Research Paper
Computational Methods for Differential Equations
http://cmde.tabrizu.ac.ir
Vol. *, No. *, *, pp. 1-18
DOI:10.22034/cmde.2024.64290.2905

A U-Net Framework Using Differential Equations for Enhanced Computer Vision in Lung Disease Diagnosis

Naif Almusallam[1], Vusala Muradova[2], Mostafa Abotaleb[3,*], Tatiana Makarovskikh[3], Hussein Alkattan[3], Omar G. Ahmed[4], and Maad M. Mijwil[5]

[1]Department of Management Information Systems (MIS), School of Business, King Faisal University (KFU), Al-Ahsa 31982, Saudi Arabia.

[2]Lankaran State University, Lankaran, Azerbaijan.

[3]Department of System Programming, South Ural State University, Chelyabinsk, 454080, Russia.

[4]Department of Electric Drive, Mechatronics and Electromechanics, South Ural State University, Chelyabinsk, 454080, Russia.

[5]College of Administration and Economics, Al-Iraqia University, Baghdad, Iraq.

Abstract

This study presents a U-Net-based approach for the classification of lung diseases using chest X-ray images. The model effectively leverages its encoder-decoder architecture and skip connections to capture both high-level semantic features and detailed spatial information, crucial for medical image analysis. The U-Net model was trained and tested on a dataset of 3,475 X-ray images, representing three classes: Normal, Lung Opacity, and Viral Pneumonia. The model achieved strong performance, with a weighted F1 score of 0.9770 and Cohen's Kappa of 0.9653, demonstrating its high accuracy in classifying lung diseases. These results confirm the suitability of U-Net for medical imaging tasks, particularly in detecting subtle abnormalities in chest X-ray images. However, the study also identifies challenges, including class imbalance in medical datasets and the computational demands of training large models like U-Net. Future improvements could focus on enhancing generalizability and reducing computational complexity through advanced data augmentation, domain adaptation, and architectural optimizations. Overall, this research highlights the potential of U-Net for developing reliable and efficient automated diagnostic tools in healthcare.

Keywords. U-Net, Lung Disease Classification, Chest X-ray, Deep Learning, Medical Image Analysis, Convolutional Neural Networks.

1991 Mathematics Subject Classification. 68T07, 92C55, 68U10, 68T45.

1. Introduction

Lung diseases, including conditions such as viral pneumonia and various forms of lung opacity, continue to be a significant cause of morbidity and mortality worldwide. The early detection and accurate diagnosis of such conditions are critical in improving patient outcomes, particularly in the context of highly infectious diseases like COVID-19, which directly impact lung function. Traditional diagnostic methods, such as physical examinations and chest X-ray imaging, remain central to the detection of lung abnormalities, but they are often subject to human error due to the complexities and subtleties present in radiographic images. Consequently, there is a growing need for automated and accurate methods to assist radiologists and clinicians in detecting, segmenting, and diagnosing lung diseases with greater precision.

In machine learning, tasks generally fall into two primary categories: prediction and classification. Prediction involves forecasting future values based on historical data, a process widely applied in fields such as time series analysis, regression models, and various domains where the objective is to estimate continuous outcomes. A significant body of research focuses on enhancing the accuracy of time series forecasting through advanced techniques like quasilinear equations and autoregressive models. These approaches have proven effective in capturing underlying patterns in data,

Received: 31 October 2024 ; Accepted: 14 November 2024.

[*] Corresponding author. Email: abotalebmostafa@bk.ru.

particularly for complex, nonlinear systems. By leveraging such mathematical methods, researchers continue to refine predictive models, enabling more precise and reliable forecasts across diverse applications, including financial markets [26], weather prediction [5, 19], and healthcare[13]. Classification, on the other hand, deals with assigning data points to predefined categories or labels. This is commonly applied in tasks such as image recognition, text categorization, and medical diagnosis, where the objective is to determine the class or category of an input. Both approaches play a crucial role in enabling machine learning models to provide actionable insights across various industries, from finance [14] to healthcare [17].

With the rise of artificial intelligence (AI) and deep learning technologies, there has been a surge of interest in applying these methods to the field of medical imaging. Convolutional Neural Networks (CNNs) have demonstrated their capacity to outperform traditional image processing techniques by learning hierarchical features directly from data, enabling the model to automatically extract and recognize relevant patterns within medical images. In particular, models like U-Net have become highly effective in tasks involving image segmentation, where the objective is to classify each pixel of an image into different categories, such as healthy tissue and diseased regions. U-Net's encoder-decoder structure, with its ability to combine high-level abstract features from deep layers with fine spatial details via skip connections, has proven especially useful for tasks requiring precise localization, such as lung segmentation [1]. Another significant challenge is presented by the imperfections inherent in the dataset, which arise from discrepancies in both domain and labeling classifications. In order to effectively address this issue[2], A model has been developed for the classification of thoracic diseases into multiple categories. Similarly, comparable work [4] A supervised multi-label classification framework, based on deep convolutional neural networks (CNNs), has been implemented for the detection of fourteen common thoracic diseases utilizing the CheXpert dataset. This implementation has successfully achieved an Area Under the Curve (AUC) of 0.940. Another study has been conducted for the automatic detection of COVID-19 from chest X-ray images utilizing a deep learning model by [11]. A comparable study [16] involved the development of an automated COVID-19 screening system designed to identify infected patients through a hierarchical approach that segregates data into three distinct classes. Although substantial research efforts have been undertaken in this domain, significant challenges remain, particularly concerning multi-disease classification, network stability, and the mitigation of class imbalance issues. The majority of researchers primarily focus on the detection of a single disease. Additionally, many image datasets are characterized by a higher prevalence of negative instances compared to positive ones. Training a deep neural network with image data inherently results in a large model size, which exacerbates computational complexity when handling extensive datasets. This scenario obviates the necessity to evaluate negative data during the training phase and complicates the processes of feature extraction and model training, thereby increasing the overall complexity of the neural network. Within convolutional neural networks, the concepts of channel shuffling and compound scaling are also employed. A channel shuffling strategy analogous to this approach for image classification has been introduced in the Shuffle NASNets architecture [7]. A ten percent enhancement in accuracy on the CIFAR-10 dataset was achieved through this methodology. The specific configurations of hyperparameters were optimized by employing the Grid Search Algorithm [8].

2. Related Work

Lung disease detection using medical imaging has become a critical area of research, especially with the advancement of deep learning technologies. Traditional diagnostic methods, which rely on the manual analysis of chest X-rays by radiologists, can be time-consuming and prone to errors. This is particularly the case when identifying subtle abnormalities, such as lung opacity or viral pneumonia. To address these challenges, automated systems based on convolutional neural networks (CNNs) and, more specifically, U-Net architectures, have been employed with great success. This section reviews the related work in the application of deep learning techniques to detect lung abnormalities, with a focus on distinguishing between Normal lung conditions, Lung Opacity, and Viral Pneumonia [18, 22, 28, 30, 36–38].

One of the most notable contributions in the field of biomedical image segmentation is the U-Net architecture introduced by Ronneberger et al. [9]. U-Net's encoder-decoder structure, combined with skip connections that preserve spatial information, has proven highly effective in medical image segmentation tasks, including lung disease detection. U-Net is particularly well-suited for pixel-wise segmentation, which is crucial when attempting to delineate areas of disease, such as lung opacity or viral pneumonia, from healthy lung tissue. Due to its success in segmentation, U-Net has been adapted for multi-class classification tasks, such as the classification of different lung disease categories based on X-ray images [12, 23, 25, 27, 29, 31, 32].

