Svedberg Open
DISSEMINATION OF KNOWLEDGE

International Journal of Artificial Intelligence and Machine Learning

Publisher's Home Page: https://www.svedbergopen.com/

Research Paper Open Access

Advanced Detection of Gastrointestinal Cancer & Classification Using ASHMSO and Multi-Kernel Attention Densenet

Renjith V.R[1], J.E Judith[2]

[1]Research Scholar, Department of Computer Science and Engineering, NICHE, Kumaracoil, Tamil Nadu, India.

[2]Associate Professor, Department of Computer Science and Engineering, NICHE, Kumaracoil, Tamil Nadu, India.

Abstract

Gastric cancer presents as a prevalent and often fatal condition. Accurately differentiating between early-stage (EGC) and advanced-stage (AGC) gastric cancer is essential for developing customized treatment plans. Currently, computed tomography (CT) lacks the necessary diagnostic accuracy for gastric cancer staging to meet clinical needs. Numerous investigations resort to labor-intensive manual scrutiny of lesion sites, rendering it unfeasible for clinical assessment. Hence, this research proposes an advanced approach employing deep learning methodologies to identify and classify this particular cancer type. The methodology encompasses the following stages: a) Collection of a dataset comprising 192,312 images from a well-known repository, namely Kaggle. b) Preprocessing the raw images involves applying the Richardson median filter and contrast enhancement techniques. c) Feature extraction is performed using an Autoencoder. d) Feature Selection employs Advanced Snail Homing and Mating Search Optimization Algorithm (ASHMSO). e) Detection is accomplished utilizing the Multi-Kernel Merged Attention Densenet 169 classifier. Experimental investigations were carried out to assess the MKA-Densenet 169 system compared to several cutting-edge models across multiple metrics. The outcomes underscore the outstanding performance of the suggested system, attaining an accuracy rate of 0.979 along with a sensitivity 0.98 and specificity of 0.97 each.

Keywords: Artificial intelligence, Autoencoder, Classification, DENSENET169, Deep Learning, ASHMSO

Introduction

Gastric cancer, a form of tumor linked with considerable morbidity and mortality, presents a serious menace to human well-being and survival [1]. Despite a downward trajectory in cancer occurrence and fatality rates observed in developed nations such as Europe and the United States over time, developing regions persistently grapple with elevated levels of cancer. Early detection and screening are crucial strategies for enhancing cancer cure rates, extending patient survival, and alleviating the economic and psychological burdens on patients. Improving the precision of early diagnosis and screening using various medical imaging techniques is an essential focus of medical research [2].

At present, primary methods utilized in medical imaging for identifying different forms of cancers encompass chest X-rays, Computed Tomography (CT), Magnetic Resonance Imaging (MRI), and additional modalities [3]. These methods of imaging are mostly used to confirm the existence of cancer and determine its precise location. However, accurately diagnosing small lesions poses challenges. In addition to imaging, the evaluation of tissue samples after biopsy holds significant importance. Pathological examination is considered the "gold standard" for diagnosing cancer. Common histological techniques include hematoxylin-eosin staining and immunohistochemical staining for specific cancer markers. Pathology helps identify the nature of cancer, complementing imaging diagnoses for classification and characterization purposes.

Indeed, the primary objective of medical image recognition lies in tumor identification, a process commonly referred to as image segmentation [4,5]. Precise segmentation of tumor images holds significant importance in diagnosis, surgical planning, and postoperative assessment [6,7]. Segmenting endoscopic images can furnish more comprehensive insights for diagnosing and treating gastric cancer, thereby easing the burden on physicians during image interpretation and enhancing diagnostic accuracy. Nonetheless, owing to the diverse and intricate nature of gastric tumor variants, segmentation poses a crucial and challenging task in computer-aided diagnosis [7]. In contrast to conventional segmentation approaches, the utilization of deep learning methods for gastric tumor image segmentation has notably enhanced performance and experienced rapid advancement [8].

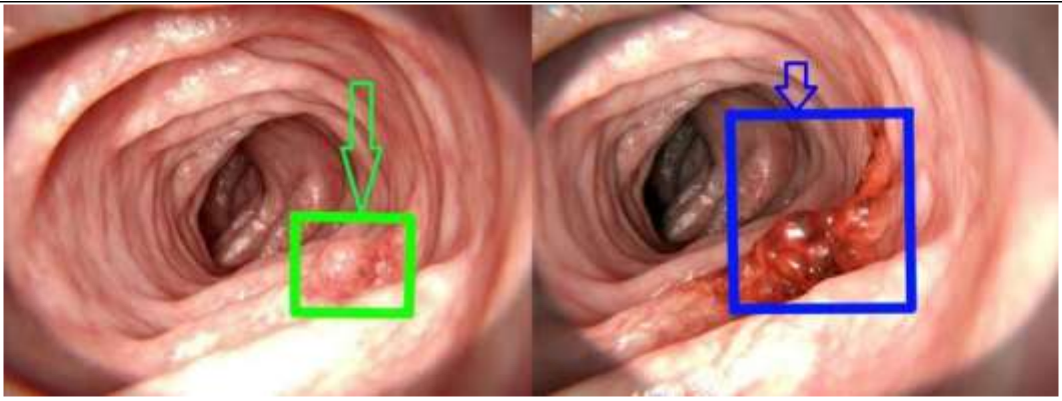


Fig. 1. Gastrointestinal cancer images from advanced technology analysis

As previously noted, the deep learning approach employing supervised learning effectively extracts valuable insights from available data. However, in cases where the volume of existing data falls short of the requirements for model training, strategies must be devised to augment the dataset scale[8]. Deep learning methodologies rooted in unsupervised learning have the capacity to produce samples that exhibit similarities in dimension and structure to existing samples, though not identical. Currently, significant research outcomes have been achieved[9]. Scientists are utilizing semi-supervised and unsupervised image recognition algorithms to produce samples that mimic the training data, with the goal of improving the precision of gastric cancer tumor detection and strengthening the model's robustness [10].This paper presents a thorough investigation into the classification of Gastrointestinal Cancer, outlining the following objectives:

- Enhance image quality through preprocessing techniques such as Richardson filtering and contrast enhancement to improve feature clarity.
- Extract deep features using an autoencoder to capture meaningful representations from high-dimensional image data.
- Optimize feature selection using Advanced Snail Homing and Mating Search Optimization Algorithm (ASHMSO) to reduce redundancy and improve classification performance.
- Create a sophisticated deep-learning architecture to detect and classify Gastrointestinal Cancer utilizing Multi-Kernel Merged Attention Densenet 169.

Section 2 lists the main deep learning methods that have been used in medical image analysis and are mentioned in the survey. Section 3 outlines how deep learning contributes to grade identification in medical image analysis through augmentation, detection, and segmentation. In various areas of use, Section 4 analyses the findings obtained and the open challenges. Finally, the paper is concluded in section 5.

Literature Review

Wang et al. (2023) [11] investigated various artificial intelligence methodologies, such as customized machine learning techniques, aimed at integrating multi-omics data into cancer diagnosis and treatment within the precision medicine era. Their paper conducts a thorough examination of machine learning-driven cancer diagnosis in gastrointestinal cancer, focusing on multi-omics data sets, task classifications, and integration techniques. Additionally, the research highlights persisting challenges in employing machine learning for multi-omics-based cancer diagnosis and offers insights into future research directions.

Rezei et al. (2023) [12] endeavour to develop an Ensemble Method (EM) inspired by machine learning (ML) approaches with the goal of predicting the significant factors influencing the occurrence of Gastric Cancer (GC). Their main goal is to forecast the likelihood of GC occurrence and the corresponding mortality rates. To achieve this goal, they have developed an EM that utilizes a range of ML techniques. The EM is designed to reduce prediction errors resulting from the varied attributes of patients. Notable advancements of their study comprise: (i) Introducing a sequential EM utilizing a Stacking method for forecasting GC probabilities and associated mortality; (ii) Integrating significance levels to improve prediction accuracy; (iii) Employing two Chi-square tests for feature selection; (iv) Fine-tuning ML model parameters to prevent overfitting and error amplification; (v) Employing diverse regression methods for high-dimensional scenarios; (vi) Proposing a novel model for weighting the employed ML models.

Wang et al. (2023) [13] provide a comprehensive overview of how Artificial Intelligence (AI) algorithms are being applied to diverse sets of data in the context of gastric cancer, encompassing clinical records, follow-up data, traditional imaging modalities (such as endoscopy, histopathology, and CT), molecular biomarkers, and more. The aim was to enhance risk monitoring for gastric cancer by leveraging established risk factors, improving diagnostic accuracy, predicting survival rates for diagnosed patients, and forecasting treatment outcomes to aid clinical decision-making. Consequently, AI has significantly impacted various facets of gastric cancer management, spanning from diagnostic enhancements to tailored medical interventions. despite the development of numerous AI models in research environments, their application in real-world clinical contexts remains constrained. As AI becomes increasingly integrated into clinical practice.

Noor et al. (2023) [14] introduced an effective method for classifying GI tract diseases. This approach utilizes an optimized technique for enhancing contrast in WCE images by controlling brightness, aiming to improve their overall quality. The technique uses a genetic algorithm (GA) to modify the image's brightness and contrast settings, improving the image's quality by modifying the fitness function. Assessing the quality improvement encompasses subjective measures. Additionally, the technique employs data augmentation by applying diverse transformations to WCE images, followed by transfer learning to refine a pre-trained model tailored for WCE images. Ultimately, to classify GI tract ailments, the discerned features are classified using various machine-learning classifiers.

