

Article

A Multi-Scale Unsupervised Feature Extraction Network with Structured Layer-Wise Decomposition

Yusuf Şevki Günaydın ¹ and Baha Şen ^{2,*}¹ Department of Computer Engineering, Ankara Yıldırım Beyazıt University, Ankara 06010, Türkiye; yusufsevkgunaydin@aybu.edu.tr² Department of Computer Engineering, National Defense University, Ankara 06654, Türkiye

* Correspondence: baha.sen@msu.edu.tr

Abstract

Recent developments in deep learning have underscored prizing effective feature extraction in scenarios with limited or unlabeled data. This study introduces a novel unsupervised multi-scale feature extraction framework based on a multi-branch auto-encoder architecture. The proposed method decomposes input images into smooth, detailed and residual components, using variational loss functions to ensure that each branch captures distinct and non-overlapping representations. This decomposition enhances the information richness of input data while preserving its structural integrity, making it especially beneficial for grayscale or low-resolution images. Experimental results on classification and image segmentation tasks show that the proposed method enhances model performance by enriching input representations. Its architecture is scalable and adaptable, making it applicable to a wide range of machine learning tasks beyond image classification and segmentation. These findings highlight the proposed method's utility as a robust, general-purpose solution for unsupervised feature extraction and multi-scale representation learning.

Keywords: feature extraction; multi-scale; deep learning; image decomposition



Academic Editor: Douglas O'Shaughnessy

Received: 25 May 2025

Revised: 19 June 2025

Accepted: 24 June 2025

Published: 26 June 2025

Citation: Günaydın, Y.Ş.; Şen, B. A Multi-Scale Unsupervised Feature Extraction Network with Structured Layer-Wise Decomposition. *Appl. Sci.* **2025**, *15*, 7194. <https://doi.org/10.3390/app15137194>

Copyright: © 2025 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) license (<https://creativecommons.org/licenses/by/4.0/>).

1. Introduction

In recent years, rapid technological advances have led to a dramatic increase in the volume of accessible data. This surge in data availability has contributed significantly to the development of computer-based systems designed for a wide range of purposes, particularly those that aim to improve human well-being through intelligent decision making, automation, and data-driven optimization. Moreover, the growing abundance of data and the increasing computational power have sped up the progress of artificial intelligence research, especially in deep learning, enhancing its capabilities and enabling its widespread adoption in almost all domains, such as image classification [1,2], denoising [3], object detection [4,5], image segmentation [6,7], decision support systems [8,9], medical imaging [10,11], facial expression recognition [12,13], and remote sensing [14,15]. However, the diverse range of problems addressed by machine learning models often introduces significant complexity and analytical challenges, with model performance still dependent on the quality and relevance of features extracted from raw data. The performance of the model remains dependent on the quality and relevance of the features extracted from the raw data. Moreover, real-world data are rarely perfect and often contain issues such as noise, missing values, or low resolution. Limited data, such as grayscale images, low-quality signals, or abbreviated text, constrains the extractable features in some situations.

These limitations make it more difficult for machine learning models to perform, as the quality of input features plays a crucial role in the learning process. Consequently, robust and effective feature extraction is important for improving the performance and success of machine learning models.

Researchers have developed many traditional feature extraction techniques based on mathematical descriptors to derive meaningful information from data. Notable examples include Histogram of Oriented Gradients (HOG) [16], Scale-Invariant Feature Transform (SIFT) [17], Gabor filters [18], Gaussian filters [19], and Local Binary Patterns (LBP) [20]. These methods are computationally efficient and well-suited for scenarios with limited computational resources or small-scale datasets. However, their performance is often constrained by their reliance on manually designed features, which may not generalize well across diverse tasks or data modalities.

With the advent of deep learning, feature extraction has undergone a major transformation. Modern approaches now enable models to automatically learn hierarchical and task-specific representations directly from raw data. Among the most widely used architectures are Convolutional Neural Networks (CNNs) [21], which use convolutional filters to learn spatial features and are widely applied to image, video, and speech data; Transformers [22], which utilize self-attention mechanisms to model long-range dependencies and have demonstrated impressive performance in various vision tasks; and Autoencoders [23], which compress input data into low-dimensional embeddings through unsupervised encoder–decoder frameworks.

Within these deep learning paradigms, multi-scale feature extraction has emerged as a critical strategy, especially for visual tasks that require a simultaneous understanding of both global structures and local fine-grained patterns. The hierarchical architecture of CNNs naturally supports multi-scale analysis, enabling the model to learn feature representations at different levels of abstraction. Liu et al. [24] demonstrated the value of this approach in medical image classification, while Ranjbarzadeh et al. [25] proposed a deep learning multi-route feature extraction architecture for improved breast tumor segmentation.

Domain-specific implementations of multi-scale methods further emphasize their practical effectiveness. Barburiceanu et al. [26] introduced a texture-aware CNN for plant disease classification in agricultural images. Ma et al. [27] developed a superpixel-wise, multi-scale model for hyperspectral image classification, enhancing robustness to noise. Likewise, Zhang et al. [28] improved traffic sign recognition by integrating multi-scale mechanisms that enriched spatial representations across object classes.

The conceptual roots of multi-scale learning trace back to classical computer vision techniques, such as Gaussian and Laplacian pyramids, which inspired modern hierarchical models. For instance, Feature Pyramid Networks (FPNs) [29] integrate features across multiple resolutions to improve object detection and segmentation. These strategies have since evolved and been adapted to multimodal and temporal data contexts. Lu et al. [30] proposed a fusion framework for Visual Question Answering (VQA) that integrates multi-scale linguistic features at the word, phrase, and sentence levels. Lei et al. [31] developed a spatio-temporal fusion method based on multi-scale feature extraction to capture structural features in dynamic visual data. Zhang et al. [32] developed an attention-guided multi-scale approach for lung cancer detection in CT scans. In image denoising, Jia et al. [33] introduced a residual-based multi-scale network that effectively suppresses noise in CT imagery. For image fusion tasks, Liu et al. [34] employed deep networks to learn shared multi-scale representations from multimodal images, facilitating more coherent and accurate fusion. In geophysical imaging, the SR-RDFAN-LOG network [35] used residual dense and attention-based multi-scale modules to enhance ultrasonic logging image resolution, an essential component in oil and gas exploration.

The benefits of multi-scale processing are not limited to the visual domain. In speech emotion recognition, Liu et al. [36] multi-scale CNN blocks to extract richer temporal patterns, resulting in improved classification performance. Text detection in complex environments has also benefited from multi-scale strategies. EMANet [37], for example, achieved high accuracy and speed in digitizing ancient texts for IoT systems through enhanced feature extraction and scale fusion.

In medical imaging, multi-scale methods continue to play a pivotal role. MEF-UNet [38] addressed challenges such as low contrast and blurred boundaries in ultrasound image segmentation through selective feature extraction and multi-scale fusion. A more recent vessel segmentation method [39] for retinal fundus images combined multi-scale feature learning with disentangled representation techniques, leveraging dilated convolutions, channel attention at skip connections, and an image reconstruction branch to separate informative content from background noise. This approach achieved state-of-the-art performance on several benchmarks, further underscoring the power of multi-scale processing in clinical diagnostics.

In the domain of remote sensing change detection, a novel framework [40] combined multi-scale feature extraction with specialized interaction and guidance modules to enhance edge preservation and semantic difference detection between bitemporal images—effectively addressing issues like seasonal variation and limited ground truth. For urban structure analysis, ME-FCN [41] incorporated tailored multi-scale fusion modules to improve building footprint extraction in complex cityscapes. Additionally, probabilistic Latent Semantic Hashing (pLSH) [42] leveraged multi-scale representations to learn topic-aware binary encodings for large-scale remote sensing image retrieval, demonstrating superior performance in unsupervised settings. Similarly, in Earth observation, a multi-scale feature extraction network [43] was designed to exploit complementary spatial, directional, and spectral cues from multi-source remote sensing data, improving classification accuracy in heterogeneous landscapes.

