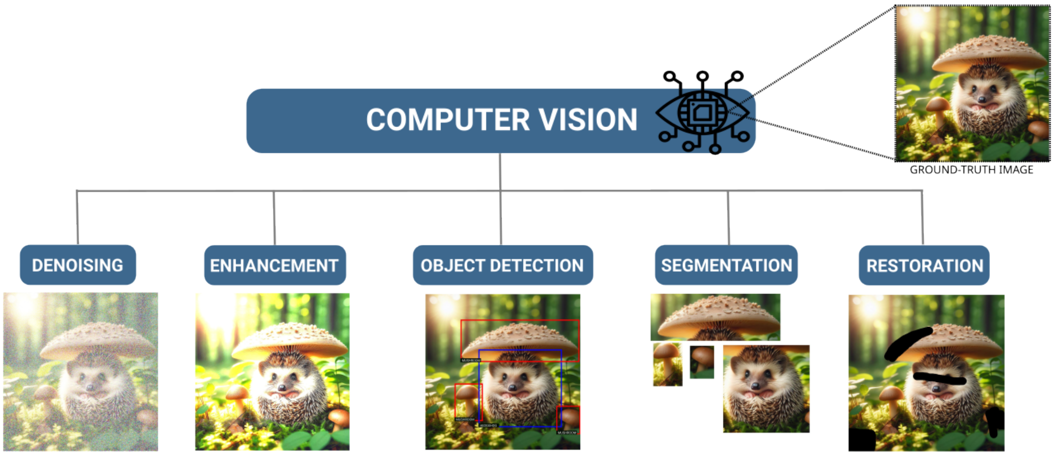

Figure 1. Example use cases of different tasks in computer vision: denoising for removing unwanted noise, enhancement for ground-truth image's quality improvement, object detection for identification and labelling, segmentation for image partitioning for further analysis, and restoration for missing parts inpainting (ground-truth image generated by DALL-E 3).

3.1. Denoising

Image denoising is the process of removing unwanted noise from digital images to enhance their visual quality and improve the accuracy of subsequent analysis or processing tasks. Noise in images can arise from various sources, including sensor limitations, transmission errors, or environmental factors during image capture. The goal of denoising algorithms is to distinguish between the true signal representing the underlying scene and the undesirable noise components, and then attenuate or eliminate the noise while preserving important image features. This typically involves applying filters or statistical techniques tailored to suppress noise, thus resulting in cleaner and more visually appealing images [29].

Image denoising using FC has emerged as a promising approach to address the challenges posed by noise in digital images, offering advantages over traditional methods. By leveraging the memory effects and long-range interactions inherent in fractional calculus, researchers have developed novel denoising algorithms capable of preserving image details while effectively suppressing noise. Several works in the literature have explored this approach, with more contributions continuously being made [30–36].

Due to their approximation capabilities, NNs have emerged as powerful tools in image denoising by learning the underlying structure of clean images and effectively differentiating between noise and true image features. These NNs are normally trained on pairs of noisy and clean images, where they learn to map noisy inputs to their corresponding clean versions. Convolutional neural networks (CNNs) are particularly well suited for image denoising tasks due to their ability to automatically extract hierarchical features from images. Through iterative training processes, NNs find the optimal parameters that minimise the difference between the denoised output and the clean ground-truth image [37]. This approach has shown remarkable success in various applications, such as medical imaging [38], surveillance systems [39], ocean biodiversity monitorisation [40], and others.

In the literature, only two works proposing the combination of fractional calculus (FC) and NN-based techniques for image denoising were found [41,42]. In [41], the authors propose formulating denoising as a variational problem, aiming to minimise a functional that incorporates fidelity to the observed noisy image and smoothness of the denoised image, by integrating non-integer order derivatives. These are integrated into the regularisation term of the variational model, facilitating the preservation of edges and textures while reducing noise, denoted as Fractional-order Total Variation. In [42], the authors use weights given by FDEs to propagate the feature maps from one layer to the next, giving rise to the Fractional Optimal Control Network.

3.1.1. Fractional-Order Total Variation

Total Variation (TV) regularisation is a core technique for image denoising that preserves important features such as edges and textures, with the ability to effectively reduce noise while maintaining sharp transitions between regions of an image. Unlike simpler techniques, such as linear smoothing or median filtering, which can blur edges and details, TV regularisation exploits the inherent sparsity in the gradient of the image, penalising rapid changes in pixel intensity.

TV regularisation can be formulated using the L_2 norm as [41]

$$\mathcal{L}_{\text{TV}} = \sum_{i=1}^{I} \sum_{j=1}^{J} \sqrt{(X'(i+1,j) - X'(i,j))^2 + (X'(i,j+1) - X'(i,j))^2} \tag{20}$$

or using the L_1 norm as

$$\mathcal{L}_{\text{TV}} = \sum_{i=1}^{I} \sum_{j=1}^{J} |X'(i+1,j) - X'(i,j)| + |X'(i,j+1) - X'(i,j)| \tag{21}$$

where I and J are the width and length (number of pixels in the horizontal and vertical directions) of the image, respectively, i and j are the coordinates of a pixel in the image, and X' is the denoised image [41]. The difference $X'(i+1,j) - X'(i,j)$ calculates the finite horizontal gradient, corresponding to the change in pixel value along the horizontal direction. Similarly, $X'(i,j+1) - X'(i,j)$ is the finite vertical gradient, representing the difference between adjacent pixels along the vertical direction.

The idea is straightforward. In addition to the classical loss function, which measures the *difference* between the ground truth and the result obtained from the neural network—$\mathcal{L}_{\text{error}}$—a new loss function is introduced to aid in image denoising and artefact reduction. This new loss function is defined as

$$\mathcal{L}_{\text{total}} = \mathcal{L}_{\text{error}} + \lambda \mathcal{L}_{\text{TV}}, \tag{22}$$

where λ is the regularisation hyperparameter. This parameter balances the fit of the model to the observed data (data fidelity term) with the smoothness or sparsity enforced by the regularisation term. A small regularisation parameter might lead to overfitting, where the model closely matches the training data but does not generalise well to new data. Conversely, a large regularisation parameter could overly smooth or simplify the solution, potentially causing the loss of important features or details. In essence, λ controls the significance of the TV loss in the optimisation process. This exploits the inherent sparsity in the gradient of the image, penalising rapid changes in pixel intensity [43].

TV fails to fully use the information from neighbouring pixels, which can lead to artefacts in images. To address this limitation, fractional-order differences have been proposed for processing this gradients. The fractional variations inherently incorporate information from neighbouring pixels, allowing them to draw data not only from adjacent pixels but also from more distant ones. Consequently, fractional-order differences can theoretically capture richer pixel information and reduce artefacts.

Taking this into account, in [41], the authors proposed a new NN-based denoising model that incorporates Fractional-order TV (FTV) regularisation into the loss function. To the best of our knowledge, this was the first time FTV was used in conjunction with a NN; however, multiple studies have compared NN-based methods to FTV filters [44–46].

FTV is therefore an extension of TV regularisation proposed in [47], by introducing a fractional exponent α to the TV term.

In [41], the authors proposed a modified FTV regularisation that computes the fractional gradient in eight directions around each pixel, namely x^-, y^-, x^+, y^+, Left-Down Direction (*LDD*), Right-Up Direction (*RUD*), Left-Up Direction (*LUD*), and Right-Down Direction (*RDD*):

$$\begin{aligned}\mathcal{L}_{\text{FTV}} = \sum_{i=1}^{I}\sum_{j=1}^{J} |\mathcal{D}^{\alpha}_{x^+}X'| + |\mathcal{D}^{\alpha}_{y^+}X'| + |\mathcal{D}^{\alpha}_{x^-}X'| + |\mathcal{D}^{\alpha}_{y^-}X'| + |\mathcal{D}^{\alpha}_{LDD}X'| + \\ |\mathcal{D}^{\alpha}_{RUD}X'| + |\mathcal{D}^{\alpha}_{LUD}X'| + |\mathcal{D}^{\alpha}_{RDD}X'|,\end{aligned} \tag{23}$$

where each term D^{α} corresponds to applying a fractional differential mask (using an approximation to the Grünwald–Letnikov derivative) in the corresponding direction, x^-, y^-, x^+, y^+, LDD, RUD, LUD, or RDD. For example, in the horizontal direction x, we have that

$$\mathcal{D}^{\alpha}_{x^+}X' = \begin{bmatrix} \cdots & \cdots & \cdots & \cdots \\ 0 & 0 & 0 & \\ C_{S_{-1}} & C_{S_0} & \cdots & C_{S_n} \\ 0 & 0 & \cdots & 0 \\ \cdots & \cdots & \cdots & \cdots \end{bmatrix}, \mathcal{D}^{\alpha}_{x^-}X' = \begin{bmatrix} \cdots & \cdots & \cdots & \cdots \\ 0 & 0 & 0 & \\ C_{S_n} & C_{S_{n-1}} & \cdots & C_{S_{-1}} \\ 0 & 0 & \cdots & 0 \\ \cdots & \cdots & \cdots & \cdots \end{bmatrix}. \tag{24}$$

Here, n is the last entry, with entries derived from the Grünwald–Letnikov derivative given by

$$C_{S_k} = \frac{1}{\Gamma(-\alpha)}\left[\frac{\Gamma(k-\alpha+1)}{(k+1)!}\cdot\left(\frac{\alpha}{4}+\frac{\alpha^2}{8}\right) + \frac{\Gamma(k-\alpha)}{k!}\cdot\left(1-\frac{\alpha^2}{4}\right) + \frac{\Gamma(k-\alpha-1)}{(k-1)!}\cdot\left(-\frac{\alpha}{4}+\frac{\alpha^2}{8}\right)\right], \tag{25}$$

with k as the entry of the fractional differential mask being computed [41].

FTV regularisation is then added to the loss function of deep learning methods with the aim of preserving texture and enhancing details [41].

$$\mathcal{L}_{\text{total}} = \mathcal{L}_{\text{error}} + \lambda\mathcal{L}_{\text{FTV}}, \tag{26}$$

The method of adding FTV to the loss function of NN-based models for image denoising is still very new and there has not been much research in this area yet. One significant study that used this method is found in [48], which focused on classifying environmental sounds. This shows that the field has many unexplored areas, such as understanding how the choice of the α value affects the results and whether it is possible to include more than one FTV regularisation term with different α values in the loss function.

3.1.2. Fractional Optimal Control Network

The Fractional Optimal Control Network (FOCNet) is an NN architecture designed for denoising tasks, leveraging the principles of Fractional Ordinary Differential Equations (F-ODEs) to propagate features depth-wise within the network. This architecture exploits the memory persistence inherent in F-ODEs, enabling enhanced denoising performance compared to traditional NN approaches. The underlying methodology of FOCNet involves solving a Fractional Optimal Control problem [42]:

$$\begin{aligned} \min_{\boldsymbol{\theta}(t)} \quad & \frac{1}{2}\int_{\omega}(\Phi(\boldsymbol{u}(T,s)) - \boldsymbol{x}(s))^2 ds \\ \text{s.t.} \quad & \mathcal{D}^{\alpha}_{t}\boldsymbol{u}(t,s) = f(\boldsymbol{u}(t,s),\boldsymbol{\theta}(t)), \\ & \boldsymbol{u}(0,s) = \Psi(\boldsymbol{y}(s)), t \in [0,T] \end{aligned} \tag{27}$$

where $\boldsymbol{y}(s)$ represents the input image to be denoised, s is a pixel, $\boldsymbol{x}(s)$ denotes the corresponding ground-truth image, $\Phi(\cdot)$ and $\Psi(\cdot)$ denote linear transformations (such as convolutions) that are predefined, and $\boldsymbol{u}(t,s)$ represents the control input. The function $f(\boldsymbol{u}(t,s),\boldsymbol{\theta}(t))$ describes the dynamics of the system, parameterised by an NN [42].

FOCNet conceptualises the NN as an infinite-depth architecture, where to each layer's output, the previous layers' outputs are added, multiplied by a weight w_k given by discretising an F-ODE using Grünwald–Letnikov [42], as shown in Figure 2:

$$\boldsymbol{u}_{t+1} = \sum_{k=0}^{t} w_k \boldsymbol{u}_k + \sigma(\boldsymbol{\theta}_t \boldsymbol{u}_t), \quad \text{with} \quad w_k = (-1)^{t-k+2}\binom{\alpha}{t-k+1}. \tag{28}$$

where $\boldsymbol{u}_{t+1}$ is the output of layer $t+1$, k denotes all previous layers, $0 < k < t$, and σ is a nonlinear operation given by convolution followed by batch normalisation and a Rectified Linear Unit [42].

This approach enables the propagation of features from one layer to another throughout the network, enabling effective feature extraction and denoising. Thus, the usage of F-ODE allows us assign weights to each layers' contribution to the end result [42].

The goal of FOCNet is to optimise the denoising process by minimising the difference between the denoised image and the ground-truth image. This is achieved through the iterative adjustment of the parameters $\boldsymbol{\theta}(t)$ of the NN, guided by the solutions to the Fractional Optimal Control problem, (27). To discretise the fractional-order dynamic system inherent in FOCNet, the Grünwald–Letnikov fractional derivative definition is employed [42].

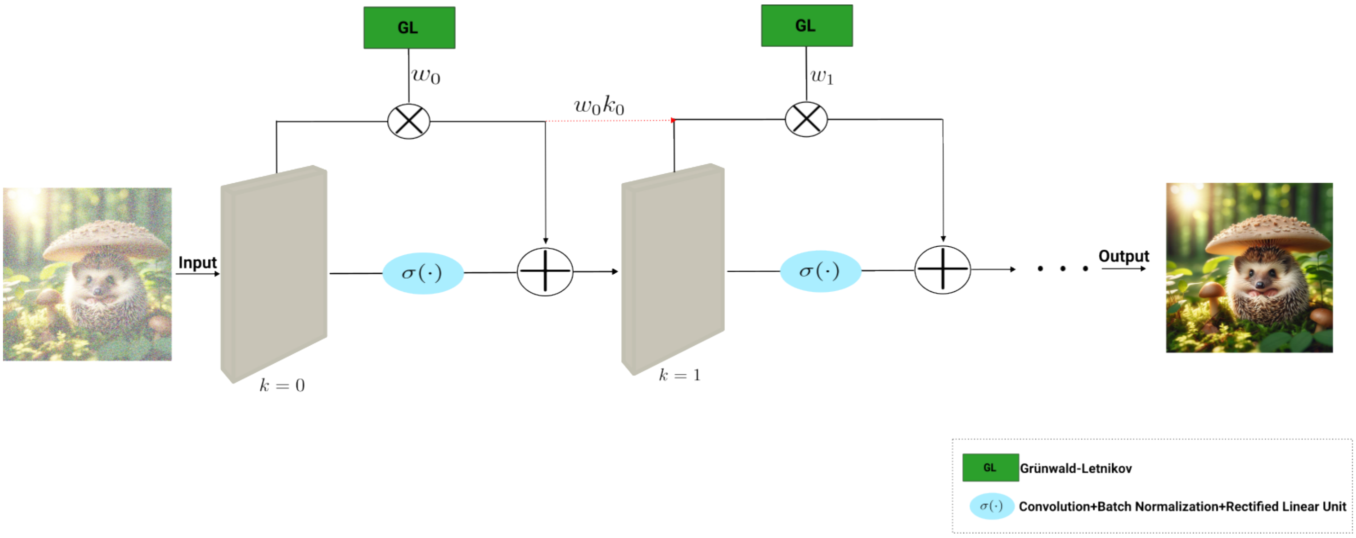

Figure 2. Architecture of FOCNet.

In comparison to traditional denoising NNs (not FC-based), the FDE enables the NN to assign weights to each layer's contribution to the end result. This is because F-ODEs provide a mathematical framework for describing the dynamics of the system (denoising process), including the propagation of features and the influence of different network layers on the final output [42].