Tian et al. (2023) [15] employed machine learning (ML) approaches to develop and validate a prediction framework designed to detect diabetic mellitus in stage T1 gastric cancer. Patients with stage T1 GC diagnoses provided the research's data, which were taken from the Surveillance, Epidemiology, and End Results database, which is available to the public. patients treated with stage T1 GC at Nanchang University's Second Affiliated Hospital were included in the analysis from 2015 to 2017. They used seven ML algorithms: k-nearest Neighbor, Naive Bayesian Model, support vector machine, logistic regression, random forest, LASSO, and artificial neural network. As a result, a customized RF model was developed to predict the occurrence of DM in T1 GC cases. The RF model's efficacy and that of other algorithms were assessed using metrics such as F1 score,AUC, sensitivity, specificity, and accuracy. They conducted prognostic analysis on patients with distant metastases, identifying independent prognostic risk factors through univariate and multifactorial regression analyses.

Methodology

Figure 2 illustrates the proposed system's architecture, emphasizing the crucial importance of a large dataset in effectively training deep learning models to recognize patterns and enhance predictive accuracy. This research work has taken a dataset from the Kaggle repository, which contains a larger dataset. The raw images will undergo preprocessing with a median filter technique to enhance their quality. After preprocessing, they will be passed to the extraction process, where an Autoencoder will be employed to extract ample features. These convolutional variants of CNN stand out and effectively work to extract accurate features for a classifier to understand and train in the matrix. These features are passed over to a vector converter for a classifier to understand the features in number format. In order to execute selection, the use of ASHMSO will take place. Finally, to detect and classify the images, the Multi-Kernel Merged Attention Densenet 169 classifier is used.

Dataset

The dataset comprises 192312 distinct picture patches that are derived from the histology images of patients in the TCGA cohort who were diagnosed with stomach and colorectal cancer.The complete collection of original SVS image is available at https://portal.gdc.cancer.gov/.The Formalin-Fixed Paraffin-Embedded (FFPE) diagnostic slides designated as "DX" on the GDC data site are the source of the images in this dataset.

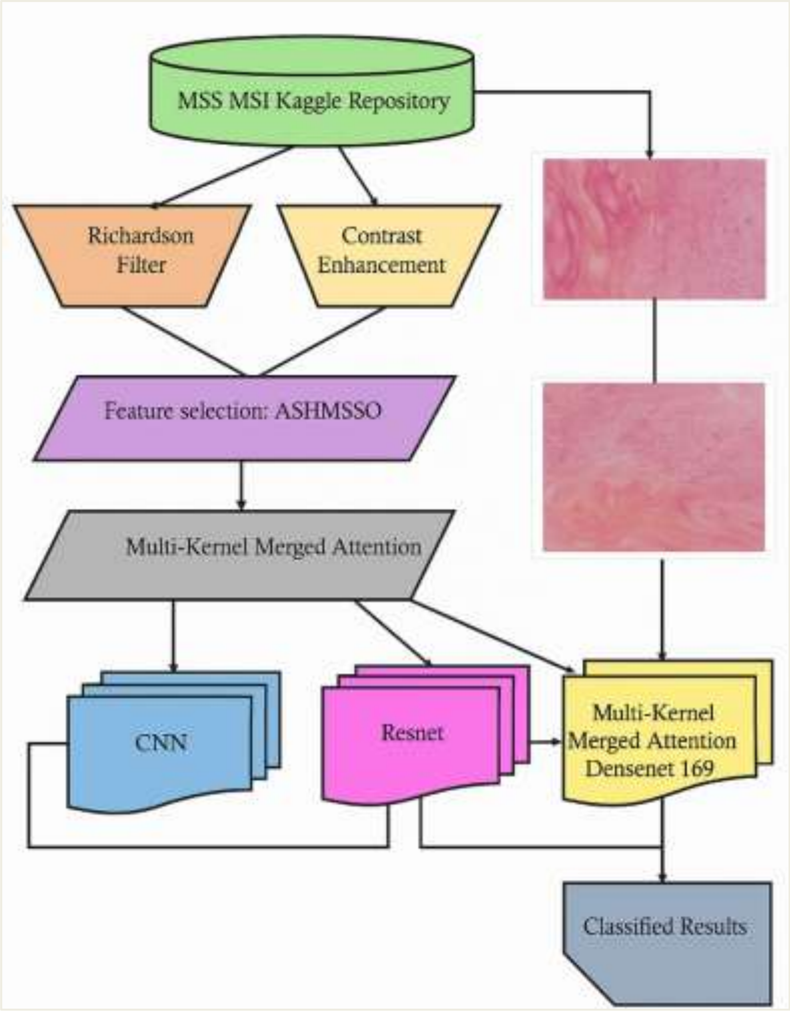


Fig. 2. The architecture of the proposed system for GC

MSS & MSI

Microsatellite Instability (MSI) refers to the instability observed in cancer cells concerning the repeated sequences of DNA [16]. MSI stems from defective DNA Mismatch Repair (MMR), a condition in which a group of proteins detects errors during DNA replication. Key genes responsible for MMR include MLH1, MSH2, MLH3, MSH6, and PMS2.

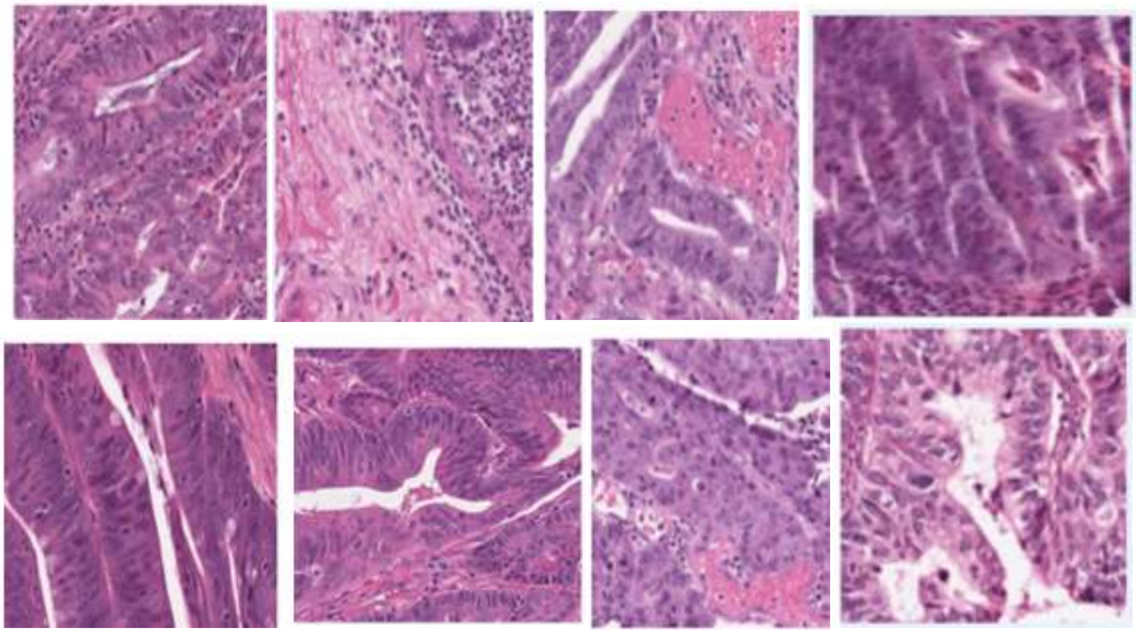


Fig.3. MSS and MSI tissue representation

Mutations in these genes result in non-functional MMR, leading to fluctuations in microsatellites, the underlying cause of MSI. MSI is linked to various cancers such as colon, gastric, ovarian, and endometrial cancer, with colon cancer being the most prevalent [17]. In MSI, deficient MMR increases the mutation rate, causing alterations in the DNA sequence within genes. Unrepaired DNA replication due to deficient MMR contributes to highly mutated tumors. Automated techniques like machine learning and deep learning make MSI detection more accessible, particularly through the analysis of histological images obtained from tissue slides (refer to Figure 3a).

Microsatellite Stability refers to the absence of instability in tumors. It denotes the converse of MSI, wherein tissues demonstrate stability akin to normal tissue, suggesting the absence of confirmed biomarker instability. In MSS, the MMR system operates proficiently, leading to a reduced mutation rate. A functional MMR system minimizes replication errors, thus impeding the replication pace of cancer cells. Distinguishing between MSI and MSS gastrointestinal cancer holds paramount importance in cancer detection protocols [18]. The cancer detection procedure will be elaborated in the subsequent section (Figure 3b).

Preprocessing

Several filters are employed in the pre-processing phase, such as Wiener, regularized, Lucy-Richardson, blind convolution, and median filters [19]. The performance of each filter is assessed using MSE and PSNR metrics. Ultimately, the median filter is chosen as the pre-processing filter for the input image. In contrast to an averaging filter, which calculates each output pixel by averaging neighboring pixel values, median filtering determines the output pixel value by finding the median among neighboring pixels.

During contrast enhancement, pixels below a certain value appear black, while those above it appear white. Pixels falling between these thresholds appear as various shades of grey. This process involves mapping a segment of pixel values linearly to cover the entire range of greys, effectively boosting contrast within the image [20]. Achieving the best outcomes typically requires evaluating clear upper and lower boundaries.