Collectively, these studies demonstrate the breadth and versatility of multi-scale feature extraction, highlighting its transformative impact across disciplines such as medical imaging, geospatial analysis, speech recognition, and image processing.

Building on these foundations, this study introduces a novel, unsupervised feature extraction framework based on autoencoders to construct multi-scale representations from grayscale image data. The proposed model decomposes each input into three distinct components:

- A smooth layer that captures coarse structural features,
- A detail layer preserving fine textures and high-frequency information,
- A residual layer isolating remaining, often less informative, content.

This decomposition is achieved through an encoding stage using convolutional filters, followed by three parallel decoding branches—each responsible for reconstructing one of the components. To ensure each branch captures distinct, non-overlapping features, dedicated variational loss functions are applied independently. The reconstructed components are then summed to recover the original image, thereby validating the effectiveness of the decomposition strategy.

A key advantage of this approach lies in its fully unsupervised training paradigm, which removes the dependency on labeled data. Moreover, the model enhances grayscale input by generating expressive three-channel outputs, which are more compatible with existing pre-trained convolutional networks requiring three channel input.

In contrast to many task-specific feature extractors embedded within custom model architectures, the proposed framework operates as a general-purpose preprocessing step. It can be seamlessly integrated with a wide array of downstream machine learning models

and tasks, providing richer, multi-scale representations without necessitating architectural changes.

In summary, this study offers the following key contributions:

- Introduction of a novel unsupervised feature extraction framework based on an autoencoder structure that decomposes input data into smooth, detailed, and residual layers without requiring labeled data.
- Design of a multi-branch autoencoder architecture that enables transforming single-channel inputs into rich, multi-scale representations to enhance machine learning task performance.
- Implementation of a layer-specific variational loss strategy to preserve semantic consistency across decomposed components while ensuring accurate reconstruction of the original input.
- Demonstration that the proposed method can be used as a flexible preprocessing step for various models, particularly pre-trained networks that require three-channel input.

We organized the rest of the paper as follows. We provide a comprehensive description of the proposed method in Section 2. Section 3 outlines the experimental setup along with the datasets used for the evaluation and presents and analyzes the experimental results, highlighting the effectiveness of the proposed approach. Section 4 discusses the experimental results of the proposed method. Section 5 concludes the study by summarizing the key findings and offering directions for future research.

2. Materials and Methods

In this section, we present a detailed explanation of the proposed unsupervised multi-scale feature extraction method.

2.1. Proposed Model Architecture

The methodology is based on an autoencoder architecture that uses images as input and is specifically designed to decompose the input data into three distinct layers: a smooth layer capturing coarse structural information, a detail layer representing fine-grained features, and a residual layer containing the remaining, often irrelevant, information as shown in Figure 1. This layer-wise decomposition aligns naturally with the autoencoder's core objective of reconstructing the input from a compact and structured representation. Unlike standard CNNs, which are typically optimized for classification and lack explicit reconstruction mechanisms, or Transformers, which are computationally intensive and less suited for localized spatial decomposition, the autoencoder enables interpretable, additive reconstruction with dedicated branches for each type of structural component. This makes it a particularly suitable choice for our goal of semantically meaningful image separation and reconstruction.

The model first applies convolutional operations to compress the input and after it splits into three branches responsible for reconstructing each of the target layers. To ensure that these layers represent the different aspects of the input data in a meaningful way, specific variational loss functions are applied. These losses enforce the separation of information across layers while preserving the overall fidelity of the reconstruction. The sum of these three outputs is expected to closely approximate the original input, allowing for effective unsupervised decomposition without the need for labeled data. This approach enables the generation of multi-scale representations even from single-channel images, potentially increasing the amount of usable information for downstream machine learning tasks.

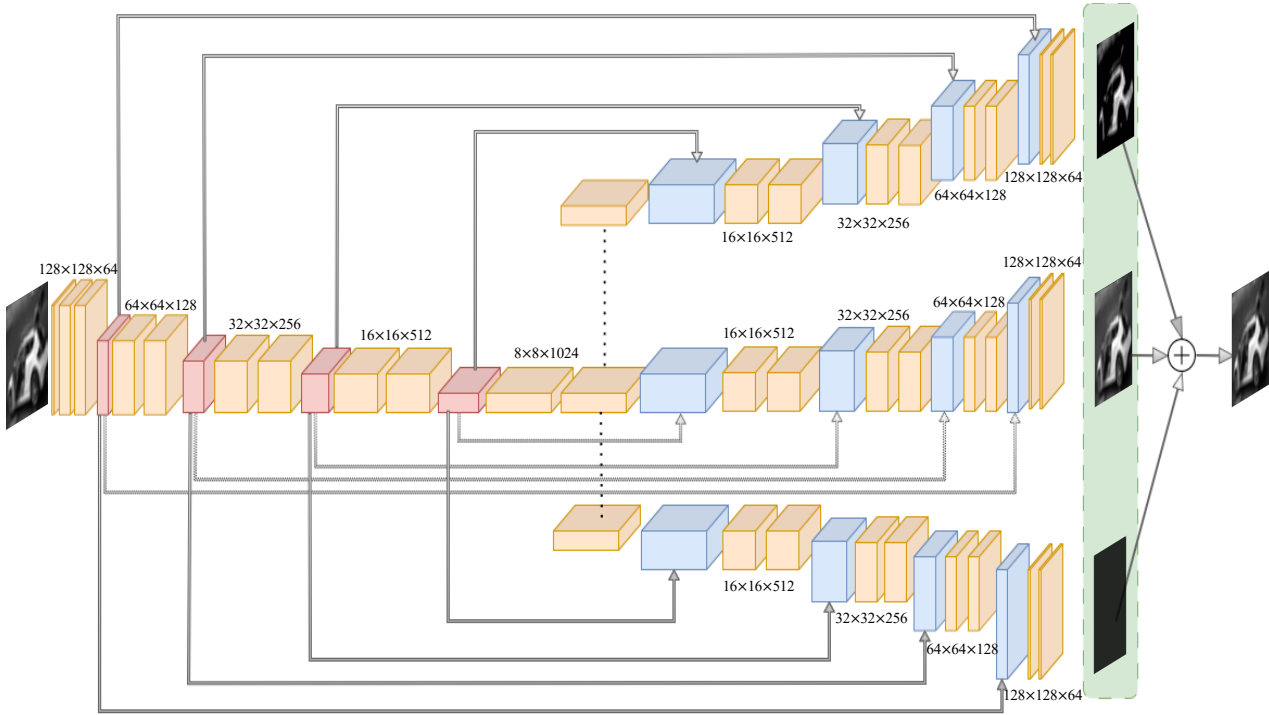


Figure 1. The proposed method architecture. Yellow blocks represent convolutional layers, red blocks indicate max pooling operations, and blue blocks correspond to upsampling layers.

2.2. Loss Functions

To guide the network branches to learn distinct yet complementary feature representations, we select loss functions grounded in well-established theory and practice from image processing and deep learning. Because of its simplicity and effectiveness in penalizing overall reconstruction error, as described in the foundational deep learning literature [44], researchers widely adopt the pixel-wise Mean Squared Error (MSE) loss of image reconstruction tasks. To encourage spatial smoothness in the smooth layer and reduce high-frequency noise, we employ a squared gradient penalty that approximates the Total Variation (TV) regularization principle, which has proven effective for noise removal while preserving edges in images [45]. Conversely, the detail layer directly uses the L1 norm of the gradient to impose a first-order TV loss, thus preserving important edges and fine structures through sharp transitions [46]. These complementary norms balance smoothness and edge preservation. Lastly, residual layer loss uses a logarithmic penalty that controls the scale of activations robustly, promoting sparsity and suppressing unstructured noise components. Together, this set of loss functions enables the multi-branch auto-encoder to disentangle smooth, detailed, and residual features effectively.

$$Loss(\hat{f}) = MSE + \mathcal{L}_{smooth} + \mathcal{L}_{detail} + \mathcal{L}_{residual} \quad (1)$$

This formulation includes a pixel-wise mean squared error (MSE) between the reconstructed image and the ground truth, along with three specialized regularization terms for smooth, detail, and residual layers, respectively, in the corresponding decoder branches.

