

Article

Intelligent Method for COVID-19 Diagnosis: Construction and Comparative Analysis of ResKAN18

Dan Li ^{1,2} , Zan Yang ^{1,2} , Yanan Li ³ and Wei Nai ^{1,2,*} 

¹ School of Electronic Information, Huzhou College, Huzhou 313000, China; lidan@zjhzu.edu.cn (D.L.); yangzan@zjhzu.edu.cn (Z.Y.)

² Huzhou Key Laboratory for Urban Multidimensional Perception and Intelligent Computing, Huzhou College, Huzhou 313000, China

³ Medical School, Tianjin University, Tianjin 300192, China; liyanan_2021@tju.edu.cn

* Correspondence: niwei@zjhzu.edu.cn; Tel.: +86-18918525982

Abstract

In response to the challenge of balancing accuracy and generalization in medical image classification using convolutional neural networks (CNNs), this paper proposes ResKAN18, a hybrid structure that embeds the learnable spline function of the Kolmogorov–Arnold network (KAN) into the ResNet18 classification head for intelligent diagnosis of COVID-19 in chest X-ray images. ResKAN18 includes three variants: ResKAN18—Large (four layers of KAN, hidden-layer dimensions [256, 128, 64]), ResKAN18—Standard (four layers of KAN, hidden-layer dimensions [128, 64, 32]), and ResKAN18—Simple (three layers of KAN, hidden-layer dimensions [64, 32]), which can achieve a flexible balance between accuracy and efficiency with different depths of KAN. A systematic comparison has been conducted between four classic CNN baselines including ResNet18, VGG16, DenseNet121, ShuffleNetV2, and three ResKAN18 variants on a benchmark dataset containing 3880 chest X-rays (COVID-19, normal, viral pneumonia). The results have shown that ResKAN18—Large can achieve an accuracy of 98.80% on the independent test set, which is 1.21% higher than ResNet18 and 0.69% higher than DenseNet121—its parameter count is 13.97M, inference delay is 8.25 ms/image, and training–validation accuracy difference is only 1.50%. The accuracy and performance stability of the dataset under random partitioning conditions are superior to the other two variants and all classic CNN baselines. All ResKAN18 variants have achieved zero missed diagnoses for COVID-19, while ResNet18 has shown missed diagnoses (0.9944). Taking into account the trade-off between accuracy, generalization, and inference efficiency, ResKAN1—Large is recommended as the default configuration, while for edge deployment scenarios with severely limited resources, ResKAN18—Simple can provide a cost-effective alternative with an extremely low latency of 2.50 ms/image and only 3.1% parameter increment compared to ResNet18.

Keywords: COVID-19; chest X-ray image; Kolmogorov–Arnold network (KAN); ResNet18; medical image classification; deep learning



Academic Editor: Maryam Ravan

Received: 12 July 2026

Revised: 21 August 2026

Accepted: 24 August 2026

Published: 26 August 2026

Copyright: © 2026 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the [Creative Commons Attribution \(CC BY\)](https://creativecommons.org/licenses/by/4.0/) license.

1. Introduction

Since the outbreak of the coronavirus disease (COVID-19) at the end of 2019, chest X-ray photography has become an important auxiliary diagnostic tool due to its convenient operation and fast imaging advantages [1–3]. Nucleic acid amplification testing (such as RT-PCR) is considered as the “gold standard” for pathogen diagnosis; however, its detection cycle is relatively long and there are limitations such as false negative

rates [4,5]. In this context, medical imaging examinations have become an important means of epidemic screening and auxiliary diagnosis. However, manual medical image reading relies on physician experience and faced significant efficiency bottlenecks during the peak of the pandemic [6,7].

Deep learning (DL) has made significant progress in the field of medical image analysis [8,9]. In COVID-19 diagnosis, the CNN structure based on ImageNet pre-training has been widely adopted [10,11], with reported classification accuracy exceeding 95% [12,13]. Among them, AlexNet pioneered CNN by introducing local connections and parameter sharing [14]; GoogLeNet has improved computational efficiency through the Inception module [15]; ResNet series effectively alleviates the gradient vanishing problem through residual connections and has become one of the most widely used backbone networks in medical image classification [16]; DenseNet achieves feature reuse through dense connections and performs outstandingly in parameter efficiency [17]; visual geometry group (VGG) series is known for its simple serial structure and large receptive field [18]; and ShuffleNet achieves lightweight deployment of the model through channel shuffling operations [19]. However, these methods still face several core challenges [13,20,21]:

(1) A lack of interpretability: Most high-performance CNN models act as “black boxes” with difficult-to-trace internal decision logic, seriously hindering clinical doctors’ trust and adoption of model diagnostic results [22,23]. Although post-attribution techniques such as gradient-weighted class activation mapping (Grad-CAM) [24,25] and local interpretable model-agnostic explanation (LIME) [26] are widely used to generate visualized heat maps, they are additional explanations to the trained model rather than inherent and understandable decision mechanisms of the model.

(2) Insufficient generalization stability: Many models perform well on specific datasets, but their performance may significantly degrade when faced with external data from different medical institutions or equipment models, and fluctuations in performance between validation and test sets often indicate overfitting risks [27,28].

(3) Imbalanced classes: In multi-class scenarios such as distinguishing COVID-19, other types of pneumonia, and normal images, the sensitivity and specificity of models in recognizing different classes are often uneven, and their ability to distinguish classes with similar manifestations is weak [21,29].

(4) High calculation cost: Many high-performance models have huge parameter quantities and slow inference speeds, and are difficult to be integrated into lightweight diagnostic systems in primary healthcare institutions or mobile devices [13,20,21].

To address the above challenges, researchers have explored various technological approaches. In response to the scarcity of annotated data, researchers have adopted methods such as data augmentation [1] and generative adversarial network (GAN) synthesis data to expand the training samples [30,31]; in response to the imbalanced data, researchers have designed targeted loss functions [29] and ensemble learning frameworks [32,33]; and in response to privacy protection and distributed training requirements, cutting-edge paradigms such as federated learning [12] and self-supervised learning [34–36] have also been introduced. However, most of these methods start from the data level or training strategy level and fail to fundamentally change the structure design of the model.

The recently proposed Kolmogorov–Arnold network (KAN) provides a new architectural solution to the aforementioned challenges. Its theoretical basis is derived from the Kolmogorov–Arnold representation theorem, which states that any multivariate continuous function can be expressed as the superposition of a finite number of univariate continuous functions and addition operations [37]. Unlike traditional multilayer perceptrons (MLPs) that use pre-set activation functions on fixed nodes, KAN places learnable spline functions at the network edge, which has stronger function approximation ability, higher parameter

efficiency, and inherent interpretability [38–42]. These characteristics demonstrate unique potential in scientific computing [43,44] and preliminary exploration of medical image analysis [45,46], providing new possibilities for constructing medical image classification models that combine high performance and interpretability. However, the original KAN was mainly designed to handle structured vector inputs, and it remains an open and challenging research direction of combining its excellent function learning and representation capabilities with CNN's powerful hierarchical visual feature extraction capabilities so as to tackle high-dimensional and spatially complex medical image classification tasks [47].

In response to the above issues, this paper proposes a ResKAN18 hybrid structure, which uses ResNet18 as the feature extraction backbone to extract multi-level spatial features of chest X-ray images, and replaces traditional fully connected classification heads with multilayer KAN modules to nonlinearly map 512 dimensional feature vectors to disease class probabilities. Three variants of the proposed ResKAN18 structure can achieve a flexible balance between accuracy and efficiency by adjusting the KAN depth (ResKAN18—Simple is three layers, ResKAN18—Standard is four layers, and ResKAN18—Large is also four layers but with wider hidden-layer dimensions).

Compared to the widely used CNN models, the main innovation of ResKAN18 proposed in this paper is reflected in three aspects:

(1) Structure level: For the first time, the learnable spline function of KAN is embedded into the CNN classification head, breaking through the linear expression limitations of fully connected layers and forming a collaborative paradigm of CNN feature abstraction combined with KAN nonlinear mapping.

(2) Regularization level: The local support of B-splines and L1 regularization form a structured sparse prior, resulting in training–validation accuracy gaps for all variants that are superior to the four classic CNN baselines in experiments.

(3) Deployment level: Through three deep configurations, covering the full spectrum of requirements from edge lightweight to high-precision diagnosis, users can flexibly choose according to hardware conditions.

This paper conducts a systematic comparative experiment of seven models (four classic CNN baselines and three ResKAN18 variants) to evaluate the comprehensive performance of ResKAN18 from classification accuracy, performance stability under intra dataset partitioning, COVID-19 specific recognition, and computational efficiency, providing data support for model selection in different clinical scenarios.

2. Research Methodologies

2.1. ResNet18 Feature Extraction Backbone

The performance of deep CNNs typically improves as their layers deepen. However, when the depth of the network exceeds a specific threshold, the model optimization process will face significant challenges, mainly manifested as gradient vanishing or gradient explosion, as well as network degradation. In other words, increasing depth leads to an increase in both training and testing errors. ResNet18 is a lightweight version of a deep residual network, which effectively solves the problems mentioned above that are commonly found in deep CNNs by introducing the residual learning mechanism. As the fundamental implementation of this paradigm, ResNet18 effectively stabilizes the training of deep networks by introducing structured shortcut connections and residual mapping, and achieves performance breakthroughs in multiple visual tasks.

2.1.1. Residual Learning Principles

The core innovation of ResNet18 lies in the reconstruction of traditional direct-mapping learning into learning of residual signals. The basic assumption is that, compared

to directly fitting a potential complex target mapping with stacked nonlinear layers, it is more feasible to optimize the residual between the target mapping and the input. This will transform the learning objective into the modeling process of incrementally correcting the input signal.

The idea of residual learning is implemented through standardized “residual blocks”. Each residual block contains two parallel paths. One is the residual path, which is composed of several convolutional layers, batch normalization layers, and activation functions connected in series, responsible for learning and outputting the incremental correction of input features, namely the residual; the other is the identity shortcut path: passing the input features as they are without any transformation. The outputs of the two paths are fused by element wise addition, and the resulting activation value serves as the final output of the block. This addition operation is the core of residual learning, which ensures that even if the output value of the residual path approaches zero, the network can still transmit and maintain the input signal without loss, so that theoretically, at least no performance loss will be introduced as the network depth increases.

The mathematic definition of residual block is as follows. If the input is x and the mapping learned by the residual path is $F(x, \{W_i\})$, which is usually a composite function of 2–3 layers of convolution, batch normalization, and rectified linear unit (ReLU) activation function, then the output of the residual block is:

$$y = F(x, \{W_i\}) + x \quad (1)$$

where x is directly passed through an identity shortcut. This addition operation is the core of residual learning.

From the optimization perspective, the gradient backpropagation of Equation (1) has special advantages. If the loss function is L , then the gradient of the input with respect to the loss is:

$$\frac{\partial L}{\partial x} = \frac{\partial L}{\partial y} \cdot \frac{\partial F}{\partial x} + \frac{\partial L}{\partial y} \quad (2)$$

The second term $\partial L / \partial y$ in Equation (2), which is the gradient of identity mapping, ensures that the gradient signal can be directly and almost without attenuation transmitted back to the shallow network. Even if the gradient $\partial F / \partial x$ of the residual path in deep networks approaches zero, the identity mapping term can still maintain gradient flow, fundamentally alleviating the problem of gradient vanishing. This is also the fundamental reason why ResNet can train hundreds of deep networks.

2.1.2. Network Structure of ResNet18

ResNet18 contains a total of 18 network levels with learnable parameters. Its structure adopts a staged design, gradually reducing the spatial resolution of the feature map in different stages while increasing its channel dimension. The first layer of the network uses 7×7 convolution combined with max pooling for fast downsampling, followed by stacking two residual blocks in each of the four stages. The first residual block in the second to fourth stages performs spatial downsampling through convolution with a step length of 2 and expands the number of channels to 128, 256, and 512 in sequence. When there is a dimension mismatch between the two ends of the shortcut connection, a 1×1 convolution is introduced for linear projection to align the feature dimensions. Finally, a 512-dimensional feature vector is generated through global average pooling and input into a fully connected layer to complete the classification task.

The core advantage of ResNet18 lies in the systematic optimization brought by its residual structure. To name a few, identity quick connections provide low impedance feed-back paths for gradients, effectively alleviating the problem of gradient vanishing in deep

networks and enabling the trainability of extremely deep networks; the residual learning mechanism transforms the optimization objective from fitting complex mappings to learning relatively simple residual functions, fundamentally eliminating network degradation and significantly reducing training and testing errors; by stacking small-sized convolution kernels and replacing traditional fully connected layers with global average pooling, the model achieves effective control of parameter quantity and improved computational efficiency while maintaining high representation ability, and has a certain regularization effect to reduce the risk of overfitting; in addition, the design of standardized residual blocks endows the network with good modularity and scalability, making it easy to scale systematically by adjusting the number of channels and stacking depth; at the same time, the introduction of quick connections smooths the loss function landscape, reduces sharp minima, and enhances the stability and convergence of the optimization process.

2.1.3. Disadvantages of Fully Connected Layer Classification Heads

Although ResNet18 has achieved significant success in feature extraction through residual learning, there are still structural limitations in the design of its end global average pooling fully connected layer. This design compresses rich spatial topological information into a single channel direction statistic before classification, resulting in the loss of spatial structural information; the subsequent fully connected layer only performs global linear projection, and its parameterized form solidifies the feature interaction mode, lacking the ability to adaptively model the complex, nonlinear, and sparse associations between different semantic features. This limits the parameter efficiency of the model under limited data and may constrain its ability to approximate upper bounds of more complex decision functions. Therefore, this module is considered a relatively rigid component in the structure, with weaker representation and flexibility than the convolution and residual modules with strong inductive bias in the frontend.

2.2. Kolmogorov–Arnold Network: Learnable Nonlinear Mapping Mechanism

The Kolmogorov–Arnold network (KAN) is an emerging neural network (NN) paradigm, whose structure design is directly rooted in the Kolmogorov–Arnold representation theorem. This theorem theoretically guarantees that any continuous multivariate function can be accurately represented by the superposition of a finite number of univariate functions. The KAN has engineered this mathematical principle by replacing fixed activation functions and linear weights in traditional multilayer perceptrons with learnable univariate functions, achieving significant breakthroughs in parameter efficiency, model interpretability, and scientific computational adaptability.

2.2.1. Kolmogorov–Arnold Representation Theorem and Network Structure Innovation

The principle of the KAN can be decomposed into two interrelated levels: mathematical foundation and architectural innovation.

(1) Mathematical Foundation

The Kolmogorov–Arnold representation theorem constitutes the theoretical core of KAN. It points out that any continuous multivariate function defined on a bounded domain can be precisely decomposed into a sum of a finite number of univariate continuous functions. This conclusion transforms the approximation problem of high-dimensional complex functions into a series of one-dimensional function construction problems, theoretically establishing the possibility of achieving universal function approximation through single variable nonlinear transformations alone, providing a strict theoretical guarantee for the sparse and modular design of KAN.

(2) Architectural Innovation

KAN has achieved a paradigm shift from “nonlinear nodes” to “nonlinear edges” and fundamentally reconstructed the basic computing units of traditional feedforward NNs. The comparison of basic computing units between traditional MLP and KAN is detailed in Table 1.

Table 1. Comparison of basic computing units between MLP and KAN.

Component	MLP	KAN
Edges	Perform only scalar multiplication (weights), which is a linear operation.	The core computing unit carries a learnable univariate nonlinear function (such as a spline function).
Nodes	The core computing units apply fixed, non-learning nonlinear activation function to the input weighted sum.	Perform simple summation operations to add the output values of the functions from all input edges.
Nonlinear source	Fixed activation functions configured at the nodes.	Functions that can adaptively learn configured on the edges.
Parameter storage	Weight matrices stored in the connections between nodes.	Parameters are stored in the adjustable coefficients of the univariate functions on the edges.