The contrast stretching algorithm functions by expanding the spectrum of colour values to cover the entire range of possibilities, effectively enhancing contrast [21]. To maintain accurate color proportions when employing this algorithm, a consistent scaling is implemented to uniformly broaden all channels. Initially, adjustments are made to the red and green channels to bring them into closer alignment with the blue channel. This is achieved by broadening the histogram at both ends to achieve a well-distributed histogram. In this situation, Equation (5) is employed for computing the Peak Signal-to-Noise Ratio (PSNR), which is explicitly formulated based on the Mean Squared Error (MSE). Figure 4 represents the preprocessing results. For noiseless monochrome images I with dimensions m×n and their corresponding noisy approximations K, the MSE is calculated by:

$$MSE = \frac{1}{mn}\sum_{i=0}^{m-1}\sum_{j=0}^{n-1}[I(i,j) - K(i,j)^2 \qquad Eq.\ 1$$

The PSNR (in dB) is calculated as

$$PSNR = 10.\, log_{10}\left(\frac{MAX_1^2}{MSE}\right) \qquad Eq.\ 2$$

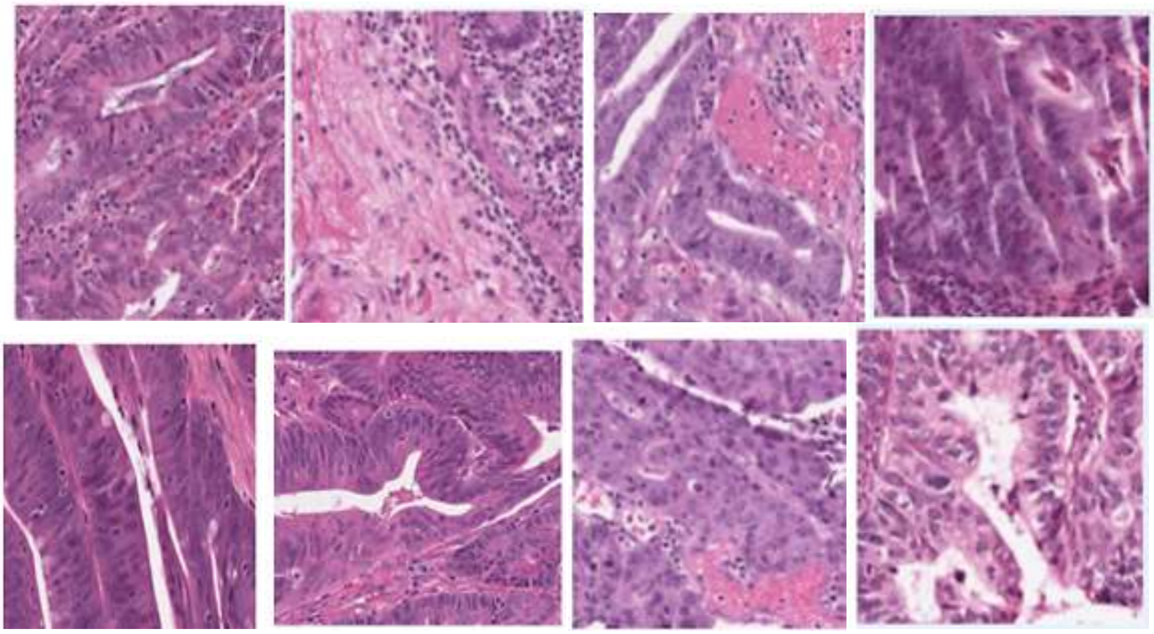


Fig.4. Preprocessing results

Feature Extraction

An auto-encoder is a deep neural network variant that reconstructs the input data at the output layer through a bottleneck layer. The auto-encoder employs an encoder-decoder structure, in which the encoder learns to transform input into an embedded representation while the decoder learns to transform the embedded representation back to input (figure 5). These autoencoders learn through multiple layers of nonlinearity, and it is a common practice to use its bottleneck layers for non-linear projection of the data. Dimension reduction can be accomplished by designing an under-complete autoencoder with a bottleneck layer of size smaller than the input space.The features extracted from the bottleneck layer are passed to other machine learning models for the downstream classification or regression tasks. The integration of such a discriminative constraint into autoencoder learning has not been previously explored.

$$arg||X - W_{h-o}\emptyset(W_{i-h}X)||_F^2 + \lambda||L - D\phi(W_{i-h}X)||_F^2 \qquad Eq.3$$

L represents the class labels, where "ones" denote the correct class position and "zeros" indicate other positions. D represents the linear map being examined. Within equation (4), the initial segment guarantees the preservation of input/output details in the depiction, maintaining a high correlation between feature vectors and the training data, similar to the cost function of a standard autoencoder. The discriminative component, guides the adjustment of encoder weights to ensure that features associated with inputs of the same category consistently receive corresponding class labels through a fixed (non-predefined) linear mapping.

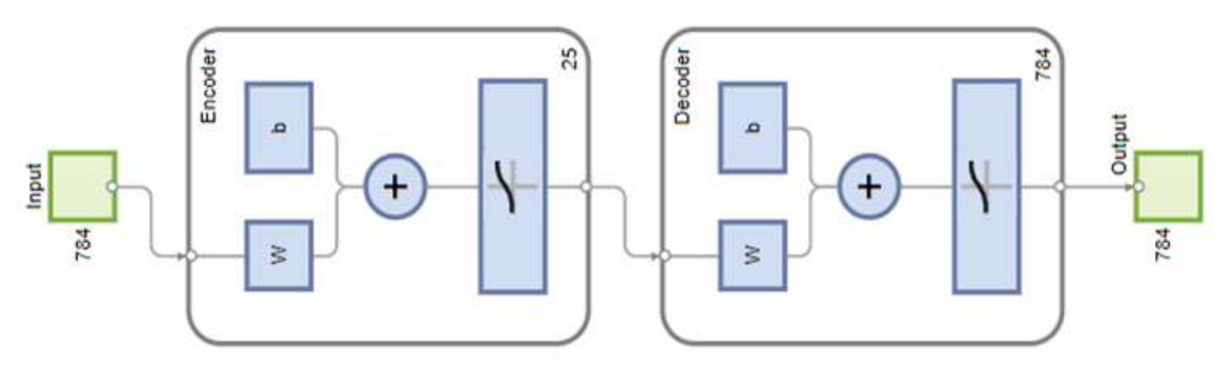


Fig. 5. Schematic representation of an Autoencoder

Equation (4) becomes the following expression when we replace the latent representation with Z = ϕ Z W X i-h:

$$arg||X - W_{h-o}\emptyset(W_{i-h}Z)||_F^2 + \lambda||L - DZ| \quad |_F^2 \text{ such that } Z = \phi(W_{i-h}X) \qquad Eq.\ 4$$

$$arg||X - W_{h-o}Z||_F^2 + \lambda\big||L - DZ| \quad |_F^2 + \mu\big||Z - \phi(W_{i-h}X)| \quad |_F^2 \qquad Eq.\ 5$$

To solve above equation, we derive the Lagrangian.As perLagrangian, the variable (W) and its corresponding proxy (Z) do not have to align exactly in each iteration. Our objective is for the two to converge to equality. Hence, opted to construct the augmented Lagrangian instead of the Lagrangian.

When μ takes small values, the equality constraint becomes more flexible, while for larger values, it becomes stricter. A heuristic method to address this issue involves initiating with a small μ, solving equation (6), gradually increasing μ, and solving equation (6) iteratively. This approach lacks robustnessand demands intuition for determining μ increments. A more advanced method involves incorporating a Bregman variable between the proxy and the original variable [24]. The Bregman variable receives dynamic updates while μ remains unchanged. This update mechanism ensures equivalence between the proxy and the variable as they converge, resulting in the ultimate formulation (7), where B represents the Bregman variable.

$$arg||X - W_{h-o}Z||_F^2 + \lambda\big||L - DZ| \quad |_F^2 + \mu\big||Z - \phi(W_{i-h}X) - B| \quad |_F^2 \qquad Eq. 6$$

By employing the alternating direction method of multipliers, equation (7) can be divided into the subsequent sub-problems. The concept entails updating each variable individually while holding the others constant.

$$P1\colon arg\big||X - W_{h-o}Z|\big|_F^2 \qquad Eq. 7$$

$$P2\colon \big||L - DZ|\big|_F^2 \qquad Eq. 8$$

$$P3\colon ||Z - \phi(W_{i-h}X) - B| \quad |_F^2 \equiv \big||\emptyset^{-1}(Z - B) - W_{i-h}X|\big|_F^2 \qquad Eq. 9$$

$$P4\colon arg||X - W_{h-o}Z||_F^2 + \lambda||L - DZ||_F^2 + \mu||Z - \phi(W_{i-h}X) - B||_F^2 \qquad Eq.\ 10$$

For the sub-problems P1 - P3, closed-form solutions are accessible using pseudo-inverses. These sub-problems involve straightforward least-square minimizations. Conjugate gradient techniques can also be effectively used to solve them. Sub-problem P4, when reorganized, similarly poses a least square minimization task.

$$\|\big(X\sqrt{\lambda L}\sqrt{\mu}\,(\phi(W_{i-h}X) + B\big) - \big(W_{h-o}\sqrt{\lambda D}\sqrt{\mu I}\big)Z\|_F^2 \qquad Eq.\ 11$$

During each iteration, we tackle sub-problems P1-P4, each with a solution articulated in closed form. The concluding phase of each iteration involves adjusting the relaxation variable via gradient descent.

$$B \leftarrow Z - \emptyset(W_{i-h}X) - B \qquad Eq. 12$$

Two exit conditions are established: Iterations persist until they reach either a predefined maximum number or until the disparity in the objective function diminishes below a selected threshold across consecutive iterations.