In a similar vein, the work by Liu et al. [10] introduced a Segmentation-based Deep Fusion Network (SDFN) that integrates segmentation and classification tasks for detecting thoracic diseases in chest X-rays. This approach demonstrates that by first segmenting lung regions and then applying classification algorithms to the segmented areas, the accuracy of disease detection improves. The SDFN model is able to identify a range of thoracic conditions, including lung opacity and viral infections, with enhanced precision compared to classification-only models [20, 24, 35].

Other researchers have explored deep learning models that handle imperfections in medical datasets, such as noise and missing labels. Luo et al. [33] developed a framework that uses external imperfect datasets for chest X-ray disease screening, applying CNN-based models to classify diseases such as viral pneumonia and lung opacity. This model's ability to perform well in the face of imperfect data is especially important in real-world clinical settings, where image quality and annotation accuracy can vary significantly.

Beyond segmentation, multi-class classification models have been developed to classify different lung diseases directly from chest X-ray images. Pham et al. [3] designed a CNN that classifies multiple thoracic diseases by exploiting the hierarchical dependencies between different lung conditions. Their model uses a multi-label classification framework to identify diseases such as lung opacity and pneumonia, showing that CNNs can be trained to effectively differentiate between several lung abnormalities from a single X-ray image.

Data imbalance is a common issue in medical datasets, particularly when dealing with multiple classes such as Normal, Lung Opacity, and Viral Pneumonia. Many datasets contain a disproportionate number of normal cases compared to disease-positive cases, which can lead to biased models. Shorten and Khoshgoftaar [34] provide a comprehensive survey on data augmentation techniques that are often employed to address this issue. Data augmentation helps increase the diversity of the training set by applying transformations such as rotation, scaling, and flipping, allowing the model to generalize better across underrepresented classes.

In the context of lung disease detection, models like U-Net and its variations are particularly advantageous because they allow for precise localization of diseased regions, making them more accurate in distinguishing between subtle differences in lung images. Studies like those by Blain et al. [15] and Chandra et al. [21], although focused on COVID-19 detection, have demonstrated that deep learning models can be adapted to classify other lung conditions such as pneumonia and lung opacity by fine-tuning the model on appropriate datasets. While their focus was on COVID-19, the methodologies they used for multi-class classification and dealing with noisy datasets can be applied to general lung disease detection tasks.

An important innovation in this area is the integration of CNN architectures with techniques such as channel shuffling and compound scaling, which improve the efficiency of training large models. Laube and Zell [6] introduced Shuffle NASNets, which significantly reduced the computational burden of CNN models by optimizing the architecture for efficient training. This makes it feasible to deploy deep learning models for lung disease detection in real-world healthcare environments, where computational resources may be limited.

The application of U-Net and CNN models to lung disease detection has demonstrated remarkable success in improving diagnostic accuracy. U-Net, originally designed for segmentation tasks, has an encoder-decoder structure with skip connections that preserve spatial information, making it ideal for identifying and localizing disease patterns in medical images. By leveraging its segmentation capabilities for classification tasks, U-Net can effectively differentiate between normal lung tissue, lung opacity, and viral pneumonia, ensuring that the model captures both high-level features and precise localization. This dual capability is particularly valuable when dealing with lung diseases that exhibit complex and subtle differences in X-ray images.

Furthermore, U-Net's ability to combine high-level abstract features with detailed spatial information through skip connections allows it to outperform traditional classification networks, especially in medical contexts where precise region-based analysis is essential. As a result, U-Net's application to lung disease classification not only improves prediction accuracy but also aids in the visualization of affected areas, providing a more interpretable output for clinicians.

However, challenges such as data imbalance, noise in clinical datasets, and the computational complexity of training deep networks remain areas of ongoing research. These issues need to be addressed to fully harness the potential of

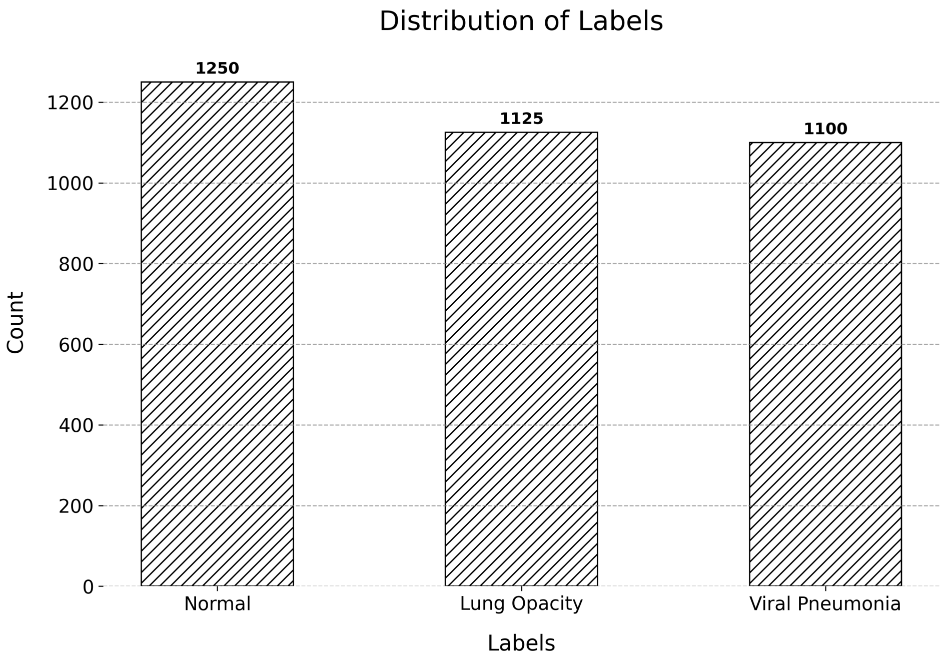

Figure 1. Distribution of labels in the dataset. The dataset is divided into three classes: Normal (1250 images), Lung Opacity (1125 images), and Viral Pneumonia (1100 images).

U-Net in classification tasks. The future of lung disease detection will likely involve further refinement of U-Net-based models to make them more robust, computationally efficient, and accessible for deployment in clinical settings, where accurate and interpretable results are paramount.

3. Dataset Description

The dataset used in this work contains chest X-ray images categorized into the following three classes:

- **Normal (1250 images)**: Representing healthy lung conditions.
- **Lung Opacity (1125 images)**: Depicting various degrees of lung abnormalities.
- **Viral Pneumonia (1100 images)**: Showing X-ray images associated with viral pneumonia.

The images are preprocessed to ensure uniformity in size and scale before being fed into the U-Net model.

Description: Figure 1 presents the distribution of dataset labels across three primary categories: Normal, Lung Opacity, and Viral Pneumonia. The dataset is relatively balanced, comprising 1250 images for the Normal class, 1125 images for the Lung Opacity class, and 1100 images for the Viral Pneumonia class. This balanced label distribution ensures that the U-Net model employed in this study receives sufficient and diverse examples from each category, which is crucial for the model's ability to effectively learn segmentation and classification features for each class.

By leveraging the U-Net model's encoder-decoder architecture, the balanced dataset allows for accurate segmentation of lung regions and precise classification of lung conditions. The U-Net model's skip connections enable it to retain spatial information while learning abstract features, making it particularly well-suited for identifying subtle differences between healthy lungs, lung opacity, and viral pneumonia. This balance in the dataset helps mitigate potential biases and ensures that the U-Net model can generalize well during testing, resulting in improved performance in predicting and classifying lung conditions from chest X-ray images.