This paradigm shift allows each layer of KAN to be viewed as a “univariate function matrix”, and the depth of the network is achieved by stacking such function layers, effectively avoiding the pathological function forms that may occur in the original two-layer representation theorem, and improving engineering feasibility and generalization performance.

2.2.2. Implementation of B-Spline Parameterization

To achieve learnable univariate functions on the edges, the KAN uses B-spline as its core function parameterization tool. B-spline constructs the objective function through a linear combination of a set of local polynomial basis functions, which ensures theoretical rigor and engineering practicality. Let the input of the l -th layer KAN be $\mathbf{x}_l \in \mathbb{R}^{d_l}$, and the output be $\mathbf{x}_{l+1} \in \mathbb{R}^{d_{l+1}}$. The calculation of the KAN layer can be defined as:

$$\mathbf{x}_{l+1}^{(j)} = \sum_{i=1}^{d_l} \phi_{l,i,j}(\mathbf{x}_l^{(i)}), \quad j = 1, 2, \dots, d_{l+1} \quad (3)$$

where $\phi_{l,i,j}$ is a learnable univariate function that connects input neuron i to output neuron j . Unlike traditional MLP where each neuron uses a unified activation function such as ReLU, KAN learns a function independently on each edge, which gives it stronger representational power.

In specific implementation, each $\phi_{l,i,j}$ is parameterized using B-splines:

$$\phi_{l,i,j}(x) = \sum_{m=0}^{G+k-1} c_{l,i,j,m} B_{m,k}(x) \quad (4)$$

where G is the number of grid intervals (in this paper $G = 5$), k is the spline order (in this paper $k = 3$, i.e., cubic B-spline), $B_{m,k}(x)$ is the m -th k -order B-spline basis function, and $c_{l,i,j,m}$ are the coefficients to be trained. The B-spline basis function is defined by the Cox-de Boor recursive formula:

$$B_{m,0}(x) = \begin{cases} 1, & t_m \leq x < t_{m+1} \\ 0, & \text{otherwise} \end{cases} \quad (5)$$

$$B_{m,k}(x) = \frac{x - t_m}{t_{m+k} - t_m} B_{m,k-1}(x) + \frac{t_{m+k-1} - x}{t_{m+k-1} - t_{m+1}} B_{m+1,k-1}(x) \quad (6)$$

where $\{t_m\}$ is the knot sequence, which defines the support interval of the spline basis function. The knot position is dynamically adjusted during the training process through a grid update strategy.

Substitute Equation (4) into Equation (3) and the complete input–output mapping of the KAN layer can be obtained as:

$$x_{l+1}^{(j)} = \sum_{i=1}^{d_l} \sum_{m=0}^{G+k-1} c_{l,i,j,m} \cdot B_{m,k}(x_l^{(i)}) \quad (7)$$

Equation (7) indicates that the output of the KAN layer is a linear combination of B-spline transformations of the input dimensions. The core advantages of B-spline parameterization are reflected in three aspects: firstly, the B-spline function itself has high-order smoothness and differentiability everywhere, which ensures that the entire network can be stably and efficiently optimized end-to-end through a gradient-based backpropagation algorithm; secondly, B-spline has excellent local adaptability—by adjusting the grid density of their domain and the order of the polynomial used, it can flexibly control the function fitting behavior, thus achieving a balance between capturing local fine features of the data and maintaining a global smooth trend; finally, this parameterization approach has inherent efficiency, as the complexity of the function is determined only by a relatively small set of basis function coefficients—this fundamentally avoids the parameter explosion problem caused by the use of large and dense weight matrices in traditional multilayer perceptrons.

2.2.3. Core Advantages of the KAN Compared to MLP

Compared with traditional multilayer perceptrons, the architectural innovation of the KAN has endowed it with significant advantages in multiple aspects. Table 2 systematically compares and elaborates on the specific manifestations and underlying mechanisms of these core advantages.

Table 2. Core advantages of the KAN compared to MLP.

Advantage	Description	Explanation
Parameter efficiency and scaling characteristics	Using locally adaptive and learnable univariate functions can express function relationships in a sparser and more fundamental way. At the same level of accuracy, the number of parameters is much lower than that of MLP, and follows a better scaling law, resulting in faster performance growth with scale.	The inherent sparsity and structural priors enable the KAN to achieve high-precision approximation in a minimalist parameterized form, providing the possibility for training larger-scale models or deploying them in resource-constrained environments.
Model interpretability and transparency	Having inherent interpretability: 1. Function visualization: The univariate functions on each edge can be directly plotted, clearly showing how each input variable is transformed. 2. Structure pruning: A refined subnet can be obtained through sparse training, revealing key feature-interaction paths. 3. Symbolic potential: The learned smooth functions are easily approximated by symbols and transformed into mathematical expressions that are readable by humans.	The transformation from a “black box” to a “white box” or “gray box” has been achieved, supporting interactive scientific discovery and decision-making-process auditing, especially suitable for science and engineering fields that require strict interpretability.

Table 2. Cont.

Advantage	Description	Explanation
Adaptability to scientific computing and low-data mechanism	Suitable for structured problems: 1. Scientific modeling: capable of efficiently and accurately solving partial differential equations, and directly discovering potential physical laws and mathematical forms from data. 2. Low-sample learning: High parameter efficiency effectively suppresses overfitting and performs robustly under few sample conditions. 3. Symbolic reasoning: Proficient in learning and combining basic mathematical operations to achieve precise and interpretable function mappings.	The data-driven capabilities of NNs has been integrated with structured priors for scientific problems.
Theoretical rigor and structure guarantee	Instantiating the Kolmogorov–Arnold representation theorem directly in its structure, the width, depth, and function of each layer of the KAN have clear mathematical explanations, making it a model of “theory guided design”.	A rigorous mathematical foundation for model generalization ability has been established, reducing the blindness of structure search, and providing a more solid theoretical framework for understanding and analyzing network behavior.

It can be seen that the KAN achieves collaborative breakthroughs in parameter efficiency, model interpretability, scientific problem adaptation, and theoretical rigor by transforming profound mathematical theorems into innovative NNstructures. It is not intended to completely replace traditional DL models, but rather to provide a powerful new paradigm for solving key bottleneck problems that have long existed, especially in the fields of scientific computing and engineering that require high accuracy, strong interpretability, and the ability to learn from limited data.

2.3. ResKAN18 Hybrid Structure

The ResKAN18 model proposed in this paper deeply integrates ResNet18 which has powerful hierarchical feature extraction capability, and the KAN which has a flexible, efficient, and interpretable-function learning mechanism, aiming to construct a new diagnostic structure that not only has high classification accuracy, but also has advantages in performance stability and decision-process transparency.

2.3.1. Design Motivation and Theoretical Principle

In the research on chest X-ray recognition of COVID-19, ResNet18 shows important academic and application value because of its excellent balance between performance, efficiency and practicality [16]. With its lightweight structure and residual learning mechanism, it not only exhibits high classification accuracy and stability on small-sample medical data, but also effectively alleviates the risk of overfitting during training. Its small number of parameters and fast inference speed enable efficient real-time screening in clinical environments, especially suitable for scenarios with limited hardware resources. Meanwhile, the enhanced interpretability through visualization technology can help establish clinical trust and assist in locating lesion areas. In addition, the robustness demonstrated by the model on small-scale and imbalanced datasets further supports its rapid deployment in the early stages of the epidemic or primary healthcare.

However, despite its outstanding performance in feature extraction, the standard classification head of ResNet18 consisting of a global average pooling layer and a fully connected layer at the end has inherent limitations. In the process of converting spatial feature maps into class probability vectors, ResNet18 essentially performs a fixed pattern of global linear projection and weighted summation. This structure not only leads to a high degree of

compression and loss of rich spatial topological information, but its rigid parameterization form also limits the ability to finely model complex, nonlinear, and potentially sparse interactions between high-dimensional features, which may become a bottleneck that restricts the final representation and generalization performance of the model. To overcome this bottleneck and explore more expressive and interpretable classification paradigms, the ResKAN18 model is proposed in this paper. The core innovation of ResKAN18 lies in inputting the deep visual features extracted by the ResNet18 backbone network into a specially designed KAN module to replace the traditional fully connected classification module. The importance of this structure integration strategy is mainly reflected in the following three aspects:

Firstly, ResKAN18 achieves a paradigm shift from fixed linear mapping to adaptive nonlinear function learning. The KAN module can dynamically and accurately model the complex mapping relationship between 512-dimensional feature vectors and the final disease class probability through its learnable spline functions on the edges. Compared to the limited linear combination ability of fully connected layers, the KAN, with its theoretically guaranteed universal function approximation ability, can more flexibly capture subtle and nonlinear dependencies between features and diagnostic results, thus potentially uncovering deeper discriminative patterns.

Secondly, ResKAN18 significantly enhances potential interpretability. The learning process of the KAN module is directly reflected in the optimization of a series of univariate functions. This endogenous, function-level explanation mechanism is in stark contrast to traditional CNNs that typically rely on post-attribution methods, providing a new architectural foundation for building more transparent and trustworthy medical AI systems.

Finally, ResKAN18 is expected to improve performance stability while remaining lightweight. The KAN is known for its high parameter efficiency, which enables powerful function-fitting capabilities with relatively fewer parameters. This helps to reduce the risk of overfitting of the model on small-scale medical datasets. Meanwhile, its smooth optimization feature based on spline functions may make the learned decision boundaries more robust, thereby improving the performance consistency of the model on independent test sets, which is crucial for the reliability of clinical deployment.

2.3.2. Network Structure and Three Variants

To investigate the impact of KAN depth on performance and lay the foundation for subsequent comparative experiments, this paper has designed three different-depth KAN classification modules while keeping the ResNet18 feature extraction backbone unchanged, forming three ResKAN18 variants: ResKAN18—Large (four layers of KAN, hidden-layer dimensions [256, 128, 64]), ResKAN18—Standard (four layers of KAN, hidden-layer dimensions [128, 64, 32]), and ResKAN18—Simple (three layers of KAN, hidden-layer dimensions [64, 32]). These three variants will form a seven-model systematic comparison framework with four classic CNN baselines including ResNet18, VGG16, DenseNet121, and ShuffleNetV2 in subsequent experiments.

ResKAN18 is an innovative hybrid structure that combines the feature extraction capabilities of deep residual networks with the interpretable function approximation properties of KAN. The network adopts a dual-mode block design: the front-end uses the convolutional backbone network of ResNet18 for hierarchical feature extraction, while the back-end replaces the traditional fully connected layer with a multilayer KAN classifier to achieve nonlinear mapping from high-dimensional feature space to target classes. The front-end part of ResKAN18 strictly follows the standard ResNet18 and consists of four feature-learning stages, each stage consisting of multiple residual blocks stacked together. These residual blocks are fused with convolutional transformations through identity short-

cut connections, effectively alleviating the problem of gradient vanishing and gradually extracting multi-scale features from low-level textures to high-level semantics. At the backend of the network, spatial feature maps are aggregated into channel descriptors through global average pooling operations, generating a 512-dimensional feature vector. The back-end part of the structure completely reconstructed the classification head design, using a multilayer KAN network to form the classifier (the specific number of layers varies depending on the variant). Each layer of the KAN performs nonlinear transformations on input features through learnable univariate functions, while nodes only perform simple summation and aggregation. This design transforms the classification process from traditional black box matrix multiplication to transparent function combinations, where functions on each edge can be independently visualized and interpreted. The entire structure enhances the interpretability and parameter efficiency through the KAN module while maintaining strong feature extraction capabilities.

The network structure of ResKAN18—Large is shown in Table 3. The KAN module configuration for ResKAN18—Simple is $512 \rightarrow 64 \rightarrow 32 \rightarrow 3$ (three layers of KAN), while the KAN module configuration for ResKAN18—Standard is $512 \rightarrow 128 \rightarrow 64 \rightarrow 32 \rightarrow 3$ (four layers of KAN). The ResNet18 feature extraction backbone of the three variants is completely identical. The parameter counts of ResKAN18—Simple, ResKAN18—Standard and ResKAN—Large are 11.53M, 13.01M and 13.97M. These three values are much smaller than VGG16 whose parameter count is 134.27M and are on the same level as ResNet18 whose parameter count is 11.18M. This parameter quantity distribution enables the three variants to cover a complete spectrum ranging from “almost no increase in computational cost” (ResKAN18—Simple only increases parameter quantity by 3.1% compared to ResNet18) to “moderately increasing cost for accuracy improvement” (ResKAN18—Large increases parameter quantity by 24.9% compared to ResNet18), providing flexible choices for different deployment scenarios such as resource constraints and accuracy sensitivity.

Table 3. Network structure of ResKAN18—Large.

Network Module	Submodule	Output Dimension	Configuration Description and Core Functions
Feature-extraction module	Initial convolutional layer	$112 \times 112 \times 64$	7×7 convolution, step size 2, output 64 channels for preliminary feature extraction and downsampling
	Initial pooling layer	$56 \times 56 \times 64$	3×3 max pooling with a stepsize of 2
	Stage 1 residual block	$56 \times 56 \times 64$	2 standard residual blocks, each containing 23×3 convolutional layers (64 channels)
	Stage 2 residual block	$28 \times 28 \times 128$	2 residual blocks, the first block downsamples and outputs 128 channels
	Stage 3 residual block	$14 \times 14 \times 256$	2 residual blocks, the first block downsamples and outputs 256 channels
	Stage 4 residual block	$7 \times 7 \times 512$	2 residual blocks, the first block downsamples and outputs 512 channels
	Global average pooling	512	Aggregate the $7 \times 7 \times 512$ feature maps into a 512-dimensional feature vector, which directly serves as the input dimension for the KAN classifier

Table 3. Cont.

Network Module	Submodule	Output Dimension	Configuration Description and Core Functions
KAN classification module	KAN layer 1	256	Input 512 dimensions, output 256 dimensions
	KAN layer 2	128	Input 256 dimensions, output 128 dimensions
	KAN layer 3	64	Input 128 dimensions, output 64 dimensions
	KAN layer 4	3	Input 64 dimensions, output 3 dimensions

2.3.3. Model-Operation Principles and Key Optimization Factors

Based on the above structure design, the operation of ResKAN18 can be summarized as a two-stage process:

(1) Hierarchical feature extraction (ResNet18 backbone). The input chest X-ray image ($224 \times 224 \times 3$) is sequentially processed through the initial convolutional layer (7×7 , step length = 2), max pooling layer (3×3 , step length = 2), and four residual stages (each containing 2 residual blocks, with channel numbers of 64, 128, 256, and 512) of ResNet18, gradually reducing the spatial resolution from 224×224 to 7×7 and increasing the channel number from 3 to 512. The end global average pooling layer aggregates the 7×7 feature maps of each channel into a single scalar, generating a 512-dimensional feature vector f_{CNN} . This feature vector condenses the global semantic information of the input image and serves as the input for subsequent KAN classifiers.

(2) Fine nonlinear mapping (KAN classification module). The 512-dimensional feature vector f_{CNN} is sequentially passed through multiple layers of KAN. The configuration of the three variants is as follows: ResKAN18—Large is $512 \rightarrow 256 \rightarrow 128 \rightarrow 64 \rightarrow 3$ (four layers of KAN), ResKAN18—Standard is $512 \rightarrow 128 \rightarrow 64 \rightarrow 32 \rightarrow 3$ (four layers of KAN), and ResKAN18—Simple is $512 \rightarrow 64 \rightarrow 32 \rightarrow 3$ (three layers of KAN). Each layer of KAN independently performs B-spline transformation on the input dimensions according to Equation (7), and then sums and aggregates them at the nodes. The final output is a 3D logits vector, which is normalized by Softmax to three probability distributions: COVID-19, normal, and viral pneumonia.

The three key factors for achieving optimization results are as follows:

(1) Dynamic grid update. Every 5 training epochs, the KAN module recalculates the node sequence of B-splines based on the feature distribution of the current batch of data. This mechanism ensures that the support interval of the spline basis function closely follows the data manifold, which is equivalent to continuously optimizing the “coordinate scale” of the feature space during the training process, making subsequent nonlinear transformations more accurate.