The general form of the loss is expanded as follows:

$$Loss(\hat{f}) = \sum_{i,j} (\hat{f}_{i,j} - f_{i,j})^2 + \lambda_1 \sum_{i,j} \left(\nabla f_{i,j}^{(1)} \right)^2 + \lambda_2 \sum_{i,j} \left| \nabla f_{i,j}^{(2)} \right| + \lambda_3 \sum_{i,j} \log(1 + f_{i,j}^{(3)}) \quad (2)$$

Here, $\hat{f}$ is the reconstructed image composed of the three branches $f^{(1)}$, $f^{(2)}$, and $f^{(3)}$, which correspond to the smooth, detail, and residual layers, respectively. The first term

penalizes reconstruction errors. The second term enforces spatial smoothness using the squared gradient magnitude of the smooth layer $f^{(1)}$. The third term uses the L_1 norm of the gradient to preserve fine structures and edges in the detail layer $f^{(2)}$. The final term encourages sparsity in the residual branch $f^{(3)}$ to separate noise and unstructured information.

We begin by defining the continuous form of the smoothness loss based on the squared 2D spatial gradient:

$$\begin{aligned} \text{Loss}(f^{(1)}, f^{(2)}, f^{(3)}) = & \sum_{i,j} \left((f_{i,j}^{(1)} + f_{i,j}^{(2)} + f_{i,j}^{(3)}) - f_{i,j} \right)^2 \\ & + \lambda_1 \sum_{i,j} \left[\left(\frac{\partial f_{i,j}^{(1)}}{\partial x} \right)^2 + \left(\frac{\partial f_{i,j}^{(1)}}{\partial y} \right)^2 \right] \\ & + \lambda_2 \sum_{i,j} \left[\left| \frac{\partial f_{i,j}^{(2)}}{\partial x} \right| + \left| \frac{\partial f_{i,j}^{(2)}}{\partial y} \right| \right] \\ & + \lambda_3 \sum_{i,j} \log(1 + f_{i,j}^{(3)}) \end{aligned} \quad (3)$$

To encourage spatial coherence in the output feature maps, we introduce smooth layer loss, which penalizes sharp variations between neighboring activations. This regularization is particularly beneficial in tasks such as image synthesis, segmentation, or depth estimation, where the output is expected to vary smoothly over space.

In Equation (3), $f_{i,j}^{(1)}$ represents the activation at spatial location (i, j) on a given feature map, and the terms $\frac{\partial f_{i,j}^{(1)}}{\partial x}$ and $\frac{\partial f_{i,j}^{(1)}}{\partial y}$ denote the partial derivatives of the activation values in the horizontal and vertical directions, respectively. This expression corresponds to the squared norm of the 2D gradient vector, which serves to penalize large changes in the feature map between adjacent positions.

In practice, neural networks operate on discrete grid-based data, so we approximate the continuous partial derivatives using forward finite differences:

$$\frac{\partial f_{i,j}^{(1)}}{\partial x} \approx f_{i+1,j}^{(1)} - f_{i,j}^{(1)}, \quad \frac{\partial f_{i,j}^{(1)}}{\partial y} \approx f_{i,j+1}^{(1)} - f_{i,j}^{(1)} \quad (4)$$

These approximations in Equation (4) estimate the local change in activation values by computing the difference between a pixel and its immediate neighbor in the corresponding direction. Forward difference usage is computationally efficient and is widely used in discrete image processing.

Substituting these approximations into Equation (2), we obtain the discretized version of the smooth layer loss in Equation (5).

$$\mathcal{L}_{\text{smooth}} \approx \lambda_1 \sum_{i,j} \left[\left(f_{i+1,j}^{(1)} - f_{i,j}^{(1)} \right)^2 + \left(f_{i,j+1}^{(1)} - f_{i,j}^{(1)} \right)^2 \right] \quad (5)$$

This formulation penalizes high-frequency components in the spatial structure of the feature map by minimizing the squared differences between adjacent pixels. As a result, it promotes locally smooth outputs and reduces visual artifacts such as noise. The hyperparameter λ_1 controls the relative strength of this regularization term in the total loss function.

The entire loss term is differentiable and compatible with automatic differentiation frameworks. In the implementation, it can be computed efficiently using convolutional filters or element-wise operations on tensor slices.

In addition to encouraging global smoothness, it is often desirable to preserve important local details such as edges and texture boundaries. To this end, we introduce a detail layer loss, which penalizes the absolute magnitude of the spatial gradients. This formulation aligns with the concept of total variation (TV) regularization, a well-known technique in signal processing and computer vision.

In Equation (3), the loss of detail layer computes the total variation by summing the absolute values of the first-order derivatives across both spatial directions. It encourages the solution to be smooth in part while allowing for sudden changes when necessary. As in the previous section, we approximate the partial derivatives on a discrete grid using forward finite differences in Equation (6).

$$\frac{\partial f_{i,j}^{(2)}}{\partial x} \approx f_{i+1,j}^{(2)} - f_{i,j}^{(2)}, \quad \frac{\partial f_{i,j}^{(2)}}{\partial y} \approx f_{i,j+1}^{(2)} - f_{i,j}^{(2)} \quad (6)$$

Substituting these into Equation (3) yields the discretized form of the loss of detail layer in Equation (7).

$$\mathcal{L}_{\text{detail}} \approx \lambda_2 \sum_{i,j} \left[\left| f_{i+1,j}^{(2)} - f_{i,j}^{(2)} \right| + \left| f_{i,j+1}^{(2)} - f_{i,j}^{(2)} \right| \right] \quad (7)$$

This discretized formulation corresponds to the first-order total variation in two dimensions. It is widely used to preserve structure while reducing minor variations and noise. The hyperparameter λ_2 controls the strength of the regularization. Higher values increase edge preservation and denoising at the potential cost of over-smoothing fine textures.

To further regulate the scale and dynamics of the output values, we introduce a residual layer loss based on a logarithmic penalty. This function discourages large magnitudes in the output activations while maintaining numerical stability, particularly for small values.

The residual loss is defined as:

$$\mathcal{L}_{\text{residual}} = \lambda_3 \sum_{i,j} \log(1 + f_{i,j}^{(3)}) \quad (8)$$

In Equation (8), $x_{i,j}$ denotes the activation value at position (i, j) , and λ_3 is a regularization coefficient. The logarithmic function applies a sublinear penalty that is growing more slowly than linear or quadratic penalties so that it is useful when it is desirable to compress the influence of larger activation values while still penalizing them.

The use of $\log(1 + x_{i,j})$ ensures that the function remains well-defined for non-negative $x_{i,j}$, and avoids singularities or instability near zero. This form is related to robust cost functions used in statistics and optimization. The hyperparameter λ_3 controls the strength of the regularization.

2.3. Model Description

In our multi-scale feature extraction framework, we propose a novel approach that enables neural networks to distinguish between different layers of information in spatial resolutions while filtering out redundant data. This methodology aims to progressively expand the depth and capacity of the network while maintaining a consistent architectural pattern across all scales.

We begin by feeding the input data into the encoder component of our model. Within the encoder, each feature extraction block consists of convolutional layers, batch normalization, and ReLU activation functions. These blocks are designed to extract features on multiple spatial scales. As the network progresses deeper, the receptive fields of the convolutional layers expand, enabling the capture of increasingly abstract representations. The encoder part, in particular, is responsible for capturing contextual information through repeated convolutional operations and downsampling via max-pooling operations. The details of the encoding part of the model are shown in Table 1.

Table 1. The details of encoder part of the model.