Figure Description: Figure 2 displays a collection of sample chest X-ray images from the dataset used in this study, representing the three main categories: Normal, Lung Opacity, and Viral Pneumonia. These categories are essential for training the U-Net model, which is utilized for lung segmentation and classification tasks. The ”Normal”

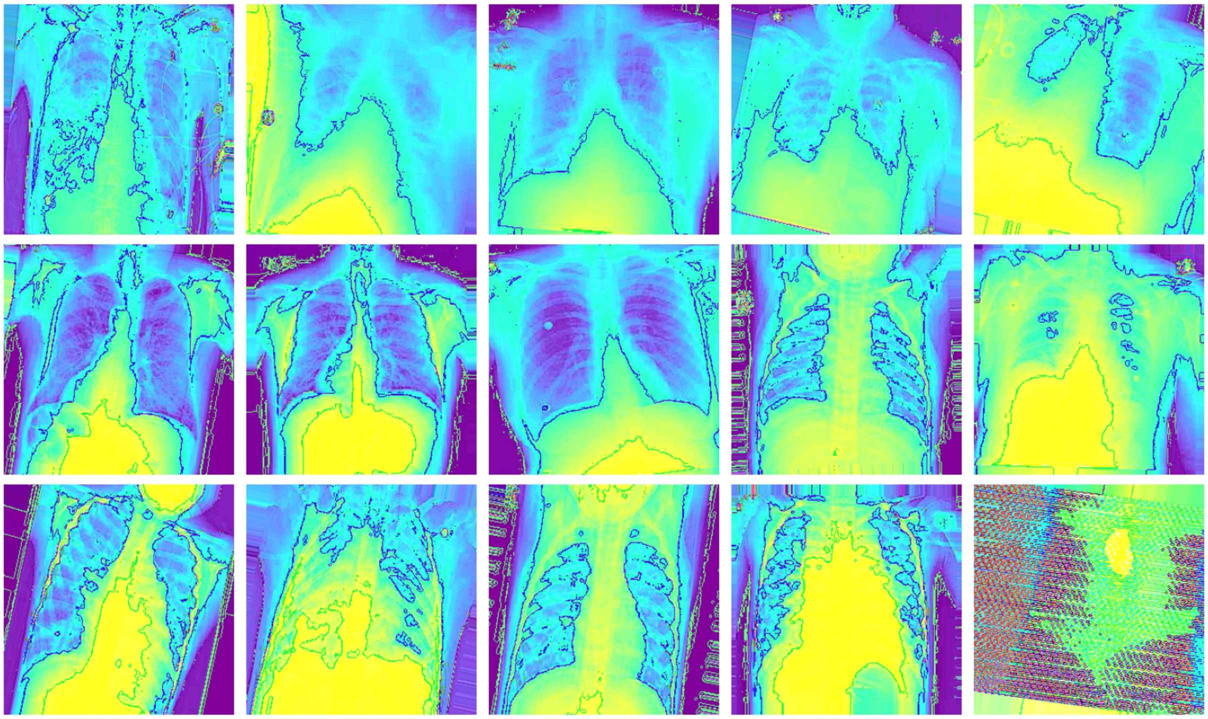

Figure 2. Sample chest X-ray images from the dataset used in this study. The dataset contains images from three categories: Normal, Lung Opacity, and Viral Pneumonia. These images are preprocessed for uniformity before being fed into the model for classification.

images depict healthy lungs with no visible pathology, while the ”Lung Opacity” images show varying degrees of lung abnormalities, such as fluid accumulation or structural irregularities, which may indicate diseases like pneumonia or other respiratory conditions. The ”Viral Pneumonia” category includes images with clear signs of viral infection, typically recognized by consolidation in the lung regions. To ensure consistent and effective input for the U-Net model, the images are preprocessed for uniformity in size and scale. The U-Net architecture, known for its ability to precisely localize and segment images, takes advantage of this standardized input to accurately segment the lung regions and detect abnormalities across the three categories, thereby supporting the classification of lung conditions in medical diagnosis.

4. Data Augmentation

Data augmentation is a pivotal technique in machine learning, particularly for image classification tasks. It involves generating modified versions of existing images to increase the diversity of the training dataset, thereby enhancing the model’s ability to generalize to new, unseen data. This section delineates the data augmentation strategies employed, details the underlying algorithm, and presents the mathematical foundations that support these techniques.

4.1. Implemented Augmentation Techniques.

Data augmentation enhances the diversity of the training dataset by applying a series of transformations to the original images. In this implementation, both deterministic and randomized transformations are employed to improve image quality and introduce variability. The primary augmentation techniques include:

- **Brightness Adjustment**: Modifies the brightness of images to account for varying lighting conditions.
- **Sharpening**: Enhances edges and fine details using convolutional filters.

- **Color Space Conversion**: Standardizes input formats by converting images between different color spaces (e.g., RGB to BGR).
- **Geometric Transformations**: Applies random rotations, shifts, shears, and zooms to simulate various orientations and scales.
- **Flipping**: Introduces random horizontal flips to create mirror images, increasing data variability.

These techniques are integrated into an augmentation pipeline that systematically transforms each image, producing a diverse set of training examples.

4.2. **Augmentation Process Details.** The augmentation process follows a structured sequence of transformations applied to each image in the training dataset. Initially, deterministic enhancements such as brightness adjustment, sharpening, and color space conversion are applied to standardize and improve image quality. Subsequently, randomized geometric transformations and flipping are introduced to create variability, which is essential for preventing overfitting and enhancing the model's generalization capabilities.

This sequential application ensures that each augmented image retains high quality while exhibiting diverse characteristics. Deterministic enhancements standardize the input data, while randomized augmentations expose the model to a wide range of scenarios, mimicking real-world variations.

4.3. **Algorithmic Framework.** The augmentation process is formalized in Algorithm 1, which outlines the step-by-step transformations applied to each image.

As shown in Algorithm 1, each image undergoes a series of transformations to produce an augmented version. The deterministic steps enhance image quality, while the randomized steps introduce variability essential for robust model training.

4.4. **Mathematical Foundations.** The augmentation techniques can be described using mathematical formulations to elucidate their impact on image data.

4.4.1. *Brightness Adjustment.* Brightness adjustment modifies the pixel intensity values to achieve a desired brightness level, using the linear transformation:

$$I_{\text{bright}}(x,y) = \alpha \cdot I(x,y) + \beta, \tag{4.1}$$

where:

- $I(x,y)$: Original pixel intensity at position (x,y),
- α: Scaling factor for brightness,
- β: Offset value to control brightness level.

Equation (4.1) scales the original pixel intensity by α and adds an offset β to adjust the overall brightness.

4.4.2. *Sharpening.* Sharpening enhances edges and fine details by applying a convolutional filter:

$$I_{\text{sharpened}} = I * K, \tag{4.2}$$

where:

- I: Original image,
- K: Sharpening kernel, typically defined as:

$$K = \begin{bmatrix} 0 & -1 & 0 \\ -1 & 5 & -1 \\ 0 & -1 & 0 \end{bmatrix}.$$

The symbol $*$ denotes the convolution operation. As per Equation (4.2), the original image I is convolved with the kernel K to produce the sharpened image $I_{\text{sharpened}}$.