(2) Gradient Protection for residual connections. The identity mapping of ResNet18 shown in Equation (1) ensures that the gradient signal can be directly transmitted back to the shallow layer, as shown in Equation (2), effectively avoiding the risk of gradient vanishing caused by the additional depth of the KAN module. This allows ResKAN18 to be trained end-to-end with the same learning rate (0.0001) and optimizer (Adam) as all seven models, without the need to adjust the training strategy.

(3) Structured regularization of B-splines. The local support of B-splines naturally limits the complexity of each univariate function. Combined with the explicit addition of spline coefficient L1 regularization in the loss function, this mechanism helps to suppress overfitting and improve the generalization performance of the model.

To ensure the reproducibility of the study, Algorithm 1 fully describes the forward propagation, loss calculation, gradient backpropagation, and dynamic grid update process of ResKAN18.

Algorithm 1 ResKAN18 algorithm applied to chest X-ray image classification

Input:

chest X-ray image $X=\{x_1, \dots, x_N\}$,
 corresponding label $Y=\{y_1, \dots, y_N\}$,
 training rounds $E=30$,
 batch size $B=256$,
 grid update frequency $f=5$

Output:

The model parameters $\Theta=\{\theta_{\text{ResNet}}, \{c_{\{l,i,j,m\}}\}\}$ after training

```

1: Initialize ResNet18 backbone using ImageNet pre-trained weights
   (weights = ResNet18_Weights.IMAGENET1K_V1)
2: Randomly initialize the spline coefficients  $c_{l,i,j,m}$  of each layer in KAN
3: Note: The configuration of the KAN layer in three variants is as follows
   (L is the total number of layers):
   ResKAN18--Large: L = 4, dimension 512→256→128→64→3
   ResKAN18--Standard: L = 4, dimension 512→128→64→32→3
   ResKAN18--Simple: L = 3, dimension 512→64→32→3
4: for epoch = 1 to E do
5:   for each mini-batch (x, y) in X, Y do
6:     f_cnn ← ResNet18_backbone(x)
7:     x_1 ← f_cnn
8:     x_list ← [x_1]
9:     u_list ← []
10:    for l = 1 to L do
11:      for j = 1 to d_{l+1} do
12:         $u_{\{l,j\}} \leftarrow \sum_{i=1}^{d_l} \sum_{m=0}^{G+k-1} c_{\{l,i,j,m\}} \cdot B_{\{m,k\}}(x_1^{\{i\}})$ 
13:         $x_{\{l+1\}^{\{j\}}} \leftarrow u_{\{l,j\}}$ 
14:      end for
15:      x_list.append(x_{l+1})
16:      u_list.append(u_l)
17:    end for
18:    p ← Softmax(x_{L+1})
19:    L ← CrossEntropy(p,y) +  $\lambda \cdot \sum_{l=1}^L \sum_{i,j,m} |c_{\{l,i,j,m\}}|$ 
20:    L.backward()
21:    optimizer.step()
22:  end for
23:  if epoch mod f = 0 then
24:    Update the grid using the input x_l and unreduced output u_l of each layer
      (i.e., data stored in x_list and u-list) saved in the current mini-batch
25:    for l = 1 to L do
26:      grid_l ← update_grid(x_l)
27:      c_l ← curve2coeff(x_l, u_l)
28:    end for
29:  end if
30:  Evaluate the accuracy and loss of the current model on the validation set
31: end for
32: return  $\Theta$ 

```

3. Experiment Results and Analysis

3.1. Data Acquisition and Experiment Environment Setup

The dataset used in this paper is a publicly available chest X-ray imaging database, aimed at promoting the development and validation of AI-based automatic detection models for COVID-19 and viral pneumonia. This dataset has been systematically orga-

nized and contains a total of 3880 chest X-ray images, covering the following three classes: COVID-19-positive cases (1198 images), viral pneumonia cases (1343 images), and normal images (1339 images), as shown in Figure 1. By providing annotated data of comparable scale and balanced classes, this dataset helps support robust and scalable AI-driven diagnostic research. The data is sourced from the shared dataset disclosed in reference [48], which provides an important benchmark resource for intelligent analysis of chest X-ray images.

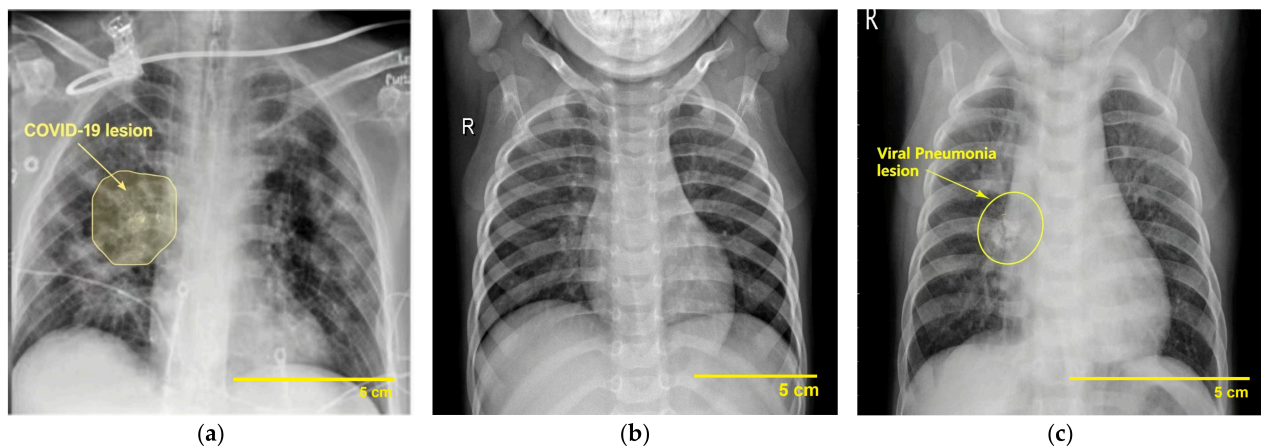


Figure 1. Chest X-ray image examples of: (a) COVID-19; (b) normal; (c) viral pneumonia.

It should be noted that the current clinical research has gradually turned to the common image of coronavirus and the systemic characterization of COVID-19 sequelae. However, the COVID-19 chest X-ray dataset used in this paper is not intended to mine the specific image lesions of COVID-19, but is used as a recognized standardized benchmark dataset in the field of multi-classification of respiratory images. The dataset contains three types of balanced samples, namely, COVID-19, normal, and ordinary viral pneumonia samples. The imaging distinguishing characteristics of each type are clear, which can fully test the discrimination accuracy, performance stability under intra dataset partitioning and class-balanced recognition ability of the proposed hybrid NN structure. At the same time, this dataset is a commonly used public benchmark in the field, which can achieve quantitative horizontal comparison with existing mainstream CNN models. The core innovation of this paper is the ResKAN18 general classification structure for medical images. The structure is only validated on the chest X-ray dataset related to COVID-19; expanding it for intelligent diagnosis of other chest diseases such as tuberculosis and lung tumors is a potential research direction that needs to be experimentally validated in the future.

In order to ensure the reliability of the model evaluation, a standard three-dataset partitioning strategy has been used to preprocess all 3880 images: 70% of the samples are randomly selected as the training set for learning model parameters; 15% of the samples are used as validation sets to monitor performance and perform hyperparameter tuning during the training process; and the remaining 15% of the samples are used as an independent test set for unbiased evaluation of the final generalization ability of the models to be evaluated, completing 30 complete training epochs. This partitioning strategy is carried out at the overall level, and attention is paid to maintaining the independence of samples between the training set, validation set, and test set to avoid data leakage and ensure scientific and rigorous evaluation results. To ensure complete reproducibility of the experiment, fixed random seeds were used for all random operations.

The experimental hardware system in this paper adopts a high-performance computing platform, equipped with Intel’s 13th-generation i9-13900KF processor (with a main frequency of 3.00 GHz), 64 GB DDR4 memory, and NVIDIA GeForce RTX 4080 SUPER graphics card (16 GB GDDR6X video memory), combined with a 3.68 TB NVMe solid-state storage system, providing sufficient parallel computing power and data throughput bandwidth for DNN training and inference.

To comprehensively evaluate the superiority of the ResKAN18 structure and systematically explore the impact of KAN depth on performance, as mentioned before, this paper has designed seven model systematic comparative experiments. The comparative models are divided into two groups: (1) Four classic CNN baselines: VGG16 [18] (classic deep serial structure whose parameter count is 134.27M), DenseNet121 [17] (dense connectivity structure whose parameter count is 6.96M), ShuffleNetV2 [19] (lightweight network whose parameter count is 1.26M), and ResNet18 [16] (direct baseline in this paper whose parameter count is 11.18M). (2) Three ResKAN18 variants: ResKAN18—Simple, ResKAN18—Standard, and ResKAN18—Large which is recommended as the default configuration in this paper.

All models are trained and evaluated under identical data partitioning (training:validation:testing = 70%:15%:15%), data preprocessing flow (size adjustment, data augmentation strategy, normalization parameters), optimizer configuration (Adam), and basic training hyperparameters (30 rounds, learning rate 0.0001, batch size 256) to ensure fairness in comparison. It should be noted that VGG16, DenseNet121, and ShuffleNetV2 all use ImageNet pretrained weights provided officially by PyTorch 2.7.1+cu118 for initialization, while ResNet18 and ResKAN18 variants also use ImageNet pre-trained weights. The hyperparameter settings for ResNet18 and ResKAN18 series networks are detailed in Table 4.

Table 4. Hyperparameter settings of ResNet18 and ResKAN18 series.

Model	Parameter Category	Parameter Name ¹	Value Set	Explanation
ResNet18	Model structure	Number of output classes	3	Corresponding to 3 classes: COVID-19, normal and viral pneumonia
	Training parameters	Learning rate	0.0001	Initial learning rate of Adam optimizer
		Batch size	256	The number of samples in each training batch
		Optimizer	Adam	The 2 optimized weight coefficients are 0.9 and 0.999 respectively
		Loss function	CrossEntropyLoss	Multi-class cross-entropy loss
		Training epochs	30	Complete number of training epochs
	Data preprocessing	Random horizontal flipping probability	0.5	Data augmentation, with a 50% probability of flipping the image horizontally
		Random vertical flipping probability	0.2	Data augmentation, with a 20% probability of flipping the image vertically
		Normalized mean	[0.5063, 0.5063, 0.5063]	Channel mean calculated based on the training set
		Normalized standard deviation	[0.2390, 0.2390, 0.2390]	Channel standard deviation calculated based on the training set

Table 4. Cont.

Model	Parameter Category	Parameter Name ¹	Value Set	Explanation
ResKAN18 series	Structure parameters	Hidden-layer dimension of the KAN	ResKAN18—Large: [256, 128, 64]	The feature dimension of the intermediate layer in the network
		Grid size	5	The number of grid points for B-spline function
		Spline order	3	Using a 3rd-order B-spline function
		Basic activation function	SiLU	Sigmoid linear unit
		Grid range	[−1, 1]	The domain of spline functions
	Training parameters	Grid update frequency	Every 5 epochs	Dynamically adjust the spacing of spline grids
		Grid adjustment coefficient	0.02	Control the magnitude of grid updates
		Activate regularization coefficient	1.0	Control the sparsity of spline coefficients
		Entropy regularization coefficient	1.0	Control the uniformity of activation weight distribution
		Basic scaling factor	1.0	Scaling factor during initialization of basic weights
		Spline scaling factor	1.0	Scaling factor during spline weight initialization

¹ All models share training parameters (learning rate, batch size, optimizer, loss function, number of training epochs, data augmentation strategy, normalization parameters) in the same data partitioning (training:validation:testing table) among all seven models. The three ResKAN18 variants share the same structure parameters (grid size, spline order, base activation function, grid range, grid update frequency, etc.), with differences only in the KAN hidden-layer dimension.

3.2. Evaluation Indicators

This paper systematically monitors and compares the convergence speed, stability, and performance under intra dataset partitioning of ResNet18 and ResKAN18 during the learning process by training a series of indicators, including training accuracy, validation accuracy, training loss, validation loss, training F1 macro average, validation F1 macro average, training area under curve (AUC)—one-vs-rest (OvR), and validation AUC—OvR. Moreover, this paper also provides a multi-dimensional quantitative evaluation and comparative analysis of the final classification ability of ResNet18 and ResKAN18 by using a comprehensive performance indicator system, including accuracy, macro precision, macro recall, macro F1, weighted precision, weighted recall, weighted F1, AUC—OvR, AUC—One-vs-One (OvO), Cohen’s Kappa, Matthews Correlation, Log Loss, and a normalized confusion matrix. The specific indicators are shown in Tables 5 and 6, the former is used to analyze training dynamics, while the latter is used to determine the overall performance.

Table 5. Key performance indicators of training dynamics.

Indicator Name	Explanation	Function in Model Evaluation
Training accuracy	The proportion of samples correctly predicted by the model on the current training set.	It reflects the fitting degree of the model to known data. During the training process, its value usually gradually increases. If the value remains too high and the validation accuracy is low, it is a typical signal of overfitting.
Validation accuracy	The proportion of samples correctly predicted by the model on an independent validation set.	It is the core indicator for evaluating the generalization ability of a model, used to monitor the performance of the model on unseen data. Its value is the direct basis for selecting the best model and adjusting hyperparameters, and its trend with the number of training epochs is crucial.
Training loss	The loss function value calculated by the model on the training set.	It is a direct measure of the fitting degree of a model to training data. In an ideal scenario, it should steadily decline and flatten out with training, indicating that the model is effectively learning from the data.
Validation loss	The loss function value calculated by the model on the validation set.	It is a key indicator for determining whether a model is overfitting or underfitting. Its value usually decreases and then increases with training, and the lowest point usually corresponds to the model state with the best generalization performance, which is the main basis for determining the timing of “early stopping”.
Training F1 macro average	The macro average F1 score calculated by the model on the training set.	It comprehensively evaluates the balance performance of the model on the training data for each class. Combining observation with training accuracy can more comprehensively reflect the learning quality of the model when processing various classes.
Validation F1 macro average	The macro average F1 score calculated by the model on the validation set.	It is an important indicator for evaluating the comprehensive classification robustness of a model on unseen data. In datasets with imbalanced classes, it better reflects the actual generalization ability of the model than validation accuracy and is one of the key references for model selection.
Training AUC-OvR	The average area under the receiver operating characteristic (ROC) curve calculated by the “one-vs-rest (OvR)” strategy for the model on the training set.	It measures the ability of a model to rank the classes of training data, focusing on the probability mass assigned to each sample by the model. A high value indicates that the model has learned good class discrimination features on the training set.
Validation AUC-OvR	The average area under the ROC curve calculated by the “OvR” strategy for the model on the validation set.	It is an indicator for evaluating the generalization ability of a model. Due to the relative insensitivity of AUC to class imbalance, it can effectively measure the ability of the model to correctly sort unknown samples, and is often used together with validation F1 for final model performance evaluation.

Table 6. Overall key performance indicators.