The literature has extensively demonstrated the benefits of using multiple scales of the same image to extract diverse features, thus enhancing the NN's capacity for feature extraction. Building upon this, in [42], the authors extended FOCNet to incorporate multi-scale representations, giving rise to a multi-level architecture.

The multi-scale FOCNet architecture comprises multiple hierarchical levels, with each level representing a distinct scale and containing a dedicated FOCNet, (27). This modification enables the network to capture and retain both previous features and features across different scales, facilitating the long-term memory mechanisms inherent in FDEs. Unlike the standard FOCNet (27), the denoising process in the multi-scale version incorporates an additional step. This step involves the application of a function $g(x) = wT(x)$, where $w \in 0, 1$ and $T(\cdot)$ represents a pooling or unpooling operation, enabling the propagation of contributions from lower-level layers to higher-level layers. The denoising process within the multi-scale FOCNet can be formulated as [42]

$$
\begin{cases}
D_t^{\alpha}\boldsymbol{u}(t,s,l_1) = f(\boldsymbol{u}(t,s,l_1), g(\boldsymbol{u}(t,s,l_{1+1}), \boldsymbol{\theta}_1(t))) \\
D_t^{\alpha}\boldsymbol{u}(t,s,l_2) = f(\boldsymbol{u}(t,s,l_2), g(\boldsymbol{u}(t,s,l_{2\pm 1}), \boldsymbol{\theta}_2(t))) \\
\dots \\
D_t^{\alpha}\boldsymbol{u}(t,s,l_i) = f(\boldsymbol{u}(t,s,l_i), g(\boldsymbol{u}(t,s,l_{i+1}), \boldsymbol{\theta}_i(t))) \\
\boldsymbol{u}(0,s,l_1) = \Psi\boldsymbol{y}(s) \\
\boldsymbol{u}(0,s,l_i) = T_{\downarrow}\boldsymbol{u}(1,s,l_{i-1}) \\
1 \leq l_i \leq k \\
0 \leq t \leq T,
\end{cases}
\tag{29}
$$

where l_i represents the FOCNet level i (with l_1 denoting the original level), $\boldsymbol{\theta}_i$ denotes the parameters of the corresponding level, $\boldsymbol{u}_t^{l_i \pm 1}$ denotes either the upper-level feature $\boldsymbol{u}_t^{l_i+1}$ or the lower-level features $\boldsymbol{u}_t^{l_i-1}$, and $T\downarrow$ is a pooling operation [42].

The computation to obtain the result of each layer is now more complex:

$$
\boldsymbol{u}_{t+1}^{l_i} = \sum_{k=0}^{t} w_k \boldsymbol{u}_k^{l_i} + \sigma(\boldsymbol{\theta}_i(\boldsymbol{u}_t^{l_i} + g(\boldsymbol{u}_t^{l_i \pm 1}))),
\tag{30}
$$

aiming to strengthen the NN by promoting cross-level feature interactions [42], as shown in Figure 3.

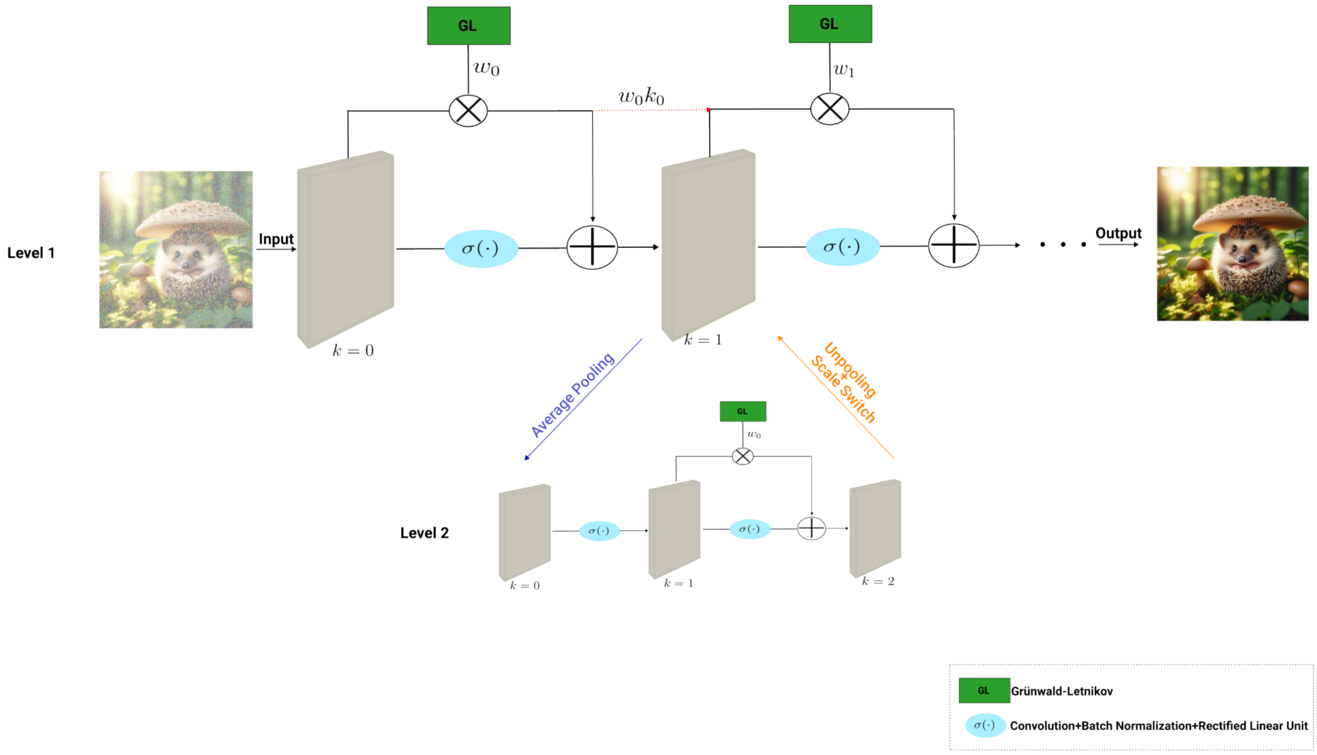

Figure 3. Schematic representation of a multi-scale FOCNet with two levels.

The work presented in [42] prompts several research questions and potential directions for further investigation. Firstly, the choice of fractional derivative definition remains largely unexplored and warrants discussion or justification to provide insight into why a particular definition was selected over others. Moreover, considering that FOCNet extracts scaling features at various levels and combines them, there is potential for enhancing its performance by employing different strategies for combining these features. Currently, in FOCNet, each level's features contribute equally, but introducing weighting factors could be an intriguing avenue to explore. Adjusting the contribution of features at different levels could potentially improve the network's ability to capture and leverage hierarchical information effectively.

3.2. Enhancement

Image enhancement is the process of improving the visual quality or perception of digital images by manipulating their attributes such as brightness, contrast, sharpness, and colour balance. Unlike denoising, which specifically targets noise reduction, image enhancement aims to enhance the overall appearance of images to make them more visually appealing or suitable for specific applications. Enhancement techniques can range from simple adjustments like histogram equalisation or contrast stretching to more advanced algorithms such as image fusion or super-resolution. These techniques can be used to highlight important features, improve visibility in low-light conditions, or adapt images for specific display or analysis requirements [49].

Image enhancement using FC is a well-explored field with several works showing the advantages over traditional methods that rely on integer calculus. The construction of masks and filters with fractional orders has opened the possibility of attaining in-between behaviours of traditional masks, offering enhanced flexibility and performance in image enhancement tasks [10,50–56].

In image enhancement, NNs learn from image datasets to understand what constitutes an enhanced or improved version of an image. CNNs are a very popular architecture that, by analysing these image pairs, learns to identify patterns and relationships between low-quality features and their desired improvements. During image enhancement, the CNN takes a low-quality image as input and processes it through its layers, making adjustments to brightness, contrast, noise levels, and other visual aspects. Furthermore, NNs have also shown high efficacy in learning masks and filters to perform tasks such as edge enhancement, texture synthesis, and artefact removal [49]. The use of NNs for image enhancement has emerged as a powerful tool in several applications, such as the detection of diseases in plants [57], downsampling of temperature climatic maps [58], and enhancement of medical imaging [59].

The authors could only find two works leveraging NNs for improving image enhancement in the literature, which will be herein discussed [18,60].

In [18], the authors propose using NNs to learn the best fractional order for the masks, demonstrating promising results in image enhancement tasks.

In [60], the authors propose using fractional Rényi entropy to enhance images before feeding them into an NN for image segmentation purposes. The results show that a higher performance in segmentation is achieved when compared with other methods.

3.2.1. Neural Fractional-Order Adaptive Masks

Masks (or kernels) and filters are mathematical operators applied to images that enhance the quality of images, easing the extraction of relevant features such as patterns and structures. Masks are matrices that are applied to the pixels in an image to perform a specific operation, such as edge detection and convolution; two very popular masks are Sobel and Lapalace [27].

The Sobel mask consists of two 3 × 3 convolution masks, one to detect horizontal changes and the other to detect vertical changes in intensity within an image [27]:

$$\boldsymbol{S}_x = \begin{bmatrix} +1 & 0 & -1 \\ +2 & 0 & -2 \\ +1 & 0 & -1 \end{bmatrix} * \boldsymbol{y}(s) \tag{31}$$

$$\boldsymbol{S}_y = \begin{bmatrix} +1 & +2 & -1 \\ 0 & 0 & 0 \\ +1 & -2 & -1 \end{bmatrix} * \boldsymbol{y}(s) \tag{32}$$

where $\boldsymbol{y}$ is the input image, $\boldsymbol{S}_x$ and $\boldsymbol{S}_y$ are the image with the horizontal and vertical gradients, respectively, and $*$ is a convolution operation. The resulting gradient magnitude image is obtained by combining the horizontal and vertical gradient images using the Euclidean norm [27]:

$$S = \sqrt{S_x^2 + S_y^2} \tag{33}$$

This operation highlights regions of significant intensity variation, effectively detecting edges in the image. The Sobel mask is commonly used in edge detection and various other image processing tasks due to its simplicity and effectiveness. Sobel masks belong to the category of first-order masks since they compute the first-order derivative of image intensity with respect to spatial coordinates [27].

In contrast, the Laplace mask computes the second-order spacial derivative of image intensity, having the same mask for vertical and horizontal convolution [27]:

$$L_x = \begin{bmatrix} 0 & +1 & 0 \\ +1 & -4 & +1 \\ 0 & 1 & 0 \end{bmatrix} * y(s) \tag{34}$$

The Sobel and Laplace masks are both widely used for edge detection; however, their approach and characteristics differ, each offering unique advantages and disadvantages. The Sobel mask, which consists of a separate masks for horizontal S_x and vertical gradients S_y, excels at detecting edges with a clear orientation, providing detailed information about the direction of intensity changes within an image. Its structured design makes it robust to noise and suitable for detecting edges in noisy environments. However, the Sobel mask struggles to detect edges at corners or junctions accurately, as it only emphasises the dominant direction of change at each pixel. In contrast, the Laplace mask detects edges regardless of their orientation and is sensitive to abrupt intensity changes, making it effective in this case. Yet, it is more sensitive to noise compared to the Sobel mask, leading to potential false edges detections in noisy images [10,27].

Due to the advantages and disadvantages inherent in the first- and second-order masks, coupled with the established theoretical foundations of fractional calculus, the exploration and refinement of fractional-order masks has been an active and growing field of research [10,61]. These masks offer a balance, leveraging the precision of higher-order derivatives while retaining the adaptability and noise resilience typically associated with lower-order operators.

A problem that persists in fractional-order masks lies in their uniform treatment of the entire image with the same fractional order. This approach can lead to the excessive enhancement of low-spatial frequency content, potentially overshadowing the subtler details within the image, while simultaneously failing to adequately boost high-frequency components [18].

Keeping this in mind, in [18], the authors propose the Adaptive Fractional-order Differential (AFD) mask. This approach uses an NN to dynamically determine the optimal order of differentiation α. The goal is to train an NN to optimise the orders of the mask for any given image.

The AFD mask comprises two 3 × 3 convolutional matrices, horizontal A_x and vertical A_y, derived from the Grünwald–Letnikov derivative definition (see [18] for details on the derivation process):

$$A_x(s) = \begin{bmatrix} 0 & \frac{\alpha_s^2 - \alpha_s}{2} & 0 \\ 0 & -\alpha_s & 0 \\ 0 & 0 & 0 \end{bmatrix} * y(s) \tag{35}$$

$$A_y(s) = \begin{bmatrix} 0 & 0 & 0 \\ \frac{\alpha_s^2 - \alpha_s}{2} & -\alpha_s & 1 \\ 0 & 0 & 0 \end{bmatrix} * y(s). \tag{36}$$

The α_s values for each pixel of the image are determined by an NN with learnable parameters θ. These values are computed from the average gradient of the pixel in eight directions $M(i,j)$. To train the NN, the authors propose using a training dataset generated by employing a piece-wise function that dictates the α_s order of each pixel. This function is specified by the AFD Algorithm (AFDA) [18]:

$$\alpha_s = \begin{cases} \dfrac{M(i,j) - t_g}{M(i,j)} & \text{if } M(i,j) \geq t_g \text{ and } \dfrac{M(i,j) - t_g}{M(i,j)} \geq \dfrac{AG_{ed} - Q}{AG_{ed}} \\ \dfrac{AG_{ed} - Q}{AG_{ed}} & \text{if } M(i,j) \geq t_g \text{ and } \dfrac{M(i,j) - t_g}{M(i,j)} < \dfrac{AG_{ed} - Q}{AG_{ed}} \\ \dfrac{Q - AG_{tex}}{Q} & \text{if } 2 < M(i,j) < t_g \text{ and } \dfrac{M(i,j)}{t_g} \geq \dfrac{Q - M_{tex}}{Q} \\ \dfrac{M(i,j)}{t_g} & \text{if } 2 < M(i,j) < t_g \text{ and } \dfrac{M(i,j)}{t_g} < \dfrac{Q - AG_{tex}}{Q} \\ 0 & \text{if } 0 \leq M(i,j) \leq 2 \end{cases} \tag{37}$$

where t_g is a hyperparameter for the gradient threshold for edges, Q is the mean gradient of image $\boldsymbol{y}$, and AG_{ed} and AG_{tex} are the average gradients of edges and textures, respectively. The computation of $M(i,j)$ is as follows [18]:

$$M(i,j) = \frac{\begin{array}{c}|8\boldsymbol{y}(i,j) - \boldsymbol{y}(i-1,j-1) - \boldsymbol{y}(i-1,j) - \boldsymbol{y}(i-1,j+1) - \\ \boldsymbol{y}(i,j-1) - \boldsymbol{y}(i,j+1) - \boldsymbol{y}(i+1,j-1) - \boldsymbol{y}(i+1,j) - \boldsymbol{y}(i+1,j+1)|\end{array}}{8}. \tag{38}$$

So, to generate the NN training dataset, the AFDA is used to compute the NN's input values, $M(i,j)$, and their corresponding α_s orders, which serve as the ground-truth output. Subsequently, the NN is trained to minimise the error between these ground-truth α_s values and the predicted $\hat{\alpha}_s$ values, for example using the Mean Squared Error [18], as shown in Figure 4.

