DOI: 10.22037/jds.v42i1.45245
Original Article
ISSN 2645-4351

An Attention-Based Residual Connection Convolutional Neural Network for Classification Tasks in Computer Vision

Shahab Kavousinejad[a,b]

[a]Assistant Professor, Dept. of Orthodontics, School of Dentistry, Shahid Beheshti University of Medical Sciences, Tehran, Iran.
[b]Dentofacial Deformities Research Center, Research Institute of Dental Sciences, Shahid Beheshti University of Medical Sciences, Tehran, Iran
Correspondence to Shahab Kavousinejad (Email: dr.shahab.k93@gmail.com).
(*Submitted: 5 May 2024 – Revised version received: 22 May 2024 – Accepted: 25 May 2024– Published online: Spring 2024*)

Abstract

Objectives: In the field of medical and dental image analysis, the development of advanced deep learning architectures for precise classification tasks has become essential. The present study aims to introduce an innovative Attention-based Residual Connection Convolutional Neural Network (ARN-CNN) designed for accurate classification of medical images using the Med-MNIST (Medical Modified National Institute of Standards and Technology) dataset.

Methods: Attention mechanisms and residual connections were integrated into the ARN-CNN model to enhance feature extraction and prediction accuracy. The model's performance was evaluated through a comparative analysis with state-of-the-art CNN architectures on the challenging MNIST medical dataset, based on key metrics, including accuracy, precision, recall, and F1 score.

Results: The ARN-CNN model achieves a classification accuracy of 99.96% and a loss of 0.0037. These results showcase the superior performance of ARN-CNN in improving classification accuracy and its potential for enhancing medical image analysis.

Conclusion: The study demonstrates the crucial role that residual connections and attention processes play in capturing intricate details and maximizing information flow in the network. It highlights the potential of deep learning techniques for revolutionizing medical image analysis and laying the foundation for future investigation into automated medical and dental diagnosis and treatment in healthcare.

Keywords: Artificial intelligence, Machine learning, Deep learning

How to cite:
Kavousinejad S. An Attention-Based Residual Connection Convolutional Neural Network for Classification Tasks in Computer Vision. J Dent Sch 2024;42(1):14-25.

Introduction

The rapid advancement of machine learning techniques, specifically deep learning, has revolutionized various domains, including medical and dental image analysis and classification. [1, 2] One such significant application is the classification of medical and dental datasets, which plays a crucial role in assisting healthcare professionals in diagnosing and treating various diseases.[3] The classification of biomedical images holds promise for reducing diagnosis time and improving overall performance.[4]

Convolutional neural networks (CNNs) have gained significant traction in the field of image classification tasks in recent years.[5] However, to improve the performance of CNNs, researchers have explored several techniques, such as attention mechanisms[6] and residual connections[7]. Attention mechanisms allow models to focus on informative regions of an image[8], while residual connections help alleviate the vanishing gradient problem and enable better information flow through the network.[9] The primary objective of this research is to showcase the unique contributions of the ARN-CNN model in enhancing feature extraction and prediction accuracy in medical and dental image classification. The integration and combining of residual connections and attention processes in CNN models—which have not been thoroughly studied in the context of medical picture classification—makes this work innovative. The ARN-CNN aims to maximize the extraction of significant features from medical pictures by utilizing attention processes and residual connections, which should ultimately result in more accurate classification results. This method highlights the unique qualities of the ARN-CNN model and how it might progress the field of medical picture categorization.

The contributions of this paper are as follows:

- We introduce a novel ARN-CNN model that combines attention mechanisms and residual connections for image classification. This model represents a unique approach to enhance feature extraction and prediction accuracy.
- We describe the experimental setup on the Med-MNIST dataset[10], including the dataset used, evaluation metrics employed, and implementation details. The dataset contains 58,954 medical images with six classes. This dataset provides developers with the opportunity to train and test their models, enabling them to develop algorithms that can accurately identify medical conditions.[11] The advancements made through these algorithms have the potential to greatly enhance diagnosis in the fields of medicine[12, 13] and dentistry[14], leading to improved healthcare outcomes.

• We report the test results, which include accuracy, precision, recall, and F1-score metrics. These metrics provide a comprehensive evaluation of the performance of the ARN-CNN model in medical image classification.

This is a fundamental study aimed at introducing and evaluating a new convolutional neural network architecture that can be beneficial for future machine vision projects in the field of dentistry.