Indicator Name	Explanation	Function in Model Evaluation
Accuracy (%)	The proportion of samples whose model prediction results are completely consistent with the true labels in all samples.	It provides an intuitive summary of the overall classification performance of the model, reflecting its overall ability to make correct judgments.
Macro precision	The arithmetic mean of the accuracy of each class (the proportion of samples predicted to belong to that class that truly belong to that class), treating each class as equally important.	It evaluates the predictive accuracy of the model on various independent classes, emphasizing fair evaluation of each class, regardless of the sample size.
Macro recall	The arithmetic mean of the recall rate of each class (the proportion of samples belong to that class that predicted to belong to that class), treating each class as equally important.	It evaluates the ability of the model to identify or capture samples from various classes, with equal attention to the risk of omission for each class.
Macro F1	The harmonic mean of macro precision and macro recall.	It provides a single, balanced score to comprehensively evaluate the balanced performance of the accuracy and recall of the model in each class.
Weighted precision	The average value obtained by weighting the accuracy based on the proportion of samples in each class.	It evaluates the overall prediction accuracy of the model in datasets with imbalanced sample sizes, where classes with larger sample sizes have a greater impact on the indicators.
Weighted recall	The average value obtained by weighting the recall rate based on the proportion of samples in each class.	It evaluates the overall recognition ability of the model in datasets with imbalanced sample sizes, focusing more on the recall of classes with larger sample sizes.
Weighted F1	The harmonic mean of weighted precision and weighted recall.	It evaluates the comprehensive performance more in line with the reality of data distribution and balanced in a dataset with imbalanced classes.
AUC-OvR	The average value of multiple area under ROC curves, adopting the “OvR” strategy, treating each class as a positive class and treating the remaining classes as negative classes.	It measures the overall ability of the model to distinguish each specific class from all other classes, and it is insensitive to the selection of classification thresholds.
AUC—one-vs-one (OvO)	The average value of area under binary ROC curves for every combination of 2 classes, adopting the “OvO” strategy.	It finely measures the discriminative power of the model in distinguishing between any two classes, suitable for scenarios with subtle differences between classes.
Cohen’s Kappa	A statistical measure used to measure the consistency between the predicted results of the model and the true labels, on the basis of excluding consistency caused by random coincidences.	It evaluates the actual classification consistency of the model beyond the level of random guessing, and it is particularly beneficial for datasets with imbalanced class distributions.
Matthews correlation	A comprehensive indicator that takes into account true positives (TPs), true negatives (TNs), false positives (FPs), and false negatives (FNs), essentially acting as a correlation coefficient between the predicted results and the true labels.	It fairly evaluates the classification consistency within the range $[-1, 1]$ even if the sample size of each class varies greatly, and it is considered one of the most reliable single indicators for class imbalance problems.
Log loss	A indicator that measures the quality of the predicted probability output by a model; for each sample, the closer the predicted probability given by the model is to its true label, the lower the log loss value.	It directly evaluates the calibration level and confidence of the model probability estimation, its lower value means that the model not only predicts the right class, but also has a high degree of confidence in its own predictions.
Normalized confusion matrix	A matrix that converts the numbers of each row (representing a true class) of the standard confusion matrix into a percentage of the total sample size of that row, and the sum of the values in each row is 100%.	It visually and fairly compares the classification of different classes. It eliminates the influence of sample size differences and clearly shows the proportion of samples from a true class that are incorrectly assigned to other classes by the model, thus quickly identifying the main sources of confusion.

3.3. Comparative Analysis of Convergence During the Training Process

To gain a deeper understanding of the impact of the KAN module on training dynamics, this section selects the baseline model ResNet18 and the recommended default configuration ResKAN18—Large for in-depth comparative analysis of the training process. The selection criteria are as follows: ResNet18 represents the typical behavior of traditional CNN classification heads, while ResKAN18—Large represents the deepest configuration in the proposed KAN enhancement structure (four layers of KAN, hidden-layer dimensions [256, 128, 64]). The comparison between the two can most clearly reveal the impact mechanism of the KAN module on convergence speed, training stability, and performance under intra dataset partitioning. The complete performance comparison of the remaining five models (VGG16, DenseNet121, ShuffleNetV2, ResKAN18—Simple, ResKAN18—Standard) will be presented in Section 3.5.

Tables 7–9 and Figures 2–5 show a deep comparison of the training process between ResNet18 and ResKAN18—Large. The results show that the validation accuracy of the two models after multiple iterations can approach or exceed 98%, and both have the ability to recognize COVID-19 images, but there are significant differences in convergence speed, training stability and performance under intra dataset partitioning between the two models.

Table 7. Comparison of overall key performance indicators of training dynamics.

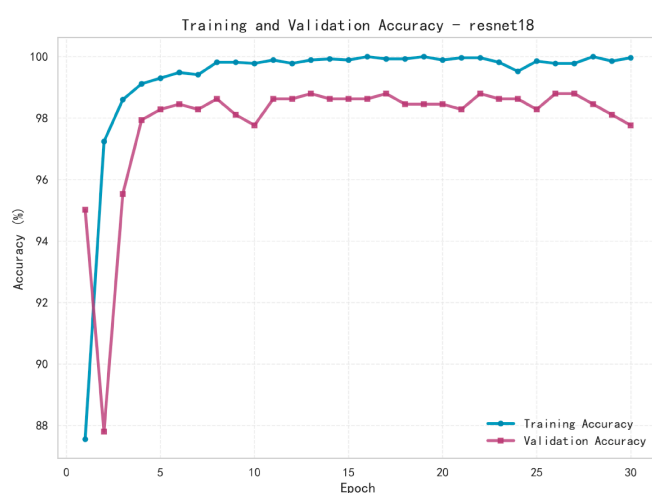
Training Epoch	Model	Training Accuracy (%)	Validation Accuracy (%)	Training Loss	Validation Loss	Training F1 Macro Average	Validation F1 Macro Average	Training AUC-OvR	Validation AUC-OvR
1	ResNet18	87.56	95.02	0.3448	0.1646	0.8763	0.9511	0.9734	0.9965
	ResKAN18—Large	86.45	92.96	0.9994	0.7584	0.8636	0.9319	0.9676	0.9837
2	ResNet18	97.24	87.80	0.0714	0.2732	0.9732	0.8777	0.9975	0.9949
	ResKAN18—Large	94.33	95.02	0.6800	0.3356	0.9447	0.9517	0.9861	0.9893
5	ResNet18	99.30	98.28	0.0239	0.0613	0.9932	0.9830	0.9999	0.9988
	ResKAN18—Large	98.16	98.63	0.0715	0.0505	0.9821	0.9866	0.9991	0.9976
10	ResNet18	99.78	97.77	0.0082	0.0576	0.9979	0.9780	1.0000	0.9988
	ResKAN18—Large	99.45	99.14	0.0184	0.0329	0.9946	0.9914	0.9996	0.9985
15	ResNet18	99.89	98.63	0.0032	0.0340	0.9989	0.9865	1.0000	0.9990
	ResKAN18—Large	99.96	98.97	0.0047	0.0517	0.9996	0.9899	1.0000	0.9988
20	ResNet18	99.89	98.45	0.0021	0.0401	0.9989	0.9848	1.0000	0.9990
	ResKAN18—Large	99.78	98.97	0.0052	0.0526	0.9979	0.9898	1.0000	0.9995
25	ResNet18	99.85	98.28	0.0074	0.0352	0.9986	0.9831	1.0000	0.9987
	ResKAN18—Large	99.93	98.11	0.0029	0.1034	0.9993	0.9815	1.0000	0.9996
30	ResNet18	99.96	97.77	0.0027	0.0571	0.9996	0.9780	1.0000	0.9991
	ResKAN18—Large	99.78	98.28	0.0058	0.0549	0.9978	0.9831	1.0000	0.9996

Table 8. Statistics comparison of overall training performance.

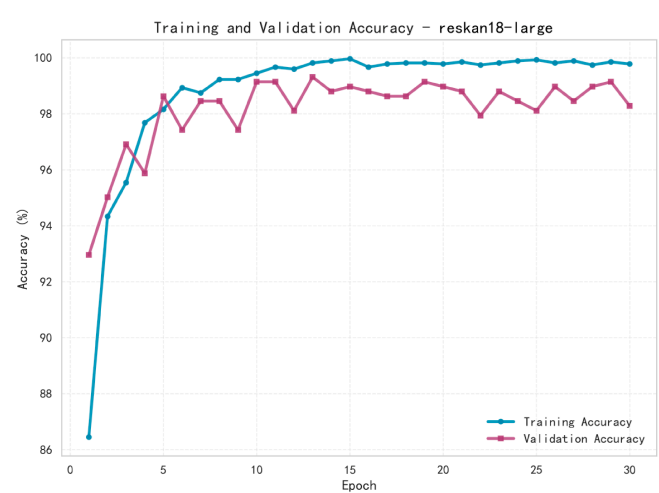
Statistics Indicator	Model	Training Accuracy (%)	Validation Accuracy (%)	Training Loss	Validation Loss	Training F1 Macro Average	Validation F1 Macro Average	Training AUC-OvR	Validation AUC-OvR
Average	ResNet18	99.56	97.98	0.0096	0.0660	0.9957	0.9804	0.9995	0.9992
	ResKAN18—Large	99.71	98.45	0.0253	0.0881	0.9972	0.9848	0.9997	0.9995
Standard deviation	ResNet18	3.34	1.87	0.0612	0.0472	0.0334	0.0187	0.0056	0.0021
	ResKAN18—Large	3.05	1.31	0.1855	0.1604	0.0304	0.0135	0.0085	0.0044
Median	ResNet18	99.93	98.80	0.0033	0.0429	0.9993	0.9880	1.0000	0.9990
	ResKAN18—Large	99.89	97.94	0.0067	0.0580	0.9989	0.9799	0.9999	0.9990
Minimum	ResNet18	87.19	87.80	0.0014	0.0320	0.8728	0.8757	0.9734	0.9949
	ResKAN18—Large	87.52	91.41	0.0008	0.0335	0.8776	0.9168	0.9676	0.9812
Maximum	ResNet18	100.00	99.14	0.3390	0.2695	1.0000	0.9914	1.0000	0.9995
	ResKAN18—Large	100.00	98.97	0.9890	0.7536	1.0000	0.9899	1.0000	0.9996

Table 9. Convergence characteristics and stability analysis.

Convergence Characteristic	ResNet18	ResKAN18—Large	Comparative Analysis
Initial convergence speed	Verification accuracy in the 1st epoch: 95.02%	Verification accuracy in the 1st epoch: 92.96%	ResNet18 converges faster in the early stages.
Stable convergence period	Reached stability around the 15th epoch	Reached stability around the 10th epoch	ResKAN18—Large enters the stable status earlier.
Final convergence level	Verification accuracy: 97.77%	Verification accuracy: 98.28%	ResKAN18—Large has slightly better final performance (+0.51%).
Overfitting degree (training–validation accuracy difference)	2.19%	1.50%	ResKAN18—Large has a slightly higher tendency towards overfitting.
Smoothness of loss curve	Significant fluctuation in validation loss	Small fluctuation in validation loss	The loss curve of ResKAN18—Large is smoother.



(a)



(b)

Figure 2. Comparison of accuracy curves: (a) ResNet18; (b) ResKAN18—Large.

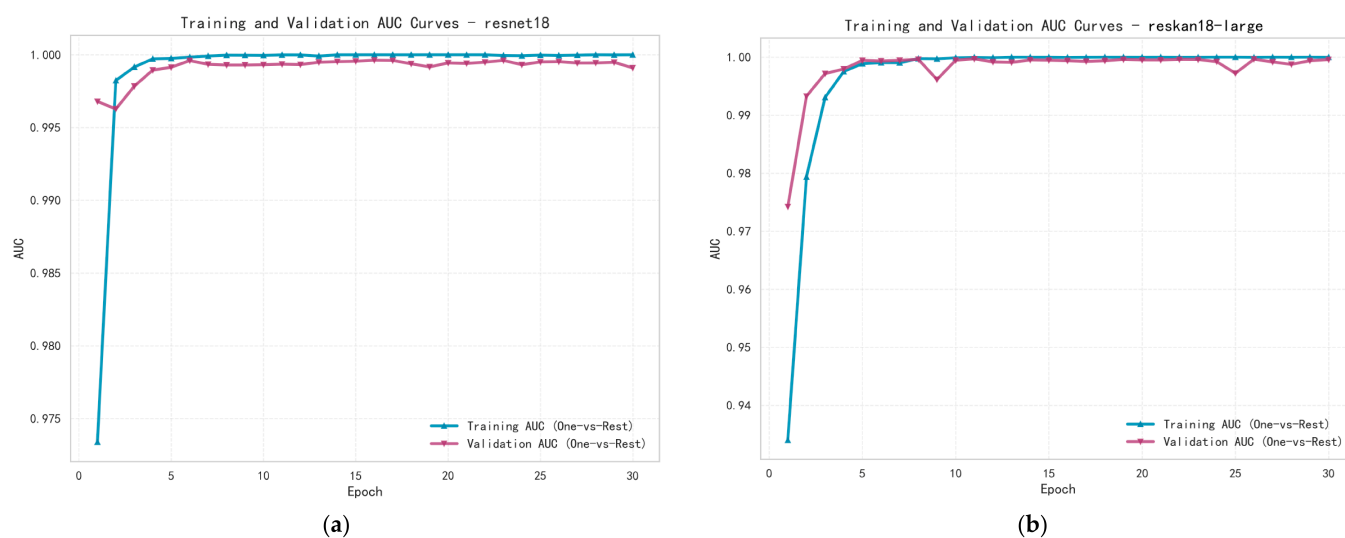


Figure 3. Comparison of AUC-OvR curves: (a) ResNet18; (b) ResKAN18—Large.

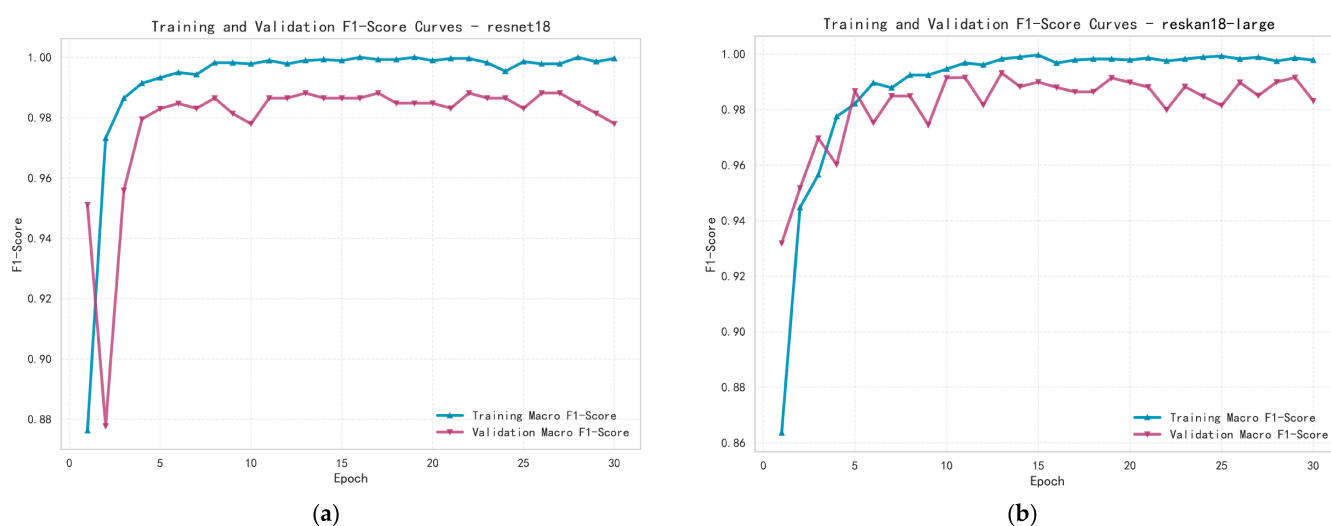


Figure 4. Comparison of macro F1 curves: (a) ResNet18; (b) ResKAN18—Large.