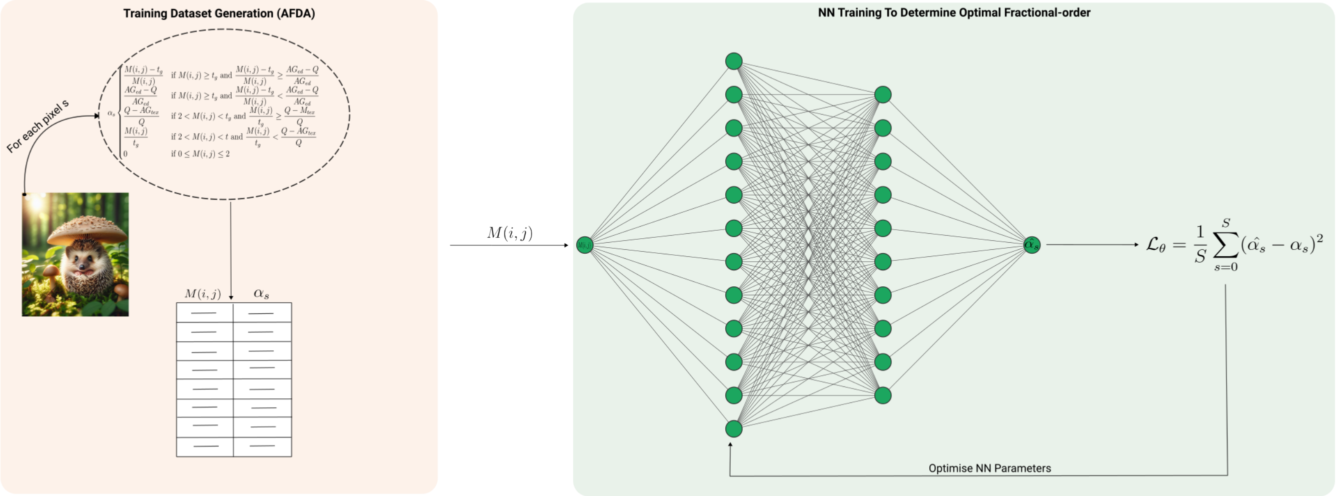
Training Dataset Generation (AFDA)
For each pixel s
$M(i,j)$
α_s
NN Training To Determine Optimal Fractional-order
$M(i,j)$
α_s
$\mathcal{L}_\theta = \frac{1}{S}\sum_{s=0}^{S}(\hat{\alpha}_s - \alpha_s)^2$
Optimise NN Parameters

Figure 4. Training process of Neural Fractional-Order Adaptive Masks

Upon the completion of training, the NN is capable of outputting the α_s orders of the masks to treat each corresponding pixel in the target image, providing the average gradient of the pixel as input [18].

The experimental results presented in [18] demonstrate that the proposed masks yield higher contrast, clearer edges, and enhance smooth areas and texture within the images. Additionally, it is observed that after training the NN to compute the mask orders, the computational requirements for determining these orders are reduced, leading to improved performance compared to using the AFDA by itself.

Although the approach presented in [18] demonstrates improvements over the manual computation of derivative orders, there remains considerable scope for enhancement and exploration. Firstly, the authors employ a relatively simple NN architecture, leaving room for experimentation with more complex architectures that could potentially yield better results. Additionally, the rationale behind the choice of the Grünwald–Letnikov definition is not explicitly addressed by the authors, which poses an open question for further investigation. Furthermore, the learned derivative orders may be constrained due to the training dataset being generated by the AFDA, which could limit the NN's ability to learn optimal orders without extensive prior knowledge input. This suggests a need to explore alternative approaches to dataset generation and NN training to broaden the capabilities and flexibility of the model.

3.2.2. Fractional Rényi Entropy

Rényi entropy $R_\alpha(\boldsymbol{y})$ [62] is a measure of uncertainty or randomness within a probability distribution, commonly employed in information theory and computer vision to quantify diversity and uncertainty in pixel intensities or image features. For a greyscale image $\boldsymbol{y}$, the Rényi entropy is defined as [63]

$$R_\alpha(\boldsymbol{y}) = \frac{1}{1-\alpha} \log\left(\sum_{s=0}^{255} p_s^\alpha\right), \quad \alpha \in [0, \infty] \tag{39}$$

where p_s is a normalised histogram of pixel intensities, and $\alpha \in \mathbb{N}$ is a parameter governing the focus on different parts of the probability distribution. When $\alpha \to 1$, Rényi entropy reduces to Shannon entropy [64], indicating overall uncertainty in the image. For $\alpha \neq 1$, Rényi entropy highlights various aspects of the distribution's structure. As $\alpha \to 0$, it emphasises the most frequent pixel intensities akin to min-entropy [65], suitable for capturing dominant image features. Conversely, as $\alpha \to \infty$, it emphasises rare pixel intensities akin to max-entropy [65], which is valuable for detecting subtle textures or anomalies in the image [63].

Fractional Rényi entropy extends the concept of Rényi entropy by extending α to non-integer values, allowing for a more refined characterisation of the uncertainty and complexity within a probability distribution. This generalisation enables a spectrum of entropy measures between the limit cases of $\alpha = 1$ and $\alpha = 0$ [60].

In [60], the authors point out that image contrast and quality are major factors in the quality of image segmentation techniques. To address this, the authors propose using fractional Rényi entropy for image enhancement before employing a CNN for segmentation, as shown in Figure 5.

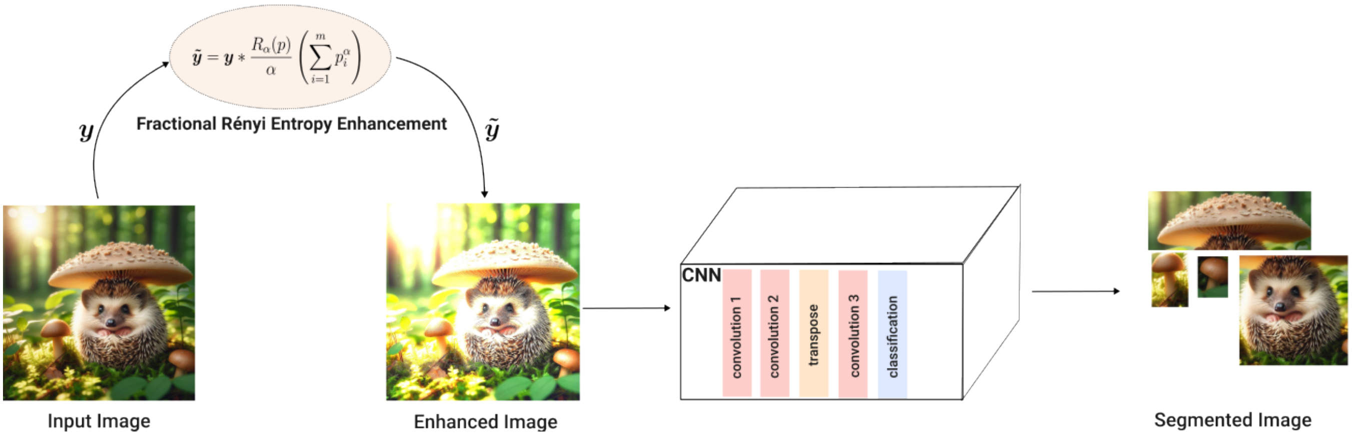

Figure 5. Image enhancement with fractional Rényi entropy before using a CNN for image segmentation.

The enhanced image $\tilde{\boldsymbol{y}}$ is obtained from the input image $\boldsymbol{y}$ through pixel-wise multiplication formulated as follows (for details, see [60]):

$$\tilde{\boldsymbol{y}} = \boldsymbol{y} * \frac{R_\alpha(p)}{\alpha}\left(\sum_{i=1}^{m} p_i^\alpha\right). \tag{40}$$

The fractional-order α is determined experimentally on the training dataset [60].

The findings in [60] suggest that employing fractional Rényi entropy enhances the robustness of the model against inhomogeneous intensity values and preserves spatial relationships between image pixels. Although promising, several questions remain unanswered, such as whether there are less time-consuming strategies for selecting the α value and whether the α value could be adaptively chosen for each image region. Furthermore, fractional Rényi entropy can provide advantages to other well-established networks, such as Transformers. Addressing these questions could further improve the effectiveness and efficiency of the proposed approach.

3.3. Object Detection

Object detection is the process of locating and classifying objects within digital images or video frames. The primary objective of object detection is to accurately identify and locate instances of predefined object classes within the image, using bounding boxes or outlines delineating their positions [28]. Then, a label or category is assigned to each detected object [66].

FC for object detection remains a relatively unexplored frontier, with limited research available in this area. Existing work focuses on exploiting fractional-order moments for feature extraction [67] and employing fractional-order populational and evolutionary optimisation strategies to refine the localisation of objects within images [68].

The field of object detection has undergone a revolutionary transformation with the widespread adoption of NNs, enabling the autonomous identification and location of objects. Numerous research efforts have propelled the development of several NN architectures with enhanced accuracy, faster inference speeds, or reduced computational costs. Among the most widely used architectures is the region-based convolutional neural network (RCNN) family [69], which includes variants such as Fast RCNN [70] and Faster RCNN [71]. These leverage a blend of convolutional layers for feature extraction and region proposal algorithms to pinpoint potential object locations. By scrutinising these regions individually, they are able to classify objects and predict bounding boxes. Moreover, the emergence of one-stage detectors, such as YOLO (You Only Look Once) [72] and SSD (Single-Shot Detector) [73] has enabled real-time object detection by directly predicting object classes and bounding boxes in a single pass. Through the usage of extensive datasets annotated with object labels, NNs acquire the ability to generalise across diverse object categories and accommodate variations in scale, orientation, and occlusion, thereby solidifying their indispensable role across applications spanning satellite surveillance [74,75], public parking management [76], robotics [77], and pest management [78].

The application of FC in NN-based techniques for object detection is still relatively new but shows promising results [67,79–81].

In [79], inspired by previous work [67], the authors propose leveraging fractional-order Legendre moments for feature extraction. These features are then used by an NN for object detection based on the extracted feature maps.

Additionally, in [80,81], the authors propose fractional-order population-based and evolution-based optimisation algorithms to improve the optimisation process of NN parameters. However, since this survey paper only focuses on methodologies that directly modify the NN architecture or preprocess input images before feeding them into the NNs, these works are not be covered.

Fractional-Order Legendre Moment Invariants

Image moments are mathematical descriptors that are used to characterise the spatial distribution and properties of intensity values within an image. They are computed by integrating the intensities of the pixels in an image $y(i,j)$, where i and j are the spatial coordinates. The image moment $M_{p,q}$ is defined as [82]

$$M_{p,q} = \sum_{i=1}^{I} \sum_{j=1}^{J} i^p j^q y(i,j), \tag{41}$$

where p and q are non-negative integers representing the order of the moment. These moments provide insights into various image attributes, such as the centroid, area, orientation, and higher-order shape characteristics. Since these are translation dependent, central moments $\mu_{p,q}$ are often preferred, as they are invariant to translation [82]:

$$\mu_{p,q} = \sum_{i=1}^{I} \sum_{j=1}^{J} (i - \bar{i})^p (j - \bar{j})^q y(i,j), \tag{42}$$

where $\bar{i}$ and $\bar{j}$ are the centroid of an image computed as $\bar{i} = \dfrac{M_{10}}{M_{00}}$ and $\bar{j} = \dfrac{M_{01}}{M_{00}}$. Image moments are widely used for shape analysis and object recognition, as they provide valuable information about the location, size, and centres of objects within an image.

There are several moments commonly used in image analysis, each with unique properties and applications. Integer-order moments, such as Zernike, Legendre, and Chebyshev moments, are among the most widely employed due to their effectiveness in capturing different aspects of image content [82].

Legendre moments are a class of orthogonal moments used in image analysis to capture shape information and structural features within an image. These moments are derived from Legendre polynomials and are computed by integrating the pixel intensities of the image weighted by Legendre polynomials of integer degree, mathematically formulated as [82]

$$L_{pq} = \sum_{i=1}^{I} \sum_{j=1}^{J} P_p(i) P_q(j) y(i,j), \tag{43}$$

where $P_p(i)$ and $P_q(j)$ are Legendre polynomials of degree p and q, respectively. Legendre moments offer advantages such as orthogonality, compactness, and rotational invariance, making them well suited for tasks such as pattern recognition, shape analysis, and image retrieval. Additionally, their robustness to noise and illumination variations enhances their utility in real-world applications. Legendre moments also provide a concise representation of image content while preserving important geometric and structural information, contributing to the development of efficient and effective image-processing techniques [83]. The Legendre polynomials $P_p(i)$ can be computed using a recurrence formula given by

$$P_{p+1}(i) = \frac{(2p+1)(2i-1)}{p+1} L_p(i) - \frac{p}{p+1} L_{p-1}(i), \;\; p \geq 1, \tag{44}$$

with $P_0(i) = 1$ and $P_1(i) = 2i - 1$. $P_q(j)$ can be computed reciprocally.

Although integer-order moments offer valuable insights into the overall shape and spatial distribution, they may lack the sensitivity required to accurately represent intricate features [67,83].

Fractional-order moments extend the concept of integer-order moments to non-integer values of α and β, offering a more refined characterisation of image properties. These moments $M_{\alpha,\beta}$ are formulated as

$$M_{\alpha,\beta} = \sum_{i=1}^{I} \sum_{j=1}^{J} i^{\alpha} j^{\beta} y(i,j), \tag{45}$$

where α and β are non-integer values. Fractional-order moments offer enhanced sensitivity to subtle variations in image structure and texture, enabling more precise analysis and interpretation. This heightened sensitivity allows for improved accuracy in locating Regions of Interest within the image [67].

Fractional-order Legendre moments are an extension of integer-order Legendre moments (43). Through allowing non-integer values for the degree of Legendre polynomials, these moments provide a more flexible and adaptive framework for image analysis. The fractional Legendre moment $L_{\alpha,\beta}$ is formulated as [67]

$$L_{\alpha,\beta} = \sum_{i=1}^{I}\sum_{j=1}^{J} P_\alpha(i)P_\beta(j)y(i,j), \tag{46}$$

where the Legendre polynomials $P_\alpha(i)$ can be computed using a generalisation of recurrence Formula (47) and introducing the change in variable $i = 2t^a - 1$ (for details, see [79]):

$$P^a_{\alpha+1}(t) = \frac{(2\alpha+1)(2t^a-1)}{\alpha+1} L^a_\alpha(t) - \frac{\alpha}{\alpha+1} P^a_{\alpha-1}(t), \;\; \alpha \geq 1, \tag{47}$$

with $P^\alpha_0(t) = 1$ and $P^\alpha_1(t) = 2t^\alpha - 1$. $P^\beta_{q+1}(t)$ can be computed reciprocally.

The fractional-order Legendre moments can be extended to be used in three dimensions, enabling the representation of the features of three-dimensional (3D) images much used in medical imaging. Thus, the 3D fractional Legendre moment $L_{\alpha,\beta,\gamma}$ can be formulated as

$$L_{\alpha,\beta,\gamma} = \sum_{i=1}^{I}\sum_{j=1}^{J}\sum_{k=1}^{K} P_\alpha(i)P_\beta(j)P_\gamma(k)y(i,j,k), \tag{48}$$

where $P_\alpha(i)$, $P_\beta(j)$, and $P_\gamma(k)$ are the fractional-order Legendre polynomials of degrees α, β, and γ along the i, j, and k axes, respectively, and $y(i,j,k)$ is a 3D image.

Motivated by the application of fractional-order moments for the classification of 2D objects, in [79] the authors propose using 3D fractional-order moments (for formulation details, see [79]) as an input descriptor of an NN, thus giving rise to a new NN architecture, the Fractional-Order Lagrange Moments Deep NN (FrOLM-DNN).

The 3D fractional-order Legendre moments serve as an input descriptor for an NN. For each image, the moments are computed, forming a descriptor vector containing moments up to order r, where r is a user-selected hyperparameter. Subsequently, this descriptor vector is fed into the input layer of an NN, enabling the network to learn and classify the object within the image accurately. In this way, the 3D fractional-order Legendre moments function as feature extractors, facilitating effective object classification [79], as shown in Figure 6.