Layer	Type	Input Channels	Output Channels	Kernel Size	Stride	Padding
Conv1	Conv2d	1	64	3×3	1	1
Conv2	Conv2d	64	64	3×3	1	1
Pool1	MaxPool2d	-	-	2×2	2	0
Conv3	Conv2d	64	128	3×3	1	1
Conv4	Conv2d	128	128	3×3	1	1
Pool2	MaxPool2d	-	-	2×2	2	0
Conv5	Conv2d	128	256	3×3	1	1
Conv6	Conv2d	256	256	3×3	1	1
Pool3	MaxPool2d	-	-	2×2	2	0
Conv7	Conv2d	256	512	3×3	1	1
Conv8	Conv2d	512	512	3×3	1	1
Pool4	MaxPool2d	-	-	2×2	2	0
Conv9	Conv2d	512	1024	3×3	1	1
Conv10	Conv2d	1024	1024	3×3	1	1

Following the encoding phase, as shown in Table 2, the decoder component of our model, composed of expanding paths, facilitates the reconstruction of high-resolution features from the encoded representations. Skip connections are used to bridge the encoder and decoder pathways, allowing for the integration of low-level spatial information from earlier layers with high-level semantic features learned in deeper layers. Our architecture includes three symmetric expanding paths, each designed to extract and reconstruct different levels of information from the encoded data. At the end of each path, distinct loss functions are applied to guide the reconstruction process. The first path aims to recover a smooth approximation of the original data, the second emphasizes the preservation of fine details, and the third is structured to capture redundant or residual information. The fusion of these three outputs is intended to accurately reconstruct the input data, ensuring both structural coherence and representational richness.

Table 2. The details of each layer for the decoder part of the model.

Layer	Type	Input Channels	Output Channels	Kernel Size	Stride	Padding
ConvTrans1	ConvTrans2d	1024	512	3×3	2	1
Conv1	Conv2d	512	512	3×3	1	1
ConvTrans2	ConvTrans2d	512	256	3×3	2	1
Conv2	Conv2d	256	256	3×3	1	1
ConvTrans3	ConvTrans2d	256	128	3×3	2	1
Conv3	Conv2d	128	128	3×3	1	1
ConvTrans4	ConvTrans2d	128	64	3×3	2	1
Conv4	Conv2d	61	64	3×3	1	1
Conv5	Conv2d	64	1	3×3	1	1

3. Experiments

In this section, we examine the effectiveness of the proposed unsupervised multi-scale feature extraction technique, which decomposes input images into smooth, detail, and residual layers using an autoencoder architecture. We also provide comprehensive information on the datasets, tasks, and experimental setup.

3.1. Dataset and Task

To evaluate the performance of the proposed method, two tasks were conducted: a multiclass classification task on the CIFAR-10 dataset [47] and an image segmentation task on a medical CT dataset [48]. CIFAR-10, a widely recognized benchmark developed by the Canadian Institute for Advanced Research, contains 60,000 colored images with a resolution of 32×32 pixels, equally distributed across 10 diverse classes. All CIFAR-10 images were preprocessed by resizing to 128×128 pixels using bicubic interpolation, converting them to single-channel grayscale, and transforming them into tensor format. The training set was used to train the model, while the test set was reserved for performance evaluation. The diverse nature of CIFAR-10 helps expose the model to various visual patterns, improving feature learning. Although hyperspectral and RGB image datasets provide richer information, practical constraints such as computational resources and hardware limitations restricted their use in this study. Therefore, CIFAR-10 was selected as a computationally efficient dataset to demonstrate the method's capabilities.

In addition, to demonstrate the effectiveness of the proposed method in a more complex and practical application, an image segmentation task was performed using a high-resolution medical CT dataset [49] focused on lung segmentation. This dataset consists of 267 manually segmented CT scans with a resolution of 512×512 pixels. Accurate lung segmentation is a critical preprocessing step in tasks such as lesion detection and disease classification. Including this dataset provides additional validation of the applicability of the proposed method in real-world medical imaging scenarios, highlighting its robustness in higher-resolution grayscale data.

3.2. Experimental Settings

The proposed method was developed using the Python 3.11.5 programming language, which offers extensive support for scientific computing and deep learning research. The PyTorch 2.0.1 deep learning library was selected for the implementation of the proposed deep learning model. The implementation was carried out in an environment equipped with an AMD Ryzen 9 7950X 16-core processor, 32 GB of RAM, and an 8 GB GeForce RTX 3080 Ti GPU. To evaluate the performance of the model, several key metrics were used, including precision, recall, F1 score, test set accuracy for the classification and confusion matrices. Together, these metrics offer a comprehensive view of the model's classification performance.

3.3. Experimental Configurations

To evaluate the proposed method, several pre-trained network architectures that accept three-channel inputs were used. The ResNet-18 [50], ResNet-50 [51], and MobileNetV2 [52] architectures were used to analyze the impact of newly generated features on classification performance. ResNet-18 and ResNet-50 are deep residual networks designed to address the problem of vanishing gradient through residual connections, which allows the training of deep networks. ResNet-18 consists of 18 layers, while ResNet-50, with 50 layers, enables more complex feature extraction. MobileNetV2 is a lightweight architecture optimized for efficient performance on mobile and embedded devices. Using separable convolutions to reduce computational complexity while maintaining high classification accuracy.

The evaluation process began with the extraction of the residual layer from the output of the proposed method. The smooth and detailed components were then merged with the original input to form a new three-channel representation, which served as input for the ResNet-18, ResNet-50, and MobileNetV2 networks to evaluate the classification performance. In addition, a separate test was conducted using the same networks, where no pre-preprocessing was applied to the input images. Instead, a grayscale version of each image was replicated across all three channels to form the input. The details of the hyperparameters used in the training process of the proposed model are shown in Table 3.

In our experiments, λ_1 , λ_2 , and λ_3 values for proposed loss functions were assigned fixed values to maintain consistency and ensure stable convergence during training. These values were empirically determined through preliminary experiments in a validation subset of the training data. The aim was to balance the influence of each branch so that the smooth and detail layers capture meaningful structural information while the residual layer suppresses noise or irrelevant patterns. Improper tuning of the regularization coefficients can lead to ineffective decomposition of the image features across the respective layers. In such cases, most of the image information can be gathered in a single layer, causing the other layers to become redundant or uninformative. This undermines the model's ability to decompose the image into smooth, detailed, and residual components as intended. This parameter tuning represents a challenging aspect of the proposed method. Therefore, careful adjustment of these parameters is essential to ensure that each layer captures distinct and meaningful features, ultimately enhancing the quality of feature extraction and improving model performance.

Table 3. Hyperparameter details of the proposed method.

Parameters	Values
Input Size	128×128
Initial Learning Rate	0.00001
Optimizer	Adam
Batch Size	16
Cost Function	$\text{MSE} + \mathcal{L}_{\text{smooth}} + \mathcal{L}_{\text{detail}} + \mathcal{L}_{\text{residual}}$
λ_1 (Smoothness Loss Weight)	1
λ_2 (Detail Loss Weight)	4
λ_3 (Residual Loss Weight)	3
Epoch	100

Besides classification, the proposed method was further evaluated on an image segmentation task using a lung CT dataset. The pre-trained multibranch autoencoder model, originally trained on CIFAR-10, was used to decompose each CT image into smooth, detailed, and residual components. For this experiment, the residual layer was excluded, which is typically the source of redundant information and noise, and the original image was used in its place. These components were then combined and used as input to a U-Net [53] segmentation model. In addition, we performed an ablation study by training the segmentation network in different combinations of smooth layer, detail layer, and original image to validate the contribution of each component.

These experiments assessed the impact of the proposed method on the extractable information from the data and evaluated how these newly generated representations influence the performance of machine learning models built upon them.