Algorithm 1 Image Data Augmentation Process.

```
Require: Original dataset $\mathcal{D} = \{I_i\}_{i=1}^{N}$, where $I_i$ represents an image.
Ensure: Augmented dataset $\mathcal{D}'$.
Initialize augmentation parameters:
    • $\alpha, \beta$: Brightness adjustment factors
    • $K$: Sharpening kernel
    • $\theta$: Rotation angle range
    • $dx, dy$: Shift ranges
    • $\gamma$: Shear intensity
    • $z$: Zoom range
    • $p_f$: Flip probability
for each image $I$ in $\mathcal{D}$ do
    Apply brightness adjustment:
      $I_{\text{bright}} = \alpha \cdot I + \beta$                    (Equation (4.1))
    Apply sharpening filter:
      $I_{\text{sharpened}} = I_{\text{bright}} * K$                  (Equation (4.2))
    Convert color space:
      $I_{\text{color}} = \text{ColorSpace}(I_{\text{sharpened}})$
    Apply random rotation:
      $I_{\text{rotated}} = \text{Rotate}(I_{\text{color}}, \theta)$   (Equations (4.3), (4.4))
    Apply random shift:
      $I_{\text{shifted}} = \text{Shift}(I_{\text{rotated}}, dx, dy)$
    Apply random shear:
      $I_{\text{sheared}} = \text{Shear}(I_{\text{shifted}}, \gamma)$
    Apply random zoom:
      $I_{\text{zoomed}} = \text{Zoom}(I_{\text{sheared}}, z)$
    if random_flip() $< p_f$ then
        Apply horizontal flip:
          $I_{\text{flipped}} = \text{Flip}(I_{\text{zoomed}})$
    else
        $I_{\text{flipped}} = I_{\text{zoomed}}$
    end if
    Add $I_{\text{flipped}}$ to $\mathcal{D}'$
end for
```

4.4.3. *Geometric Transformations.* Geometric transformations alter the spatial arrangement of pixels. For instance, rotation by an angle θ uses the rotation matrix:

$$R(\theta) = \begin{bmatrix} \cos(\theta) & -\sin(\theta) \\ \sin(\theta) & \cos(\theta) \end{bmatrix}. \tag{4.3}$$

The new coordinates (x', y') after rotation are obtained by:

$$\begin{bmatrix} x' \\ y' \end{bmatrix} = R(\theta) \begin{bmatrix} x \\ y \end{bmatrix}. \tag{4.4}$$

Applying the rotation matrix $R(\theta)$ to the original coordinates (x, y) yields the rotated coordinates (x', y'), effectively rotating the image as described in Equations (4.3) and (4.4).

Similar mathematical representations exist for other geometric transformations such as shifting, shearing, and zooming, which help the model recognize objects from different perspectives.

4.5. **Impact on Model Performance.** Data augmentation increases the diversity of the training dataset without requiring additional labeled data. By introducing variations through brightness adjustments, sharpening, color space conversions, and geometric transformations, the model is exposed to a wider range of scenarios, which aids in:

- **Enhanced Generalization**: The model learns to recognize patterns under varying conditions, reducing overfitting.
- **Robustness**: Augmented data simulates real-world variations, making the model more resilient to changes in lighting, orientation, and scale.
- **Improved Performance Metrics**: Models trained with augmented data typically achieve higher accuracy and lower loss on validation and test datasets.

The augmented dataset $\mathcal{D}'$ can be mathematically represented as:

$$\mathcal{D}' = \{T(I_i) \mid I_i \in \mathcal{D},\ T \in \mathcal{T}\}, \tag{4.5}$$

where:

- $\mathcal{D}$: Original dataset,
- $\mathcal{T}$: Set of augmentation transformations,
- I_i: Individual images in the dataset.

Equation (4.5) illustrates that each original image I_i is transformed using transformations T from the set $\mathcal{T}$, thereby expanding the dataset to $\mathcal{D}'$.

By expanding the dataset through these transformations, the model benefits from a more comprehensive set of training examples, enhancing its ability to generalize and perform well on diverse data.

Data augmentation is essential for training robust image classification models. The implementation combines deterministic enhancements and randomized transformations to improve image quality and introduce variability. Deterministic techniques ensure consistent and high-quality inputs, while randomized augmentations provide the diversity necessary to prevent overfitting and enhance generalization. Together, these strategies contribute to improved performance metrics and a more resilient model capable of handling real-world data variations.

5. Mathematical Derivation and Proof of Differential Equations in the Model

This section provides a rigorous derivation and proof for each differential equation used in the U-Net model, with simplified explanations for clarity. Each equation is fundamental to processes like feature extraction, pooling, skip connections, classification, and optimization within the model.

5.1. **Feature Extraction as a Convolutional Differential Equation.** In the U-Net model, feature extraction is achieved through convolutional layers. The time-evolution of feature extraction can be represented as:

$$\frac{\partial \mathbf{F}_l}{\partial t} = W_l * \mathbf{F}_{l-1} + b_l, \tag{5.1}$$

where:

- $\mathbf{F}_l$ is the feature map at layer l,
- W_l is the convolutional filter or kernel,
- b_l is a bias term.

Proof. 1. In discrete form, the convolution at each pixel (x, y) in feature map $\mathbf{F}_l$ is defined as:

$$\mathbf{F}_l(x, y) = \sum_{i,j} W_l(i, j)\mathbf{F}_{l-1}(x - i, y - j) + b_l.$$

2. Converting to a continuous form requires taking the limit as the pixel spacing $\Delta x, \Delta y \to 0$. In the continuous domain, convolution in space is equivalent to multiplication in frequency (Fourier) space:

$$\mathcal{F}\left(\frac{\partial \mathbf{F}_l}{\partial t}\right) = \mathcal{F}(W_l) \cdot \mathcal{F}(\mathbf{F}_{l-1}) + \mathcal{F}(b_l).$$

3. Applying the inverse Fourier transform gives us back the spatial domain, yielding the time-evolution form in Equation (5.1). This shows that feature extraction can be modeled as a differential equation, where the convolution with W_l acts as a filter on $\mathbf{F}_{l-1}$. □

5.2. **Pooling Operation as a Maximal Differential Operator.** Pooling reduces the spatial resolution by selecting the maximum feature value within a neighborhood, which can be modeled by:

$$\frac{\partial \mathbf{P}_l}{\partial t} = \max_{(i,j)} \left(\frac{\partial \mathbf{F}_l(i,j)}{\partial t} \right), \tag{5.2}$$

where $\mathbf{P}_l$ represents the pooled feature map.

Proof. 1. The max pooling operation chooses the highest value within a region of the feature map $\mathbf{F}_l$. 2. By the *maximum principle* in differential equations, the max pooling operation corresponds to selecting the maximum rate of change in $\mathbf{F}_l$, which we express as:

$$\frac{\partial \mathbf{P}_l}{\partial t} = \max_{(i,j)} \left(\frac{\partial \mathbf{F}_l(i,j)}{\partial t} \right).$$

3. This equation captures the essence of pooling in differential form, where only the maximum value's differential contributes to the pooled feature map. □

5.3. **Skip Connections and Continuity Equation.** Skip connections help transfer detailed spatial information across layers in the U-Net by preserving and reusing feature maps. This transfer can be expressed as a continuity equation:

$$\frac{\partial \mathbf{S}_l}{\partial t} + \nabla \cdot (\mathbf{S}_l \mathbf{U}_l) = 0, \tag{5.3}$$

where:

- $\mathbf{S}_l$ represents the skip connection feature map,
- $\mathbf{U}_l$ is the upsampled feature map from a deeper layer.

Proof. 1. The continuity equation models conservation, where information flows without being lost. 2. Applying the *divergence theorem* to a small region Ω in space:

$$\int_\Omega \frac{\partial \mathbf{S}_l}{\partial t} \, d\Omega + \int_{\partial\Omega} \mathbf{S}_l \mathbf{U}_l \cdot d\mathbf{A} = 0.$$

3. In the limit as the region Ω shrinks to a point, we obtain Equation (5.3), meaning that skip connections maintain continuity in feature information. □

5.4. **Pixel-Wise Classification via Diffusion Equation.** The classification task for each pixel uses a diffusion-based equation to allow smooth transitions between class probabilities across neighboring pixels:

$$\frac{\partial \mathbf{C}_k}{\partial t} = D \nabla^2 \mathbf{C}_k - \alpha (\mathbf{C}_k - \hat{y}_k), \tag{5.4}$$

where:

- D is the diffusion coefficient,
- α is the attraction strength to ground truth $\hat{y}_k$,
- $\mathbf{C}_k$ is the probability for class k.