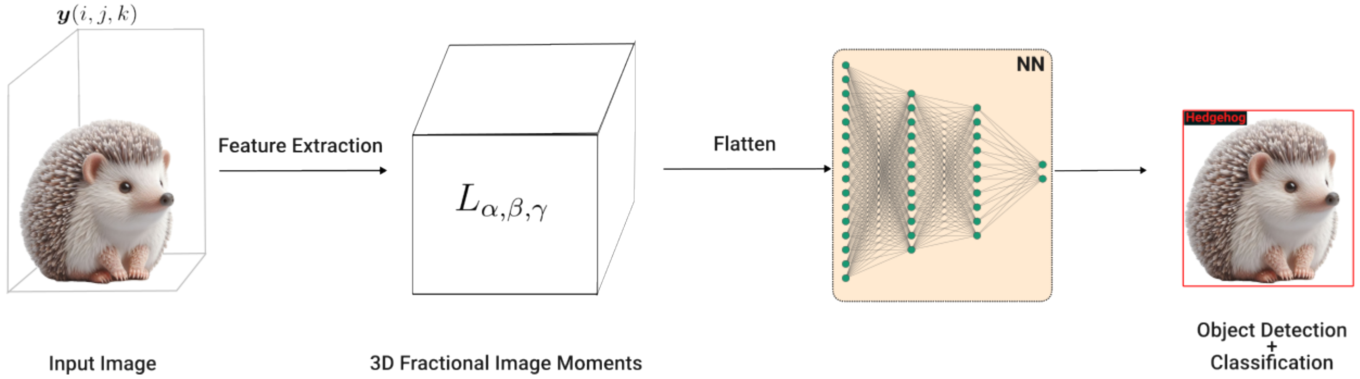

Figure 6. Architecture of FrOLM-DNN for object detection and classification of 3D image (input image generated by DALL-E 3).

The main motivation for employing fractional-order moments lies in their additional parameters, which offer the potential for improved results tailored to specific use cases. The experimental results described in [79] demonstrate that the integration of 3D fractional-order moments with an NN leads to improved classification accuracy compared to using them in isolation.

However, an open question raised in [79] is the impact of selecting the moments order r on the results and how to effectively determine the value of this hyperparameter. One potential avenue for future research involves devising a strategy to automate and optimise the selection of this parameter. By developing such a strategy, researchers can streamline the process of hyperparameter tuning and potentially improve the overall performance of the classification system.

3.4. Segmentation

Image segmentation is the process of partitioning a digital image into multiple segments or regions based on certain criteria such as colour, intensity, texture, or spatial proximity. The goal is to break an image into smaller parts; these segments often correspond to objects or regions of interest within the image, allowing for further analysis or manipulation at a more granular level [28].

The development of FC methods for image segmentation is a well-explored and rising field, with various approaches explored in the recent literature. Some methods employ fractional-order optimisation algorithms to refine segmentation accuracy, while others introduce novel loss functions incorporating fractional orders. Additionally, established segmentation algorithms have been extended to accommodate fractional-order operations, enhancing performance on images with intricate textures [12,84–88].

CNNs are particularly well suited for segmentation tasks due to their ability to effectively capture spatial dependencies and hierarchies of features within images. One of the most well-known architectures is U-Net, which employs an encoder–decoder structure with skip connections to preserve spatial information and enhance segmentation accuracy [89]. Through training on annotated datasets, NNs learn to delineate boundaries and assign pixel-level labels to different regions within an image, effectively segmenting objects from the background or distinguishing between different object classes. Several successful applications can be found in the literature, including the medical analysis of chest scans [90], fingerprint security devices [91], and the detection of road cracks [92].

While some research endeavours have incorporated FC into NN-based methods for image segmentation, the majority have primarily relied on fractional-order population-based optimisation algorithms to enhance the optimisation process of NN parameters [93,94]. However, this survey paper focuses on methodologies that directly modify the NN architecture or preprocess input images before feeding them into the NNs. In light of this, we highlight two notable works [17,95].

In [17], the authors use FOCNet [42] for image segmentation. In contrast, [95] proposes the use of fractional-order differentiation active contour models for segmentation, employing an NN to solve fractional-order PDEs and thereby reduce the computational complexity associated with fractional-order active contour models.

3.4.1. Active Contour Detection with Fractional-Order Regularisation Term

Active contour models are tools particularly useful for segmenting objects with complex or ambiguous boundaries. These models are represented as a parametric curve or contour that evolves over time to minimise an energy functional. The contour is attracted to features of interest in the image while being constrained by factors such as smoothness and shape.

Level-set functions (LSF) are improved active contours that implicitly represent the contour as the zero level set of a higher-dimensional function defined over a larger domain that includes the entire image. The evolution of the contour is described by the evolution of the level-set function, governed by a fractional-order partial differential equation (FPDE). The level-set methods evolve smoothly over the entire domain, and the contour is extracted as the zero level set at each time step. Level-set methods offer advantages in handling topological changes and complex contour deformations, making them suitable for tasks where the object boundaries are ill defined or undergo significant changes over time [95].

Despite their differences, level-set methods and active contour models share the goal of accurately segmenting images by evolving contours to capture object boundaries.

Level-set methods can be seen as a generalisation of active contour models, where the contour evolution is described implicitly through the evolution of a level-set function. This connection allows for the incorporation of active contour energy terms into level-set formulations, enhancing their versatility, robustness, and adaptability [95].

Variational level-set methods extend basic level-set techniques by incorporating variational principles into the formulation, enabling the optimisation of a variational energy functional to evolve contours over time. Through formulating the segmentation problem as an optimisation task, variational level-set methods provide a systematic framework for integrating various constraints, prior knowledge, and image features into the segmentation process. This approach offers improved convergence, stability, and flexibility, and has been successfully applied to challenging segmentation problems involving complex object shapes, noisy images, and topological changes [96].

The first instance of combining FDEs with active contour methods was introduced in [97]. In this work, a fractional-order differentiation active contour model was proposed, employing variational level-set methods. The energy function proposed in this model comprises three terms: a fractional-order fitting term, a regularisation term, and a penalty term. The fractional-order term enables a more precise representation of the image and improves the robustness to noise, while the penalty term ensures stable evolution [95,97].

Despite these advantages, this method incurs a significant computational burden [95]. To address computational cost, [95] proposes using cellular neural networks (CeNNs) [98] to solve FPDEs instead of finite difference numerical schemes (see [95] for the mathematical formulation). CeNNs, which are composed of locally connected neurons arranged in a grid-like structure, offer stability, noise robustness, and time efficiency in computing solutions to FPDEs for active contour methods [95].

Due to the usage of FPDEs, which describe the temporal evolution of a system, an open area of research is the application of this approach for object tracking in temporal image sequences. Additionally, considering the popularity of physics-informed neural networks [99], it would be interesting to explore replacing CeNNs with this architecture in future studies.

3.4.2. FOCNet for Segmentation

In [17], the authors recognise the potential of FOCNet beyond denoising tasks. They propose to use the weighted skip connections of FOCNet to improve the image segmentation performance, as shown in Figure 2. To facilitate segmentation using FOCNet, the authors suggest employing the Dice Coefficient as the loss function, which quantifies the similarity between the ground truth and predicted segmentation masks. Hence, the distinction between FOCNet as proposed in [42] and in [17] lies in the choice of loss function and the nature of the training data. For denoising tasks, the input comprises a noisy image, with the corresponding ground truth being the denoised version of the image [42]. Conversely, in segmentation tasks, the input consists of the original image, while the ground truth represents the segmented image [17].

The usage of FOCNet for segmentation has demonstrated computational efficiency and outperformed other segmentation methods. The presence of fractional derivatives that determine the weights of the connections enables the propagation of information from shallower layers [17].

3.5. Restoration

Image restoration is the process of recovering an undistorted version of a digital image from a damaged or corrupted input. Degradation in images can occur due to various factors such as sensor limitations, time degradation, or environmental factors. The goal is to complete the degraded images, making them suitable for analysis, interpretation, or presentation purposes. Restoration/inpainting techniques aim to reverse or mitigate the effects of degradation by applying mathematical models, filters, or learning-based algorithms to estimate and recover the original image underlying.

The usage of FC for enhancing image restoration and inpainting has received considerable attention in the literature, with numerous studies showcasing its performance advantages. These works demonstrate the efficacy of FC in modelling the intricate spatial and temporal variations present in image structures, leveraging information from surrounding areas to achieve more accurate restoration and inpainting results. By seamlessly integrating fractional-order techniques, these approaches effectively capture the nuanced in-between behaviours often overlooked by traditional integer-order methods [100–104].

The field of restoration has risen in popularity mostly due to architectures based on Variational Autoencoders (VAEs) [105] and Generative Adversarial Networks (GANs) [106] that offer novel approaches to image reconstruction. VAEs are renowned for their ability to learn rich probabilistic models of data, enabling them to effectively capture complex distributions in image space. By encoding input images into a latent space and then decoding them back to their original form, VAEs can learn to inpaint images while also providing uncertainty estimates [107]. On the other hand, GANs introduce a competitive training scheme between a generator network and a discriminator network, resulting in the generation of highly realistic images. In the context of restoration, GANs excel in producing visually convincing results by learning intricate details and textures from training data [108]. The success of NN architectures in image restoration has sparked various applications, including restoring damaged historical texts [109], removing unwanted objects [110], and the completion of medical scans [111].

The combination of FC and NN-based methods for image restoration/inpainting is almost unexplored, with only one work available [112]. Given the potential already demonstrated in the literature with the several works that are not using NNs, it comes easily that this is a promising field.

In their work [112], the authors introduce an innovative approach that leverages fractional-order wavelet transforms to enhance feature extraction within an NN encoder. Their methodology extends beyond traditional techniques by integrating fractional-order operations into the encoding process, enabling more nuanced and comprehensive feature representation. Moreover, the authors propose a novel strategy for image generation, where multiple fractional-order encoders are employed to produce different representations of the same image. These representations are subsequently merged to create a single composite image, with enhanced detail and richness.

Fractional Wavelet Scattering Networks

GANs [106] and VAEs [105] represent two influential paradigms in the realm of generative modelling, a field aiming to understand and replicate the underlying structure of data. GANs operate on a game-theoretic framework where a generator network competes against a discriminator network, iteratively improving the generation of data until it becomes indistinguishable from real data. On the other hand, VAEs adopt a probabilistic approach, aiming to encode and decode data by modelling the underlying probability distribution. While GANs excel in generating high-quality, realistic samples, VAEs offer a principled way to learn latent representations of data and perform tasks such as data compression and synthesis. Both methods have witnessed tremendous advancements and applications across various domains, revolutionising tasks like image generation, data augmentation, and anomaly detection.

GANs and VAEs are notorious for their challenging training dynamics, including unstable training and issues like blurred images or model collapse. To address these challenges, Generative Scattering Networks (GSNs) were introduced, leveraging wavelet scattering networks (ScatNets) as encoders and CNNs as decoders [113].

ScatNets are mathematical models designed to extract meaningful features from signals, particularly in the domain of generative modelling. They operate by performing a cascade of operations on the input signal $\boldsymbol{y}(t)$ in each layer of the ScatNet S_n: wavelet transform, in which $\boldsymbol{y}(t)$ is convolved with wavelet function $\psi_\lambda(t)$ at different scales λ, $S_1\boldsymbol{y}(t) = |\boldsymbol{y}(t)\psi_\lambda(t)|$; modulus nonlinearity, in which after each wavelet transform, the modulus operation is applied element-wise $S_n\boldsymbol{y}(t) = |S_{n-1}\boldsymbol{y}(t)|$; and pooling, in which pooling operations are performed to aggregate information across scales $S_{pool}\boldsymbol{y}(t) = T_{\uparrow}|S_n\boldsymbol{y}(t)|$. Through this hierarchical process, ScatNets create a series of increasingly invariant and abstract representations of the input signal, capturing both local and global structures.

After extracting the features with a ScatNet, Principal Component Analysis (PCA) is applied to reduce the dimensional resulting in the latent space vector $\boldsymbol{z}$. Then, the decoder uses a CNN for deconvolution and to output the predicted image $\tilde{\boldsymbol{y}}$, as shown in Figure 7.

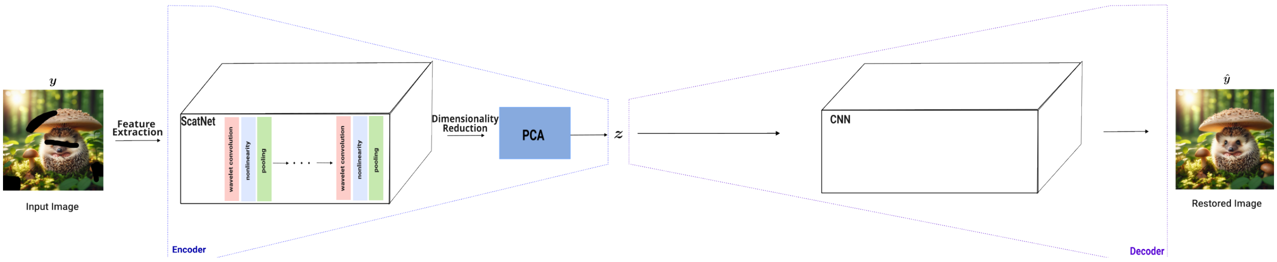

Figure 7. Architecture of a GSN with a ScatNet and PCA encoder, and a CNN decoder.

GSNs simplify training by avoiding the need to learn ScatNet parameters, yet they may suffer from reduced image quality due to limitations in ScatNets' expressiveness and overfitting induced by PCA in dimensionality reduction [112].

In response to these limitations, Generative Fractional Scattering Networks (GFRSNs) were proposed as an extension of GSNs in [112]. GFRSNs aim to address the overfitting issue by introducing a more suitable dimensional reduction method, thus enhancing GSN performance.

GFRSN includes an encoder composed of novel components, a Fractional Wavelet Scattering Network (FrScatNet) and a Feature Map Fusion (FMF) dimensional reduction method [112].

FrScatNet extends the concepts of ScatNets to non-integer order wavelet transforms by introducing a fractional convolution operator Θ_α to the wavelet transform operation formulated as [112]

$$S_1\boldsymbol{y}(t) = |\boldsymbol{x}(t)\Theta_\alpha\psi_\lambda| = e^{-\frac{j}{2}t^2\cot(\theta)[\boldsymbol{y}(t)e^{\frac{j}{2}t^2\cot(\theta)}\psi_\lambda(t)]}, \tag{49}$$

where α is the fractional order, and $\theta = \dfrac{\alpha\pi}{2}$ is the rotation angle. Note that FrScatNet can be reduced to a ScatNet when $\alpha = 1$.

In [112], after employing FrScatNet for feature extraction, the authors propose using an FMF method instead of PCA, as used in ScatNets. The rationale behind this choice is that PCA fails to consider the semantic differences present in the features extracted by FrScatNets across different layers, thus overlooking the hierarchical information contained within the features. The reduced dimensional feature map after applying FMF to the features extracted by FrScatNet is the latent space $\boldsymbol{z}$ (for more details on FMF, see [112]).

FrScatNets are equipped with a hyperparameter α that determines the fractional order of the convolution. Since α can be chosen arbitrarily, different α values lead to the extraction of distinct features. Consequently, using FrScatNets with varying α values results in the generation of multiple feature vectors. To fully exploit this diversity in feature extraction, the authors propose merging the image predictions obtained from different α values using an image fusion technique. This approach allows for embedding the input in different fractional-order domains to enhance the quality of the generated images. The image fusion method proposed can be formulated as [112]

$$\tilde{\boldsymbol{y}}_{\alpha_1,\alpha_2} = \omega\tilde{\boldsymbol{y}}_{\alpha_1} + (1-\omega)\tilde{\boldsymbol{y}}_{\alpha_2}, \tag{50}$$

where $\tilde{\boldsymbol{y}}_{\boldsymbol{\alpha}_1}$ and $\tilde{\boldsymbol{y}}_{\boldsymbol{\alpha}_2}$ are image predictions generated by feature extraction using an FrScatNet with fractional-orders α_1 and α_2, respectively. The hyperparameter ω acts as a weighting factor that determines the contribution of each predicted image [112], Figure 8.