3.4. Experimental Results

The performance of the proposed model was evaluated using the CIFAR-10 test dataset. To assess its effectiveness, standard classification metrics are used, including precision,

recall, F1-score, accuracy, and confusion matrix. Precision and recall were calculated using Equations (10) and (11), where TP denotes true positives, TN denotes true negatives, FP denotes false positives, and FN denotes false negatives. The F1-score was calculated using Equation (12).

$$\text{Accuracy} = \frac{\text{TP} + \text{TN}}{\text{TP} + \text{TN} + \text{FP} + \text{FN}} \quad (9)$$

$$\text{Precision} = \frac{\text{TP}}{\text{TP} + \text{FP}} \quad (10)$$

$$\text{Recall} = \frac{\text{TP}}{\text{TP} + \text{FN}} \quad (11)$$

$$\text{F1} = 2 * \frac{\text{Precision} * \text{Recall}}{\text{Precision} + \text{Recall}} \quad (12)$$

The outputs of the proposed method are used as inputs for the specified deep learning models and compared with the models' performance when using grayscale images as input. Across all models, the proposed method consistently outperforms the grayscale baseline. ResNet50 achieves an accuracy of 86.32% when using the proposed method, compared to 85.45% with grayscale inputs. Similarly, ResNet18 shows an improvement from 85.43% to 86.02%, and MobileNetV2 shows the most pronounced gain, increasing from 83.02% to 85.57%. These results suggest that the proposed method provides a more informative input representation, leading to better classification performance across different architectures, as shown in Table 4.

Table 4. Classification performance on CIFAR-10 test dataset.

Model	Proposed Method				Grayscale Images			
	Accuracy (%)	Precision	Recall	F1-Score	Accuracy (%)	Precision	Recall	F1-Score
ResNet18	86.02	0.86	0.8602	0.86	85.43	0.8561	0.8563	0.8560
ResNet50	86.32	0.8631	0.8632	0.8629	85.45	0.8541	0.8545	0.8540
MobileNetv2	85.57	0.8567	0.8557	0.8559	83.02	0.8331	0.8302	0.8308

Figures 2–4 present comparative analyses of precision, recall, and F1-score. These comparisons illustrate the performance differences between using the output of the proposed method as input to deep learning models and using grayscale images as input to the same models ResNet18, ResNet50, and MobileNetV2.

Figure 5 illustrates the confusion matrices for each experimental configuration, highlighting the impact of the proposed method compared to grayscale image inputs across three deep learning architectures: ResNet18, ResNet50, and MobileNetV2. A comparative analysis reveals that the proposed feature extraction approach consistently enhances classification performance. With ResNet18, for example, the number of correctly classified samples increased significantly in classes such as plane (from 861 to 875) and bird (from 768 to 796). Similarly, ResNet50 demonstrated improvements in the classifications of plane (841 to 868), deer (820 to 832), and cat (722 to 741). MobileNetV2 exhibited particularly more gains, with correct predictions increasing in the car (907 to 951), plane (794 to 844), deer (818 to 850), and ship (887 to 930) classes. When aggregated across all classes, the total number of correctly classified samples increased by 67 for ResNet18 (from 8683 to 8750), 69 for ResNet50 (from 8717 to 8786), and 132 for MobileNetV2 (from 8597 to 8729). These results underscore the effectiveness of the proposed representation in providing more discriminative features, thereby enhancing model performance across a variety of architectures and particularly in categories where inter-class confusion is typically higher, such as car and truck or cat and dog.

	Proposed Method + ResNet18			Grayscale Inputs + ResNet18		
	precision	recall	F1-score	precision	recall	F1-score
plane	0.887	0.868	0.877	0.883	0.841	0.862
car	0.940	0.945	0.943	0.930	0.939	0.934
bird	0.798	0.794	0.796	0.801	0.784	0.792
cat	0.734	0.741	0.738	0.713	0.722	0.717
deer	0.825	0.832	0.829	0.841	0.820	0.830
dog	0.841	0.780	0.809	0.806	0.766	0.786
frog	0.858	0.908	0.882	0.853	0.912	0.882
horse	0.915	0.909	0.912	0.896	0.904	0.900
ship	0.922	0.931	0.926	0.902	0.934	0.918
truck	0.909	0.924	0.917	0.916	0.923	0.919

Figure 2. Precision, Recall, and F1-score for ResNet18 on CIFAR-10 Dataset.

	Proposed Method + ResNet50			Grayscale Inputs + ResNet50		
	Precision	Recall	F1-score	Precision	Recall	F1-score
plane	0.887	0.868	0.877	0.883	0.841	0.862
car	0.940	0.945	0.943	0.930	0.939	0.934
bird	0.798	0.794	0.796	0.801	0.784	0.792
cat	0.734	0.741	0.738	0.713	0.722	0.717
deer	0.825	0.832	0.829	0.841	0.820	0.830
dog	0.841	0.780	0.809	0.806	0.766	0.786
frog	0.858	0.908	0.882	0.853	0.912	0.882
horse	0.915	0.909	0.912	0.896	0.904	0.900
ship	0.922	0.931	0.926	0.902	0.934	0.918
truck	0.909	0.924	0.917	0.916	0.923	0.919

Figure 3. Precision, Recall and F1-score for ResNet50 on CIFAR-10 Dataset.

	Proposed Method + MobileNetV2			Grayscale Inputs + MobileNetV2		
	Precision	Recall	F1-score	Precision	Recall	F1-score
plane	0.883	0.844	0.863	0.880	0.794	0.835
car	0.934	0.951	0.943	0.945	0.907	0.925
bird	0.833	0.763	0.796	0.795	0.726	0.759
cat	0.707	0.745	0.725	0.667	0.732	0.698
deer	0.809	0.850	0.829	0.796	0.818	0.807
dog	0.782	0.789	0.785	0.760	0.767	0.764
frog	0.870	0.886	0.878	0.797	0.877	0.835
horse	0.911	0.884	0.897	0.889	0.872	0.880
ship	0.919	0.930	0.924	0.916	0.887	0.901
truck	0.919	0.915	0.917	0.884	0.922	0.903

Figure 4. Precision, Recall and F1-score for MobileNetV2 on CIFAR-10 Dataset.