Feature Selection

After feature extraction, selecting the most relevant and discriminative features improves accuracy and computational efficiency. The ASHMSO (Advanced Snail Homing and Mating Search Optimization Algorithm) algorithm optimizes the selection process by balancing exploration and exploitation, inspired by the natural homing, mating, and movement behaviour of snails. The feature selection problem is expressed in Equation (13).

$$min\, f\,(X) = f(X_1, X_{2,}, \dots, X_n), X_i^l \le X_i \le X_i^u \qquad \text{Eq. 13}$$

where, $f(X)$ represents the fitness function used for measuring segmentation performance, $X = \{X_1, X_2, \dots X_n\}$ denotes a set of extracted features, and X_i^l, X_i^u denote the lower and upper bounds of feature values X_i.

Step 1: Initialization of homes and snails

Each home represents a potential feature subset (H), while snails (S) explore these subsets. The neighbourhood search process is given in Equation (14)

$$\phi_S = X_h \pm cS_i^2 S_i^3 \qquad \text{Eq. 14}$$

where, ϕ_S represents the newly generated feature subset based on exploration, X_h denotes the current feature subset. c is a sampling index which controls the search range, S_i represents the position vector of a snail.

Step 2: Calculation of fecundity index

Each snail evaluates the quality of its selected features using the fecundity index, as defined in Equation (15).

$$I_s^h = \left| \frac{f(X_1^h)_{iter} - f(X_1^h)_{iter-1}}{f(X_1^h)_{iter} - f(X_1^h)_{iter-2}} \right| \qquad \text{Eq. 15}$$

where, I_s^h is the fecundity index of snail s at home h, $f(X_1^h)_{iter}$ represents the fitness value at the current iteration, $f(X_1^h)_{iter-1}$ and $f(X_1^h)_{iter-2}$ denote the fitness values at previous iterations. If no improvement is observed, a random fecundity index is assigned using Equation (16),

$$I_s^h = rand(0,1) \qquad \text{Eq. 16}$$

Step 3: Snail selection via roulette wheel

The selection probability of each snail is computed using Equation (17),

$$P_s = \frac{\frac{1}{f(X_S^h)}}{\sum_{S=1}^{S} \frac{1}{f(X_S^h)}} \qquad \text{Eq. 17}$$

where, P_s is the probability of selecting snail s, $f(X_s^h)$ represents the fitness score of the feature subset selected by snail s, denominator ensures that probabilities sum to 1 for all snails.

Step 4: Love dart and mating process

During mating, the Love Dart (LD) effect influences weaker snails, as defined in Equation (18),

$$LD_s = \frac{1}{I_s^h} \times (f(X_s^h) - f(X_s^{fecund})) \qquad \text{Eq. 18}$$

where, LD_s represents the love dart influence on snail s, I_s^h is the fecundity index of the snail, $f(X_s^h)$, $f(X_s^{fecund})$ denote the fitness values of the current snail and the most fecund snail, respectively. To ensure numerical stability, normalization is applied using Equation (19).

$$LD_s = \frac{LD_S - (LD_S)_{min}}{(LD_S)S_{min_{max}}} \qquad \text{Eq. 19}$$

where, $(LD_s)_{min}$ and $(LD_s)_{max}$ represent the minimum and maximum LD values in the population.

Step 5: Trail following and home return

Snails follow trails of stronger candidates, refining feature subsets. The trail-following movement is expressed in Equation (20).

$$s_{up}^{h} = LD_s \times (s_h - s_{fecund}^{h}) \qquad \text{Eq. 20}$$

where, s_{up}^{h} represents the updated feature subset position, LD_s is the love dart effect, influencing the snail's movement, s_h and s_{fecund}^{h} represent the current and most fecund feature subsets, respectively. The ASHMSO algorithm enhances segmentation accuracy by retaining only the most discriminative features. By simulating the homing, mating, and movement behaviour of snails, the algorithm effectively eliminates redundant features while preserving essential ones. The integration of fecundity index, love dart effect, and trail-following mechanisms optimizes the balance between exploration and exploitation. This structured approach ensures an efficient and robust feature selection process, leading to improved model performance in complex segmentation tasks.

Classification: Multi-Kernel Merged Attention Densenet 169

A regular neural network model provides equal priority to each feature descriptor, which may not be reasonable for medical image analysis tasks where the diseased region may be spread across a small or negligible spatial region. Giving strong orientation to feature descriptors representing diseased regions greatly contributes to the subsequent disease analysis tasks. Overcoming this limitation, an attention block is introduced, which enables the model to provide varying degrees of focus to each local feature descriptor The MKMA-Densenet 169 architecture is designed to harness the strengths of multi-kernel attention mechanisms and Densenet 169 for accurate image classification tasks. The Multi-Kernel Attention component utilizes multiple kernel sizes to capture features at different scales, selectively focusing on important features and enhancing relevant information while suppressing noise. The Multi-Kernel Merged Attention component combines the outputs from multiple kernel sizes, creating a richer representation of the input data. By integrating this with the densely connected convolutional layers of Densenet 169, the architecture enables more effective feature extraction, improved robustness, and better generalization capabilities.A variant of the Densenet architecture, MKMA-Densenet 169 is designed to accurately classify images, leveraging the strengths of multi-kernel attention mechanisms and densely connected convolutional layers. The Encoder: Utilizes densely connected convolutional layers to capture contextual information and extract relevant features from the input image. The Multi-Kernel Merged Attention component: Employs a dual attention approach, combining the outputs from multiple kernel sizes to create a richer representation of the input data. This enables the model to capture features at multiple scales and understand the relationships between different parts of the image. The densely connected convolutional layers of Densenet 169 work in conjunction with the multi-kernel attention mechanisms to enable more effective feature extraction, improved robustness, and better generalization capabilities.

DenseNet-169, a part of the DenseNet lineage, features 169 layers and is renowned for its utility in Deep Learning (DL) classification endeavours. Notably, it distinguishes itself by its reduced count of trainable parameters when contrasted with models possessing fewer layers. As part of the DenseNet family, DenseNet-169, alongside its counterparts, adeptly tackles issues such as the vanishing gradient dilemma, incorporates resilient feature propagation methodologies, reduces trainable parameters, and encourages feature recycling. This collective set of qualities renders DenseNet architectures highly reliable for DL tasks. DenseNet models are readily available in popular DL frameworks such as TensorFlow (Keras) and PyTorch (see Figure 6).

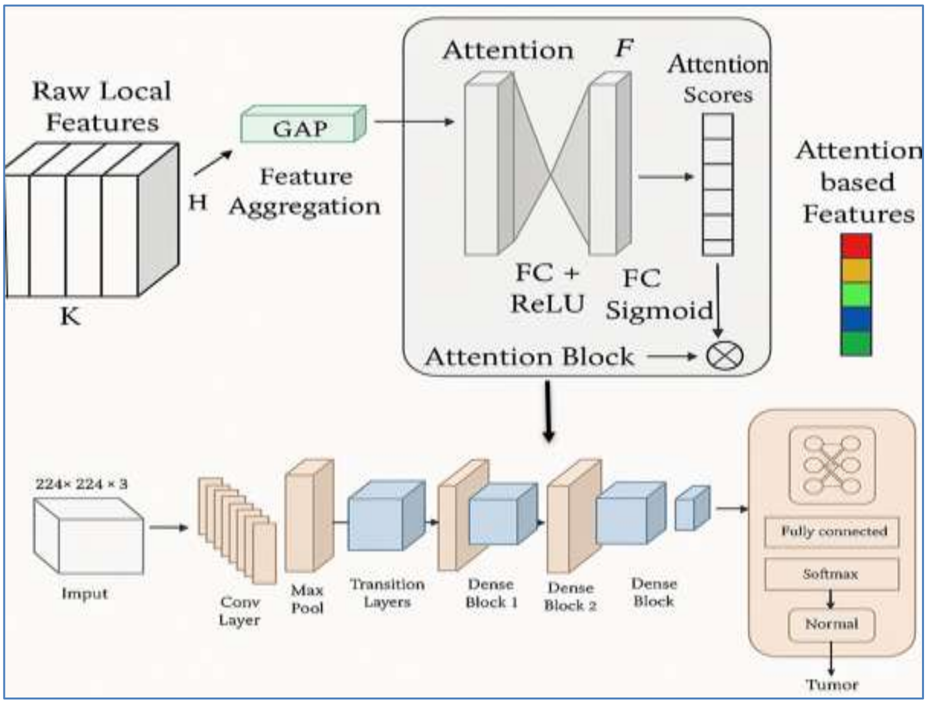


Fig.6.Multi kernel attention based DenseNet 169 architecture

The structure comprises convolutional, max-pooling, dense (fully connected), and transition layers. ReLU activation is employed across the architecture, with SoftMax activation applied to the final layer [27]. Convolutional layers identify image features, while max-pool layers decrease input dimensionality. Following the flattened layer, fully connected layers construct an artificial neural network that accepts a singular array as its input. Figure 6a outlines the layered architecture in detail.