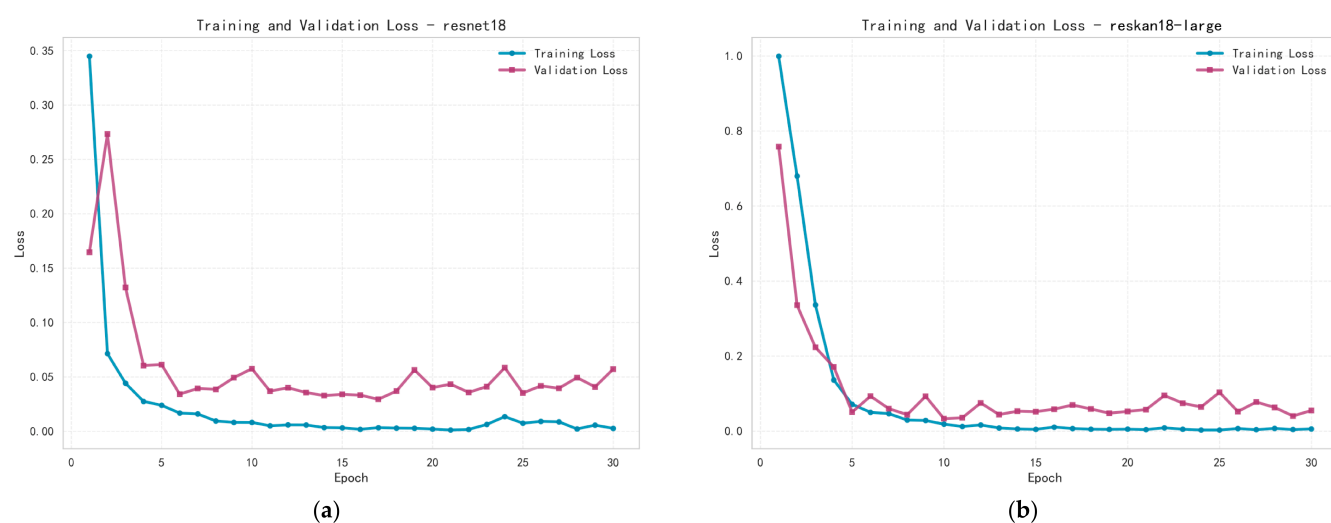


Figure 5. Comparison of loss curves: (a) ResNet18; (b) ResKAN18—Large.

As shown in Figures 2–5, ResNet18 exhibited faster convergence speed in the early stages of training (with a validation accuracy of 95.02% in the first epoch), but its pro-

cess fluctuated greatly, with a sudden drop in validation accuracy to 87.80% in the second epoch, indicating sensitivity to initialization and insufficient stability. Although ResKAN18—Large has a slightly slower initial convergence rate (92.96% validation accuracy in the first epoch), the training process is smoother, and ultimately outperforms ResNet18's validation accuracy with a validation accuracy of 98.28%. It also performs better in validation F1 macro average and validation loss, as shown in Tables 10–12.

Table 10. Comparison of overall key performance among the training set, validation set and test set of seven models.

Indicator	Dataset	ResNet18	VGG16	DenseNet121	ShuffleNetV2	ResKAN18—Simple	ResKAN18—Standard	ResKAN18—Large
Accuracy (%)	Training set	99.93	99.96	99.93	99.67	99.93	99.78	99.85
	Validation set	97.77	98.45	98.45	97.94	97.94	98.45	98.28
	Test set	97.59	97.77	98.11	97.77	98.45	98.45	98.80
Macro precision	Training set	0.9993	0.9996	0.9993	0.9964	0.9992	0.9979	0.9985
	Validation set	0.9787	0.9848	0.9848	0.9793	0.9812	0.9856	0.9831
	Test set	0.9766	0.9787	0.9822	0.9785	0.9851	0.9848	0.9883
Macro recall	Training set	0.9993	0.9996	0.9993	0.9964	0.9992	0.9979	0.9985
	Validation set	0.9780	0.9848	0.9848	0.9793	0.9805	0.9856	0.9831
	Test set	0.9766	0.9787	0.9822	0.9785	0.9851	0.9848	0.9883
Macro F1	Training set	0.9993	0.9996	0.9993	0.9964	0.9992	0.9979	0.9985
	Validation set	0.9780	0.9848	0.9848	0.9793	0.9805	0.9856	0.9831
	Test set	0.9766	0.9787	0.9822	0.9785	0.9851	0.9848	0.9883
Weighted precision	Training set	0.9993	0.9996	0.9993	0.9964	0.9992	0.9979	0.9985
	Validation set	0.9787	0.9848	0.9848	0.9793	0.9811	0.9856	0.9831
	Test set	0.9766	0.9787	0.9822	0.9785	0.9851	0.9848	0.9883
Weighted recall	Training set	0.9993	0.9996	0.9993	0.9964	0.9992	0.9979	0.9985
	Validation set	0.9780	0.9848	0.9848	0.9793	0.9805	0.9856	0.9831
	Test set	0.9766	0.9787	0.9822	0.9785	0.9851	0.9848	0.9883
Weighted F1	Training set	0.9993	0.9996	0.9993	0.9964	0.9992	0.9979	0.9985
	Validation set	0.9780	0.9848	0.9848	0.9793	0.9804	0.9856	0.9831
	Test set	0.9766	0.9787	0.9822	0.9785	0.9851	0.9848	0.9883
AUC-OvR	Training set	1.0000	1.0000	1.0000	0.9999	1.0000	1.0000	1.0000
	Validation set	0.9991	0.9995	0.9995	0.9973	0.9990	0.9998	0.9996
	Test set	0.9992	0.9989	0.9988	0.9958	0.9987	0.9995	0.9998
AUC-OvO	Training set	1.0000	1.0000	1.0000	0.9999	1.0000	1.0000	1.0000
	Validation set	0.9991	0.9995	0.9995	0.9973	0.9990	0.9998	0.9996
	Test set	0.9992	0.9989	0.9988	0.9958	0.9987	0.9995	0.9998
Cohen's Kappa	Training set	0.9989	0.9994	0.9989	0.9945	0.9989	0.9967	0.9978
	Validation set	0.9665	0.9768	0.9767	0.9691	0.9690	0.9767	0.9741
	Test set	0.9639	0.9663	0.9715	0.9664	0.9768	0.9768	0.9819
Matthews correlation	Training set	0.9989	0.9994	0.9989	0.9945	0.9989	0.9967	0.9978
	Validation set	0.9665	0.9768	0.9767	0.9692	0.9690	0.9770	0.9743
	Test set	0.9640	0.9665	0.9719	0.9664	0.9768	0.9770	0.9820
Log loss	Training set	0.00285	0.00097	0.00374	0.06495	0.00314	0.00696	0.00543
	Validation set	0.05712	0.07732	0.03011	0.11893	0.08710	0.04967	0.05492
	Test set	0.05428	0.11696	0.06496	0.13357	0.06623	0.05829	0.04857

Table 11. Comparison of accuracy, recall and F1 scores for each class in the test set of seven models.

Class	Indicators	ResNet18	VGG16	DenseNet121	ShuffleNetV2	ResKAN18—Simple	ResKAN18—Standard	ResKAN18—Large
COVID-19	Accuracy	0.9888	0.9882	1.0000	1.0000	1.0000	1.0000	1.0000
	Recall	0.9944	1.0000	1.0000	0.9943	1.0000	1.0000	1.0000
	F1	0.9916	0.9940	1.0000	0.9972	1.0000	1.0000	1.0000
Normal	Accuracy	0.9847	0.9614	0.9522	0.9626	0.9794	0.9581	0.9902
	Recall	0.9554	0.9803	0.9950	0.9763	0.9744	0.9946	0.9758
	F1	0.9698	0.9707	0.9731	0.9694	0.9769	0.9760	0.9830
Viral pneumonia	Accuracy	0.9565	0.9854	0.9951	0.9741	0.9760	0.9951	0.9744
	Recall	0.9802	0.9575	0.9528	0.9641	0.9807	0.9623	0.9896
	F1	0.9682	0.9713	0.9735	0.9691	0.9783	0.9784	0.9819

Table 12. Comprehensive comparison of performance and complexity of test sets for seven models.

Models	KAN Configurations	Parameter Quantity (M)	Test Accuracy (%)	Macro F1	AUC-OvR	Training–Validation Accuracy Gap
ResNet18	-	11.18	97.59	0.9766	0.9992	2.19%
VGG16	-	134.27	97.77	0.9787	0.9989	2.24%
DenseNet121	-	6.96	98.11	0.9822	0.9988	2.15%
ShuffleNetV2	-	1.26	97.77	0.9785	0.9958	2.06%
ResKAN18—Simple	[64, 32]	11.53	98.45	0.9851	0.9987	1.84%
ResKAN18—Standard	[128, 64, 32]	13.01	98.45	0.9848	0.9995	1.84%
ResKAN18—Large	[256, 128, 64]	13.97	98.80	0.9883	0.9998	1.50%

In terms of performance stability under intra dataset partitioning, the training–validation accuracy gap of ResKAN18—Large is 1.50%, which is better than ResNet18, and the validation loss curve is smoother and less volatile, indicating that its generalization performance is more stable. From the perspective of convergence efficiency, ResNet18 is faster in the initial stage but takes about 15 epochs to reach stability, while ResKAN18—Large enters a stable state in the 10th epoch.

In terms of clinical applicability, ResKAN18—Large may provide more consistent diagnostic reliability in practical deployment due to its lower performance fluctuations and robust optimization features that enter the stable status earlier. Its adaptive mechanism may also be more adaptable to small sample data, and the potential interpretability value of the KAN module can help enhance clinical trust. In summary, ResKAN18—Large demonstrates advantages in performance stability under intra dataset partitioning, final comprehensive performance, especially suitable for medical scenarios that require high diagnostic consistency and model credibility.

3.4. Comparison of Overall Comprehensive Performance of Seven Models

Table 10 shows the comprehensive performance indicators of seven models on the training set, validation set, and test set. The following is a systematic analysis of the comprehensive performance of each model from seven dimensions.

(1) Accuracy and Macro F1 Analysis

From the perspective of testing accuracy, ResKAN18—Large ranks first with 98.80%, followed by ResKAN18—Simple and ResKAN18—Standard (both at 98.45%), then followed by DenseNet121 (98.11%), ResNet18 (97.59%), VGG16, and ShuffleNetV2 (both at 97.77%). ResKAN18—Large is 0.69% higher than the classic CNN baseline DenseNet121 and 1.21% higher than the direct baseline ResNet18.

From the macro F1 score, ResKAN18—Large ranks first with 0.9883, followed closely by ResKAN18—Simple with 0.9851, all outperforming DenseNet121 (0.9822), VGG16 (0.9787), ShuffleNetV2 (0.9785), and ResNet18 (0.9766). Macro precision and macro recall exhibit the same ranking, indicating that ResKAN18—Large achieves optimal balance across all classes.

(2) Weighted Indicator Analysis

Weighted precision, weighted recall, and weighted F1 all show rankings consistent with the macroindicators, with ResKAN18—Large leading overall with 0.9883. Due to the similar sample sizes of the three classes in this dataset, the difference between the weighted indicator and the macro indicator is minimal, which verifies the class balance of the dataset and indicates that the macro indicator fully reflects the true performance of the model in each class.

(3) AUC Analysis

ResKAN18—Large ranks first on AUC-OvR with 0.9998, and ResKAN18 variants all outperform DenseNet121 (0.9988), ResNet18 (0.9992), VGG16 (0.9989), and ShuffleNetV2 (0.9958). AUC-OvO presents a completely consistent ranking. All models have an AUC exceeding 0.995, indicating excellent discriminative ability, while ResKAN18—Large reaches the highest level with a near-perfect 0.9998.

(4) Consistency Index Analysis

The Cohen's Kappa (0.9819) and the Matthews correlation (0.9820) of ResKAN18—Large test set are both the highest among the seven models, outperforming ResKAN18—Standard (0.9768, 0.9770), ResKAN18—Simple (0.9768, 0.9768), DenseNet121 (0.9715, 0.9719), ResNet18 (0.9639, 0.9640), VGG16 (0.9663, 0.9665), and ShuffleNetV2 (0.9664, 0.9664). Due to the fact that these two indicators are not excessively affected by the number of class samples, the above results validate the robustness of ResKAN18—Large classification consistency.

(5) Probability Calibration Analysis

The log loss of ResKAN18—Large test set is 0.0486, which is the lowest among the seven models and better than ResKAN18—Standard (0.0583), ResNet18 (0.0543), DenseNet121 (0.0650), VGG16 (0.1170), and ShuffleNetV2 (0.1336). This indicates that ResKAN18—Large not only has the highest classification accuracy, but also has the most calibrated prediction probability. The confidence level of the model output highly matches the actual accuracy, which is particularly important for clinical decision support.

(6) Performance Stability Analysis under Intra Dataset Partitioning

The training–validation accuracy gaps (from 1.50% to 1.84%) of all three ResKAN18 variants are superior to all four classic CNN baselines (from 2.06% to 2.24%). The training–validation accuracy gap of ResKAN18—Large is the smallest (1.50%), while for ResKAN18—Simple it is 1.84%, both outperforming the best performing classic CNN baselines ShuffleNetV2 (2.06%) and DenseNet121 (2.15%). This result fully demonstrates the universal effectiveness of KAN structured regularization in suppressing overfitting, and is independent of the model parameter size—VGG16 has the largest parameter size (134.27M) but the largest training–validation accuracy gap (2.24%), while ResKAN18—Large has the smallest difference (1.50%) with a parameter size of 13.97M. In terms of the validation–test accuracy gap, ResKAN18—Large is 0.52%, which is at an excellent level (<2%). It should be noted that the training–validation and validation–test accuracy gaps mentioned above only reflect the stability of the performance under random partitioning of this dataset, and cannot be directly used to infer the generalization performance in external datasets or cross-domain scenarios.

(7) COVID-19-Specific Performance Analysis

Table 11 shows the accuracy, recall, and F1 score of seven models in each class.

To visually demonstrate the changes in fine-grained classification performance before and after the introduction of the KAN module, Figures 6 and 7 show the normalized confusion matrices of the baseline model ResNet18 and the optimal model ResKAN18—Large on training set, validation set, and test set respectively. Compared to the standard confusion matrix, the normalized confusion matrix normalizes the total number of rows (true classes) to 100%, allowing for fair comparison of classification performance between different classes without being affected by differences in the number of samples in each class.

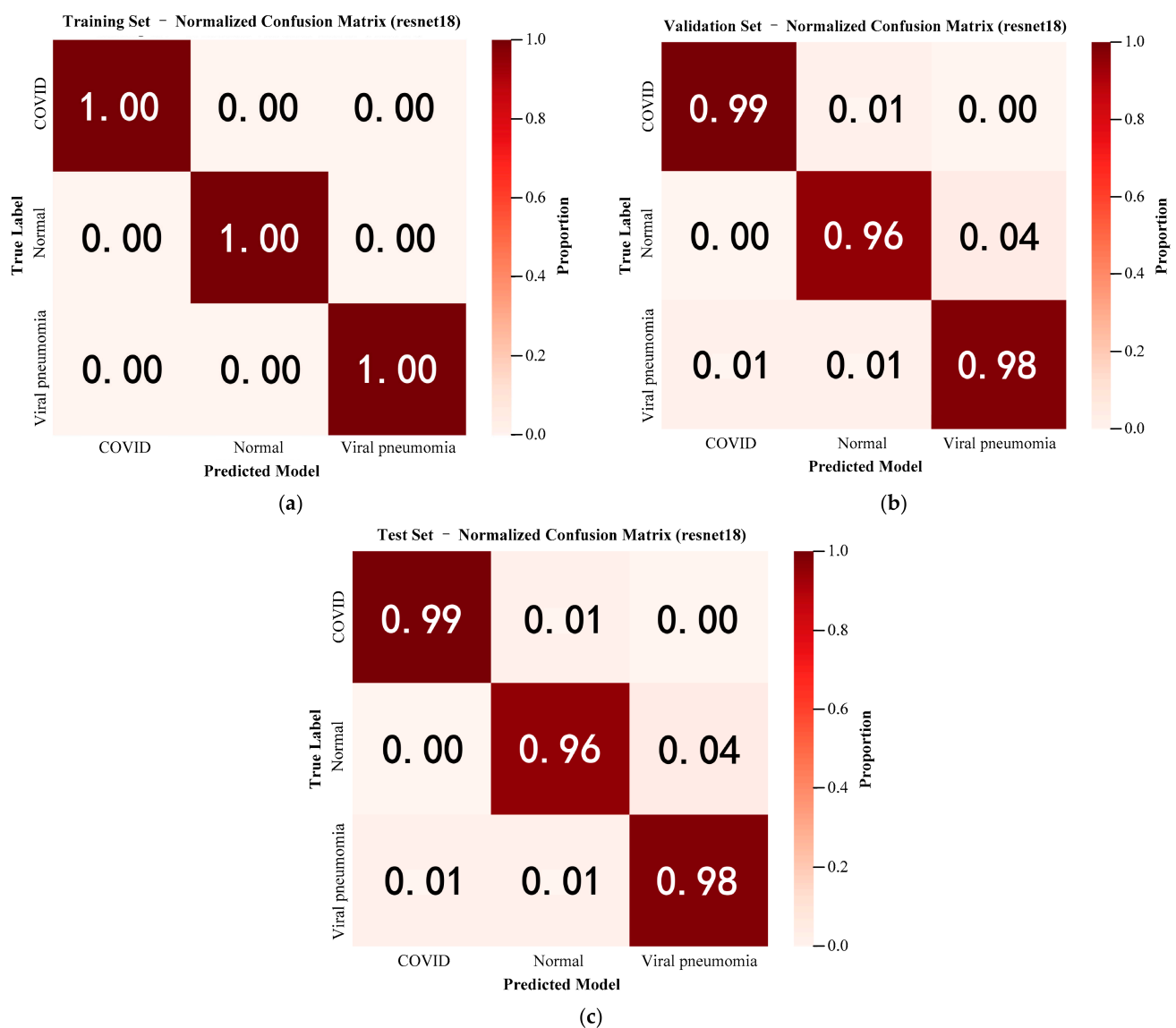


Figure 6. Normalized confusion matrices of ResNet18 in: (a) training set; (b) validation set; (c) test set.