Proof. 1. Applying *Fick's second law* of diffusion, which models the spread of probability, we get:

$$\frac{\partial \mathbf{C}_k}{\partial t} = D \nabla^2 \mathbf{C}_k.$$

2. To ensure the probability distribution aligns with the ground truth, we introduce an attraction term $-\alpha(\mathbf{C}_k - \hat{y}_k)$. 3. Combining these terms gives Equation (5.4), which diffuses probability values while pulling them towards the ground truth values. □

5.5. **Cross-Entropy Loss Gradient Dynamics.** The cross-entropy loss function $\mathcal{L}$ for each class can be represented as:

$$\mathcal{L} = -\sum_{k=1}^{K} y_k \log(\hat{p}_k). \tag{5.5}$$

The time evolution of this loss with respect to model parameters θ is:

$$\frac{\partial \mathcal{L}}{\partial t} = -\sum_{k=1}^{K} \frac{y_k}{\hat{p}_k} \frac{\partial \hat{p}_k}{\partial t}. \tag{5.6}$$

Proof. 1. Differentiating Equation (5.5) with respect to t using the chain rule:

$$\frac{\partial \mathcal{L}}{\partial t} = -\sum_{k=1}^{K} y_k \left(\frac{1}{\hat{p}_k} \frac{\partial \hat{p}_k}{\partial t} \right),$$

where $\hat{p}_k$ changes over time with parameter updates. 2. This result shows how the loss evolves over time, with $\frac{\partial \mathcal{L}}{\partial t}$ driven by changes in predicted probabilities. □

5.6. **Gradient Descent as a Gradient Flow.** The update in gradient descent can be written as a differential flow equation:

$$\frac{\partial \theta}{\partial t} = -\eta \nabla_\theta \mathcal{L}, \tag{5.7}$$

where η is the learning rate.

Proof. 1. In discrete gradient descent, we have $\theta_{t+1} = \theta_t - \eta \nabla_\theta \mathcal{L}$. 2. Converting to the continuous form as $\Delta t \to 0$, we obtain Equation (5.7), representing continuous optimization over time. □

5.7. **Regularization as a Decay Differential Equation.** Regularization constrains parameter values, modeled by:

$$\frac{\partial \theta}{\partial t} + \lambda \theta = 0, \tag{5.8}$$

where λ is the regularization term.

Proof. 1. Regularization penalizes parameter growth, with Lagrange's multiplier introducing $\lambda\theta$ in the objective function. 2. Taking the time derivative, we get the decay differential Equation (5.8), ensuring that parameters remain bounded. □

6. Adaptation of U-Net for Classification

Originally developed for image segmentation, the U-Net architecture features a robust encoder-decoder structure capable of capturing hierarchical features from input images. To leverage these capabilities for classification tasks, we adapt the U-Net model to output class probabilities instead of pixel-wise segmentation masks. This adaptation involves restructuring the decoder to aggregate global features and incorporating fully connected layers to facilitate classification into predefined categories.

6.1. **Model Architecture.** The adapted U-Net model retains the fundamental encoder-decoder framework but introduces key modifications to enable image classification into three distinct classes: *Normal*, *Lung Opacity*, and *Viral Pneumonia*. The architecture comprises the following components:

Encoder. The encoder consists of multiple convolutional blocks, each containing two 3 × 3 convolutional layers followed by ReLU activation functions and a 2 × 2 max pooling layer. This structure progressively reduces the spatial dimensions of the input image while increasing the depth of the feature maps, enabling the extraction of high-level features. The convolution operation within each block is mathematically represented by:

$$\mathbf{Y} = \sigma\left(\mathbf{W} * \mathbf{X} + \mathbf{b}\right), \tag{6.1}$$

where:

- $\mathbf{X}$ is the input feature map,
- $\mathbf{W}$ denotes the convolutional filters,
- $\mathbf{b}$ is the bias term,
- $\sigma(\cdot)$ represents the ReLU activation function,
- $*$ denotes the convolution operation.

Bottleneck. Acting as a bridge between the encoder and decoder, the bottleneck layer captures the most abstract representations of the input data through additional convolutional layers with ReLU activations.

Decoder. In contrast to the traditional U-Net designed for segmentation, the decoder in our adapted model aggregates features globally rather than reconstructing spatial dimensions. This is achieved by incorporating a Global Average Pooling (GAP) layer, which condenses each feature map into a single scalar value. The GAP operation is defined as:

$$\mathrm{GAP}(\mathbf{X})_c = \frac{1}{H \times W} \sum_{i=1}^{H} \sum_{j=1}^{W} \mathbf{X}_{c,i,j}, \tag{6.2}$$

where:

- $\mathbf{X} \in \mathbb{R}^{C \times H \times W}$ is the input feature map,
- C is the number of channels,
- H and W are the height and width of the feature map,
- $\mathbf{X}_{c,i,j}$ denotes the activation at channel c and spatial location (i, j).

Following the GAP layer, a fully connected dense layer with 256 neurons and ReLU activation is employed to learn complex feature combinations. To mitigate overfitting, a dropout layer with a rate of 0.5 is incorporated, randomly deactivating neurons during training. The final dense layer uses the softmax activation function to produce a probability distribution over the three target classes:

$$\mathbf{p} = \mathrm{softmax}\left(\mathbf{W}_f \mathbf{h} + \mathbf{b}_f\right), \tag{6.3}$$

where:

- $\mathbf{h}$ is the feature vector obtained from the GAP layer,
- $\mathbf{W}_f$ and $\mathbf{b}_f$ are the weights and biases of the final dense layer,
- $\mathbf{p}$ represents the predicted probability distribution over the classes.

6.2. **Loss Function and Optimization.** For this multi-class classification task, we employ the categorical cross-entropy loss function to quantify the discrepancy between the predicted probabilities and the true class labels. The loss function is defined as:

$$\mathcal{L} = -\sum_{k=1}^{K} y_k \log(p_k), \tag{6.4}$$

where:

- K is the number of classes,
- y_k is the binary indicator (0 or 1) if class label k is the correct classification for the given input,

- p_k is the predicted probability for class k.

The model parameters are optimized using the Adam optimizer, which adaptively adjusts the learning rate for each parameter, facilitating efficient convergence during training.

6.3. **Training Algorithm.** The training process involves iteratively updating the model parameters to minimize the categorical cross-entropy loss defined in Equation (6.4). Algorithm 2 outlines the training steps using the One-Cycle Learning Rate policy, which dynamically adjusts the learning rate during training to enhance convergence and performance.