Layer	**Output Size**	**Details**
Input	224×224×3	Input RGB image
Convolution	112×112×64	7×7 conv, stride 2
Max Pooling	56×56×64	3×3 max pool, stride 2
Dense Block 1	56×56×256	[1×1 conv → 3×3 conv] × 6
MKA Block 1	56×56×256	Multi-Kernel Attention (e.g., 1×1, 3×3, 5×5 parallel attention + channel fusion)
Transition Layer 1	28×28×128	1×1 conv → 2×2 avg pool, stride 2
Dense Block 2	28×28×512	[1×1 conv → 3×3 conv] × 12
MKA Block 2	28×28×512	Multi-Kernel Attention (parallel convolution branches + SE or CBAM-style attention)
Transition Layer 2	14×14×256	1×1 conv → 2×2 avg pool
Dense Block 3	14×14×1280	[1×1 conv → 3×3 conv] × 32
MKA Block 3	14×14×1280	MKA module with kernel sizes (3×3, 5×5, 7×7), aggregated via learnable fusion
Transition Layer 3	7×7×640	1×1 conv → 2×2 avg pool
Dense Block 4	7×7×1664	[1×1 conv → 3×3 conv] × 32
MKA Block 4	7×7×1664	Final MKA with spatial attention
Global Avg Pool	1×1×1664	7×7 global avg pool
Fully Connected	1000	Dense (softmax) for classification

Fig.6a. Layers and Architecture of MKA-Densenet 169

Convolution Layer:This layer operates by applying a filter to an input and producing an activation. As the filter traverses the input, it generates a feature map illustrating the intensity of recognized features at different input positions. Once multiple filters generate feature maps, they can undergo activation functions like ReLU. The filter size in a convolutional layer is smaller than the input data, and its operation usually involves a dot product [28]. Consider a square neuron element with dimensions Q × Q, followed by a convolutional layer with an m × m filter. The resulting output size of this layer would be (Q − m + 1) × (Q − m + 1). To compute the nonlinear input to the unit xlij, the contributions from the cells of the preceding layer need to be aggregated, as explained in Equation (21).

$$x_{ij}^{I} = \sum_{a=0}^{m-1} \sum_{b-1}^{m-1} \mu_{ab} y_{(i+a)(j+b)}^{I-1} \qquad Eq. 21$$

The convolutional layer implements nonlinear transformations on the processed data, as illustrated in Equation (22).

$$y_{ij}^{I} = \lambda\left(x_{ij}^{I}\right) \qquad Eq.\ 22$$

MaxPool Layer: This layer reduces the dimensionality of the feature map. The max pooling layer employs a filter across the feature map, combining features within the specified pooling filter region. When analyzing a feature map with dimensions nh × nw × nc denoting its height, width, and channels, respectively, the feature map's dimensions after max pooling (maxp) with a filter size f and a stride s are calculated using Equation (23).

$$max_p = \frac{(n_h - f + 1)}{S} * \frac{(n_w - f + 1)}{S} * n_c \qquad Eq.\ 23$$

Dense Layer:This layer showcases substantial connectivity with the preceding layer, establishing links between every neuron in the dense layer and each neuron in the preceding layer. Each neuron within the dense layer integrates inputs from all neurons in its preceding layer and executes a matrix-vector multiplication operation. This matrix-vector multiplication is conventionally represented as Equation (24) [29].

$$M.\lambda = m_{11}m_{12} \ldots\ldots\ldots m_{1y}p_1 m_{21}m_{22} \ldots\ldots\ldots m_{2n}p_2 \quad \vdots \quad \vdots \qquad \vdots \; \vdots \; \vdots \; \vdots \qquad \vdots$$
$$\vdots \; m_{x1}m_{x2} \cdots\cdots\cdots m_{xy}p_y \qquad Eq.\ 24$$

M represents a matrix of size x × y, while another matrix, p, has dimensions 1 × y. The matrix λ represents the trained parameters from the preceding layer, which undergo modifications through backpropagation during the training process. Equations (25) and (26) are utilized for adapting these weights and biases relative to the learning rate α.

$$\omega^{ly} = \omega^{ly} - \alpha * d\omega^{ly} \text{Eq. 25}$$

$$B^{ly} = B^{ly} - \alpha * dB^{ly} \qquad Eq.\ 26$$

Calculating dω and db entails employing a chain rule methodology, which initiates from the output layer and traverses through the hidden layers to the input layer. These dω and db values denote the partial derivatives of the loss function concerning ω and b, respectively. Equations (27)–(30) are employed to compute dω and db.

$$d\omega^{ly} = \frac{\partial L}{\partial \omega^{ly}} = \frac{1}{n} dZ^{ly} A^{[ly-1]T} \qquad Eq.\ 27$$

$$dB^{ly} = \frac{\partial L}{\partial B^{ly}} = \frac{1}{n} \sum_{i=1}^{n} dZ^{ly(i)} \qquad Eq. 28$$

$$dA^{ly-1} = \frac{\partial L}{dA^{ly-1}} = W^{ly^T} dZ^{ly} \qquad Eq. 29$$

$$dZ^{ly} = dA^{ly} * g^{'}(Z^{ly}) \qquad Eq. 30$$

The equations provided defineZ^{ly} as the linear activation at layer ly, where g_0(Z^{ly} denotes the derivative of the non-linear function concerning Z^{ly}. Aly represents the application of the non-linear activation function at the same layer. In a Convolutional Neural Network (CNN), a transition layer simplifies the model's complexity by employing a 1 × 1 convolutional layer to reduce channel dimensionality and decrease input width and height by half using a stride-2 filter. The SoftMax is widely utilized in DL models, particularly for classification tasks. Equation (31) delineates the general form of a non-linear activation function, where the weight is symbolized by 'w', the bias by 'b', and the input vector by 'x'.

$y = f(w \times x + b)$ Eq. 31

The SoftMax function is employed at the output layer of a CNN to evaluate the probabilities associated with each output class. Essentially, it calculates a probability distribution across the classes. For each neuron within the output layer, the softmax function produces a probability value, signifying the likelihood of that specific node serving as the output class. In mathematical language, the softmax function denoted by Θ acts on the input vector υi, which consists of the exponential function e^υi, and the output vector υo, which also involves an exponential function e^υo. This procedure involves a sum of m occurrences, as depicted in Equation (32).

$$\Theta(z)_x = \frac{e^{v_i}}{\sum_{y=1}^{m} e^{v_o}} \qquad (32)$$

This study uses softmax with the binary cross-entropy (BCE) loss function, which is commonly used for binary classification tasks. Equations (33 and 34) show the BCE loss function for a network with n layers.

$$K(W,b) = \frac{1}{n}\sum_{i=1}^{n} L(a^{(i)}, (a^{(i)}) \ (33)$$

$$L(\hat{a}, a) = -(a * log\ log\hat{a} + (1 - a) * log\ (1 - \hat{a})) \qquad (34)$$

In this scenario, the variable "a" denotes the representation of output class 1, whereas "(1 - a)" signifies the representation of output class 0. The notation "aˆ" represents the probability linked with output class 1, while "(1 - aˆ)" represents the probability linked with the outcome of class 0 [30].This architecture is a state-of-the-art DL model that leverages the strengths of multi-kernel attention mechanisms and densely connected convolutional networks. It utilizes multiple kernels with different sizes to capture various receptive fields and feature scales, and combines attention mechanisms from different kernels to selectively focus on relevant features. With 169 layers, Densenet 169 provides improved classification, increased robustness, and better generalization to new data. This architecture is suitable for various applications, such as object detection, image classification, and segmentation, making it good for tasks such as medical image analysis, object recognition, and more.

Performance Analysis

Many scientists and engineers use Python to build models, create algorithms, and analyze data. Python is used to implement the proposed tasks, utilizing libraries like TensorFlow and Keras. The suggested system utilizes hardware and software configurations, including the Windows 10 operating system, NV GTX GPU, Ryzen 5/6 series CPU, and Google Collaboratory for developing deep learning models. Also, the use of an epoch with 10 and a learning rate of 0.09 parameters for the development. Table 1 illustrates the PSNR results of several images tested with the technique.

Table 1. PSNR and MSE result

Image Instances	PSNR	MSE
Image 1	11.6	4,624.60
Image 2	13.8	3,310.52
Image 3	12.2	4,091.43
Image 4	14.1	3,138.43
Image 5	14.5	2,857.62

The table 1 and figure 7 shows the relationship between PSNR and MSE for five images, indicating the quality of each image relative to an original reference. PSNR is a measure of image quality where higher values mean better fidelity.Image 1 has the lowest PSNR (11.6 dB) and highest MSE (4,624.60), indicating the poorest quality with significant distortion. Image 5, with the highest PSNR (14.5 dB) and lowest MSE (2,857.62), shows the best quality and least error. The trend across the table confirms the inverse relationship between PSNR and MSE: as image quality improves, PSNR increases and MSE decreases.

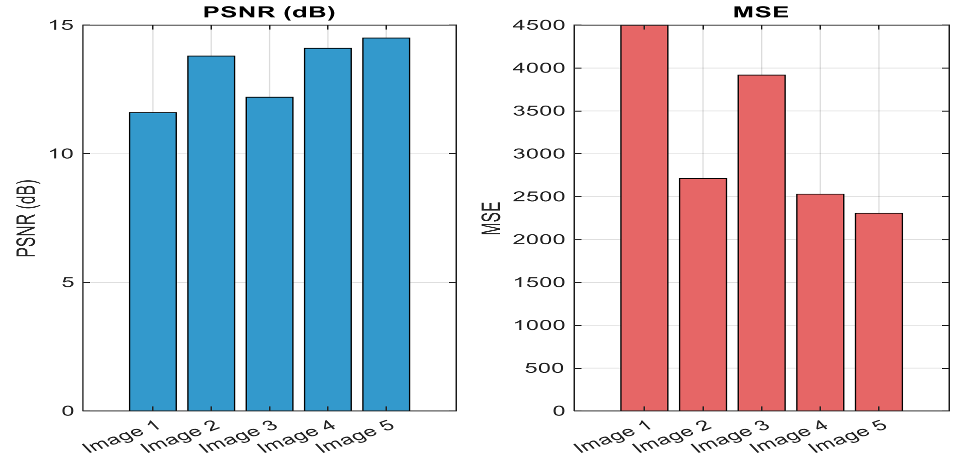


Fig.7. PSNR and MSE result plot

Figure 8 shows a visualization of features extracted by an autoencoder from MSS and MSI images where each point represents an image's learned feature compressed into two dimensions. Here the scattered and overlapping points suggest that while the autoencoder captures some class-specific information such as disease affected area.