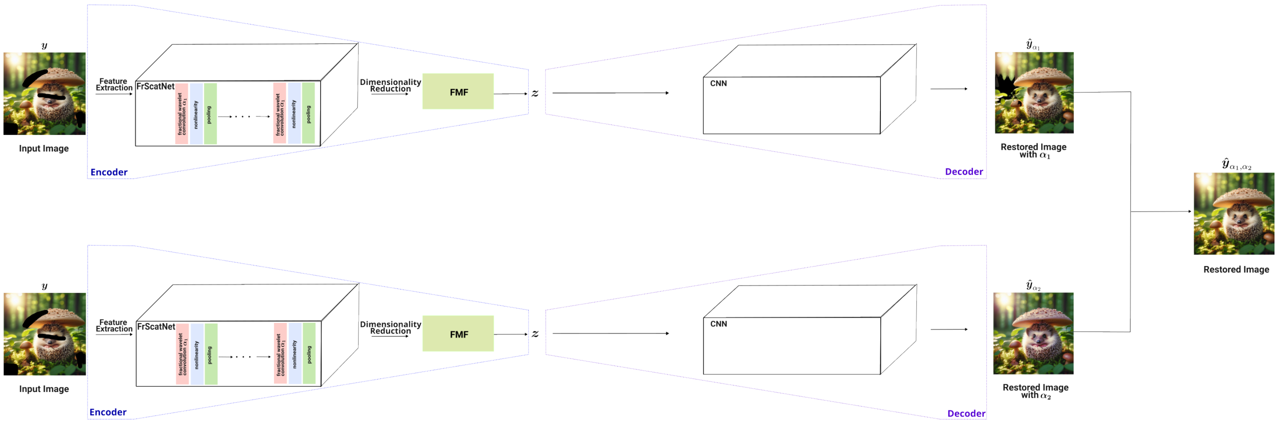

Figure 8. Architecture of a GFRSN with an FrScatNet and FM encoder, and a CNN decoder. Using two GFRSNs with different fractional-orders, one can enhance the predicted image $\widetilde{y}$ by merging the outputs from both orders, $\widetilde{y}_{\alpha_1}$ and $\widetilde{y}_{\alpha_2}$.

The introduction of GFRSN in [112] opens up several research gaps that warrant further investigation. First, there is a need to explore the methodologies to select the appropriate α value for FrScatNets. Identifying which α values yield the most distinct features would be beneficial, as combining such features could potentially enhance the completion of the predicted output. Furthermore, while the proposed FMF method takes the average of the third-layer's feature maps, there is room for exploring more sophisticated weighting strategies. Using alternative weight strategies could potentially improve the efficacy of the dimensionality reduction technique used in GFRSNs.

3.6. Compression

Contemporary advancements in deep learning have propelled the field to achieve remarkable performances across a spectrum of tasks, encompassing image classification, semantic segmentation, object detection, pose detection, and beyond. This progress is underpinned by the evolution of increasingly complex architectures, housing millions, and even billions of trainable parameters, thereby augmenting model efficacy. However, the proliferation of such massive architectures poses challenges concerning memory and computational resources, hindering seamless deployment in edge computing devices and other resource-constrained environments.

Reducing the computational cost of NN architectures for denoising, image restoration, segmentation, object detection, and enhancement is a critical endeavour in computer vision research. Given the increasing demand for real-time and resource-efficient solutions, optimising these architectures for computational efficiency is essential for practical deployment in various applications. Techniques for reducing computational cost typically focus on minimising the number of operations, parameters, or memory footprint required by the models while maintaining satisfactory performance.

In our survey of the literature, we uncovered two works that harness FC to alleviate the computational burden associated with NN-based computer vision tasks [19,114].

In [114], the authors introduce fractional max-pooling (FMP), a novel technique designed to address the limitations of traditional max-pooling methods. Through incorporating fractional-order principles, fractional max-pooling allows for overlapping windows and mitigates information loss during the pooling process, thereby enhancing feature preservation while reducing the reduction size. Conversely, in [19], the authors present a groundbreaking approach by deriving fractional convolutional filters tailored for popular integer-order filters such as Gaussian, Sobel, and Laplacian. This innovative formulation not only improves filter performance but also offers a compelling advantage: a reduction in the number of parameters is required, which remains constant regardless of filter size. Moreover, these fractional filters exhibit intermediate behaviours, providing a versatile tool for capturing nuanced features within the image data.

3.6.1. Fractional Max-Pooling

Max-pooling is a fundamental operation in CNNs used for downsampling feature maps, reducing computational complexity, and extracting dominant features. Given an input feature map with dimensions $I_{in} \times J_{in} \times D_{in}$, where I_{in}, J_{in}, and D_{in} are the width, height, and number of channels, respectively, max-pooling partitions the input into non-overlapping regions and computes the maximum value within each region to produce the output [114].

Let $l \times l$ denote the size of the pooling window, where l is typically a small integer; usually, $l = 2$ is used as default. The output dimensions of the feature map after max-pooling are given by $I_{out} = \left\lfloor \frac{I_{in}}{l} \right\rfloor$ and $J_{out} = \left\lfloor \frac{J_{in}}{l} \right\rfloor$. The pooling regions $P_{i,j}$ are computed by dividing the input feature map $I_{in} \times I_{in}$ by the size of the output feature map $I_{out} \times I_{out}$ (considering $J_{in} = I_{in}$ and $J_{out} = I_{out}$) [114]:

$$P_{i,j} \subset 1, 2, \ldots, {I_{in}}^2 \text{ with } (i, j) \in 1, \ldots, {I_{out}}^2, \tag{51}$$

where the size of the pooling windows can be computed as

$$P_{i,j} = [2i - 1, 2i] \times [2j - 1, 2j]. \tag{52}$$

Then, for each pooling region, the maximum pixel value of the input feature map $In_{P_{i,j}}$ will be kept and used to compose the output feature map $Out_{i,j}$:

$$Out_{i,j} = \max(In_{P_{i,j}}). \tag{53}$$

Max-pooling introduces translation invariance and reduces the spatial dimensions of the feature maps, which helps in controlling overfitting and improving computational efficiency. Through retaining only the maximum activations within each pooling region, max-pooling focuses on preserving the most salient features while discarding less relevant information, facilitating hierarchical feature learning in CNNs [114].

The limitations of max-pooling in CNNs are indeed well recognised in the literature [115–117]. One major drawback is its fixed pooling window size, which may not adequately capture diverse spatial patterns within feature maps, especially in scenarios where objects vary significantly in scale or orientation. Additionally, the non-overlapping nature of max-pooling leads to a loss of spatial information between adjacent pooling regions, potentially discarding valuable details crucial for tasks like object localisation. Moreover, default max-pooling results in a rapid reduction in the size of hidden layers $\frac{I_{in}}{I_{out}} \approx 2$, necessitating the use of multiple stacked convolutional layers to achieve significant depth. While some methods have been proposed to address this issue [115,116], they still result in a halving of the size of hidden layers, highlighting the need for a gentler approach to spatial pooling [114].

A potential solution lies in adopting a more flexible approach to pooling that reduces the size of hidden layers by a smaller factor. By incorporating additional layers of pooling, each with a smaller reduction factor, we can observe the input image at different scales, potentially leading to an easier recognition of distinctive features indicative of specific object classes. This approach could lead to more effective feature extraction and improve the performances of CNNs in various computer vision tasks [114].

In [114], the authors introduce FMP to address the issue of controlling the reduction in the spatial size of images by a fractional-order $1 < \alpha < 2$. Additionally, FMP introduces flexibility by allowing overlapping pooling regions, thus preserving spatial information more effectively. The pooling regions $P_{i,j}$ in FMP can either be overlapping squares or disjoint collections of rectangles. To generate $P_{i,j}$, two hyperparameters are needed: a_i and b_j, with $i, j \in 0, \ldots, I_{out}$. In considering a_i and b_j as two increasing sequences with a step of 1 ending at I_{in}, then the overlapping pooling regions can be computed as [114]

$$P_{i,j} = [a_{i-1}, a_i - 1] \times [b_{j-1}, b_j - 1], \tag{54}$$

and the disjoint pooling regions as

$$P_{i,j} = [a_{i-1}, a_i] \times [b_{j-1}, b_j]. \tag{55}$$

The output dimensions of the feature map after FMF are given by $I_{out} = \left\lfloor \frac{I_{in}}{\alpha} \right\rfloor$ and $J_{out} = \left\lfloor \frac{J_{in}}{\alpha} \right\rfloor$, where α is the fractional-order of reduction $\frac{I_{in}}{I_{out}} \in (1, 2)$ [114].

FMP holds promise in enhancing the performance of CNNs as well as in reducing their computational cost without losing information. Several research directions stemming from this work [114] could be pursued. For instance, investigating optimal combinations of reduction ratios and overlap factors in FMF could lead to improved feature representation while minimising information loss. Additionally, exploring the integration of FMF with other techniques, such as attention mechanisms or adaptive pooling strategies, may result in synergistic improvements in both model performance and efficiency.

3.6.2. Fractional Convolutional Filters

In [19], authors propose learning reduced representations of convolutional filters through combining fractional calculus and NNs, originating fractional convolutional filters.

As seen previously, convolutional filters are used in NNs to extract features from images, with the most popular filters being Gaussian, Sobel and Laplacian. These are intimately connected through the derivatives of a Gaussian filter. The first derivative of the Gaussian corresponds to the Sobel operator, emphasising edges in a particular direction, while the second derivative of the Gaussian, known as the Laplacian of the Gaussian, enhances areas of rapid intensity change regardless of direction. Mathematically, the relationship can be expressed as follows [19]:

$$G(i,j) = \frac{1}{2\pi std^2} e^{-\dfrac{i^2+j^2}{2std^2}}, \quad \frac{\partial G(i,j)}{\partial i} \propto j e^{-\dfrac{i^2+j^2}{2std^2}}, \quad \frac{\partial^2 G(i)}{\partial i^2} = 4i^2 e^{-i^2} - 2e^{-i^2}, \tag{56}$$

where std is the standard deviation.

Mathematically, using the Grünwald–Letnikov definition and considering the first 15 terms of the Taylor series, the authors extend this to the fractional-order derivative of order α of the Gaussian $\mathcal{D}^\alpha G(i,j)$ [19]:

$$\mathcal{D}^\alpha G(i) = \frac{A}{h^\alpha} \sum_{m=0}^{15} \frac{\Gamma(\alpha+1) G(i)}{(-1)^m \Gamma(m+1) \Gamma(1-m+\alpha)}, \tag{57}$$

where Γ is the Gamma function.

The fractional derivative of a Gaussian provides a versatile framework for deriving a range of filters, including Gaussian, Sobel, Laplacian, and more. By selecting specific values for the fractional order α, we can directly obtain the corresponding traditional filters: $\alpha = 0$ yields the Gaussian filter; $\alpha = 1$, the Sobel filter; and $\alpha = 2$, the Laplacian filter. Remarkably, by varying α, we can interpolate between these filters, creating a continuum of filter behaviours that smoothly transition between them. This flexibility enables the creation of general customised filters whose behaviour ranges between the previously referred filters [19]. A 2D fractional convolutional filter is thus defined as

$$F(i,j) = A\mathcal{D}^\alpha \mathcal{D}^\beta e^{-\dfrac{(i-i_o)^2 + (j-j_o)^2}{std^2}}, \tag{58}$$

where $\mathcal{D}^\alpha \mathcal{D}^\beta G(i,j) = \mathcal{D}^\alpha G(i) \times \mathcal{D}^\beta G(j)$. The 3D fractional convolutional filter is given by

$$F(i,j,k) = A\mathcal{D}^\alpha \mathcal{D}^\beta \mathcal{D}^\gamma e^{-\dfrac{(i-i_o)^2 + (j-j_o)^2 + (k-k_o)^2}{std^2}}, \tag{59}$$

where α, β, and γ are the orders of the fractional derivatives, and std, A, i_o, j_o, and k_o are the parameters that define the filter.

One significant computational benefit of fractional convolutional filters is that the number of parameters required to describe them remains constant, regardless of the filter size, offering a substantial reduction in the number of parameters compared to integer-order filters. For instance, instead of needing $l \times l$ parameters for an $l \times l$ pixel-wise dimension filter, fractional convolutional filters require only a maximum of 6 (2d) or 9 (3d) parameters. This reduction in parameter count simplifies the model and enhances computational efficiency, making fractional convolutional filters an attractive option for various applications in image processing and computer vision [19].

An important result from [19] is that the introduction of fractional convolutional filters in CNNs allowed us to achieve a record for the smallest model that could achieve an accuracy greater than 99% on the MNIST dataset [19]. In building upon this milestone, several promising research directions emerge. Firstly, exploring the applicability of fractional convolutional filters across diverse datasets and domains can unveil their generalisability and robustness. Furthermore, delving into the theoretical foundations and mathematical properties of fractional convolutional filters may lead to deeper insights into their underlying mechanisms and ease the development of more sophisticated variants.

4. Conclusions

In this survey paper, we conducted an extensive exploration of the integration of fractional calculus with neural network-based computer vision methodologies, focusing on denoising, enhancement, object detection, segmentation, restoration, and neural network compression tasks; see a summary in Appendix A Table A1. While fractional calculus's application in computer vision is well established, its incorporation into neural network-based approaches is still in its early stages, with limited works in the literature. Nonetheless, the results from existing studies demonstrate notable performance improvements, indicating the potential for further advancements in this area. Through our investigation, we identified several research gaps and outlined potential directions for future exploration in each of the studied works.

In real-world conditions, image capture often occurs under less-than-ideal circumstances, resulting in complex artefacts and challenges such as corruption and varying light conditions. Addressing these challenges has spurred the development of numerous methods aimed at enhancing neural network performances in computer vision tasks. The versatility of fractional calculus methods, with their increased degrees of freedom capable of modelling nuanced behaviours, holds promise in overcoming the hardships encountered in computer vision tasks.

The goal of this survey is to provide an overview of the current research and to offer intuitive explanations of the proposed methods. We aim to demystify fractional calculus and motivate its usage over integer-order methods by providing accessible explanations. We understand that fractional calculus can appear daunting, but we hope that this survey serves to dispel misconceptions and inspire researchers to contribute to the incorporation of fractional calculus into neural network-based computer vision.

In addition to the identified research gaps, we suggest investigating the influence of non-integer order derivatives on additional aspects of convolutional neural networks architectures, such as regularisation techniques, attention mechanisms, and transfer learning. Exploring these avenues could yield novel strategies for model optimisation and performance enhancement. Additionally, given the time dependence inherent in fractional calculus, we envisage that leveraging the memory capabilities could advance video object detection and generation tasks.

Furthermore, it is imperative to explore the interpretability and explainability of fractional calculus in neural network-based computer vision models. Understanding how fractional-order operations affect feature representations and decision-making processes within the network can provide valuable insights into model behaviour and foster trust among researchers and field experts. Furthermore, investigating the generalisability and robustness of fractional calculus-enhanced neural networks across diverse datasets and real-world scenarios is crucial to ensure their practical applicability. Finally, the scalability and computational efficiency of fractional calculus-based methods merit further investigation, particularly in resource-constrained environments and large-scale applications.

Author Contributions: Conceptualization, C.C.; methodology, C.C.; validation C.C., M.F.P.C. and L.L.F.; formal analysis, C.C. and L.L.F.; investigation, C.C.; writing—original draft preparation, C.C.; writing—review and editing, C.C., M.F.P.C. and L.L.F.; supervision, M.F.P.C. and L.L.F. All authors have read and agreed to the published version of the manuscript.