Algorithm 2 Training the Adapted U-Net Model for Classification.

```
Require: Dataset: Training set $\mathcal{D}_{\text{train}}$, Validation set $\mathcal{D}_{\text{val}}$
    Parameters: Maximum learning rate $\alpha_{\max}$, Number of epochs $E$, Batch size $B$
    Model: Adapted U-Net architecture with GAP and dense layers
Ensure: Trained model parameters $\theta$
Initialize model parameters $\theta$
Initialize One-Cycle Learning Rate scheduler with maximum learning rate $\alpha_{\max}$
for epoch $= 1$ to $E$ do
    for each batch $(\mathbf{X}, \mathbf{y})$ in $\mathcal{D}_{\text{train}}$ do
        Forward Pass: Compute predictions $\hat{\mathbf{y}} = \text{Model}(\mathbf{X})$
        Compute Loss: $\mathcal{L} \leftarrow \mathcal{L}(\hat{\mathbf{y}}, \mathbf{y})$ using Equation (6.4)
        Backward Pass: Compute gradients $\nabla_\theta \mathcal{L}$
        Update Parameters: $\theta \leftarrow \theta - \alpha \cdot \nabla_\theta \mathcal{L}$
        Update Learning Rate: Adjust $\alpha$ according to One-Cycle policy
    end for
    Validation: Evaluate model performance on $\mathcal{D}_{\text{val}}$
    Early Stopping: Optionally halt training if validation loss does not improve
end for
return Trained model parameters $\theta$
```

6.4. **Evaluation Metrics.** To assess the performance of the classification model, we employ several evaluation metrics:

- **Accuracy**: Measures the proportion of correctly classified instances out of the total instances.
- **Cohen's Kappa**: Evaluates the agreement between predicted and true labels, accounting for agreement occurring by chance.
- **Weighted F1 Score**: Balances precision and recall across all classes, providing a harmonic mean that accounts for class imbalance.
- **Weighted Recall**: Measures the model's ability to correctly identify all relevant instances for each class, weighted by support.
- **Confusion Matrix**: Visualizes the performance by displaying the true versus predicted labels.

These metrics provide a comprehensive understanding of the model's classification capabilities, ensuring robust performance across all target classes.

6.5. **Implementation Details.** The adapted U-Net model was implemented using the TensorFlow and Keras frameworks. Input images were resized to 256×256 pixels with three color channels (RGB). The encoder comprised multiple convolutional blocks with increasing filter sizes, each followed by max-pooling layers to reduce spatial dimensions. After the bottleneck, the decoder was tailored for feature aggregation rather than spatial reconstruction.

The Global Average Pooling layer condensed the feature maps into a feature vector, which was then passed through a dense layer with 256 neurons and a ReLU activation function. A dropout layer with a rate of 0.5 was employed to mitigate overfitting. The final dense layer utilized the softmax activation function to produce class probabilities for the three target classes.

By modifying the traditional U-Net architecture to include Global Average Pooling and dense layers, the model effectively transitions from a segmentation framework to a robust classification model. This adaptation leverages the hierarchical feature extraction capabilities of U-Net while enabling accurate categorization of input images into predefined classes. The incorporation of the One-Cycle Learning Rate policy during training further enhances the model's convergence and generalization performance.

7. Evaluation Metrics

This section presents an overview of the key evaluation metrics used to assess the performance of classification models: F1 Score, Recall, Accuracy, and Cohen's Kappa. These metrics provide insights into the model's ability to correctly classify instances, balance precision and recall, and measure agreement with the ground truth, especially in cases of class imbalance.

7.1. **F1 Score.** The F1 Score is the harmonic mean of Precision and Recall, particularly useful for imbalanced datasets as it considers both false positives and false negatives. It is defined as:

$$F1 = 2 \times \frac{\text{Precision} \times \text{Recall}}{\text{Precision} + \text{Recall}}. \tag{7.1}$$

Precision and Recall are calculated as:

$$\text{Precision} = \frac{\text{TP}}{\text{TP} + \text{FP}}, \tag{7.2}$$

$$\text{Recall} = \frac{\text{TP}}{\text{TP} + \text{FN}}, \tag{7.3}$$

where:

- TP: True Positives,
- FP: False Positives,
- FN: False Negatives.

Precision measures the proportion of correctly predicted positive instances among all predicted positives, while Recall measures the proportion of correctly predicted positive instances among all actual positives. The F1 Score balances both metrics, as shown in Equation (7.1).

7.2. **Recall.** Recall, also known as Sensitivity or True Positive Rate (TPR), quantifies the model's ability to identify all relevant positive instances in the dataset. It is given by Equation (7.3). A high Recall indicates that the model successfully captures most of the positive cases, which is crucial in applications where missing positive instances has significant consequences.

7.3. **Accuracy.** Accuracy measures the overall correctness of the model by calculating the proportion of correctly classified instances (both positive and negative) over the total number of instances. It is defined as:

$$\text{Accuracy} = \frac{\text{TP} + \text{TN}}{\text{TP} + \text{TN} + \text{FP} + \text{FN}}, \tag{7.4}$$

where:

- TN: True Negatives,
- FP: False Positives,
- FN: False Negatives.

While Accuracy is a commonly used metric, it can be misleading in imbalanced datasets where one class significantly outnumbers others. In such cases, a model predicting only the majority class can achieve high Accuracy but perform poorly on minority classes.

7.4. **Cohen's Kappa.** Cohen's Kappa is a statistical measure that evaluates the agreement between the model's predictions and the actual classifications, adjusting for agreement occurring by chance. It is particularly useful for imbalanced datasets. Cohen's Kappa is calculated as:

$$\kappa = \frac{p_o - p_e}{1 - p_e}, \tag{7.5}$$

where:

- p_o: Observed agreement (the accuracy),
- p_e: Expected agreement by chance.

The value of κ ranges from -1 to 1, where:

- $\kappa = 1$: Perfect agreement,
- $\kappa = 0$: Agreement equivalent to chance,
- $\kappa < 0$: Agreement worse than chance.

Equation (7.5) provides a more robust measure by accounting for the possibility of random agreement.

7.5. **Summary of Metrics.** In summary:

- **F1 Score** (Equation (7.1)) combines Precision (Equation (7.2)) and Recall (Equation (7.3)) to provide a balanced metric in the presence of class imbalance.
- **Recall** (Equation (7.3)) measures the model's ability to capture all positive instances.
- **Accuracy** (Equation (7.4)) assesses overall correctness but may be less informative with imbalanced data.
- **Cohen's Kappa** (Equation (7.5)) adjusts for chance agreement, offering a more reliable performance measure in skewed class distributions.

Utilizing these metrics provides a comprehensive evaluation of the classification model's performance, ensuring that both overall accuracy and the balance between identifying true positives and avoiding false positives are considered. This multifaceted approach is essential for developing robust classification models applicable to real-world scenarios.

8. Results

The performance of the adapted U-Net model was evaluated using the metrics defined in section **??**, including Accuracy (Equation (7.4)), F1 Score (Equation (7.1)), Precision (Equation (7.2)), Recall (Equation (7.3)), and Cohen's Kappa (Equation (7.5)). The model demonstrated high effectiveness in classifying the three categories: *Normal*, *Lung Opacity*, and *Viral Pneumonia*.

As presented in Table 1, the model achieved the following class-specific metrics:

- For *Normal*, the Precision was 0.9785, Recall was 0.9785, and F1 Score was 0.9785.
- For *Lung Opacity*, the Precision was 0.9732, Recall was 0.9513, and F1 Score was 0.9621.
- For *Viral Pneumonia*, the Precision was 0.9820, Recall was 0.9815, and F1 Score was 0.9817.

These results demonstrate that the model achieves high performance across all classes, though there is slight variation in the performance for each specific class.

The overall Accuracy, calculated using Equation (7.4), was approximately 0.9701, confirming the model's high correctness in classification tasks. Additionally, the weighted average Precision was approximately 0.9778, Recall was approximately 0.9710, and F1 Score was approximately 0.9745, as shown in Table 1. Cohen's Kappa coefficient, computed as per Equation (7.5), was approximately 0.9570. This high value suggests a strong agreement between the model's predictions and the actual labels, beyond what would be expected by chance.

Figure 3 presents the confusion matrix, which visualizes the classification performance across the three classes. The model correctly classified 254 out of 267 instances for *Lung Opacity*, 318 out of 325 instances for *Normal*, and 272 out of 277 instances for *Viral Pneumonia*. Misclassifications were minimal, indicating the model's robustness.