From Figure 6, it can be seen that ResNet18 has a certain degree of class confusion on all three datasets. On the test set, 0.56% samples of the COVID-19 class are misclassified as normal (0.28%) or viral pneumonia (0.28%), with a recall rate of 99.44%; 4.46% samples of the normal class are misclassified as other classes; 1.98% samples of the viral pneumonia class are misclassified. This indicates that ResNet18 has some uncertainty in distinguishing three types of chest X-ray images, especially in determining the boundary between COVID-19 and viral pneumonia.

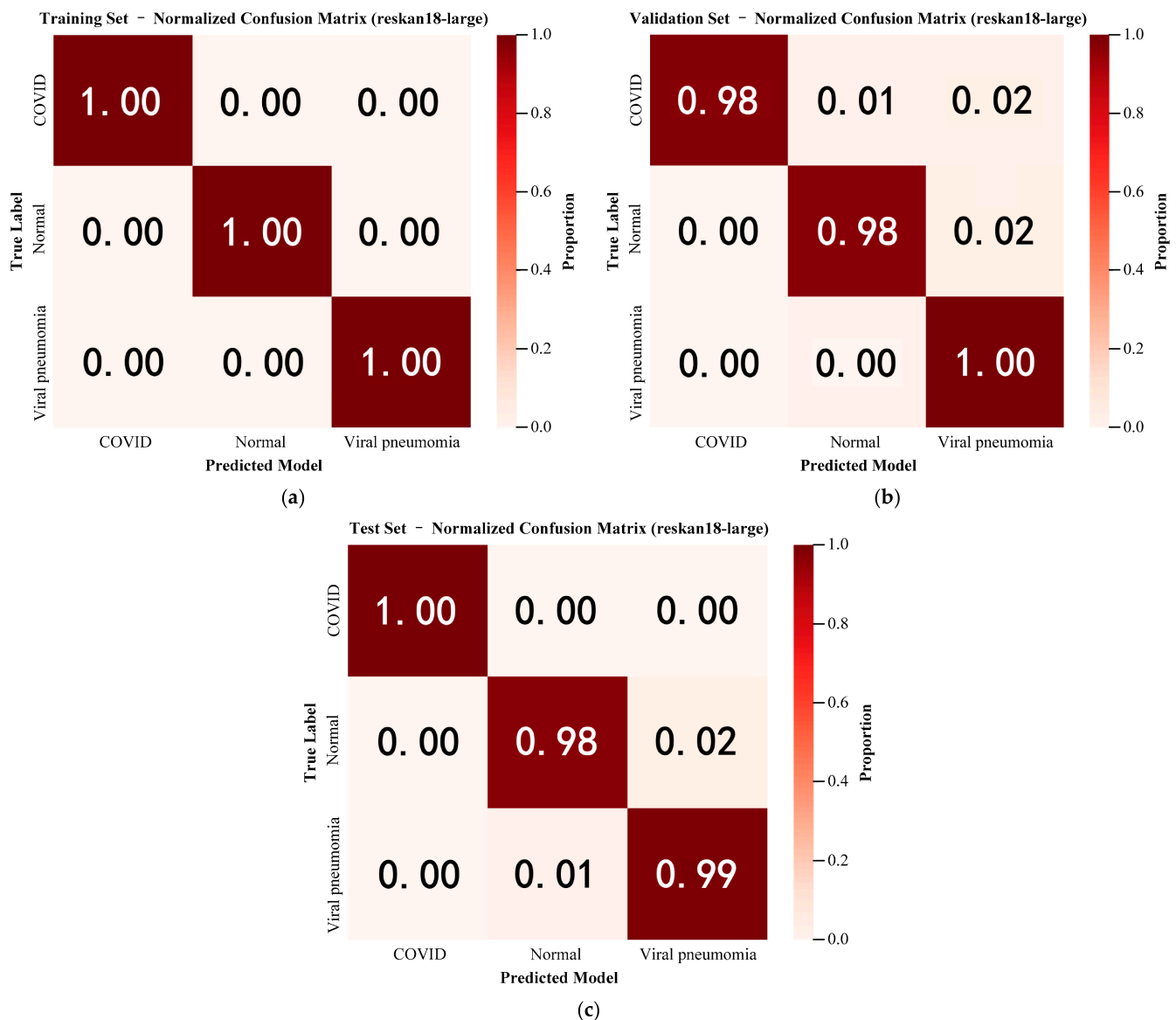


Figure 7. Normalized confusion matrices of ResKAN18—Large in: (a) training set; (b) validation set; (c) test set.

In contrast, the confusion matrix of ResKAN18—Large in Figure 7 presents clearer classification boundaries. On the test set, the classification error rate of the COVID-19 class is 0, and all 183 real COVID-19 samples are correctly identified, verifying the result of a recall rate of 1.000 in Table 11. The misjudgment rate of the normal class is only 2.42%, mainly concentrated at the boundary with viral pneumonia, and the misjudgment rate of the viral pneumonia class is 1.04%. Compared with ResNet18, ResKAN18—Large has significantly reduced misjudgment rates in all classes, with COVID-19 being particularly prominent.

From the evolution of the confusion matrix from the training set to the validation set, and then to the test set, it can be seen that the normalized confusion matrix of ResKAN18—Large maintains high consistency across all datasets, while the diagonal element values (correct classification ratio) of ResNet18 significantly decrease on both the validation set and test set. This further confirms that ResKAN18—Large has better generalization consistency than ResNet18.

Based on Tables 10 and 11, the core finding is that ResKAN18—Large ranks first in test accuracy (98.80%), macro average F1 (0.9883), and AUC (0.9998), with improvements

of 0.69%, 0.0061, and 0.0010 respectively, compared to the strongest classic CNN baseline DenseNet121. Meanwhile, ResKAN18—Large has achieved two key performance breakthroughs: (1) Optimal generalization consistency: The training–validation accuracy gap is only 1.50%, which is better than all four classic CNN baselines. (2) Zero missed diagnoses of COVID-19: All three ResKAN18 variants have achieved accuracy = recall = 1.000 on the COVID-19 class, while all four classic CNN baselines showed varying degrees of missed diagnoses.

3.5. Multi-Model Horizontal Comparative Experiment

This section further integrates the test set performance, model complexity, and training–validation accuracy gap of all seven models to form a complete horizontal comparison matrix. Table 12 summarizes the KAN configuration, parameter count, test accuracy, macro mean F1, AUC-OvR, and training–validation accuracy gap of seven models. Figures 8–11 visually illustrate the comparative relationship between seven models in four dimensions: test accuracy, macro average F1, AUC, and parameter quantity.

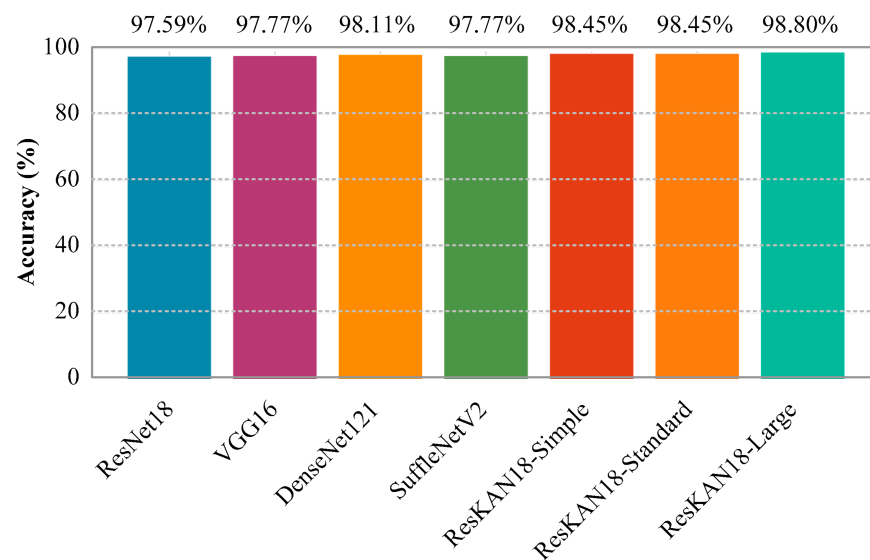


Figure 8. Test accuracy comparison across seven models.

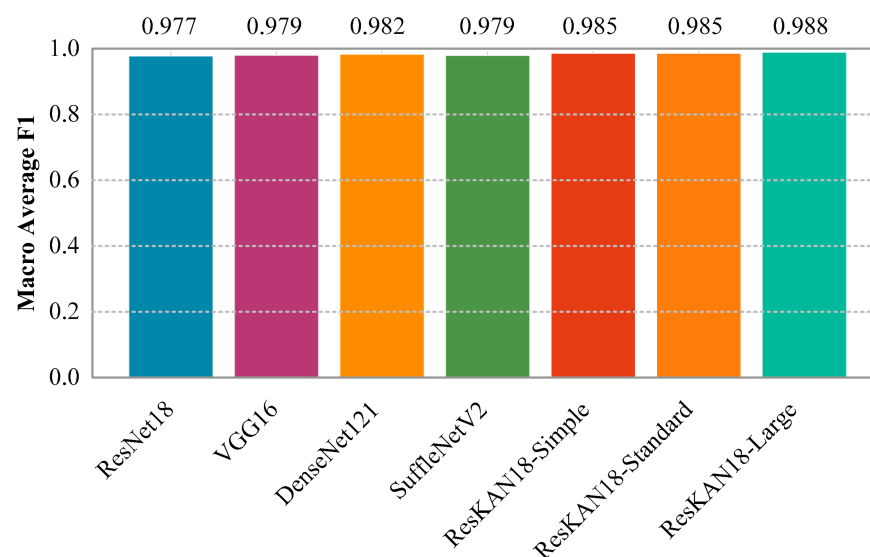


Figure 9. Macro F1score comparison across seven models.

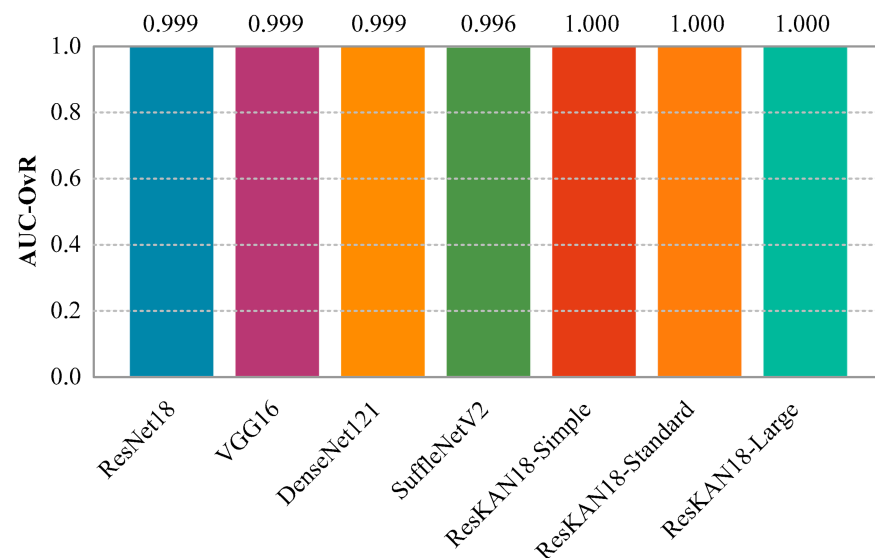


Figure 10. AUC-OvR comparison across seven models.

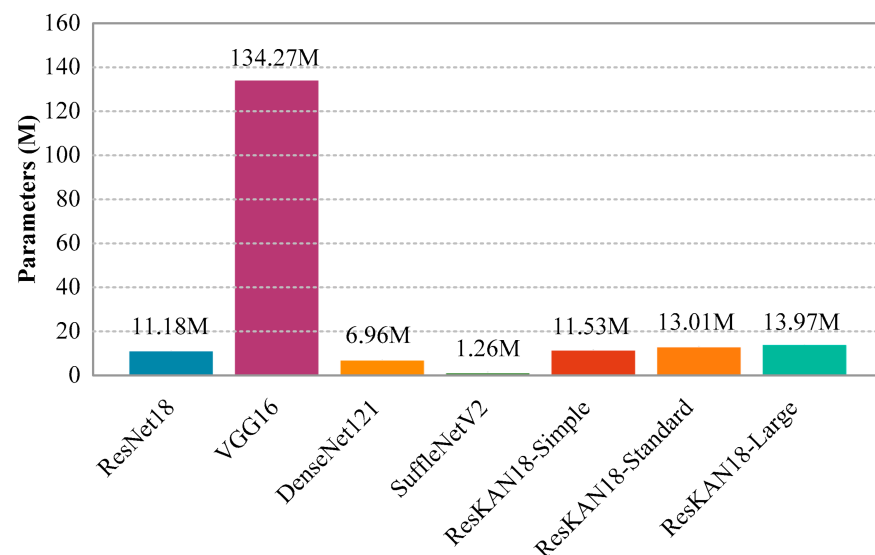


Figure 11. Model parameters comparison across seven models.

A systematic analysis of the seven model comparison data shown in Table 13 and Figures 8–11 reveals three conclusions: (1) ResKAN18—Large leads in accuracy metrics comprehensively. ResKAN18—Large ranks first in testing accuracy (98.80%), macro average F1 (0.9883), and AUC-OvR (0.9998). Under the condition that the parameter quantity is only about one tenth of VGG16, the testing accuracy exceeds 1.03%; compared to DenseNet121, it achieved an accuracy improvement of 0.69% with approximately twice the number of parameters. This demonstrates the ability of the KAN module to significantly improve classification performance while maintaining parameter efficiency. (2) ResKAN18—Simple achieves maximum revenue with minimal cost. ResKAN18—Simple achieves a testing accuracy of 98.45% with a parameter count of only 11.53M, and an ultra-fast inference delay of 2.50 ms, which is on par with the deeper ResKAN18—Standard. Compared to DenseNet121, which is the strongest among the four classic CNN baselines, ResKAN18—Simple achieves an accuracy inversion of 0.34% at the cost of increasing 4.57M parameters, and the training–validation accuracy gap is better than all baselines, demonstrating the effectiveness of structured regularization under the minimalist KAN configuration. (3) The generalization consistency of all ResKAN18 variants is better than the classic CNN baseline, and the deeper the KAN, the better the generalization. The training–

validation accuracy gap of the four classic CNN baselines is 2.06–2.24%, while the three ResKAN18 variants are 1.50–1.84%. As the depth or the width (hidden layer of dimensions) of KAN increases, the gap continues to narrow. VGG16 has the largest parameter count (134.27M) but the largest difference (2.24%), while ResKAN18—Large has a parameter count of 13.97M but the smallest difference (1.50%), proving that the regularization effect of KAN originates from the structured sparse prior of B-spline function space, rather than a reduction in parameter count.