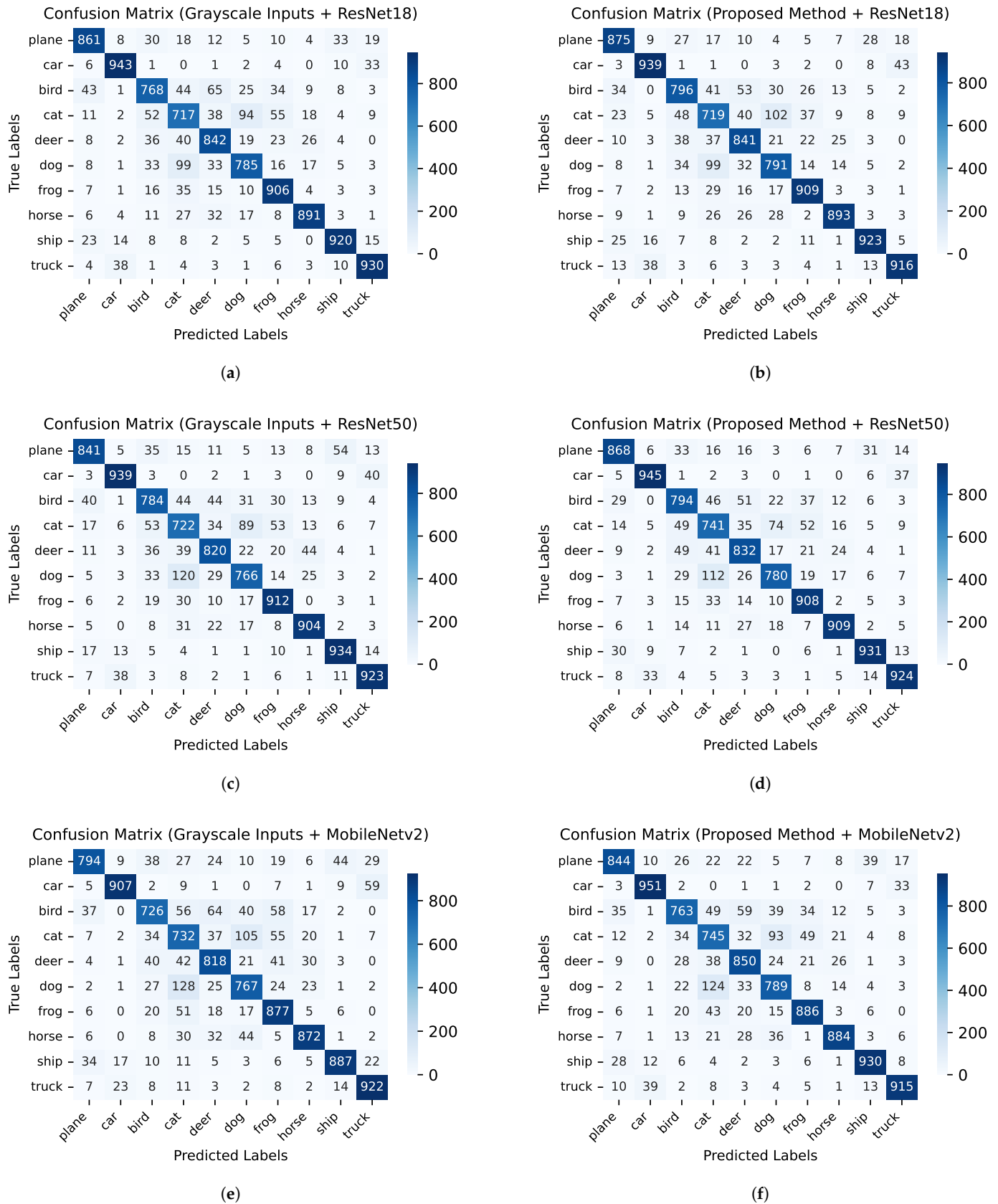


Figure 5. Confusion matrix results for classification: (a) Grayscale images given to ResNet18 as inputs, (b) Proposed Method outputs given to ResNet18 as inputs, (c) Grayscale images given to ResNet50 as inputs, (d) Proposed Method outputs given to ResNet50 as inputs, (e) Grayscale images given to MobileNetv2 as inputs, and (f) Proposed Method outputs given to MobileNetv2 as inputs.

The qualitative performance of the proposed decomposition process is illustrated in Figure 6, which shows how the original image is separated into three semantically meaningful layers: a smooth layer capturing coarse structural information, a detail layer emphasizing fine textures and edges, and a residual layer isolating noise and unstructured content. As expected, the residual layer appears mostly sparse, confirming that most relevant information is effectively captured by the first two layers.

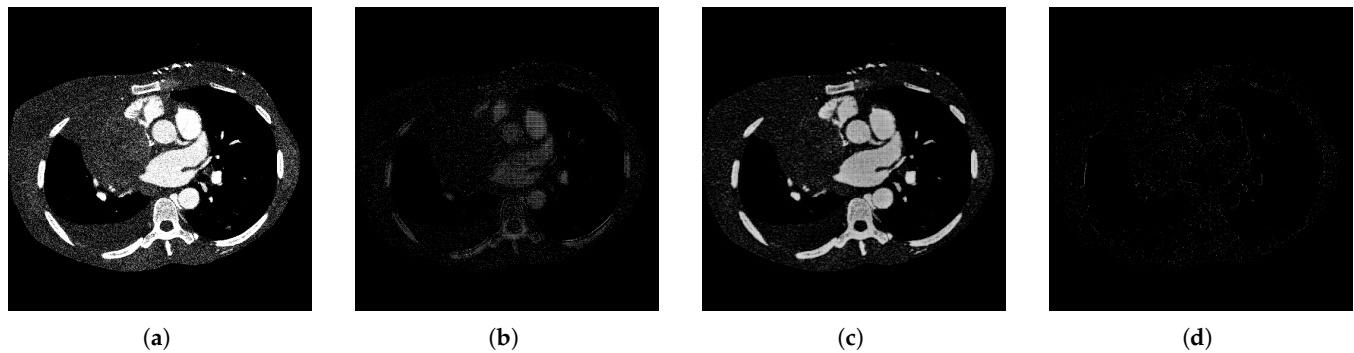


Figure 6. Visual representation of the image decomposition process using the proposed method: (a) Original input image, (b) smooth layer output, (c) detail layer output, (d) residual layer output.

The results presented in Table 5 illustrate the impact of different input combinations on segmentation performance using the U-Net architecture. Among all configurations, the U-Net model that utilized the original image together with the detail layer achieved the highest performance across all metrics, with a Dice score of 0.9778, IoU of 0.9600, and pixel accuracy of 0.9897. This indicates that the detail component significantly enhances the model's ability to capture fine-grained structures critical for accurate medical image segmentation.

$$\text{Dice} = \frac{2 \times |A \cap B|}{|A| + |B|} = \frac{2\text{TP}}{2\text{TP} + \text{FP} + \text{FN}} \quad (13)$$

$$\text{IoU} = \frac{|A \cap B|}{|A \cup B|} = \frac{\text{TP}}{\text{TP} + \text{FP} + \text{FN}} \quad (14)$$

Table 5. Segmentation performance comparison with different input combinations

Model	Dice Score	IoU	Pixel Accuracy
U-Net (Original Image)	0.8276	0.7646	0.9412
U-Net (Original + Smooth Layer)	0.7931	0.7417	0.9360
U-Net (Original + Detail Layer)	0.9778	0.9600	0.9897
U-Net (Original + Smooth + Detail)	0.9636	0.9355	0.9834

In comparison, the baseline model using only the original image achieved moderate performance (Dice: 0.8276, IoU: 0.7646, Pixel Accuracy: 0.9412), while adding only the smooth layer slightly degraded performance. Interestingly, when both the smooth and detail layers were combined with the original image, performance remained high (Dice: 0.9636), though still slightly lower than using the detail layer alone. These findings suggest that while smooth features may dilute fine details when used in isolation, they can still complement high-frequency detail features in multi-scale representations. Ultimately, the detail layer is the most influential factor in improving segmentation quality, and its integration into the input representation leads to significant gains in accuracy and robustness.

4. Discussion

The primary contribution of this study is the introduction of an unsupervised multi-scale feature extraction model inspired by scale-space theory. Rather than targeting state-of-the-art performance, the goal is to offer a modular and interpretable preprocessing technique that enriches feature representations and improves downstream task performance.

Our experiments on classification and segmentation tasks demonstrated the model's ability to enhance both the diversity and relevance of extracted features. In segmentation, the decomposed layers—particularly the detail and smooth components—proved effective in isolating structural details and suppressing irrelevant variations.

Despite using relatively small datasets and conventional models, the architecture is inherently scalable. Its fully convolutional design allows seamless operation on various image sizes that are multiple of 16 (e.g., 512×512) without structural modification. This flexibility enables re-training, fine-tuning, or plug-and-play use with pre-trained weights, making the model suitable for broader applications in medical imaging, remote sensing, and other domains requiring multi-scale analysis.

5. Conclusions

In this study, we proposed an unsupervised multi-scale feature extraction mechanism capable of decomposing image content into smooth, detailed, and redundant components. The proposed approach demonstrated promising results in improving classification performance by increasing the representational capacity of the input data. Specifically, on a 10-class dataset, our method led to accuracy improvements of up to 3 percent, underscoring its potential to support and strengthen downstream tasks.

Beyond classification, we evaluated the model on an image segmentation task using 512×512 resolution medical images. The decomposed layers, particularly the detail component proved effective in capturing fine structures essential for accurate boundary delineation, while the smooth layer helped suppress irrelevant background information. These results highlight the model's utility in tasks where spatial precision is critical, such as biomedical or remote sensing image analysis.

In addition to its quantitative benefits, the model offers several practical advantages: it preserves the structural integrity of the input, enriches feature representations without requiring supervision, and remains adaptable to datasets and architectures of varying complexity. Its fully convolutional design enables the straightforward deployment of inputs with compatible dimensions, enhancing its flexibility in real-world applications.

Future work will focus on assessing the model's effectiveness across broader domains beyond classification and segmentation, including anomaly detection, image synthesis, and remote sensing. Furthermore, investigating the sensitivity and robustness of the model to different hyperparameter settings, such as regularization weights and architectural configurations will be crucial to ensure stability across varied datasets and tasks.