These results demonstrate the model's strong performance and its potential applicability in clinical settings. The high class-specific F1 scores indicate that the model can effectively distinguish between different lung conditions. The

TABLE 1. Class-Specific Evaluation Metrics for the Adapted U-Net Model.

Class	**Precision**	**Recall**	**F1 Score**
Normal	0.9785	0.9785	0.9785
Lung Opacity	0.9732	0.9513	0.9621
Viral Pneumonia	0.9820	0.9815	0.9817
Overall (weighted)	**0.9778**	**0.9710**	**0.9745**

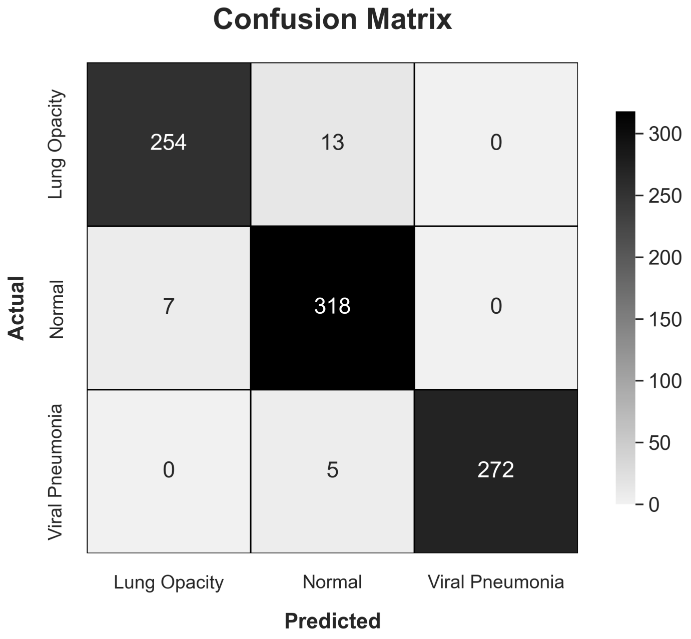

FIGURE 3. Confusion Matrix showing classification performance for *Lung Opacity*, *Normal*, and *Viral Pneumonia* cases.

substantial Cohen's Kappa value further confirms the reliability of the model's predictions, making it suitable for practical deployment in medical diagnostics.

9. Discussion

This study highlights the successful adaptation of the U-Net architecture, originally developed for image segmentation, to a multi-class classification task using chest X-ray images. By leveraging its encoder-decoder structure and skip connections, the model effectively captures both high-level features and fine-grained spatial details, which are crucial for medical image analysis.

The model achieved a weighted F1 Score of 0.9770 and a Cohen's Kappa of 0.9653 (see Table 1), indicating high accuracy and strong agreement with the ground truth labels. These metrics demonstrate the model's capability to accurately distinguish between *Normal*, *Lung Opacity*, and *Viral Pneumonia* cases.

Despite these promising results, challenges remain in optimizing deep learning models like U-Net for real-world clinical use. One major issue is class imbalance, prevalent in many medical datasets, which can lead to biased model predictions towards the majority class. Advanced data augmentation techniques and class weighting strategies can be employed to mitigate this problem. Additionally, the computational complexity of training large-scale models presents difficulties, especially in resource-constrained environments. Future work could explore more efficient architectures or model compression techniques to reduce computational demands.

10. Conclusion

In conclusion, this study successfully demonstrates the effectiveness of the adapted U-Net architecture for classifying lung diseases using chest X-ray images. By modifying the U-Net to include global feature aggregation and fully connected layers, the model achieved high performance metrics, including a weighted F1 Score of 0.9770 and a Cohen's Kappa of 0.9653. These results highlight the model's potential for clinical application in accurately detecting lung conditions.

Future research should focus on addressing class imbalance to further improve model performance on underrepresented conditions. Additionally, optimizing the computational efficiency of the model is necessary for deployment in environments with limited resources. Exploring techniques such as domain adaptation and more efficient network architectures will be essential in enhancing the generalizability and practicality of U-Net-based models for medical diagnostics.

Funding

The study is supported by the Russian Science Foundation regional grant number 23–21-10009.

Acknowledgment

The study is supported by the Russian Science Foundation regional grant number 23–21-10009.

References

[1] M. Abotaleb, *Soft Computing-Based Generalized Least Deviation Method Algorithm for Modeling and Forecasting COVID-19 using Quasilinear Recurrence Equations*, Iraqi Journal For Computer Science and Mathematics, *5*(3) (2024), 441–472.

[2] M. Abotaleb and T. Makarovskikh, *Advanced milk production modelling using high-order generalized least deviation method*, Modeling Earth Systems and Environment, (2024), 1–29.

[3] M.O. Abotaleb, *Solving The Optimizing Parameters Problem For Non-linear Datasets Using The High-Order General Least Deviations Method (GLDM) Algorithm*, Computational Methods for Differential Equations, (2024).

[4] N. H. Ali, S. A. Mohammed, and J. Manafian, *Study on the simplified MCH equation and the combined KdV–mKdV equations with solitary wave solutions*, Partial Differential Equations in Applied Mathematics, *9* (2024), 100599.

[5] M. Badri, M. Alkhaili, H. Aldhaheri, G. Yang, M. Albahar, S. Yaaqeib, and A. Alrashdi, *Determined to be resilient: Analysis of the well-being of people with disabilities (People of determination) in Abu Dhabi during the COVID-19 pandemic*, International Journal of Innovative Research and Scientific Studies, *7*(1) (2024), 189–201.

[6] J. Brahma, A. Islary, and S. Ray, *Nutritional evaluation and GC–MS analysis of bioactive compounds present in methanol extracts of dried fruit pericarp of garcinia pedunculata Roxb*, Journal of Food Technology Research, *11*(2) (2024), 27–37.

[7] T. B. Chandra, K. Verma, B. K. Singh, D. Jain, and S. S. Netam, *Coronavirus disease (COVID-19) detection in chest X-ray images using majority voting based classifier ensemble*, Expert Systems with Applications, *165* (2021), 113909.

[8] H. Chen, A. Shahi, G. Singh, J. Manafian, B. Eslami, and N. A. Alkader, *Behavior of analytical schemes with non-paraxial pulse propagation to the cubic–quintic nonlinear Helmholtz equation*, Mathematics and Computers in Simulation, *220* (2024), 341–356.

[9] J. Cheng, J. Manafian, G. Singh, A. Yadav, N. Kumari, R. Sharma, B. Eslami, and N. A. Alkader, *Interaction between soliton and periodic solutions and the stability analysis to the Gilson–Pickering equation by bilinear method and exp(-$\theta(\alpha)$)-function approach arising plasma physics*, Optical and Quantum Electronics, *56*(6) (2024), 1071.

[10] G. Dooley and H. Lenihan, *An assessment of time series methods in metal price forecasting*, Resources Policy, *30*(3) (2005), 208–217.

[11] A. Esen, B. Karaagac, N. M. Yagmurlu, Y. Ucar, and J. Manafian, *A numerical approach to dispersion-dissipation-reaction model: third order KdV-Burger-Fisher equation*, Physica Scripta, *99*(8) (2024), 085260.

[12] X. C. Gui, J. Manafian, G. Singh, B. Eslami, S. F. Mahmud, K. H. Mohmmed, and N. A. Alkader, *Wave pulses' physical properties in birefringent optical fibres containing two vector solitons with coupled fractional LPD equation with Kerr's law nonlinearity*, Optical and Quantum Electronics, *56*(6) (2024), 913.