Table 13. Comprehensive comparison of complexity, inference delay, and generalization ability of seven models.

Models	Parameter Quantity (M)	Parameter Increment Compared to ResNet18	Model Storage (MB)	FLOPs	GPU Inference Delay (ms)	CPU Inference Delay (ms)	Training–Validation Accuracy Gap
ResNet18	11.18	—	42.64	1.82 G	1.18	12	2.19%
VGG16	134.27	+1100.9%	512.21	15.47 G	40.56	320	2.24%
DenseNet121	6.96	−37.7%	26.54	2.90 G	7.46	60	2.15%
ShuffleNetV2	1.26	−88.7%	4.79	0.15 G	2.88	18	2.06%
ResKAN18—Simple	11.53	+3.1%	43.97	1.824 G	2.50	25	1.84%
ResKAN18—Standard	13.01	+16.4%	49.62	1.824 G	7.63	75	1.84%
ResKAN18—Large	13.97	+24.9%	53.29	1.824 G	8.25	95	1.50%

3.6. Analysis of Model Complexity and Reasoning Delay

This section further conducts a systematic complexity analysis of all seven models from three dimensions: parameter quantity, FLOPs, and inference delay, to evaluate the computational cost of each model in actual deployment. Table 13 summarizes the parameter count, relative parameter increment to ResNet18, model storage size, floating-point operations per second (FLOPs), graphics processing unit (GPU) inference delay, central processing unit (CPU) inference delay, and training–validation accuracy gap of seven models.

3.6.1. Parameter Quantity Analysis

The parameter distribution of the seven models shows significant differences. VGG16 has the largest parameter count (134.27M), which is about 12 times that of ResNet18, consistent with its deep serial structure design—a large number of fully connected layer parameters results in a bloated model. ShuffleNetV2 has the smallest parameter count (1.26M), only 11.3% of ResNet18, thanks to its lightweight design of channel mixing and depth-separable convolution. DenseNet121 has a parameter size of 6.96M and achieves parameter efficiency through feature reuse.

The parameter sizes of the three ResKAN18 variants are: ResKAN18—Simple 11.53M, ResKAN18—Standard 13.01M, and ResKAN18—Large 13.97M. All these three are on the same level as ResNet18. Among them, ResKAN18—Simple is almost on par with ResNet18, with only a 3.1% increase; ResKAN18—Large increases by 24.9%, but its absolute value is still much smaller than VGG16 (only 10.4% of VGG16). The parameter increment mainly comes from the spline coefficient matrix of the KAN module. The total number of spline coefficients in ResKAN18—Large’s four-layer KAN is 1,377,792, in ResKAN18—Standard’s three-layer KAN it is 606,976, and in ResKAN18—Simple’s two-layer KAN it is 279,296.

3.6.2. FLOPs Analysis

VGG16 has the highest FLOPs (15.47G), which is approximately 8.5 times that of ResNet18; the FLOPs of ShuffleNetV2 is the lowest (0.15G), about 8.2% of ResNet18; the FLOPs DenseNet121 is 2.90G, which is approximately 1.6 times that of ResNet18. The FLOPs of the three ResKAN18 variants are all 1.824 G, which is almost the same as the 1.82G of

ResNet18, indicating that replacing the fully connected layer with the KAN module does not introduce significant computational overhead.

To further support the above-measured results theoretically, the time complexity of the KAN module can be expressed as:

$$T_{\text{KAN}} = O\left(\sum_{l=0}^L d_l \cdot d_{l+1} \cdot G \cdot k\right) \quad (8)$$

where $L = 4$ for ResKAN18—Large, $L = 3$ for ResKAN18—Standard, and $L = 2$ for ResKAN18—Simple, d_l is the number of neurons in the l -th layer, $G = 5$ is the number of grid points, and $k = 3$ is the spline order. Equation (8) is on the same level of time complexity $O(\sum d_l \cdot d_{l+1})$ as the standard fully connected layer, with the only difference being the constant factor $G \cdot k = 15$, which is consistent with the measured results of FLOPs.

3.6.3. Inference Delay Analysis

The inference delay test has been conducted on NVIDIA GeForce RTX 4080 SUPER (16 GB GDDR6X) with Intel Core i9-13900KF CPU. Take the average of 100 runs for each test, with the first 20 runs being warm-up. On GPU inference delay, ResNet18 is the fastest (1.18 ms), ResKAN18—Simple is the second fastest (2.50 ms), followed by ShuffleNetV2 (2.88 ms), DenseNet121 (7.46 ms), ResKAN18—Standard (7.63 ms), ResKAN18—Large (8.25 ms), and the slowest VGG16 (40.56 ms). A similar order is presented on CPU inference delay: ResNet18 (12 ms), ShuffleNetV2 (18 ms), ResKAN18—Simple (25 ms), DenseNet121 (60 ms), ResKAN18—Standard (75 ms), ResKAN18—Large (95 ms), VGG16 (320 ms).

As for clinical reference, radiologists typically take 10–30 s to manually review a chest X-ray image [49,50]. Even on CPUs in primary healthcare institutions without dedicated graphics cards, the inference speed of ResKAN18—Simple (about 25 ms) is over 400 times faster than manual reading; ResKAN18—Large (about 95 ms) is also over 100 times faster. All seven models (except for VGG16 which is slightly slower on CPU) can meet the requirements for real-time auxiliary diagnosis.

3.6.4. Comprehensive Trade-Off Between Complexity, Accuracy, and Performance Stability Under Intra Dataset Partitioning

Based on the four dimensions of parameter quantity, inference delay, accuracy, and generalization, the seven models form a clear hierarchy:

ResNet18 has a moderate parameter count (11.18M), fastest speed (1.18 ms), but lowest testing accuracy (97.59%), and the largest training–validation accuracy gap (2.19%). It is suitable for scenarios that require extreme inference speed and stable data distribution.

VGG16 has the largest number of parameters (134.27M), the slowest speed (GPU inference delay is 40.56 ms, CPU inference delay is 320 ms), a testing accuracy of only 97.77%, the largest training–validation accuracy gap (2.24%), and the lowest overall cost-effectiveness.

DenseNet121 has a small number of parameters (6.96M), moderate testing accuracy (98.11%), and a moderate training–validation accuracy gap (2.15%); it has the best overall performance among classic CNN baselines.

ShuffleNetV2 has the smallest number of parameters (1.26M) and fast speed (GPU inference delay is 2.88 ms), but the testing accuracy is only 97.77%, and the training–validation accuracy gap is moderate (2.06%). It is suitable for ultra-lightweight mobile deployment.

ResKAN18—Simple's parameter count is almost the same as ResNet18 (an increase of 3.1%), it has the second fastest speed (GPU inference delay is 2.50 ms), the testing accuracy is 98.45% (0.86% higher than ResNet18), and the training–validation accuracy

gap is 1.84% (better than all four classic CNN baselines). It has the highest comprehensive cost performance ratio and is suitable for resource-constrained scenarios such as primary medical care and edge computing.

ResKAN18—Standard has a moderate parameter count (13.01M), moderate speed (GPU inference delay is 7.63 ms), its testing accuracy is comparable to ResKAN18—Simple (98.45%), and its training–validation accuracy gap is 1.84%. It is suitable for scenarios with certain precision requirements and moderate computing resources.

ResKAN18—Large has a parameter count of 13.97M and a GPU inference delay at 8.25 ms, it has the highest testing accuracy (98.80%) and the smallest training–validation accuracy gap (1.50%), making it suitable for extreme-precision scenarios.

The three ResKAN18 variants occupy three different working points on the Pareto front of complexity, accuracy, and performance stability under intra dataset partitioning, covering the full spectrum of requirements from “edge lightweight” to “high-precision diagnosis”, and users can flexibly choose according to hardware conditions. Taking into account the accuracy, performance stability under intra dataset partitioning, and computational efficiency, this paper recommends ResKAN18—Large as the default configuration; for scenarios where resources are severely limited, ResKAN18—Simple can provide a cost-effective alternative solution.

3.7. In-Depth Discussion

Based on the systematic comparative experiments of the seven models from Sections 3.3–3.6 this section presents a rational analysis of three core issues:

(1) The mathematical mechanism of the KAN breaking through the accuracy bottleneck of classic CNNs: The classification heads of the four classic CNN baselines all adopt a linear projection design with global average pooling and a single fully connected layer, and their expressive power is limited by the rank of the linear transformation. Although DenseNet121 enhances feature reuse through dense connections, its final classification head is still linear projection, so its testing accuracy (98.11%) is better than ResNet18 (97.59%), but still lower than ResKAN18—Simple (98.45%). ResKAN18 replaces single-layer fully connected layers with multilayer KAN modules, and each layer performs a nonlinear transformation on the input dimension by dimension using B-spline functions as shown in Equation (7). B-splines have the ability to approximate general functions [37], enabling the KAN module to fit any continuous mapping from 512-dimensional features to three types of probabilities. In the experiment, the testing accuracy of the three ResKAN18 variants (from 98.45% to 98.80%) is superior to all four classic CNN baselines (from 97.59% to 98.11%), empirically verifying the expression ability advantage of KAN compared to linear fully connected layers.

(2) The mathematical mechanism of the KAN breaking through the accuracy bottleneck of classic CNNs: Table 11 shows that three of the four classic CNN baselines have varying degrees of missed diagnosis in COVID-19 (ResNet18, ShuffleNetV2 and VGG16), but for all three ResKAN18 variants, they have all achieved an accuracy = recall rate = 1.000. This systematic difference stems from the dynamic grid update mechanism of KAN, which means that every five training cycles, the KAN module recalculates the B-spline knot sequence based on the current batch feature distribution, so that the basis function support interval closely follows the data manifold. For samples at the edge of the feature space in the COVID-19 class, B-spline basis functions can dynamically adjust the resolution to accurately characterize their position. The fully connected layer of classical CNN does not have this adaptive ability, and its linear projection weights are fixed after training has been completed. It is worth noting that even the shallowest ResKAN18—Simple can achieve a

recall rate of 1.0000, indicating that the KAN structure itself plays a core role in capturing key differences.

(3) The intrinsic regularization mechanism of overfitting suppression: Tables 12 and 13 show a counterintuitive but crucial phenomenon that the training–validation accuracy gap of all three ResKAN18 variants is better than all four classical CNN baselines, and the deeper and the wider (larger hidden dimensions) the KAN, the smaller the gap. VGG16 has the largest parameter size (134.27M) and the largest gap (2.24%), while ResKAN18—Large has the smallest gap (1.50%) with a parameter size of 13.97M. This proves that the regularization effect of KAN is not due to a reduction in the number of parameters, but rather to the structured sparse prior of the B-spline function space: each univariate function is determined by only eight coefficients, and local support naturally limits the complexity of the function. As the depth of the KAN increases, the regularization effect accumulates on more layers. Cohen’s Kappa and Matthews correlation indicators further confirm that ResKAN18—Large is significantly better than ResNet18. In terms of probability calibration, The log loss indicator also shows that ResKAN18—Large is the best among the seven models, which is 10.5% lower than ResNet18.

In summary, the three ResKAN18 variants occupy three different working points on the Pareto front: ResKAN18—Simple achieves the maximum benefit with minimal cost, while the ResKAN18—Large achieves maximum accuracy and optimal generalization with moderate cost.

4. Limitations and Future Work

4.1. Research Limitations

Although this study systematically compared and validated the superiority of ResKAN18 through seven models, there are still three limitations:

(1) The data source is single. This paper has only used one publicly available dataset with a total of 3880 chest X-ray images, and although the seven models are compared on the same dataset to ensure fairness, there is a lack of external validation across multiple centers and devices. The generalization ability of the model on different devices and patient populations has not been validated. Although the three KAN variants outperform the four comparison models in terms of the training–validation accuracy gap, this result is only obtained based on random partitioning of the same data source, and their generalization performance across domains and external datasets still needs further validation on multi-center external datasets.

(2) KAN hyperparameters rely on experience. The configuration of three KAN variants: grid size $G = 5$, spline order $k = 3$, update frequency is set as every five rounds, hidden-layer dimension is determined through preliminary search, lacking an adaptive selection mechanism. The optimal hyperparameter configuration may vary for datasets of different scales or domains.

(3) A lack of prospective clinical validation. All experiments are based on retrospective public data, and the performance of the model in real clinical environments (including complex situations such as low-quality images, improper positioning, and the presence of other lesions) has not been tested by prospective clinical trials.

4.2. Future Work Directions

In response to the above limitations, future research will be conducted from three directions:

(1) Data expansion and external validation. We hope to collaborate with multiple medical institutions to collect chest X-ray image data from multiple centers and devices, so as to construct a large-scale external validation set, to systematically evaluate the cross-

domain generalization ability of three versions, and to verify the actual performance of the Simple version on primary medical equipment.

(2) KAN hyperparameter adaptive strategy. We hope to propose hyperparameter automatic search algorithms based on Bayesian optimization or reinforcement learning (RL), which can enable parameters such as grid size, spline order, update frequency, and hidden-layer dimension to be dynamically adjusted according to data features.

(3) Embedded system integration and clinical translation. Through knowledge distillation, quantitative training, and structured pruning, we hope to further compress the model volume. We hope to integrate ResKAN18—Simple into portable X-ray devices, and the ResKAN18—Large into picture archiving and a communication system (PACS) for prospective clinical validation trials.

In addition, the feasibility of expanding the application of ResKAN18 for the diagnosis of other chest diseases such as tuberculosis and lung tumors can also be explored in the future.

5. Conclusions

This paper has constructed and validated three variants of the ResKAN18 hybrid structure and conducted systematic comparative experiments using seven models (four classic CNN baselines: ResNet18, VGG16, DenseNet121, ShuffleNetV2, and three ResKAN18 variants: ResKAN18—Simple, ResKAN18—Standard, and ResKAN18—Large). On the 3880 chest X-ray dataset, ResKAN18—Large has achieved a testing accuracy of 98.80%, with a macro average F1 of 0.9883 and an AUC-OvR of 0.9998, ranking first among all seven models. ResKAN18—Simple has achieved a testing accuracy of 98.45% and a training–validation accuracy gap of 1.84% with a parameter count of 11.53M (an increase of only 3.1% compared to ResNet18). All variants have achieved zero missed diagnoses for COVID-19 (recall rate = 1.0000), while ResNet18 has experienced missed diagnoses (recall rate = 0.9944).

In terms of implementation, ResNet18 which adopts pre-trained ImageNet is chosen in ResKAN18 variants as the feature extraction backbone, and KAN modules of different depths are used to replace the fully connected layer. Each layer adopts B-spline function for dimensional nonlinear transformation, and the spline mesh is dynamically updated every five rounds. The training configuration includes 30 rounds, a learning rate of 0.0001, a batch size of 256, an Adam optimizer, and L1 regularization.

Compared with four classic CNN baselines, ResKAN18—Large has shown a 0.69% improvement in testing accuracy compared to DenseNet121 and a 1.21% improvement compared to ResNet18; ResKAN18—Simple has improved by 0.86% compared to ResNet18 with almost the same number of parameters. The training–validation accuracy gap among the three KAN variants (from 1.50% to 1.84%) is better than all four CNN baselines (from 2.06% to 2.24%), and the deeper and the wider the KAN, the better the generalization. All KAN variants achieved zero missed diagnoses in COVID-19 recognition, while the classic CNN baseline showed varying degrees of missed diagnoses. Three variants cover the full spectrum requirements, from lightweight deployment to high-precision diagnosis, with ResKAN18—Simple achieving maximum benefits with minimal parameter increments, suitable for resource-constrained scenarios, and ResKAN18—Large achieving the highest accuracy and optimal generalization. KAN spline functions can be independently visualized, providing endogenous function-level explanations for the decision-making process. Taking into account the trade-off between accuracy, generalization, and inference efficiency, this paper recommends ResKAN18—Large as the default configuration and ResKAN18—Simple as a cost-effective alternative for resource-constrained scenarios.