Materials

The proposed attention-based residual network architecture (Figures 1 and 2) implemented using the Tensorflow library (version 2.11.0) in the Python programming language (version 3.7.5) for image classification consists of several key components:

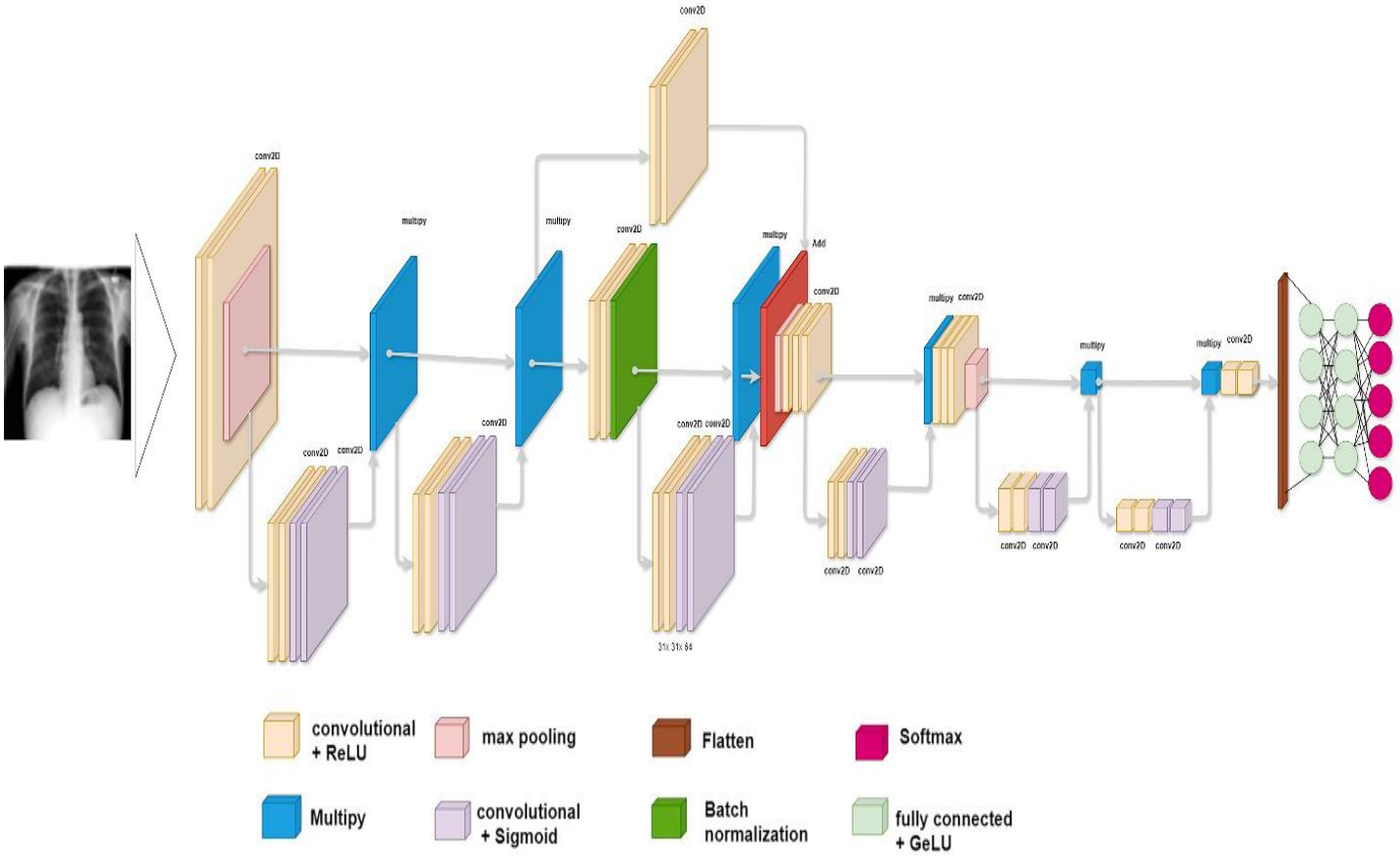

Figure 1: The proposed attention-based residual network architecture

Attention Module: This module is responsible for capturing relevant spatial information and generating attention maps. It takes the input image features and applies two convolutional layers. The first convolutional layer uses a ReLU activation function to extract relevant features, while the second convolutional layer uses a sigmoid activation function to generate attention weights. The attention weights are then multiplied elementwise with the input features to obtain attended features. By generating attention maps, the model can selectively attend to important regions and suppress irrelevant information. This attention mechanism helps improve the model's ability to capture fine-grained details and discriminative features, leading to enhanced classification performance. Figure 3 illustrates an example showcasing the functionality of the attention mechanism. Suppose we have an input and apply a convolutional layer (with a filter) with a ReLU activation function to it. Then, the output is passed through another convolutional layer with the sigmoid activation function. Finally, the output of the last layer is multiplied by the input. The resulting image will exhibit enhanced prominence in specific regions, effectively highlighting particular areas of the image.

Layer (type)	Output Shape	Param #	Connected to
input_1 (InputLayer)	(Batch size, 64, 64, 1)	0	
conv2d	(Batch size, 62, 62, 32)	320	input_1
max_pooling2d	(Batch size, 31, 31, 32)	0	conv2d
conv2d_1	(Batch size, 31, 31, 32)	1056	max_pooling2d
conv2d_2	(Batch size, 31, 31, 32)	1056	conv2d_1
multiply	(Batch size, 31, 31, 32)	0	max_pooling2d , conv2d_2