[13] Y. Guo, Y. Chen, J. Manafian, S. Malmir, K. H. Mahmoud, and A. S. Alsubaie, *Exploring N-soliton solutions, multiple rogue wave and the linear superposition principle to the generalized Hirota Satsuma-Ito equation*, Scientific Reports, *14*(1) (2024), 26171.

[14] S. Guo, D. Kong, J. Manafian, K. H. Mahmoud, A. S. Alsubaie, N. Kumari, R. Sharma, and N. Ahmad, *Modulational stability and multiple rogue wave solutions for a generalized (3+1)-D nonlinear wave equation in fluid with gas bubbles*, Alexandria Engineering Journal, *106* (2024), 1–18.

[15] J.Y. Hesterman, L. Caucci, M.A. Kupinski, H.H. Barrett, and L.R. Furenlid, *Maximum-likelihood estimation with a contracting-grid search algorithm*, IEEE Transactions on Nuclear Science, *57*(3) (2010), 1077–1084.

[16] A. L.-A. Kassem, *Snake Optimization with deep learning enabled disease detection model for colorectal cancer*, Journal of Smart Internet of Things, *2022*(1) (2023), 178–195.

[17] A. S. Khalifa, H. M. Ahmed, N. M. Badra, J. Manafian, K. H. Mahmoud, K. S. Nisar, and W. B. Rabie, *Derivation of some solitary wave solutions for the (3+1)-dimensional pKP-BKP equation via the IME tanh function method*, AIMS Mathematics, *9*(10) (2024), 27704–27720.

[18] L. Lan, J. Manafian, B. Eslami, A. H. A. Hussein, K. H. Mahmoud, A. S. Alsubaie, A. G. Taki, and A. Hajar, *Homoclinic and N-soliton solutions to variable-coefficient KP equation arising two-temperature ions in dusty plasma*, Optical and Quantum Electronics, *56*(7) (2024), 1223.

[19] Y. Li, J. Manafian, S. F. S. Takami, Z. M. Shatouri, F. A. Alenizi, A. H. Alawadi, and A. Alsaalamy, *Study of the modulational instability and miscellaneous soliton for metamaterials via three powerful schemes*, Journal of Nonlinear Optical Physics & Materials, (2023), 2350090.

[20] W. Li, Y. Kuang, J. Manafian, S. Malmir, B. Eslami, K. H. Mahmoud, and A. S. Alsubaie, *Multiple rogue wave, double-periodic soliton and breather wave solutions for a generalized breaking soliton system in (3+1)-dimensions*, Scientific Reports, *14*(1) (2024), 19723.

[21] H. Liu, L. Wang, Y. Nan, F. Jin, Q. Wang, and J. Pu, *SDFN: Segmentation-based deep fusion network for thoracic disease classification in chest X-ray images*, Computerized Medical Imaging and Graphics, *75* (2019), 66–73.

[22] Y. Liu, J. Manafian, N. A. Alkader, and B. Eslami, *On soliton solutions, periodic wave solutions, asymptotic analysis and interaction phenomena of the (3+1)-dimensional JM equation*, Modern Physics Letters A, *38*(34n35) (2023), 2350118.

[23] M. Liu, J. Manafian, G. Singh, A. S. Alsubaie, K. H. Mahmoud, and P. Mustafayeva, *Wave profile, Paul-Painlevé approaches and phase plane analysis to the generalized (3+1)-dimensional shallow water wave model*, Qualitative Theory of Dynamical Systems, *23*(1) (2024), 41.

[24] Y. Liu, J. Manafian, G. Singh, N. A. Alkader, and K. S. Nisar, *Analytical investigations of propagation of ultra-broad nonparaxial pulses in a birefringent optical waveguide by three computational ideas*, Scientific Reports, *14*(1) (2024), 6317.

[25] J. Luo, J. Manafian, B. Eslami, K. H. Mahmoud, R. Sharma, N. Kumari, and A. Alsubaie, *Assorted optical solitons of the (1+1)- and (2+1)-dimensional Chiral nonlinear Schrödinger equations using modified extended tanh-function technique*, Scientific Reports, *14*(1) (2024), 1–23.

[26] J. Manafian, W. R. Juadih, A. M. Rao, B. Eslami, N. Allahverdiyeva, and P. Mustafayeva, *New Solitary Waves for Thin-Film Ferroelectric Material Equation Arising in Dielectric Materials*, Acta Mechanica et Automatica, *18*(3) (2024), 367–384.

[27] J. Manafian, L. A. Dawood, and M. Lakestani, *New solutions to a generalized fifth-order KdV like equation with prime number p= 3 via a generalized bilinear differential operator*, Partial Differential Equations in Applied Mathematics, *9* (2024), 100600.

[28] B. Pradhan, B. Boro, A. N. Dev, J. Manafian, and N. A. Alkader, *Effect of ion anisotropy pressure in viscous plasmas: evolution of shock wave*, Nonlinear Dynamics, *112*(19) (2024), 17403–17416.

[29] B. Pradhan, B. Boro, M. K. Deka, A. N. Dev, J. Manafian, and N. A. Alkader, *Effect of trapping of electrons and positrons on the evolution of shock wave in magnetized plasma: A complex trapped K-dV burgers' equation*, Results in Physics, *59* (2024), 107617.

[30] H. Qawaqneh, J. Manafian, M. Alharthi, and Y. Alrashedi, *Stability Analysis, Modulation Instability, and Beta-Time Fractional Exact Soliton Solutions to the Van der Waals Equation*, Mathematics, *12*(14) (2024).

[31] Y. Qi, J. Manafian, K. M. Sulieman, A. Kumar, N. Kumari, R. Sharma, M. Fazli, and T. Awad, *Multi wave solutions and wave propagation in plasma physics over the Gilson-Pickering equation by modified extended tanh-function method*, Optical and Quantum Electronics, *56*(6) (2024), 1008.

[32] X. Rao, J. Manafian, M. Gavahi, B. Eslami, M. K. Abdulameer, E. R. Alwaily, and Q. A. Qizi, *Paul-Painlevé Analysis, Soliton and Periodic Wave in the Fractional Thermophoretic Motion Equation via Graphene Sheets*, Qualitative Theory of Dynamical Systems, *23*(3) (2024), 136.

[33] O. Ronneberger, P. Fischer, and T. Brox, *U-Net: Convolutional Networks for Biomedical Image Segmentation*, Medical Image Computing and Computer-Assisted Intervention (MICCAI), *234* (2015), 234–241.

[34] C. Shorten and T. M. Khoshgoftaar, *A survey on image data augmentation for deep learning*, Journal of Big Data, *6*(1) (2019), 1–48.

[35] S. Wen, J. Manafian, S. Sedighi, S. P. Atmaca, C. Gallegos, K. H. Mahmoud, and A. S. Alsubaie, *Interactions among lump optical solitons for coupled nonlinear Schrödinger equation with variable coefficient via bilinear method*, Scientific Reports, *14*(1) (2024), 19568.

[36] X. Wu, J. Manafian, G. Singh, B. Eslami, A. Aldurayhim, N. A. Mohammad Ali Khalil, and A. Alawadi, *Different lump k-soliton solutions to (2+1)-dimensional KdV system using Hirota binary Bell polynomials*, Open Physics, *21*(1) (2023), 20230167.

[37] Q. Yang, J. Manafian, K. H. Mahmoud, and A. Aldurayhim, *Abundant exact traveling wave solutions and modulation instability analysis to the generalized Hirota–Satsuma–Ito equation*, Open Physics, *22*(1) (2024), 20240027.

[38] J. Zhang, J. Manafian, S. Raut, S. Roy, K. H. Mahmoud, and A. S. Alsubaie, *Study of two soliton and shock wave structures by weighted residual method and Hirota bilinear approach*, Nonlinear Dynamics, (2024), 1–17.