Author Contributions: Conceptualization, methodology, writing paper, Y.Ş.G.; formal analysis, supervision, providing research ideas, supervision, B.Ş. All authors have read and agreed to the published version of the manuscript.

Funding: This research received no external funding.

Institutional Review Board Statement: Not applicable.

Informed Consent Statement: Not applicable.

Data Availability Statement: All data generated or analyzed during this study are included in this article and are available from the corresponding author upon reasonable request.

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

Abbreviations

The following abbreviations are used in this manuscript:

HOG	Histogram of Oriented Gradients
SIFT	Scale-Invariant Feature Transform
LBP	Locally Binary Patterns
CNN	Convolutional Neural Network
2D	2 Dimensional
ReLU	Rectified Linear Unit
GB	Gigabyte
TP	True Positives
FP	False Positives
FN	False Negatives
RGB	Red Green Blue

References

- Chen, L.; Li, S.; Bai, Q.; Yang, J.; Jiang, S.; Miao, Y. Review of Image Classification Algorithms Based on Convolutional Neural Networks. *Remote Sens.* **2021**, *13*, 4712. [\[CrossRef\]](#)
- Jiao, L.; Gao, J.; Liu, X.; Liu, F.; Yang, S.; Hou, B. Multiscale Representation Learning for Image Classification: A Survey. *IEEE Trans. Artif. Intell.* **2021**, *4*, 23–43. [\[CrossRef\]](#)
- Ilesanmi, A.E.; Ilesanmi, T.O. Methods for Image Denoising Using Convolutional Neural Network: A Review. *Complex Intell. Syst.* **2021**, *7*, 2179–2198. [\[CrossRef\]](#)
- Zou, Z.; Chen, K.; Shi, Z.; Guo, Y.; Ye, J. Object Detection in 20 Years: A Survey. *Proc. IEEE* **2023**, *111*, 257–276. [\[CrossRef\]](#)
- Li, Z.; Wang, Y.; Zhang, N.; Zhang, Y.; Zhao, Z.; Xu, D.; Ben, G.; Gao, Y. Deep Learning-Based Object Detection Techniques for Remote Sensing Images: A Survey. *Remote Sens.* **2022**, *14*, 2385. [\[CrossRef\]](#)
- Yu, Y.; Wang, C.; Fu, Q.; Kou, R.; Huang, F.; Yang, B.; Yang, T.; Gao, M. Techniques and Challenges of Image Segmentation: A Review. *Electronics* **2023**, *12*, 1199. [\[CrossRef\]](#)
- Minaee, S.; Boykov, Y.; Porikli, F.; Plaza, A.; Kehtarnavaz, N.; Terzopoulos, D. Image Segmentation Using Deep Learning: A Survey. *IEEE Trans. Pattern Anal. Mach. Intell.* **2021**, *44*, 3523–3542. [\[CrossRef\]](#)
- Gupta, S.; Modgil, S.; Bhattacharyya, S.; Bose, I. Artificial Intelligence for Decision Support Systems in the Field of Operations Research: Review and Future Scope of Research. *Ann. Oper. Res.* **2022**, *308*, 215–274. [\[CrossRef\]](#)
- Nguyen, Q.-T.; Tran, T. N.; Heuchenne, C.; Tran, K. P. Decision Support Systems for Anomaly Detection with the Applications in Smart Manufacturing: A Survey and Perspective. In *Machine Learning and Probabilistic Graphical Models for Decision Support Systems*; CRC Press: Boca Raton, FL, USA, 2022; pp. 34–61.
- Shamshad, F.; Khan, S.; Zamir, S.W.; Khan, M.H.; Hayat, M.; Khan, F.S.; Fu, H. Transformers in Medical Imaging: A Survey. *Med. Image Anal.* **2023**, *88*, 102802. [\[CrossRef\]](#)
- Pinto-Coelho, L. How Artificial Intelligence Is Shaping Medical Imaging Technology: A Survey of Innovations and Applications. *Bioengineering* **2023**, *10*, 1435. [\[CrossRef\]](#)
- Revina, I.M.; Emmanuel, W.R.S. A Survey on Human Face Expression Recognition Techniques. *J. King Saud Univ. Comput. Inf. Sci.* **2021**, *33*, 619–628. [\[CrossRef\]](#)
- Karnati, M.; Seal, A.; Bhattacharjee, D.; Yazidi, A.; Krejcar, O. Understanding Deep Learning Techniques for Recognition of Human Emotions Using Facial Expressions: A Comprehensive Survey. *IEEE Trans. Instrum. Meas.* **2023**, *72*, 5006631. [\[CrossRef\]](#)
- Aleissae, A.A.; Kumar, A.; Anwer, R.M.; Khan, S.; Cholakkal, H.; Xia, G.-S.; Khan, F.S. Transformers in Remote Sensing: A Survey. *Remote Sens.* **2023**, *15*, 1860. [\[CrossRef\]](#)
- Janga, B.; Asamani, G.P.; Sun, Z.; Cristea, N. A Review of Practical AI for Remote Sensing in Earth Sciences. *Remote Sens.* **2023**, *15*, 4112. [\[CrossRef\]](#)
- Dalal, N.; Triggs, B. Histograms of Oriented Gradients for Human Detection. In Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR), San Diego, CA, USA, 20–25 June 2005; Volume 1, pp. 886–893.
- Burger, W.; Burge, M.J. Scale-Invariant Feature Transform (SIFT). In *Digital Image Processing: An Algorithmic Introduction*; Springer International Publishing: Cham, Switzerland, 2022; pp. 709–763.
- Mehrotra, R.; Namuduri, K.R.; Ranganathan, N. Gabor Filter-Based Edge Detection. *Pattern Recognit.* **1992**, *25*, 1479–1494. [\[CrossRef\]](#)
- Ito, K.; Xiong, K. Gaussian Filters for Nonlinear Filtering Problems. *IEEE Trans. Autom. Control.* **2000**, *45*, 910–927. [\[CrossRef\]](#)