Funding: The authors acknowledge the funding by Fundação para a Ciência e Tecnologia (Portuguese Foundation for Science and Technology) through CMAT projects UIDB/00013/2020 and UIDP/00013/2020 and the funding by the FCT and Google Cloud partnership through projects CPCA-IAC/AV/589164/2023 and CPCA-IAC/AF/589140/2023. C. Coelho would like to thank FCT for the funding through the scholarship with reference 2021.05201.BD. This work was also financially supported by national funds through the FCT/MCTES (PIDDAC), under project 2022.06672.PTDC—iMAD—Improving the Modelling of Anomalous Diffusion and Viscoelasticity: solutions to industrial problems.

Institutional Review Board Statement: Not applicable.

Informed Consent Statement: Not applicable.

Data Availability Statement: Not applicable.

Conflicts of Interest: The authors declare no conflicts of interest.

Abbreviations

The following abbreviations are used in this manuscript:

List of symbols.

Symbol	**Meaning**
α, β, γ	Fractional-orders of the the derivatives
$\mathcal{D}^{\alpha} f(x)$	Fractional derivative of order α of function $f(x)$
$\mathcal{D}^{2} f(x)$	Second-order derivative of function $f(x)$
h	Step size
t_0	Arbitrary initial time
t_f	Arbitrary final time
$\boldsymbol{y}(s)$	Input image
s	Pixel
$\Phi(\cdot), \Psi(\cdot)$	Linear transformations
$\boldsymbol{u}(t,s)$	Control input
$\boldsymbol{\theta}$	NN learnable parameters
w_k	Weight of layer k
$T_{\uparrow}, T_{\downarrow}$	Pooling and unpooling operations
σ	Nonlinear operation
l_i	FOCNet of level i
$\boldsymbol{\theta}_i$	Learnable parameters of level i
$\boldsymbol{u}_t^{l_i \pm 1}$	Upper-level and lower-level features
I, J, K	Pixel-wise width, length, and depth of an image
i, j, k	Coordinates of a pixel
$\boldsymbol{X}'$	Denoised image
$\mathcal{L}(\boldsymbol{\theta})$	Loss function
λ	Hyperparameter
$\boldsymbol{S}$	Sobel filter
$\boldsymbol{L}$	Laplace filter
$\boldsymbol{A}$	AFD mask
$M(i,j)$	Average gradient of a pixel
t_g	Gradient threshold
Q	Mean gradient of an image
AG_{ed}, AG_{tex}	Average gradients of edges and textures
R_{α}	Rényi entropy

p_s	Normalised histogram of pixel intensities
M	Image moment
μ	Central moment
$\overline{x}, \overline{y}$	Image centroid
P	Legendre polynomial
L	Legendre moment
$\tilde{\mathbf{Y}}$	Generated image
S_n	ScatNet layer
S_{pool}	ScatNet pooling
Θ_α	Fractional convolution operator
$\psi_\lambda(t)$	Wavelet function at scale λ
ω	Weighting hyperparameter
$G(i,j)$	Gaussian operator
std	Standard deviation
$F(i,j)$	Fractional convolutional filter
i_o, j_o, k_o	Parameters of a filter
l	Pixel-wise dimension of a filter
I_{in}, J_{in}, D_{in}	Width, height, and channels of the input feature map
$I_{out}, J_{out}, D_{out}$	Width, height, and channels of the output feature map
$P_{i,j}$	Pooling region
In	Input feature map
Out	Output feature map
a_i, b_j	Hyperparameters for computing pooling windows

List of abbreviations.

Abbreviation	**Meaning**
HOG	Histogram of Oriented Gradients
FC	Fractional Calculus
NN	Neural Network
ML	Machine Learning
CNN	Convolutional Neural Network
FDE	Fractional Differential Equation
FPDE	Fractional Partial Differential Equation
FOCNet	Fractional Optimal Control Network
F-ODE	Fractional Ordinary Differential Equations
TV	Total Variation
FTV	Fractional-order Total Variation
LDD	Left-Down Direction
RUD	Right-Up Direction
LUD	Left-Up Direction
RDD	Right-Down Direction
AFD	Adaptive Fractional-order Differential
AFDA	Adaptive Fractional-order Differential Algorithm
RCNN	Region-based Convolutional Neural Network
YOLO	You Only Look Once
SSD	Single-Shot Detector
FrOLM-DNN	Fractional-Order Lagrange Moments Deep Neural Network
LSF	Level-Set Function
PDE	Partial Differential Equation
CeNN	Cellular Neural Network
VAE	Variational Autoencoder
GAN	Generative Adversarial Network
GSN	Generative Scattering Network
ScatNet	Wavelet Scattering Network
PCA	Principal Component Analysis
GFRSN	Generative Fractional Scattering Networks
FrScatNet	Fractional Wavelet Scattering Network
FMF	Feature Map Fusion
FMP	Fractional Max-Pooling

Appendix A. Summary Table

Table A1. Summary table of the methods discussed in this survey.

Task	Proposal	Advantages	Experimental Setup	Results
Denoising	Fractional-Order Total Variation [41]	Incorporates information from neighbouring pixels Reduces artefacts	Dataset: Google Maps images Comparison: same network with total and fractional-order total variation loss Metrics: peak signal noise ration, structural similarity index, and universal quality	Best performance metric Better performance in preserving texture details
	Fractional Optimal Control Network [42]	Propagates features depth-wise within an NN	Dataset: Set12 [118], BSD68 [119], and Urban100 [120], with three induced noise levels Comparison: BM3D [121], WNNM [122], TNRD [123], DnCNN [118], FFDNet [124], RED [125], MemNet [126], and N^3Net [127] Metric: average peak signal-to-noise ratio	Leading performance metric results Similar computational cost
Enhancement	Neural Fractional-order Adaptive Masks [18]	Captures the advantages and reduce the disadvantages of first- and second-order masks	Dataset: chest X-ray images, ultrasonic images, and pelvis radiography images, Comparison: same network with and without neural fractional-order adaptive masks Metrics: information entropy, mean absolute difference coefficient, and absolute mean brightness error	Better performance metrics Offers higher contrast and clearer edges Enhances smooth areas and texture
	Fractional Rényi Entropy [60]	Refined characterisation of the uncertainty and complexity within a probability distribution Enhances robustness and preserves spatial relationships between image pixels	Dataset: kidney magnetic resonance imaging scans Comparison: Hasan et al. [128], Li et al. [129], Ibrahim et al. [130], Alaa et al. [131], and DLSS [132] Metric: accuracy	Achieves best performance metrics Improves fine details with low contrast
Object Detection	Fractional-Order Legendre Moment Invariants [67]	Fractional-order moments have additional parameters, offering improved results	Dataset: AR database of faces Comparison: LMs [133], DTMs [134], DKMs [135], ZMs [133], OFMs [136], and BMs [137] Metrics: statistical normalisation image reconstruction error and correct classification percentages	Better performance metrics Higher noise robustness

Table A1. *Cont.*

Task	Proposal	Advantages	Experimental Setup	Results
Segmentation	Active Contour Detection With Fractional-Order Regularisation Term [95]	More precise representation of the image Improves the robustness to noise	Dataset: synthetic and medical images Comparison: GAC [138], Chan-Vese [139], Chunming Li [140], Lankton [141], Shi [142], RSFLG [143], and LPF [144] Metrics: Dice Similarity Coefficient, peak signal-to-noise ratio, Hausdorff distance, structural similarity index measure, and mean sum of squares distance	Higher robustness and effectiveness Time-efficient Easy implementation
	FOCNet For Segmentation [17]	Propagation of information from shallower layers	Dataset: Massachusetts Road Dataset [145] and Ottawa Road Dataset [146] Comparison: U-Net [89], Dlinknet [147], HsgNet [148], Dense-UNet [149], SUNet [149], and SDUNet [149] Metrics: recall, precision, Dice Coefficient, accuracy, and mean intersection over union	Better performance metrics Lesser loss of information and computationally efficient
Restoration	Fractional Wavelet Scattering Networks [112]	More suitable dimensional reduction method that considers features extracted across different layers Improves reduced image quality and overfitting	Dataset: CIFAR-10 [150] and CelebA [151] Comparison: Principal Component Analysis [152] Metrics: peak signal-to-noise ratio and structural similarity	Better performance metrics Generates better images
Compression	Fractional Max-Pooling [114]	Preserves spatial information More effective feature extraction	Dataset: MNIST [153], CIFAR-100 [150], The Online Handwritten Assamese Characters Dataset [154], CASIA-OLHWDB1.1 [155], and CIFAR-10 [150] Comparison: max-pooling Metric: accuracy	Better performance metrics Better way of encoding information
	Fractional Convolutional Filters [19]	Reduced number of parameters to describe filters	Dataset: MNIST [153], CIFAR-10 [150], ImageNet [156], and UCF101 [157] Comparison: LeNet [153], LeNet5 [153], 50-50-200-10NN [158], Best Practices [159], CNN for MNIST and All-CNN [160], MobileNetV1 [161], MobileNetV2 [162], ShuffleNet 8G [163], ShuffleNet 1G [163], HENet [164], ResNet18 [165] for CIFAR-10, and ResNet18 [165] for UCF101 Metric: accuracy	Achieved new record for the smallest model with >99% accuracy on MNIST Effective for filter compression Reduces computational cost

References

1. Szeliski, R. *Computer Vision: Algorithms and Applications*; Springer Nature: Cham, Switzerland, 2022.
2. Dey, S. *Hands-On Image Processing with Python: Expert Techniques for Advanced Image Analysis and Effective Interpretation of Image Data*; Packt Publishing Ltd.: Birmingham, UK, 2018.
3. Prince, S.J. *Computer Vision: Models, Learning, and Inference*; Cambridge University Press: Cambridge, UK, 2012.
4. Herrmann, R. *Fractional Calculus: An Introduction for Physicists*; World Scientific: Singapore, 2011.
5. Abdou, M.A. An analytical method for space–time fractional nonlinear differential equations arising in plasma physics. *J. Ocean Eng. Sci.* **2017**, *2*, 288–292. [CrossRef]
6. Alquran, M. The amazing fractional Maclaurin series for solving different types of fractional mathematical problems that arise in physics and engineering. *Partial Differ. Equ. Appl. Math.* **2023**, *7*, 100506. [CrossRef]
7. Ionescu, C.; Lopes, A.; Copot, D.; Machado, J.T.; Bates, J.H. The role of fractional calculus in modeling biological phenomena: A review. *Commun. Nonlinear Sci. Numer. Simul.* **2017**, *51*, 141–159. [CrossRef]
8. Ma, Y.; Li, W. Application and research of fractional differential equations in dynamic analysis of supply chain financial chaotic system. *Chaos Solitons Fractals* **2020**, *130*, 109417. [CrossRef]
9. Jan, A.; Srivastava, H.M.; Khan, A.; Mohammed, P.O.; Jan, R.; Hamed, Y. In vivo HIV dynamics, modeling the interaction of HIV and immune system via non-integer derivatives. *Fractal Fract.* **2023**, *7*, 361. [CrossRef]
10. Zhang, Y.; Pu, Y.; Zhou, J. Construction of fractional differential masks based on Riemann-Liouville definition. *J. Comput. Inf. Syst.* **2010**, *6*, 3191–3199.
11. Yang, Q.; Chen, D.; Zhao, T.; Chen, Y. Fractional calculus in image processing: A review. *Fract. Calc. Appl. Anal.* **2016**, *19*, 1222–1249. [CrossRef]
12. Ghamisi, P.; Couceiro, M.S.; Benediktsson, J.A.; Ferreira, N.M. An efficient method for segmentation of images based on fractional calculus and natural selection. *Expert Syst. Appl.* **2012**, *39*, 12407–12417. [CrossRef]
13. Tian, D.; Xue, D.; Wang, D. A fractional-order adaptive regularization primal–dual algorithm for image denoising. *Inf. Sci.* **2015**, *296*, 147–159. [CrossRef]
14. Coelho, C.; Costa, M.F.P.; Ferrás, L. Neural Fractional Differential Equations. *arXiv* **2024**, arXiv:2403.02737.
15. Alsaade, F.W.; Al-zahrani, M.S.; Yao, Q.; Jahanshahi, H. A Model-Free Finite-Time Control Technique for Synchronization of Variable-Order Fractional Hopfield-like Neural Network. *Fractal Fract.* **2023**, *7*, 349. [CrossRef]
16. Boroomand, A.; Menhaj, M.B. Fractional-order Hopfield neural networks. In Proceedings of the Advances in Neuro-Information Processing: 15th International Conference, ICONIP 2008, Auckland, New Zealand, 25–28 November 2008; Revised Selected Papers, Part I 15; Springer: Berlin/Heidelberg, Germany, 2009; pp. 883–890.
17. Arora, S.; Suman, H.K.; Mathur, T.; Pandey, H.M.; Tiwari, K. Fractional derivative based weighted skip connections for satellite image road segmentation. *Neural Netw.* **2023**, *161*, 142–153. [CrossRef]
18. Krouma, H.; Ferdi, Y.; Taleb-Ahmedx, A. Neural adaptive fractional order differential based algorithm for medical image enhancement. In Proceedings of the 2018 International Conference on Signal, Image, Vision and their Applications (SIVA), Guelma, Algeria, 26–27 November 2018; pp. 1–6.
19. Zamora, J.; Vargas, J.A.C.; Rhodes, A.; Nachman, L.; Sundararajan, N. Convolutional filter approximation using fractional calculus. In Proceedings of the IEEE/CVF International Conference on Computer Vision, Virtual, 11–17 October 2021; pp. 383–392.
20. Diethelm, K.; Ford, N.J. Analysis of fractional differential equations. *J. Math. Anal. Appl.* **2002**, *265*, 229–248. [CrossRef]
21. Ross, B. A brief history and exposition of the fundamental theory of fractional calculus. In *Fractional Calculus and Its Applications*; Springer: Berlin/Heidelberg, Germany, 1975; pp. 1–36. [CrossRef].
22. Machado, J.T.; Kiryakova, V.; Mainardi, F. Recent history of fractional calculus. *Commun. Nonlinear Sci. Numer. Simul.* **2011**, *16*, 1140–1153. [CrossRef]
23. Caputo, M. Linear Models of Dissipation whose Q is almost Frequency Independent—II. *Geophys. J. Int.* **1967**, *13*, 529–539. [CrossRef]
24. Grünwald, A. Ueber, Begrenzte, Derivationen und Deren Anwendung. *Z. Math. Phys* **1867**, *12*, 441–480.
25. Post, E.L. Generalized differentiation. *Trans. Am. Math. Soc.* **1930**, *32*, 723–781. [CrossRef]
26. Letnikov, A. An explanation of the concepts of the theory of differentiation of arbitrary index. *Mosc. Matem. Sb.* **1872**, *6*, 413–445.
27. Jain, R.; Kasturi, R.; Schunck, B.G. *Machine Vision*; McGraw-Hill: New York, NY, USA, 1995; Volume 5.
28. Bishop, C.M.; Bishop, H. *Deep Learning: Foundations and Concepts*; Springer Nature: Cham, Switzerland, 2023.
29. Komatsu, R.; Gonsalves, T. Comparing U-Net based models for denoising color images. *AI* **2020**, *1*, 465–486. [CrossRef]
30. Liu, Y.; Pu, Y.; Zhou, J. Design of image denoising filter based on fractional integral. *J. Comput. Inf. Syst.* **2010**, *6*, 2839–2847.
31. Liu, Y. A digital image denoising method based on fractional calculus. *J. Sichuan Univ. Eng. Sci. Ed.* **2011**, *43*, 90–95+144.
32. Azerad, P.; Bouharguane, A.; Crouzet, J.F. Simultaneous denoising and enhancement of signals by a fractal conservation law. *Commun. Nonlinear Sci. Numer. Simul.* **2012**, *17*, 867–881. [CrossRef]
33. Zheng, W.; Xianmin, M. Fractional-Order Differentiate Adaptive Algorithm for Identifying Coal Dust Image Denoising. In Proceedings of the 2014 International Symposium on Computer, Consumer and Control, Taichung, Taiwan, 10–12 June 2014; pp. 638–641.
34. Pan, X.; Liu, S.; Jiang, T.; Liu, H.; Wang, X.; Li, L. Non-causal fractional low-pass filter based medical image denoising. *J. Med. Imaging Health Inform.* **2016**, *6*, 1799–1806. [CrossRef]