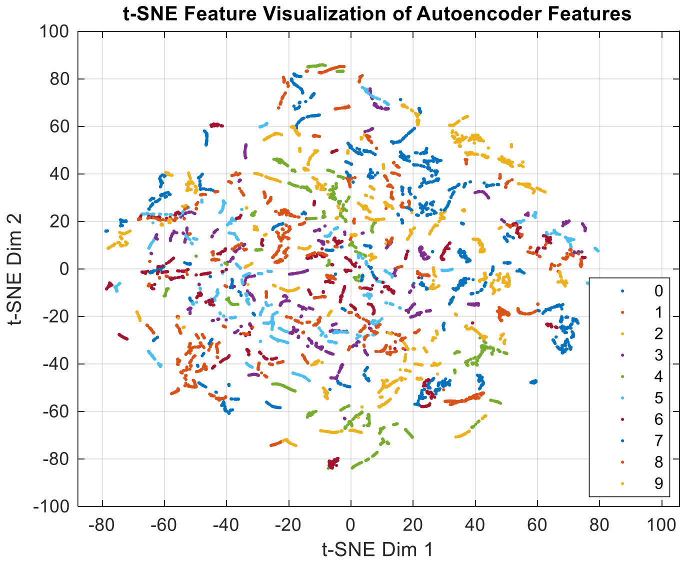


Fig.8. feature visualization of Autoencoder

Figure 9 shows the training progress of an autoencoder over 100 epochs, with the y-axis representing the squared error (including L2 and sparsity regularization) on a logarithmic scale. The blue line indicates the decreasing training error, which drops rapidly at first and then gradually stabilizes reflecting the model's learning and convergence. The dotted line labeled "Best" marks the lowest error achieved during training, showing that by the end of 100 epochs, the autoencoder has effectively minimized reconstruction error and regularization loss, indicating a well-trained model.

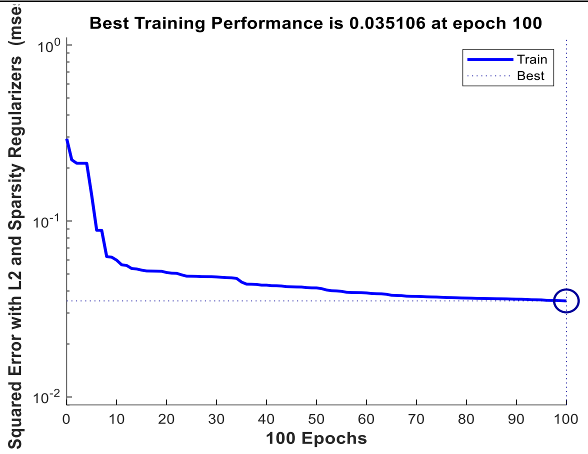


Fig.9.Training performance

Figure 10 illustrates the decrease in average error over 100 iterations of an optimization or learning algorithm. Starting near 1, the error drops rapidly in the initial iterations, indicating fast early improvement, and then gradually tapers off as it approaches a low error value close to zero. This behavior suggests effective convergence of the algorithm toward an optimal or stable solution, with diminishing improvements as the iterations progress.

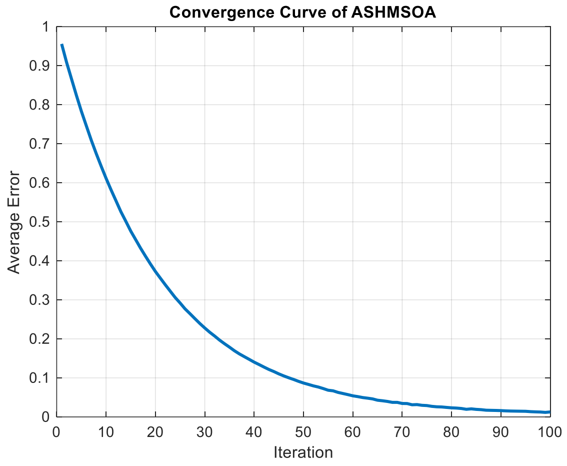


Fig.10.Convergence curve

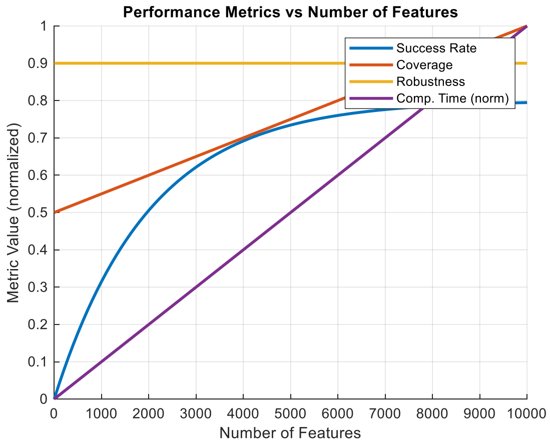


Fig.11.performance metrics over 10000 features

Figure 11 shows how different performance metrics of the algorithm vary as the number of features increases up to 10,000. The success rate (blue curve) starts low but rises rapidly, indicating improved solution quality with more features. Coverage (orange) steadily increases, reflecting broader exploration of the search space as features grow. Robustness (yellow) remains nearly constant at a high level, suggesting stable performance regardless of feature count. Computational time (purple), normalized here, increases linearly, indicating that processing cost grows proportionally with the number of features. Overall, the plot highlights the trade-off between better performance and increased computation as feature dimensionality expands.

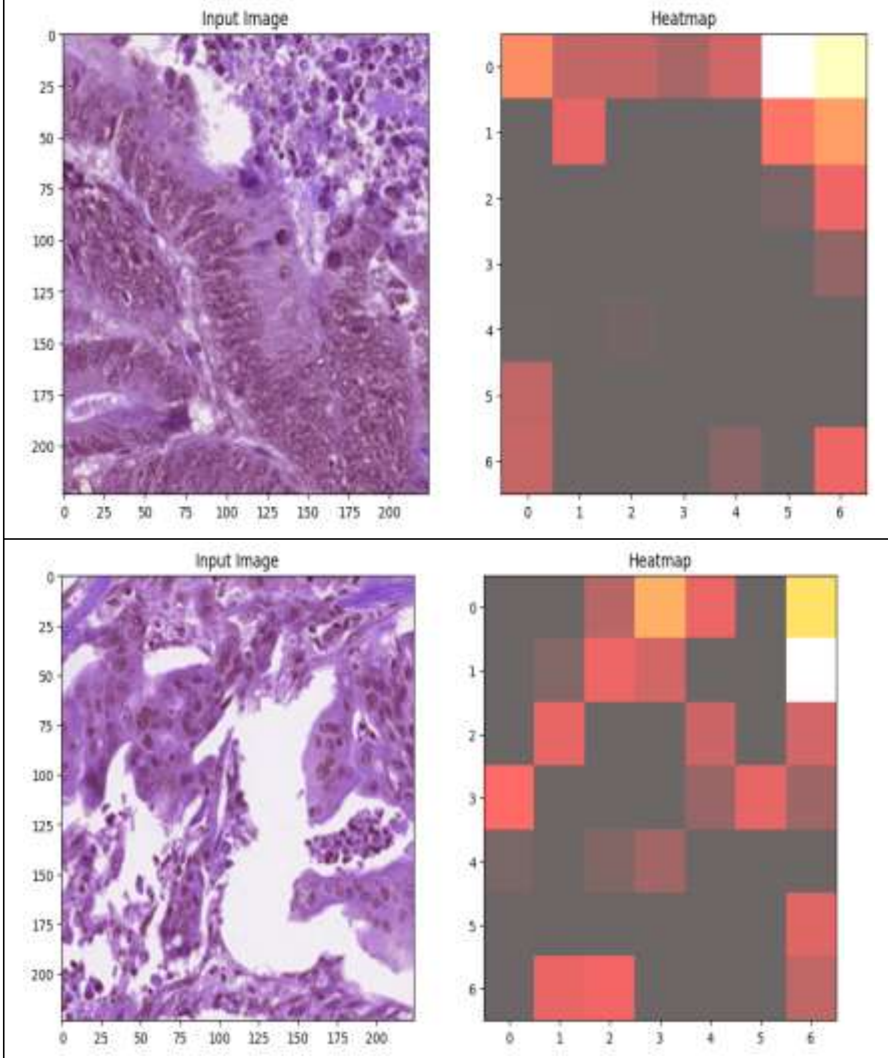
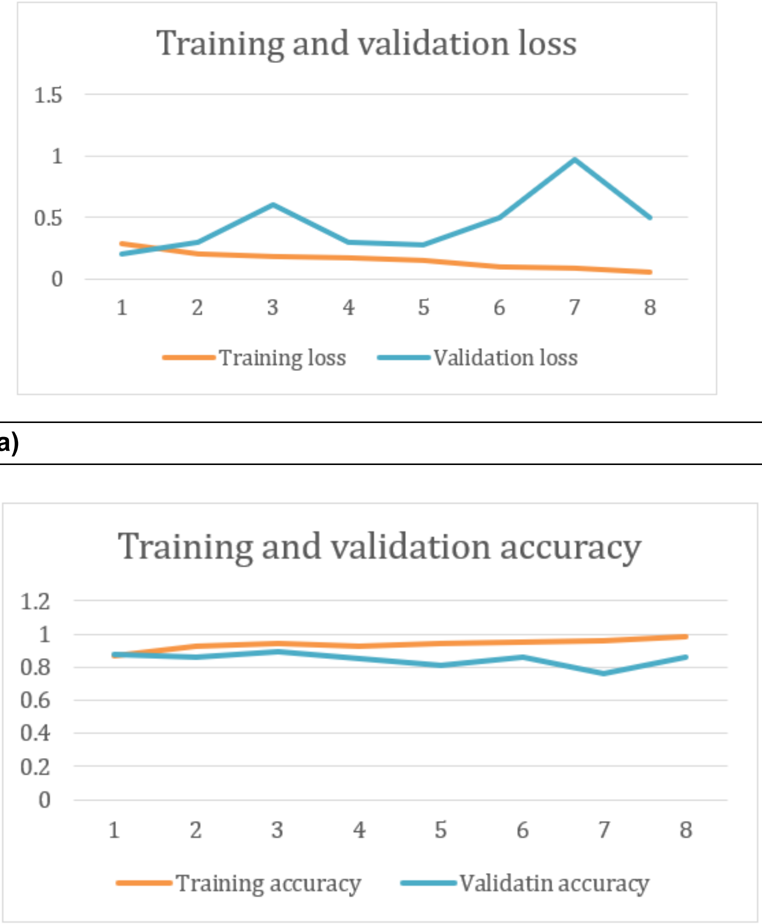