20. Lan, S.; Li, J.; Hu, S.; Fan, H.; Pan, Z. A Neighbourhood Feature-Based Local Binary Pattern for Texture Classification. *Vis. Comput.* **2024**, *40*, 3385–3409. [[CrossRef](#)]
21. Li, Z.; Liu, F.; Yang, W.; Peng, S.; Zhou, J. A Survey of Convolutional Neural Networks: Analysis, Applications, and Prospects. *IEEE Trans. Neural Netw. Learn. Syst.* **2021**, *33*, 6999–7019. [[CrossRef](#)]
22. Lin, T.; Wang, Y.; Liu, X.; Qiu, X. A Survey of Transformers. *AI Open* **2022**, *3*, 111–132. [[CrossRef](#)]
23. Pinheiro Cinelli, L.; Araújo Marins, M.; Barros da Silva, E.A.; Lima Netto, S. Variational Autoencoder. In *Variational Methods for Machine Learning with Applications to Deep Networks*; Springer International Publishing: Cham, Switzerland, 2021; pp. 111–149.
24. Liu, H.; Li, I.; Liang, Y.; Sun, D.; Yang, Y.; Yang, H. Research on Deep Learning Model of Feature Extraction Based on Convolutional Neural Network. In Proceedings of the 2024 IEEE 2nd International Conference on Image Processing and Computer Applications (ICIPCA), Shenyang, China, 28–30 June 2024; pp. 810–816.
25. Ranjbarzadeh, R.; Tataei Sarshar, N.; Jafarzadeh Ghouschi, S.; Saleh Esfahani, M.; Parhizkar, M.; Pourasad, Y.; Bendeche, M. MRFE-CNN: Multi-Route Feature Extraction Model for Breast Tumor Segmentation in Mammograms Using a Convolutional Neural Network. *Ann. Oper. Res.* **2023**, *328*, 1021–1042. [[CrossRef](#)]
26. Barburiceanu, S.; Meza, S.; Orza, B.; Malutan, R.; Terebes, R. Convolutional Neural Networks for Texture Feature Extraction: Applications to Leaf Disease Classification in Precision Agriculture. *IEEE Access* **2021**, *9*, 160085–160103. [[CrossRef](#)]
27. Ma, P.; Ren, J.; Sun, G.; Zhao, H.; Jia, X.; Yan, Y.; Zabalza, J. Multiscale Superpixelwise Prophet Model for Noise-Robust Feature Extraction in Hyperspectral Images. *IEEE Trans. Geosci. Remote. Sens.* **2023**, *61*, 5508912. [[CrossRef](#)]
28. Zhang, Y.; Lu, Y.; Zhu, W.; Wei, X.; Wei, Z. Traffic Sign Detection Based on Multi-Scale Feature Extraction and Cascade Feature Fusion. *J. Supercomput.* **2023**, *79*, 2137–2152. [[CrossRef](#)]
29. Deng, C.; Wang, M.; Liu, L.; Liu, Y.; Jiang, Y. Extended Feature Pyramid Network for Small Object Detection. *IEEE Trans. Multimed.* **2021**, *24*, 1968–1979. [[CrossRef](#)]
30. Lu, S.; Ding, Y.; Liu, M.; Yin, Z.; Yin, L.; Zheng, W. Multiscale Feature Extraction and Fusion of Image and Text in VQA. *Int. J. Comput. Intell. Syst.* **2023**, *16*, 54. [[CrossRef](#)]
31. Lei, D.; Ran, G.; Zhang, L.; Li, W. A Spatiotemporal Fusion Method Based on Multiscale Feature Extraction and Spatial Channel Attention Mechanism. *Remote. Sens.* **2022**, *14*, 461. [[CrossRef](#)]
32. Zhang, G.; Zhang, H.; Yao, Y.; Shen, Q. Attention-Guided Feature Extraction and Multiscale Feature Fusion 3D ResNet for Automated Pulmonary Nodule Detection. *IEEE Access* **2022**, *10*, 61530–61543. [[CrossRef](#)]
33. Jia, L.; Huang, A.; He, X.; Li, Z.; Liang, J. A Residual Multi-Scale Feature Extraction Network with Hybrid Loss for Low-Dose Computed Tomography Image Denoising. *Signal Image Video Process.* **2024**, *18*, 1215–1226. [[CrossRef](#)]
34. Liu, J.; Fan, X.; Jiang, J.; Liu, R.; Luo, Z. Learning a Deep Multi-Scale Feature Ensemble and an Edge-Attention Guidance for Image Fusion. *IEEE Trans. Circuits Syst. Video Technol.* **2021**, *32*, 105–119. [[CrossRef](#)]
35. Rong, Y.; Jia, M.; Zhan, Y.; Zhou, L. SR-RDFAN-LOG: Arbitrary-Scale Logging Image Super-Resolution Reconstruction Based on Residual Dense Feature Aggregation. *Geoenergy Sci. Eng.* **2024**, *240*, 213042. [[CrossRef](#)]
36. Liu, M.; Raj, A.N.J.; Rajangam, V.; Ma, K.; Zhuang, Z.; Zhuang, S. Multiscale-Multichannel Feature Extraction and Classification through One-Dimensional Convolutional Neural Network for Speech Emotion Recognition. *Speech Commun.* **2024**, *156*, 103010. [[CrossRef](#)]
37. Wang, P.; Hu, Y.; Peng, S.; Zhou, L. EMANet: An Ancient Text Detection Method Based on Enhanced-EfficientNet and Multi-Dimensional Scale Fusion. *IEEE Internet Things J.* **2024**, *11*, 32105–32116. [[CrossRef](#)]
38. Xu, M.; Ma, Q.; Zhang, H.; Kong, D.; Zeng, T. MEF-UNet: An End-to-End Ultrasound Image Segmentation Algorithm Based on Multi-Scale Feature Extraction and Fusion. *Comput. Med. Imaging Graph.* **2024**, *114*, 102370. [[CrossRef](#)] [[PubMed](#)]
39. Zhong, Y.; Chen, T.; Zhong, D.; Liu, X. Vessel Segmentation in Fundus Images with Multi-Scale Feature Extraction and Disentangled Representation. *Appl. Sci.* **2024**, *14*, 5039. [[CrossRef](#)]
40. Zhang, C.; Zhang, Y.; Lin, H. Multi-Scale Feature Interaction Network for Remote Sensing Change Detection. *Remote Sens.* **2023**, *15*, 2880. [[CrossRef](#)]
41. Sheng, H.; Zhang, Y.; Zhang, W.; Wei, S.; Xu, M.; Muhammad, Y. ME-FCN: A Multi-Scale Feature-Enhanced Fully Convolutional Network for Building Footprint Extraction. *Remote Sens.* **2024**, *16*, 4305. [[CrossRef](#)]
42. Fernandez-Beltran, R.; Demir, B.; Pla, F.; Plaza, A. Unsupervised Remote Sensing Image Retrieval Using Probabilistic Latent Semantic Hashing. *IEEE Geosci. Remote. Sens. Lett.* **2020**, *18*, 256–260. [[CrossRef](#)]
43. Liu, Y.; Ye, Z.; Xi, Y.; Liu, H.; Li, W.; Bai, L. Multi-Scale and Multi-Direction Feature Extraction Network for Hyperspectral and LiDAR Classification. *IEEE J. Sel. Top. Appl. Earth Obs. Remote. Sens.* **2024**, *17*, 9961–9973. [[CrossRef](#)]
44. Goodfellow, I.; Bengio, Y.; Courville, A. *Deep Learning*; MIT Press: Cambridge, MA, USA, 2016.
45. Rudin, L.I.; Osher, S.; Fatemi, E. Nonlinear Total Variation Based Noise Removal Algorithms. *Phys. D Nonlinear Phenom.* **1992**, *60*, 259–268. [[CrossRef](#)]
46. Zhao, H.; Gallo, O.; Frosio, I.; Kautz, J. Loss Functions for Image Restoration with Neural Networks. *IEEE Trans. Comput. Imaging* **2016**, *3*, 47–57. [[CrossRef](#)]

47. Krizhevsky, A.; Nair, V.; Hinton, G. *CIFAR-10*; Technical Report; Canadian Institute for Advanced Research: Toronto, ON, Canada, 2009; pp. 1–4.
48. Armato, S.G., III; McLennan, G.; Bidaut, L.; McNitt-Gray, M.F.; Meyer, C.R.; Reeves, A.P.; Zhao, B.; Aberle, D.R.; Henschke, C.I.; Hoffman, E.A.; et al. Data From LIDC-IDRI [Data set]. *The Cancer Imaging Archive*, 2015. Available online: <https://www.cancerimagingarchive.net/collection/lidc-idri/> (accessed on 15 June 2025).
49. Kaggle. Finding Lungs in CT Data. 2017. Available online: <https://www.kaggle.com/datasets/kmader/finding-lungs-in-ct-data/data> (accessed on 15 June 2025).
50. Liu, H.; Brailsford, T.; Bull, L. ResNet18 Performance: Impact of Network Depth and Image Resolution on Image Classification. In Proceedings of the 2024 8th International Conference on Advances in Artificial Intelligence, London, UK, 17–19 October 2024; pp. 351–356.
51. Koonce, B. ResNet 50. In *Convolutional Neural Networks with Swift for TensorFlow: Image Recognition and Dataset Categorization*; Apress: Berkeley, CA, USA, 2021; pp. 63–72.
52. Dong, K.; Zhou, C.; Ruan, Y.; Li, Y. MobileNetV2 Model for Image Classification. In Proceedings of the 2020 2nd International Conference on Information Technology and Computer Application (ITCA), Guangzhou, China, 18–20 December 2020; pp. 476–480.
53. Ronneberger, O.; Fischer, P.; Brox, T. U-Net: Convolutional Networks for Biomedical Image Segmentation. In Proceedings of the 18th International Conference on Medical Image Computing and Computer-Assisted Intervention (MICCAI 2015), Munich, Germany, 5–9 October 2015; Springer: Cham, Switzerland, 2015; Volume 9351, pp. 234–241.

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.