35. Li, D.; Jiang, T.; Jin, Q.; Zhang, B. Adaptive fractional order total variation image denoising via the alternating direction method of multipliers. In Proceedings of the 2020 Chinese Control And Decision Conference (CCDC), Hefei, China, 22–24 August 2020; pp. 3876–3881.
36. Al-Shamasneh, A.R.; Ibrahim, R.W. Image denoising based on quantum calculus of local fractional entropy. *Symmetry* **2023**, *15*, 396. [CrossRef]
37. Ilesanmi, A.E.; Ilesanmi, T.O. Methods for image denoising using convolutional neural network: A review. *Complex Intell. Syst.* **2021**, *7*, 2179–2198. [CrossRef]
38. Jifara, W.; Jiang, F.; Rho, S.; Cheng, M.; Liu, S. Medical image denoising using convolutional neural network: A residual learning approach. *J. Supercomput.* **2019**, *75*, 704–718. [CrossRef]
39. Singh, P.; Shankar, A. A novel optical image denoising technique using convolutional neural network and anisotropic diffusion for real-time surveillance applications. *J. Real-Time Image Process.* **2021**, *18*, 1711–1728. [CrossRef]
40. Chandra, I.S.; Shastri, R.K.; Kavitha, D.; Kumar, K.R.; Manochitra, S.; Babu, P.B. CNN based color balancing and denoising technique for underwater images: CNN-CBDT. *Meas. Sens.* **2023**, *28*, 100835. [CrossRef]
41. Bai, Y.C.; Zhang, S.; Chen, M.; Pu, Y.F.; Zhou, J.L. A fractional total variational CNN approach for SAR image despeckling. In Proceedings of the Intelligent Computing Methodologies: 14th International Conference, ICIC 2018, Wuhan, China, 15–18 August 2018; Proceedings, Part III 14; Springer: Cham, Switzerland, 2018; pp. 431–442.
42. Jia, X.; Liu, S.; Feng, X.; Zhang, L. Focnet: A fractional optimal control network for image denoising. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, Long Beach, CA, USA, 15–20 June 2019; pp. 6054–6063.
43. Rudin, L.I.; Osher, S.; Fatemi, E. Nonlinear total variation based noise removal algorithms. *Phys. D Nonlinear Phenom.* **1992**, *60*, 259–268. [CrossRef]
44. Yang, X.; Zhang, J.; Liu, Y.; Zheng, X.; Liu, K. Super-resolution image reconstruction using fractional-order total variation and adaptive regularization parameters. *Vis. Comput.* **2019**, *35*, 1755–1768. [CrossRef]
45. Wang, Q.; Gao, Z.; Xie, C.; Chen, G.; Luo, Q. Fractional-order total variation for improving image fusion based on saliency map. *Signal Image Video Process.* **2020**, *14*, 991–999. [CrossRef]
46. Zhang, X.; Yan, H. Medical image fusion and noise suppression with fractional-order total variation and multi-scale decomposition. *IET Image Process.* **2021**, *15*, 1688–1701. [CrossRef]
47. Jun, Z.; Zhihui, W. A class of fractional-order multi-scale variational models and alternating projection algorithm for image denoising. *Appl. Math. Model.* **2011**, *35*, 2516–2528. [CrossRef]
48. Yazgaç, B.G.; Kırcı, M. Fractional-order calculus-based data augmentation methods for environmental sound classification with deep learning. *Fractal Fract.* **2022**, *6*, 555. [CrossRef]
49. Lepcha, D.C.; Goyal, B.; Dogra, A.; Sharma, K.P.; Gupta, D.N. A deep journey into image enhancement: A survey of current and emerging trends. *Inf. Fusion* **2023**, *93*, 36–76. [CrossRef]
50. Pu, Y. Fractional calculus approach to texture of digital image. In Proceedings of the 2006 8th International Conference on Signal Processing, Guilin, China, 16–20 November 2006; Volume 2.
51. Huang, G.; Xu, L.; Chen, Q.; Men, T. Image Enhancement Using a Fractional-Order Differential. In Proceedings of the 4th International Conference on Computer Engineering and Networks: CENet2014, Shanghai, China, 19–20 July 2014; Springer: Cham, Switzerland, 2015; pp. 555–563.
52. Saadia, A.; Rashdi, A. Fractional order integration and fuzzy logic based filter for denoising of echocardiographic image. *Comput. Methods Programs Biomed.* **2016**, *137*, 65–75. [CrossRef] [PubMed]
53. Lei, J.; Zhang, S.; Luo, L.; Xiao, J.; Wang, H. Super-resolution enhancement of UAV images based on fractional calculus and POCS. *Geo-Spat. Inf. Sci.* **2018**, *21*, 56–66. [CrossRef]
54. AbdAlRahman, A.; Ismail, S.M.; Said, L.A.; Radwan, A.G. Double fractional-order masks image enhancement. In Proceedings of the 2021 3rd Novel Intelligent and Leading Emerging Sciences Conference (NILES), Giza, Egypt, 23–25 October 2021; pp. 261–264.
55. Aldawish, I.; Jalab, H.A. A Mathematical Model for COVID-19 Image Enhancement based on Mittag-Leffler-Chebyshev Shift. *Comput. Mater. Contin.* **2022**, *73*, 1307–1316.
56. Miah, B.A.; Sen, M.; Murugan, R.; Gupta, D. Developing Riemann–Liouville-Fractional Masks for Image Enhancement. *Circuits Syst. Signal Process.* **2024**, *43*, 3802–3831. [CrossRef]
57. Yogeshwari, M.; Thailambal, G. Automatic feature extraction and detection of plant leaf disease using GLCM features and convolutional neural networks. *Mater. Today Proc.* **2023**, *81*, 530–536. [CrossRef]
58. Accarino, G.; Chiarelli, M.; Immorlano, F.; Aloisi, V.; Gatto, A.; Aloisio, G. Msg-gan-sd: A multi-scale gradients gan for statistical downscaling of 2-meter temperature over the euro-cordex domain. *AI* **2021**, *2*, 600–620. [CrossRef]
59. He, C.; Li, K.; Xu, G.; Yan, J.; Tang, L.; Zhang, Y.; Wang, Y.; Li, X. Hqg-net: Unpaired medical image enhancement with high-quality guidance. *IEEE Trans. Neural Netw. Learn. Syst.* **2023**, 1–15. [CrossRef]
60. Jalab, H.A.; Al-Shamasneh, A.R.; Shaiba, H.; Ibrahim, R.W.; Baleanu, D. Fractional Renyi entropy image enhancement for deep segmentation of kidney MRI. *Comput. Mater. Contin.* **2021**, *67*, 2061–2075.
61. Ferdi, Y. Some applications of fractional order calculus to design digital filters for biomedical signal processing. *J. Mech. Med. Biol.* **2012**, *12*, 1240008. [CrossRef]

62. Rényi, A. On measures of entropy and information. In *Proceedings of the Fourth Berkeley Symposium on Mathematical Statistics and Probability, Volume 1: Contributions to the Theory of Statistics*; University of California Press: Berkeley, CA, USA, 1961; Volume 4, pp. 547–562.
63. Karmeshu. *Entropy Measures, Maximum Entropy Principle and Emerging Applications*; Springer: Berlin/Heidelberg, Germany, 2003.
64. Bromiley, P.; Thacker, N.; Bouhova-Thacker, E. Shannon entropy, Renyi entropy, and information. *Stat. Inf. Ser. (2004-004)* **2004**, *9*, 2–8.
65. Zhu, S.C.; Wu, Y.N.; Mumford, D. Minimax entropy principle and its application to texture modeling. *Neural Comput.* **1997**, *9*, 1627–1660. [CrossRef]
66. Ibraheam, M.; Li, K.F.; Gebali, F.; Sielecki, L.E. A performance comparison and enhancement of animal species detection in images with various r-cnn models. *AI* **2021**, *2*, 552–577. [CrossRef]
67. Xiao, B.; Li, L.; Li, Y.; Li, W.; Wang, G. Image analysis by fractional-order orthogonal moments. *Inf. Sci.* **2017**, *382*, 135–149. [CrossRef]
68. Kumar, S.P.; Latte, M.V. Modified and optimized method for segmenting pulmonary parenchyma in CT lung images, based on fractional calculus and natural selection. *J. Intell. Syst.* **2019**, *28*, 721–732. [CrossRef]
69. Girshick, R.; Donahue, J.; Darrell, T.; Malik, J. Rich feature hierarchies for accurate object detection and semantic segmentation. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Columbus, OH, USA, 23–28 June 2014; pp. 580–587.
70. Girshick, R. Fast r-cnn. In Proceedings of the IEEE International Conference on Computer Vision, Santiago, Chile, 7–13 December 2015; pp. 1440–1448.
71. Ren, S.; He, K.; Girshick, R.; Sun, J. Faster r-cnn: Towards real-time object detection with region proposal networks. *Adv. Neural Inf. Process. Syst.* **2015**, *28*, 91–99. [CrossRef]
72. Redmon, J.; Divvala, S.; Girshick, R.; Farhadi, A. You only look once: Unified, real-time object detection. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Las Vegas, NV, USA, 27–30 June 2016; pp. 779–788.
73. Liu, W.; Anguelov, D.; Erhan, D.; Szegedy, C.; Reed, S.; Fu, C.Y.; Berg, A.C. Ssd: Single shot multibox detector. In Proceedings of the Computer Vision–ECCV 2016: 14th European Conference, Amsterdam, The Netherlands, 11–14 October 2016; Proceedings, Part I 14; Springer: Cham, Switzerland, 2016; pp. 21–37.
74. Coelho, C.; Costa, M.F.P.; Ferrás, L.L.; Soares, A.J. Object detection with retinanet on aerial imagery: The algarve landscape. In Proceedings of the International Conference on Computational Science and Its Applications, Cagliari, Italy, 13–16 September 2021; Springer: Cham, Switzerland, 2021; pp. 501–516.
75. Tang, Z.; Liu, X.; Chen, H.; Hupy, J.; Yang, B. Deep learning based wildfire event object detection from 4K aerial images acquired by UAS. *AI* **2020**, *1*, 166–179. [CrossRef]
76. Albuquerque, L.; Coelho, C.; Costa, M.F.P.; Ferrás, L.; Soares, A. Improving public parking by using artificial intelligence. *AIP Conf. Proc.* **2023**, *2849*, 220003.
77. Gunturu, S.; Munir, A.; Ullah, H.; Welch, S.; Flippo, D. A spatial AI-based agricultural robotic platform for wheat detection and collision avoidance. *AI* **2022**, *3*, 719–738. [CrossRef]
78. Barbedo, J.G.A. Detecting and classifying pests in crops using proximal images and machine learning: A review. *AI* **2020**, *1*, 312–328. [CrossRef]
79. El Ogri, O.; Karmouni, H.; Sayyouri, M.; Qjidaa, H. 3D image recognition using new set of fractional-order Legendre moments and deep neural networks. *Signal Process. Image Commun.* **2021**, *98*, 116410. [CrossRef]
80. Zhou, M.; Li, B.; Wang, J. Optimization of Hyperparameters in Object Detection Models Based on Fractal Loss Function. *Fractal Fract.* **2022**, *6*, 706. [CrossRef]
81. Mahaveerakannan, R.; Anitha, C.; Thomas, A.K.; Rajan, S.; Muthukumar, T.; Rajulu, G.G. An IoT based forest fire detection system using integration of cat swarm with LSTM model. *Comput. Commun.* **2023**, *211*, 37–45. [CrossRef]
82. Castleman, K.R. *Digital Image Processing*; Prentice Hall Press: Saddle River, NJ, USA, 1996.
83. Deepika, C.L.; Kandaswamy, A.; Vimal, C.; Satish, B. Palmprint authentication using modified legendre moments. *Procedia Comput. Sci.* **2010**, *2*, 164–172. [CrossRef]
84. Kamaruddin, N.; Abdullah, N.A.; Ibrahim, R.W. Image segmentation based on fractional non-markov poisson stochastic process. *Pak. J. Stat.* **2015**, *31*, 557–574.
85. Tang, Q.; Gao, S.; Liu, Y.; Yu, F. Infrared image segmentation algorithm for defect detection based on FODPSO. *Infrared Phys. Technol.* **2019**, *102*, 103051. [CrossRef]
86. Kamaruddin, N.; Maarop, N.; Narayana, G. Fractional Active Contour Model for Edge Detector on Medical Image Segmentation. In Proceedings of the 2020 2nd International Conference on Image, Video and Signal Processing, Marrakesh, Morocco, 4–6 June 2020; pp. 39–44.
87. Vivekraj, A.; Sumathi, S. Resnet-Unet-FSOA based cranial nerve segmentation and medial axis extraction using MRI images. *Imaging Sci. J.* **2023**, *71*, 750–766. [CrossRef]
88. Geng, N.; Sheng, H.; Sun, W.; Wang, Y.; Yu, T.; Liu, Z. Image segmentation of rail surface defects based on fractional order particle swarm optimization 2D-Otsu algorithm. In Proceedings of the International Conference on Algorithm, Imaging Processing, and Machine Vision (AIPMV 2023), Qingdao, China, 15–17 September 2023; Volume 12969, pp. 56–59.