Author Contributions: Conceptualization, D.L. and W.N.; methodology, Z.Y. and W.N.; software, D.L. and Z.Y.; validation, D.L., Y.L. and W.N.; formal analysis, D.L., Z.Y., Y.L. and W.N.; investigation, Y.L.; data curation, D.L. and Z.Y.; writing—original draft preparation, Z.Y. and Y.L.; writing—review and editing, D.L. and W.N.; funding acquisition, W.N. All authors have read and agreed to the published version of the manuscript.

Funding: This research was funded by the General Scientific Research Project, Department of Education of Zhejiang Province under Grant No. Y202559659.

Data Availability Statement: The data presented in this paper are available in Baidu Netdisk at <https://pan.baidu.com/s/1W86xliFyYnVvkRsM-kEfqg?pwd=ceu4> (accessed on 12 August 2026), extraction code: ceu4. These data were derived from the shared dataset disclosed in the literature [48] mentioned in this paper at <https://www.kaggle.com/tawsifurrahman/covid19-radiography-database> (accessed on 12 August 2026).

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

Abbreviations

The following abbreviations are used in this manuscript:

AUC	Area under curve
CNN	Convolutional neural network
COVID-19	Coronavirus disease 2019
CPU	Central processing unit
DL	Deep learning
FLOPs	Floating-point operations per second
GAN	Generative adversarial network
GPU	Graphics processing unit
Grad-CAM	Gradient-weighted class activation mapping
KAN	Kolmogorov–Arnold network
LIME	Local interpretable model-agnostic explanation
MLP	Multilayer perceptron
NN	Neural network
OvR	One-vs-rest
PACS	Picture archiving and communication system
ReLU	Rectified linear unit
RL	Reinforcement learning
SiLU	Sigmoid linear unit
VGG	Visual geometry group

References

1. Chauhan, R.; Pokhariya, H.; Negi, P.; Bhatt, C.; Kumar, R.; Dhiman, G.; Suri, A.; Kumar, D.; Chowdhury, S. A comparative study of X-ray and CT images to using imaging and deep learning to detect the COVID19 virus. *AIP Conf. Proc.* **2025**, *3185*, 030001. [\[CrossRef\]](#)
2. Fardela, R.; Milvita, D.; Almuhyar, M.; Mardiansyah, D.; Rasyada, L.A.; Hakim, L.M. Classification of thoracic X-ray images of COVID-19 patients using the convolutional neural network. *J. Comput. Sci.* **2024**, *20*, 357–364. [\[CrossRef\]](#)
3. Das, A.; Agarwal, R.; Singh, R.; Chowdhury, A.; Nandi, D. Automatic detection of COVID-19 from chest X-ray images using deep learning model. *arXiv* **2024**, arXiv:2408.14927. [\[CrossRef\]](#)
4. Hilali-Jaghdam, I.; Elhag, A.A.; Ishak, A.B.; Elnaïm, B.M.E.; Elhag, O.E.M.; Abuhaimed, F.M.; Abdel-Khalek, S. Toward COVID-19 detection and classification using optimal efficient densenet model on chest X-ray images. *Alex. Eng. J.* **2024**, *101*, 136–146. [\[CrossRef\]](#)
5. Surendra; Manoj Kumar, M.V.; Shiva Darshan, S.L.; Prashanth, B.S. CXNet—A novel approach for COVID-19 detection and classification using chest X-ray image. *Procedia Comput. Sci.* **2024**, *235*, 2496–2497. [\[CrossRef\]](#)
6. Ahmed, S.; Subhan, F.; Su’ud, M.M.; Alam, M.M.; Waheed, A. A survey of lung nodules detection and classification from CT scan images. *Comput. Syst. Sci. Eng.* **2024**, *48*, 1483–1511. [\[CrossRef\]](#)

7. Dack, E.; Brigato, L.; McMurray, M.; Fontanellaz, M.; Frauenfelder, T.; Hoppe, H.; Exadaktylos, A.; Geiser, T.; Funke-Chambour, M.; Christel, A.; et al. An empirical analysis for zero-shot multi-label classification on COVID-19 CT scans and uncured reports. In Proceedings of the IEEE/CVF International Conference on Computer Vision Workshops (ICCVW), Paris, France, 2–6 October 2023. [\[CrossRef\]](#)
8. Mobarakeh, S.A.M.; Kazemi, K.; Aarabi, A.; Danyal, H. Empowering medical imaging with artificial intelligence: A review of machine learning approaches for the detection, and segmentation of COVID-19 using radiographic and tomographic images. *arXiv* **2024**, arXiv:2401.07020. [\[CrossRef\]](#)
9. Ali, A. Deep learning models for classification of COVID-19 cases by medical images. *arXiv* **2023**, arXiv:2310.16851. [\[CrossRef\]](#)
10. Yang, K.; Bennett, A.; Duncan, D. Robust and interpretable COVID-19 diagnosis on chest X-ray images using adversarial training. *arXiv* **2023**, arXiv:2311.14227. [\[CrossRef\]](#)
11. Lanka, N.S.D.; Arunkumar, C. Multi-class classification of COVID-19 in chest X-ray using deep convolutional neural networks. In *Advances in Data and Information Sciences*; Tiwari, S., Trivedi, M.C., Kolhe, M.L., Singh, B.K., Eds.; Lecture Notes in Networks and Systems; Springer: Singapore, 2023; Volume 522, pp. 409–419. [\[CrossRef\]](#)
12. Bag, V.V.; Gaikwad, V.D.; Patil, M.B.; Swami, K.S.; Abhang, S.P.; Antad, S.M.; Joshi-Bag, S. DRESCNN: Deep ResNet convolutional neural network based classification of X-ray images for detection of COVID-19. *J. Electr. Syst.* **2024**, *20*, 1415–1425. [\[CrossRef\]](#)
13. Dharmik, A. COVID-19 pneumonia diagnosis using medical images: Deep learning-based transfer learning approach. *JMIRx Med.* **2025**, *6*, e75015. [\[CrossRef\]](#)
14. Krizhevsky, A.; Sutskever, I.; Hinton, G.E. ImageNet classification with deep convolutional neural networks. *Commun. ACM* **2017**, *60*, 84–90. [\[CrossRef\]](#)
15. 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/CVF Conference on Computer Vision and Pattern Recognition (CVPR), Boston, MA, USA, 7–12 June 2015. [\[CrossRef\]](#)
16. He, K.; Zhang, X.; Ren, S.; Sun, J. Deep residual learning for image recognition. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), Las Vegas, NV, USA, 27–30 June 2016. [\[CrossRef\]](#)
17. Huang, G.; Liu, Z.; Van Der Maaten, L.; Weinberger, K.Q. Densely connected convolutional networks. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), Honolulu, HI, USA, 21–26 July 2017. [\[CrossRef\]](#)
18. Simonyan, K.; Zisserman, A. Very deep convolutional networks for large-scale image recognition. *arXiv* **2014**, arXiv:1409.1556. [\[CrossRef\]](#)
19. Zhang, X.; Zhou, X.; Lin, M.; Sun, J. ShuffleNet: An extremely efficient convolutional neural network for mobile devices. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), Salt Lake City, UT, USA, 18–24 June 2018. [\[CrossRef\]](#)
20. Roy, S.; Suresh, A.; Sahu, P.; Gupta, T.R. Novel pooling-based VGG-Lite for pneumonia and COVID-19 detection from imbalanced chest X-ray datasets. *arXiv* **2025**, arXiv:2504.07468. [\[CrossRef\]](#)
21. Angara, S.; Thirunagaru, S. Study of vision transformers for COVID-19 detection from chest X-rays. *arXiv* **2023**, arXiv:2307.09402. [\[CrossRef\]](#)
22. Barbano, C.A.; Renzulli, R.; Grosso, M.; Basile, D.; Busso, M.; Grangetto, M. AI-assisted diagnosis for COVID-19 CXR screening: From data collection to clinical validation. In Proceedings of the 2024 IEEE 21st International Symposium on Biomedical Imaging (ISBI), Athens, Greece, 27–30 May 2024. [\[CrossRef\]](#)
23. Khater, O.H.; Shuaib, A.S.; Haq, S.U.; Siddiqui, A.J. AttCDCNet: Attention-enhanced chest disease classification using X-ray images. In Proceedings of the IEEE 22nd International Multi-Conference on Systems and Devices (SSD), Istanbul, Turkey, 21–24 April 2025. [\[CrossRef\]](#)
24. Singh, G. Comparative analysis of vision transformers and traditional deep learning approaches for automated pneumonia detection in chest X-rays. *arXiv* **2025**, arXiv:2507.10589. [\[CrossRef\]](#)
25. Lei, Y.; Nguyen, M.; Liu, T.C.; Oh, H. Deep learning-powered classification of thoracic diseases in chest X-rays. *arXiv* **2025**, arXiv:2501.14279. [\[CrossRef\]](#)
26. Haris, M.A.H.; Saad, A.F. COVID classification system for COVID detection. *J. Eng. Technol. Appl. Sci.* **2024**, *6*, 9–18. [\[CrossRef\]](#)
27. Vamsidhar S, S.R.; Satya, B.; Gorthi, R.K. Effective-LDAM: An effective loss function to mitigate data imbalance for robust chest X-ray disease classification. *arXiv* **2024**, arXiv:2407.04953. [\[CrossRef\]](#)
28. Bahram, A.M.; Omer, S.M.; Mohammed, H.M.; Aula, S.A. Enhanced chest disease classification using an improved CheXNet framework with EfficientNetV2-M and optimization-driven learning. *arXiv* **2025**, arXiv:2512.08992. [\[CrossRef\]](#)
29. Jahromi, S.; Hajati, F.; Rezaee, A.; Nourian, J. Medical image classification on imbalanced data using ProGAN and SMA-optimized ResNet: Application to COVID-19. *arXiv* **2025**, arXiv:2512.24214. [\[CrossRef\]](#)
30. Zoph, B.; Vasudevan, V.; Shlens, J.; Le, Q.V. Learning transferable architectures for scalable image recognition. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), Salt Lake City, UT, USA, 18–24 June 2018. [\[CrossRef\]](#)

31. Ukwandu, O.; Hindy, H.; Ukwandu, E. An evaluation of lightweight deep learning techniques in medical imaging for high precision COVID-19 diagnostics. *arXiv* **2023**, arXiv:2305.19016. [[CrossRef](#)]
32. Bassi, P.R.A.S.; Decherchi, S.; Cavalli, A. Faster ISNet for background bias mitigation on deep neural networks. *IEEE Access* **2024**, *12*, 108919–108932. [[CrossRef](#)]
33. Haque, M.S.; Taluckder, M.S.; Shawkat, S.B.; Shahriyar, M.A.; Sayed, M.A.; Modak, C. Prediction of pneumonia and COVID-19 using deep neural networks. *arXiv* **2023**, arXiv:2308.10368. [[CrossRef](#)]
34. Falero, D.; Kabir, M.A.; Homaira, N. From lab to packet: A novel continual learning-based mobile application for screening COVID-19. *arXiv* **2024**, arXiv:2410.12589. [[CrossRef](#)]
35. Beghoura, I.; Benssalah, M.; Sbargoud, F. An improved CovidCovLSTM model for pneumonia-COVID-19 detection and classification. *Biomed. Eng. Appl. Basis Commun.* **2025**, *37*, 2550019. [[CrossRef](#)]
36. Shakouri, M.; Iranmanesh, F.; Eftekhari, M. DINO-CXR: A self supervised method based on vision transformer for chest X-ray classification. *arXiv* **2023**, arXiv:2308.00475. [[CrossRef](#)]
37. Liu, Z.; Wang, Y.; Vaidya, S.; Ruehle, F.; Halverson, J.; Soljačić, M.; Hou, T.Y.; Tegmark, M. KAN: Kolmogorov-Arnold networks. *arXiv* **2024**, arXiv:2404.19756. [[CrossRef](#)]
38. Kolmogorov, A.N. On the representation of continuous functions of several variables by superpositions of continuous functions of a smaller number of variables. In *Twelve Papers on Algebra and Real Functions. American Mathematical Society Translations, Series 2*; Černikov, S., Černikova, Kolmogorov, A., Mal'cev, A., Plotkin, B., Rimskiĭ-Korsakov, B., Tihomirov, V., Trohimčuk, Y., Vilenkin, N., Vituškin, A., Eds.; American Mathematical Society: Providence, RI, USA, 1961; Volume 17, pp. 369–373. [[CrossRef](#)]
39. Samadi, M.E.; Müller, Y.; Schuppert, A. Smooth Kolmogorov Arnold networks enabling structural knowledge representation. *arXiv* **2024**, arXiv:2405.11318. [[CrossRef](#)]
40. Schmidt-Heiber, J. The Kolmogorov-Arnold representation theorem revisited. *Neural Netw.* **2021**, *137*, 119–126. [[CrossRef](#)]
41. Poeta, E.; Giobergia, F.; Pastor, E.; Cerquitelli, T.; Baralis, E. A benchmarking study of Kolmogorov-Arnold networks on tabular data. *arXiv* **2024**, arXiv:2406.14529. [[CrossRef](#)]
42. Hou, Y.; Ji, T.; Zhang, D.; Stefanidis, A. Kolmogorov-Arnold networks: A critical assessment of claims, performance, and practical viability. *arXiv* **2024**, arXiv:2407.11075. [[CrossRef](#)]
43. Wang, Y.; Sun, J.; Bai, J.; Anitescu, C.; Eshaghi, M.S.; Zhuang, X.; Rabczuk, T.; Liu, Y. Kolmogorov Arnold informed neural network: A physics-informed deep learning framework for solving forward and inverse problems based on Kolmogorov Arnold networks. *arXiv* **2024**, arXiv:2406.11045. [[CrossRef](#)]
44. Shukla, K.; Toscano, J.D.; Wang, Z.; Zou, Z.; Karniadakis, G.E. A comprehensive and FAIR comparison between MLP and KAN representations for differential equations and operator networks. *arXiv* **2024**, arXiv:2406.02917. [[CrossRef](#)]
45. Li, C.; Liu, X.; Li, W.; Wang, C.; Liu, H.; Liu, Y.; Chen, Z.; Yuan, Y. U-KAN makes strong backbone for medical image segmentation and generation. *arXiv* **2024**, arXiv:2406.02918. [[CrossRef](#)]
46. Ferdaus, M.M.; Abdelguerfi, M.; Ioup, E.; Dobson, D.; Niles, K.N.; Pathak, K.; Sloan, S. KANICE: Kolmogorov-Arnold networks with interactive convolutional elements. *arXiv* **2024**, arXiv:2410.17172. [[CrossRef](#)]
47. Bodner, A.D.; Tepsich, A.S.; Spolski, J.N.; Pourteau, S. Convolutional Kolmogorov-Arnold networks. *arXiv* **2024**, arXiv:2406.13155. [[CrossRef](#)]
48. Chowdhury, M.E.H.; Rahman, T.; Khandakar, A.; Mazhar, R.; Kadir, M.A.; Mahbub, Z.B.; Islam, K.R.; Khan, M.S.; Iqbal, A.; Al Emadi, N.; et al. Can AI help in screening viral and COVID-19 pneumonia? *IEEE Access* **2020**, *8*, 132655–132676. [[CrossRef](#)]
49. Shin, H.J.; Han, K.; Ryu, L.; Kim, E.-K. The impact of artificial intelligence on the reading times of radiologists for chest radiographs. *npj Digit. Med.* **2023**, *6*, 82. [[CrossRef](#)]
50. Hong, E.K.; Roh, B.; Park, B.; Jo, J.-B.; Bae, W.; Park, J.S.; Sung, D.-W. Value of using a generative AI model in chest radiography reporting: A reader study. *Radiology* **2025**, *314*, e241646. [[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.