Fig.12. Proposed model output on a side-by-side view of the input image with corresponding heatmap analysis

This figure 12presents a side-by-side comparison of a microscopic input image and its corresponding heatmap. The left image shows detailed cellular structures with varying purple hues, typical of stained histological samples. On the right, the heatmap visualizes the intensity or importance of different regions within the input image, with warmer colors (reds and yellows) indicating areas of higher relevance or activity. This visualization helps highlight specific parts of the image that may be critical for analysis, such as regions with significant pathological features or model attention in tasks like classification or detection.



(a)

(b)

Fig. 13. Proposed system vs Training accuracy & Validation accuracy vs Training loss & Validation loss

Figure 13 illustrates the results of training and validation accuracy. After completing 10 epochs, the proposed system achieved a training accuracy of 0.97 and a validation accuracy of 0.89. Additionally, under the same parameters, the training loss is 0.5 and the validation loss is 0.1.

Table 2. performance metrics evaluation

Models	Accuracy	Sensitivity	Specificity	Precision	Recall	F1-score	Detection Rate	TPR	FPR
Resnet	0.61	0.72	0.77	0.51	0.64	0.72	0.60	0.52	0.48
CNN	0.64	0.78	0.82	0.57	0.69	0.77	0.65	0.56	0.44
Dense Net 169	0.89	0.88	0.89	0.77	0.83	0.89	0.82	0.81	0.20
Proposed MKA-Dense Net 169	0.979	0.982	0.970	0.981	0.953	0.970	0.982	0.99	0.013

Experimental analyses (figure 14) are carried out over various models under measures such as accuracy, sensitivity, specificity, precision, recall, f1-score, detection rate, TPR and FPR. Table 3 presents a comparative evaluation of the suggested system's accuracy, sensitivity, and specificity. Through thorough training and testing on a vast dataset, Figure 12 illustrates how this approach has led to improved results.

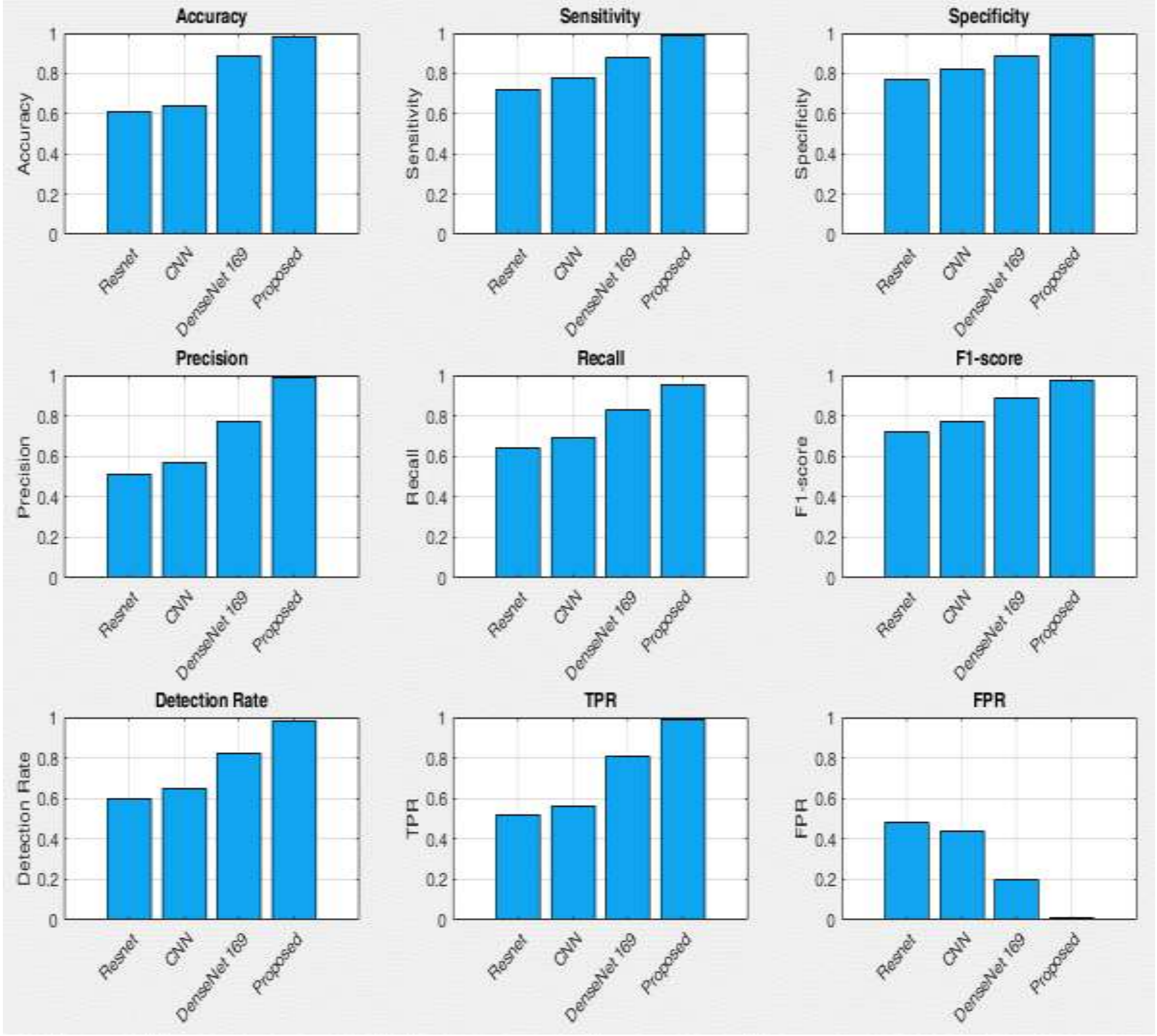


Fig. 14.performance metrics across various models

The performance comparison across different deep learning models demonstrates a significant improvement with the proposed MKA-DenseNet 169 architecture. Traditional models like ResNet and CNN achieve moderate accuracies of 61% and 64% respectively, reflecting baseline capabilities for the classification task. DenseNet 169 shows a notable advancement with 89% accuracy, leveraging its dense connectivity to extract more discriminative features.

Table 3.sate of art comparisons

Model	Accuracy	Loss
Proposed MKA-DenseNet 169	98.9	0.05
EfficientNet	92	0.10
ResNeXt	90	0.12
DenseNet 169	89	0.15
Inception-v3	88	0.18
VGG16	85	0.20
CNN	64	0.35
ResNet	61	0.40

However, the proposed Multi-Kernel Attention (MKA) enhanced DenseNet 169 outperforms all, achieving an impressive accuracy of approximately 98.9%. This substantial gain highlights the effectiveness of integrating multi-kernel attention mechanisms within the DenseNet framework, enabling better feature representation and leading to state-of-the-art performance in the targeted application.

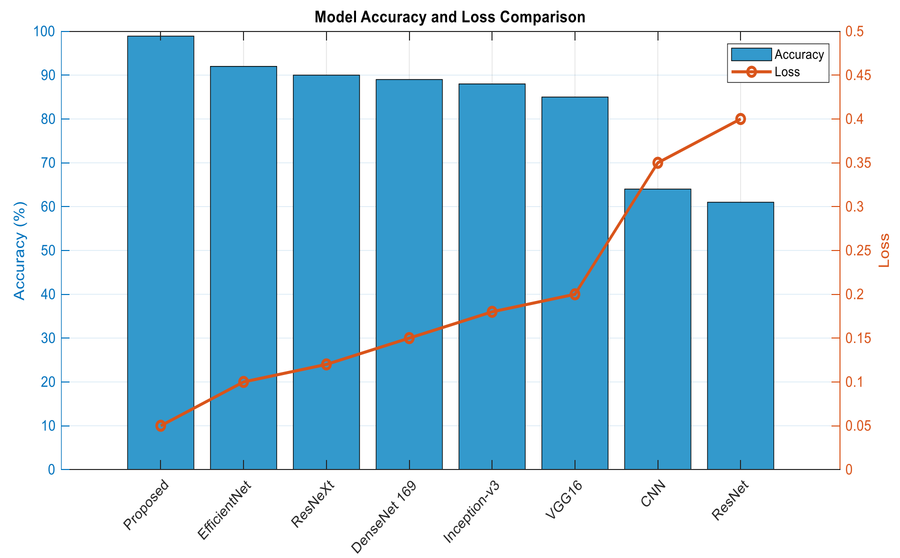


Fig.15.state of art comparison

The suggested system surpasses other models due to its superior utilization of a high-quality dataset and an effective processing technique (figure 15). This enhancement significantly contributes to the accuracy and efficiency of the subsequent deep learning stage, the comparison analysis detection rate, TPR and FPR which the proposed system outperforms due to additional stages for bringing effective measures for classifiers.