89. Ronneberger, O.; Fischer, P.; Brox, T. U-net: Convolutional networks for biomedical image segmentation. In Proceedings of the Medical Image Computing and Computer-Assisted Intervention—MICCAI 2015: 18th International Conference, Munich, Germany, 5–9 October 2015; Proceedings, Part III 18; Springer: Cham, Switzerland, 2015; pp. 234–241.
90. De Silva, M.S.; Narayanan, B.N.; Hardie, R.C. A patient-specific algorithm for lung segmentation in chest radiographs. *AI* **2022**, *3*, 931–947. [CrossRef]
91. Chhabra, M.; Ravulakollu, K.K.; Kumar, M.; Sharma, A.; Nayyar, A. Improving automated latent fingerprint detection and segmentation using deep convolutional neural network. *Neural Comput. Appl.* **2023**, *35*, 6471–6497. [CrossRef]
92. Zhang, T.; Wang, D.; Lu, Y. ECSNet: An accelerated real-time image segmentation CNN architecture for pavement crack detection. *IEEE Trans. Intell. Transp. Syst.* **2023**, *24*, 15105–15112. [CrossRef]
93. Marques, F.; De Araujo, T.P.; Nator, C.; Saraiva, A.; Sousa, J.; Pinto, A.M.; Melo, R. Recognition of simple handwritten polynomials using segmentation with fractional calculus and convolutional neural networks. In Proceedings of the 2019 8th Brazilian Conference on Intelligent Systems (BRACIS), Salvador, Brazil, 15–18 October 2019; pp. 245–250.
94. Nirmalapriya, G.; Agalya, V.; Regunathan, R.; Ananth, M.B.J. Fractional Aquila spider monkey optimization based deep learning network for classification of brain tumor. *Biomed. Signal Process. Control* **2023**, *79*, 104017. [CrossRef]
95. Lakra, M.; Kumar, S. A fractional-order PDE-based contour detection model with CeNN scheme for medical images. *J. Real-Time Image Process.* **2022**, *19*, 147–160. [CrossRef]
96. Li, C.; Xu, C.; Gui, C.; Fox, M.D. Level set evolution without re-initialization: A new variational formulation. In Proceedings of the 2005 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR'05), San Diego, CA, USA, 20–25 June 2005; Volume 1, pp. 430–436.
97. Ren, Z. Adaptive active contour model driven by fractional order fitting energy. *Signal Process.* **2015**, *117*, 138–150. [CrossRef]
98. Chua, L.O.; Yang, L. Cellular neural networks: Theory. *IEEE Trans. Circuits Syst.* **1988**, *35*, 1257–1272. [CrossRef]
99. Raissi, M.; Perdikaris, P.; Karniadakis, G.E. Physics-informed neural networks: A deep learning framework for solving forward and inverse problems involving nonlinear partial differential equations. *J. Comput. Phys.* **2019**, *378*, 686–707. [CrossRef]
100. Zhang, Y.; Pu, Y.; Hu, J.; Zhou, J. A class of fractional-order variational image inpainting models. *Appl. Math. Inf. Sci* **2012**, *6*, 299–306.
101. Bosch, J.; Stoll, M. A fractional inpainting model based on the vector-valued Cahn–Hilliard equation. *SIAM J. Imaging Sci.* **2015**, *8*, 2352–2382. [CrossRef]
102. Li, D.; Tian, X.; Jin, Q.; Hirasawa, K. Adaptive fractional-order total variation image restoration with split Bregman iteration. *ISA Trans.* **2018**, *82*, 210–222. [CrossRef] [PubMed]
103. Ammi, M.R.S.; Jamiai, I. Finite difference and legendre spectral method for a time-fractional diffusion-convection equation for image restoration. *Discret. Contin. Dyn. Syst.-Ser. S* **2018**, *11*, 103–117.
104. Gouasnouane, O.; Moussaid, N.; Boujena, S.; Kabli, K. A nonlinear fractional partial differential equation for image inpainting. *Math. Model. Comput.* **2022**, *9*, 536–546. [CrossRef]
105. Kingma, D.P.; Welling, M. Auto-encoding variational bayes. *arXiv* **2013**, arXiv:1312.6114.
106. Goodfellow, I.; Pouget-Abadie, J.; Mirza, M.; Xu, B.; Warde-Farley, D.; Ozair, S.; Courville, A.; Bengio, Y. Generative adversarial networks. *Commun. ACM* **2020**, *63*, 139–144. [CrossRef]
107. Peng, J.; Liu, D.; Xu, S.; Li, H. Generating diverse structure for image inpainting with hierarchical VQ-VAE. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, Virtual, 19–25 June 2021; pp. 10775–10784.
108. Chen, Y.; Zhang, H.; Liu, L.; Chen, X.; Zhang, Q.; Yang, K.; Xia, R.; Xie, J. Research on image inpainting algorithm of improved GAN based on two-discriminations networks. *Appl. Intell.* **2021**, *51*, 3460–3474. [CrossRef]
109. Farajzadeh, N.; Hashemzadeh, M. A deep neural network based framework for restoring the damaged persian pottery via digital inpainting. *J. Comput. Sci.* **2021**, *56*, 101486. [CrossRef]
110. Cai, X.; Song, B. Semantic object removal with convolutional neural network feature-based inpainting approach. *Multimed. Syst.* **2018**, *24*, 597–609. [CrossRef]
111. Wang, Q.; Chen, Y.; Zhang, N.; Gu, Y. Medical image inpainting with edge and structure priors. *Measurement* **2021**, *185*, 110027. [CrossRef]
112. Wu, J.; Zhang, J.; Wu, F.; Kong, Y.; Yang, G.; Senhadji, L.; Shu, H. Generative networks as inverse problems with fractional wavelet scattering networks. *arXiv* **2020**, arXiv:2007.14177.
113. Angles, T.; Mallat, S. Generative networks as inverse problems with scattering transforms. *arXiv* **2018**, arXiv:1805.06621.
114. Graham, B. Fractional max-pooling. *arXiv* **2014**, arXiv:1412.6071.
115. Krizhevsky, A.; Sutskever, I.; Hinton, G.E. Imagenet classification with deep convolutional neural networks. *Adv. Neural Inf. Process. Syst.* **2012**, *25*, 1097–1105. [CrossRef]
116. Zeiler, M.D.; Fergus, R. Stochastic pooling for regularization of deep convolutional neural networks. *arXiv* **2013**, arXiv:1301.3557.
117. Szegedy, C.; Liu, W.; Jia, Y.; Sermanet, P.; Reed, S.; Anguelov, D.; Erhan, D.; Vanhoucke, V.; Rabinovich, A. Going deeper with convolutions. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Boston, MA, USA, 7–12 June 2015; pp. 1–9.
118. Zhang, K.; Zuo, W.; Chen, Y.; Meng, D.; Zhang, L. Beyond a gaussian denoiser: Residual learning of deep cnn for image denoising. *IEEE Trans. Image Process.* **2017**, *26*, 3142–3155. [CrossRef] [PubMed]

119. Ghahremani, M.; Khateri, M.; Sierra, A.; Tohka, J. Adversarial distortion learning for medical image denoising. *arXiv* **2022**, arXiv:2204.14100.
120. Huang, J.B.; Singh, A.; Ahuja, N. Single image super-resolution from transformed self-exemplars. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Boston, MA, USA, 7–12 June 2015; pp. 5197–5206.
121. Dabov, K.; Foi, A.; Katkovnik, V.; Egiazarian, K. Image denoising by sparse 3-D transform-domain collaborative filtering. *IEEE Trans. Image Process.* **2007**, *16*, 2080–2095. [CrossRef] [PubMed]
122. Gu, S.; Zhang, L.; Zuo, W.; Feng, X. Weighted nuclear norm minimization with application to image denoising. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Columbus, OH, USA, 23–28 June 2014; pp. 2862–2869.
123. Chen, Y.; Pock, T. Trainable nonlinear reaction diffusion: A flexible framework for fast and effective image restoration. *IEEE Trans. Pattern Anal. Mach. Intell.* **2016**, *39*, 1256–1272. [CrossRef] [PubMed]
124. Zhang, K.; Zuo, W.; Zhang, L. FFDNet: Toward a fast and flexible solution for CNN-based image denoising. *IEEE Trans. Image Process.* **2018**, *27*, 4608–4622. [CrossRef] [PubMed]
125. Romano, Y.; Elad, M.; Milanfar, P. The little engine that could: Regularization by denoising (RED). *SIAM J. Imaging Sci.* **2017**, *10*, 1804–1844. [CrossRef]
126. Tai, Y.; Yang, J.; Liu, X.; Xu, C. Memnet: A persistent memory network for image restoration. In Proceedings of the IEEE International Conference on Computer Vision, Venice, Italy, 22–29 October 2017; pp. 4539–4547.
127. Plötz, T.; Roth, S. Neural nearest neighbors networks. *Adv. Neural Inf. Process. Syst.* **2018**, *31*, 1087–1098.
128. Hasan, A.M.; Meziane, F.; Aspin, R.; Jalab, H.A. Segmentation of brain tumors in MRI images using three-dimensional active contour without edge. *Symmetry* **2016**, *8*, 132. [CrossRef]
129. Li, C.; Xu, C.; Gui, C.; Fox, M.D. Distance regularized level set evolution and its application to image segmentation. *IEEE Trans. Image Process.* **2010**, *19*, 3243–3254. [CrossRef] [PubMed]
130. Ibrahim, R.W.; Hasan, A.M.; Jalab, H.A. A new deformable model based on fractional Wright energy function for tumor segmentation of volumetric brain MRI scans. *Comput. Methods Programs Biomed.* **2018**, *163*, 21–28. [CrossRef] [PubMed]
131. Al-Shamasneh, A.R.; Jalab, H.A.; Shivakumara, P.; Ibrahim, R.W.; Obaidellah, U.H. Kidney segmentation in MR images using active contour model driven by fractional-based energy minimization. *Signal Image Video Process.* **2020**, *14*, 1361–1368. [CrossRef]
132. Deep Learning Super Sampling (DLSS). Available online: https://developer.nvidia.com/rtx/dlss (accessed on 17 July 2024).
133. Teague, M.R. Image analysis via the general theory of moments. *Josa* **1980**, *70*, 920–930. [CrossRef]
134. Mukundan, R.; Ong, S.; Lee, P.A. Image analysis by Tchebichef moments. *IEEE Trans. Image Process.* **2001**, *10*, 1357–1364. [CrossRef] [PubMed]
135. Asli, B.H.S.; Flusser, J. Fast computation of Krawtchouk moments. *Inf. Sci.* **2014**, *288*, 73–86. [CrossRef]
136. Sheng, Y.; Shen, L. Orthogonal Fourier–Mellin moments for invariant pattern recognition. *JOSA A* **1994**, *11*, 1748–1757. [CrossRef]
137. Xiao, B.; Ma, J.F.; Wang, X. Image analysis by Bessel–Fourier moments. *Pattern Recognit.* **2010**, *43*, 2620–2629. [CrossRef]
138. Caselles, V.; Kimmel, R.; Sapiro, G. Geodesic active contours. *Int. J. Comput. Vis.* **1997**, *22*, 61–79. [CrossRef]
139. Chan, T.F.; Vese, L.A. Active contours without edges. *IEEE Trans. Image Process.* **2001**, *10*, 266–277. [CrossRef] [PubMed]
140. Li, C.; Kao, C.Y.; Gore, J.C.; Ding, Z. Minimization of region-scalable fitting energy for image segmentation. *IEEE Trans. Image Process.* **2008**, *17*, 1940–1949. [PubMed]
141. Lankton, S.; Tannenbaum, A. Localizing region-based active contours. *IEEE Trans. Image Process.* **2008**, *17*, 2029–2039. [CrossRef] [PubMed]
142. Shi, Y.; Karl, W.C. A real-time algorithm for the approximation of level-set-based curve evolution. *IEEE Trans. Image Process.* **2008**, *17*, 645–656. [PubMed]
143. Ding, K.; Xiao, L.; Weng, G. Active contours driven by region-scalable fitting and optimized Laplacian of Gaussian energy for image segmentation. *Signal Process.* **2017**, *134*, 224–233. [CrossRef]
144. Ding, K.; Xiao, L.; Weng, G. Active contours driven by local pre-fitting energy for fast image segmentation. *Pattern Recognit. Lett.* **2018**, *104*, 29–36. [CrossRef]
145. Mnih, V. Machine Learning for Aerial Image Labeling. Ph.D. Thesis, University of Toronto, Toronto, ON, Canada, 2013.
146. Liu, Y.; Yao, J.; Lu, X.; Xia, M.; Wang, X.; Liu, Y. RoadNet: Learning to comprehensively analyze road networks in complex urban scenes from high-resolution remotely sensed images. *IEEE Trans. Geosci. Remote Sens.* **2018**, *57*, 2043–2056. [CrossRef]
147. Zhou, L.; Zhang, C.; Wu, M. D-LinkNet: LinkNet with pretrained encoder and dilated convolution for high resolution satellite imagery road extraction. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition Workshops, Salt Lake City, UT, USA, 18–22 June 2018; pp. 182–186.
148. Xie, Y.; Miao, F.; Zhou, K.; Peng, J. HsgNet: A road extraction network based on global perception of high-order spatial information. *ISPRS Int. J. Geo-Inf.* **2019**, *8*, 571. [CrossRef]
149. Yang, M.; Yuan, Y.; Liu, G. SDUNet: Road extraction via spatial enhanced and densely connected UNet. *Pattern Recognit.* **2022**, *126*, 108549. [CrossRef]
150. Krizhevsky, A.; Hinton, G. *Learning Multiple Layers of Features from Tiny Images*; University of Toronto: Toronto, ON, Canada, 2009.
151. Liu, Z.; Luo, P.; Wang, X.; Tang, X. Deep Learning Face Attributes in the Wild. In Proceedings of the International Conference on Computer Vision (ICCV), Santiago, Chile, 7–13 December 2015.
152. Shlens, J. A tutorial on principal component analysis. *arXiv* **2014**, arXiv:1404.1100.

153. Lecun, Y.; Bottou, L.; Bengio, Y.; Haffner, P. Gradient-based learning applied to document recognition. *Proc. IEEE* **1998**, *86*, 2278–2324. [CrossRef]
154. Baruah, U.; Hazarika, S. *Online Handwritten Assamese Characters Dataset*; UCI Machine Learning Repository; UC Irvine: Irvine, CA, USA, 2011. [CrossRef].
155. Wang, D.H.; Liu, C.L.; Yu, J.L.; Zhou, X.D. CASIA-OLHWDB1: A Database of Online Handwritten Chinese Characters. In Proceedings of the 2009 10th International Conference on Document Analysis and Recognition, Barcelona, Spain, 26–29 July 2009; pp. 1206–1210. [CrossRef].
156. Deng, J.; Dong, W.; Socher, R.; Li, L.J.; Li, K.; Fei-Fei, L. ImageNet: A large-scale hierarchical image database. In Proceedings of the 2009 IEEE Conference on Computer Vision and Pattern Recognition, Miami, FL, USA, 20–25 June 2009; pp. 248–255. [CrossRef].
157. Soomro, K.; Zamir, A.R.; Shah, M. UCF101: A Dataset of 101 Human Actions Classes from Videos in the Wild. *arXiv* **2012**, arXiv:1212.0402.
158. Schölkopf, B.; Platt, J.; Hofmann, T. Efficient Learning of Sparse Representations with an Energy-Based Model. In Proceedings of the 20th Annual Conference on Neural Information Processing Systems, NIPS 2006, Vancouver, BC, Canada, 4–7 December 2007; pp. 1137–1144.
159. Simard, P.; Steinkraus, D.; Platt, J. Best practices for convolutional neural networks applied to visual document analysis. In Proceedings of the Seventh International Conference on Document Analysis and Recognition, Edinburgh, UK, 3–6 August 2003; pp. 958–963. [CrossRef].
160. Springenberg, J.T.; Dosovitskiy, A.; Brox, T.; Riedmiller, M. Striving for Simplicity: The All Convolutional Net. *arXiv* **2015**, arXiv:1412.6806.
161. Howard, A.G.; Zhu, M.; Chen, B.; Kalenichenko, D.; Wang, W.; Weyand, T.; Andreetto, M.; Adam, H. MobileNets: Efficient Convolutional Neural Networks for Mobile Vision Applications. *arXiv* **2017**, arXiv:1704.04861.
162. Sandler, M.; Howard, A.; Zhu, M.; Zhmoginov, A.; Chen, L.C. MobileNetV2: Inverted Residuals and Linear Bottlenecks. In Proceedings of the 2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition, Salt Lake City, UT, USA, 18–23 June 2018; pp. 4510–4520. [CrossRef].
163. Zhang, X.; Zhou, X.; Lin, M.; Sun, J. ShuffleNet: An Extremely Efficient Convolutional Neural Network for Mobile Devices. In Proceedings of the 2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition, Salt Lake City, UT, USA, 18–23 June 2018; pp. 6848–6856.
164. Zhu, Q.; Zhang, R. HENet:A Highly Efficient Convolutional Neural Networks Optimized for Accuracy, Speed and Storage. *arXiv* **2018**, arXiv:1803.02742.
165. He, K.; Zhang, X.; Ren, S.; Sun, J. Deep Residual Learning for Image Recognition. In Proceedings of the 2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), Las Vegas, NV, USA, 27–30 June 2016; pp. 770–778. [CrossRef].

Disclaimer/Publisher's Note: The statements, opinions and data contained in all publications are solely those of the individual author(s) and contributor(s) and not of MDPI and/or the editor(s). MDPI and/or the editor(s) disclaim responsibility for any injury to people or property resulting from any ideas, methods, instructions or products referred to in the content.