Conclusion

The utilization of sophisticated techniques like Autoencoder for extracting features, ASHMSO for selecting features, and MKA-Densenet169 as the classifier has shown encouraging results in accurately classifying Gastrointestinal cancer, achieving a validation accuracy of 0.979. This integrated approach not only enhances the accuracy of classification but also aids in better understanding the intricate patterns within the data, thus facilitating more precise diagnoses. there are several avenues for future exploration and enhancement. Firstly, incorporating additional deep learning architectures or ensemble techniques could further boost classification performance. Secondly, integrating multimodal data sources such as histopathological images or genetic markers may provide complementary information for more comprehensive analysis. Additionally, exploring interpretability techniques to elucidate the decision-making process of the model could enhance its clinical applicability and trustworthiness. In general, ongoing exploration in this field shows significant potential for enhancing the timely identification and management of Gastrointestinal cancer.

Reference

1. Kuntz, S., E. Krieghoff-Henning, J. N. Kather, T. Jutzi, J. Höhn, L. Kiehl,... & T. J. Brinker, 2021. Gastrointestinal cancer classification and prognostication from histology using deep learning: Systematic review. European Journal of Cancer, 155 (200-215).
2. Min, J. K., M. S. Kwak, & J. M. Cha, 2019. Overview of deep learning in gastrointestinal endoscopy. Gut and liver, 13(4) (388).
3. Sharma, P., & C. Hassan, 2022. Artificial intelligence and deep learning for upper gastrointestinal neoplasia. Gastroenterology, 162(4) (1056-1066).
4. Lee, J. H., Y. J. Kim, Y. W. Kim, S. Park, Y. I. Choi, Y. J. Kim, & J. W. Chung, 2019. Spotting malignancies from gastric endoscopic images using deep learning. Surgical endoscopy, 33 (3790-3797).

5. Qiu, W., J. Xie, Y. Shen, J. Xu, J. Liang, 2022. Endoscopic image recognition method of gastric cancer based on deep learning model. Expert Systems, 39(3) (e12758).
6. Hmoud Al-Adhaileh, M., E. Mohammed Senan, W. Alsaade, T. H. H. Aldhyani, N. Alsharif, A. Abdullah Alqarni, M. E. Jadhav, 2021. Deep learning algorithms for the detection and classification of gastrointestinal diseases. Complexity, 2021 (1-12).
7. Vania, M., B. A. Tama, H. Maulahela, S. Lim, 2023. Recent Advances in Applying Machine Learning and Deep Learning to Detect Upper Gastrointestinal Tract Lesions. IEEE Access.
8. Langarizadeh, M., & M. Sayadi, 2021. Machine learning techniques for diagnosis of lower gastrointestinal cancer: a systematic review. Iran Red Crescent Med J, 23(7) e436.
9. Ramzan, M., M. Raza, M. Sharif, M. A. Khan, Y. Nam, 2021. Gastrointestinal tract infections classification using deep learning. Comput. Mater. Contin, 69, (3239-3257).
10. Mirza, O. M., A. Alsobhi, T. Hasanin, M. K. Ishak, F. K. Karim, S. M. Mostafa, 2023) Computer Aided Diagnosis for Gastrointestinal Cancer Classification using Hybrid Rice Optimization with Deep Learning. IEEE Access.
11. Wang, S., S. Wang, Z. Wang, 2023. A survey on multi-omics-based cancer diagnosis using machine learning with the potential application in gastrointestinal cancer. Frontiers in Medicine, 9, 1109365.
12. BaradaranRezaei, H., A. Amjadian, M. V. Sebt, R. Askari, A. Gharaei, 2023. An ensemble method of machine learning to prognosticate gastric cancer. Annals of Operations Research, 328(1) (151-192).
13. Wang, Z., Y. Liu, X. Niu, 2023, April. Application of artificial intelligence for improving early detection and prediction of therapeutic outcomes for gastric cancer in the era of precision oncology. In Seminars in Cancer Biology. Academic Press
14. Nouman Noor, M., M. Nazir, S. A. Khan, O. Y. Song, I. Ashraf, 2023. Efficient gastrointestinal disease classification using pretrained deep convolutional neural network. Electronics, 12(7), 1557.
15. Tian, H., Z. Liu, J. Liu, Z. Zong, Y. Chen, Z. Zhang, H. Li, 2023. Application of machine learning algorithm in predicting distant metastasis of T1 gastric cancer. Scientific Reports, 13(1), 5741.
16. Kullmann, Friederike, Pamela L. Strissel, Reiner Strick, Robert Stoehr, Markus Eckstein, Simone Bertz, Bernd Wullich et al. 2023. Frequency of microsatellite instability (MSI) in upper tract urothelial carcinoma: comparison of the Bethesda panel and the Idylla MSI assay in a consecutively collected, multi-institutional cohort. Journal of Clinical Pathology 76, no. 2 (126-132).
17. SaiVenkatesh, C. H., C. Meriga, M. G. V. L. Geethika, T. Lakshmi Gayatri, V. B. K. L. Aruna, 2022. Modified resnetmodel for MSI and MSS classification of gastrointestinal cancer. In High Performance Computing and Networking: Select Proceedings of CHSN 2021 (pp. 273-282). Singapore: Springer Singapore.
18. Khan, Z., & R. Loganathan, R. 2022. Transfer Learning Based Classification of MSI and MSS Gastrointestinal Cancer (No. 7392). EasyChair.
19. Satish, Pooja, MallikarjunaswamySrikantaswamy, and NatarajKanathurRamaswamy. (2020). A Comprehensive Review of Blind Deconvolution Techniques for Image Deblurring. Traitement du Signal 37, no. 3
20. Deepika, Kumari, DeepikaPunj, and Jyoti. Spatial Domain Method for Image Analysis: A Grey-Level Computation Approach. In International Conference on Advances in Computing and Data Sciences, pp. 351-366. Cham: Springer Nature Switzerland, 2023.
21. Toresa, Dafwen, FanaWiza, Ahmad Ade Irwanda, WentiSasparitaAbiyus, EdriyansyahEdriyansyah, and TaslimTaslim. 2023. The cuckoo optimization algorithm enhanced visualization of morphological features of diabetic retinopathy. Journal of Applied Engineering and Technological Science (JAETS) 4, no. 2 (929-939).
22. Aslam, M. A., C. Xue, Y. Chen, A. Zhang, M. Liu, K. Wang, D. Cui, 2021. Breath analysis based early gastric cancer classification from deep stacked sparse autoencoder neural network. Scientific reports, 11(1), 4014.
23. Zhou, Q., B. Yong, Q. Lv, J. Shen, X. Wang, 2020. Deep autoencoder for mass spectrometry feature learning and cancer detection. IEEE Access, 8, 45156-45166.
24. Hemmerling, Scott A., Christine DeMyers, Jessica Parfait, Edwin Piñero, Melissa M. Baustian, MartijnBregman, Diana Di Leonardo et al. 2023. A community-informed transdisciplinary approach to coastal restoration planning: Maximizing the social and ecological co-benefits of wetland creation in Port Fourchon, Louisiana, USA. Frontiers in Environmental Science 11 1105671.
25. Guo, W., P. Cao, X. Wang, M. Hu, Y. Feng, 2022. Medicinal plants for the treatment of gastrointestinal cancers from the metabolomics perspective. Frontiers in Pharmacology, 13, 909755.
26. Almasinejad, P., A. Golabpour, M. R. MollakhaliliMeybodi, K. Mirzaie, A. Khosravi, 2021. A dynamic model for imputing missing medical data: A multiobjective particle swarm optimization algorithm. Journal of Healthcare Engineering, 2021.

27. Gunasekaran, H., K. Ramalakshmi, D. K. Swaminathan, M. Mazzara,. 2023. GIT-Net: an ensemble deep learning-based GI tract classification of endoscopic images. Bioengineering, 10(7), 809.
28. Lokesh, K., S. Srivastava, M. P. Kumar, S. Arun, S. Padmapriya, R. Krishnamoorthy, 2021, December. Detection of Stomach Cancer Using Deep Neural Network in Healthcare Sector. In 2021 3rd International Conference on Advances in Computing, Communication Control and Networking (ICAC3N) (pp. 521-526). IEEE.
29. Ghaleb Al-Mekhlafi, Z., E. Mohammed Senan, J. SulaimanAlshudukhi, B. Abdulkarem Mohammed, 2023. Hybrid Techniques for Diagnosing Endoscopy Images for Early Detection of Gastrointestinal Disease Based on Fusion Features. International Journal of Intelligent Systems, 2023.
30. Heo, J. J.H. Lim, H. R. Lee, J. Y. Jang, Y. S. Shin, D. Kim, C. H. Kim, 2022. Deep learning model for tongue cancer diagnosis using endoscopic images. Scientific reports, 12(1), 6281