conv2d_3	(Batch size, 31, 31, 32)	1056	multiply
conv2d_4	(Batch size, 31, 31, 32)	1056	conv2d_3
multiply_1	(Batch size, 31, 31, 32)	0	multiply , conv2d_4
conv2d_5	(Batch size, 31, 31, 64)	18496	multiply_1
batch_normalization	(Batch size, 31, 31, 64)	256	conv2d_5
conv2d_6	(Batch size, 31, 31, 64)	36928	batch_normalization
batch_normalization_1	(Batch size, 31, 31, 64)	256	conv2d_6
conv2d_7	(Batch size, 31, 31, 64)	4160	batch_normalization_1
conv2d_8	(Batch size, 31, 31, 64)	4160	conv2d_7
multiply_2	(Batch size, 31, 31, 64)	0	batch_normalization_1 , conv2d_8
conv2d_9	(Batch size, 31, 31, 64)	2112	multiply_1
add	(Batch size, 31, 31, 64)	0	multiply_2 , conv2d_9
activation	(Batch size, 31, 31, 64)	0	add
max_pooling2d_1	(Batch size, 15, 15, 64)	0	activation
conv2d_10	(Batch size, 13, 13, 64)	36928	max_pooling2d_1
conv2d_11	(Batch size, 13, 13, 64)	4160	conv2d_10
conv2d_12	(Batch size, 13, 13, 64)	4160	conv2d_11
multiply_3	(Batch size, 13, 13, 64)	0	conv2d_10 , conv2d_12
conv2d_13	(Batch size, 11, 11, 256)	147712	multiply_3
max_pooling2d_2	(Batch size, 5, 5, 256)	0	conv2d_13
conv2d_14	(Batch size, 5, 5, 256)	65792	max_pooling2d_2
conv2d_15	(Batch size, 5, 5, 256)	65792	conv2d_14
multiply_4	(Batch size, 5, 5, 256)	0	max_pooling2d_2 , conv2d_15
conv2d_16	(Batch size, 5, 5, 256)	65792	multiply_4
conv2d_17	(Batch size, 5, 5, 256)	65792	conv2d_16
multiply_5	(Batch size, 5, 5, 256)	0	multiply_4 , conv2d_17
conv2d_18	(Batch size, 5, 5, 256)	65792	multiply_5
conv2d_19	(Batch size, 5, 5, 256)	65792	conv2d_18
multiply_6	(Batch size, 5, 5, 256)	0	multiply_5 , conv2d_19
conv2d_20	(Batch size, 5, 5, 256)	65792	multiply_6
conv2d_21	(Batch size, 5, 5, 256)	65792	conv2d_20
multiply_7	(Batch size, 5, 5, 256)	0	multiply_6 , conv2d_21
conv2d_22	(Batch size, 5, 5, 125)	32125	multiply_7
flatten	(Batch size, 3125)	0	conv2d_22
dense	(Batch size, 256)	800256	flatten
dropout	(Batch size, 256)	0	dense
dense_1	(Batch size, 6)	1542	dropout

Figure 2: The proposed attention-based residual network architecture shapes and parameters

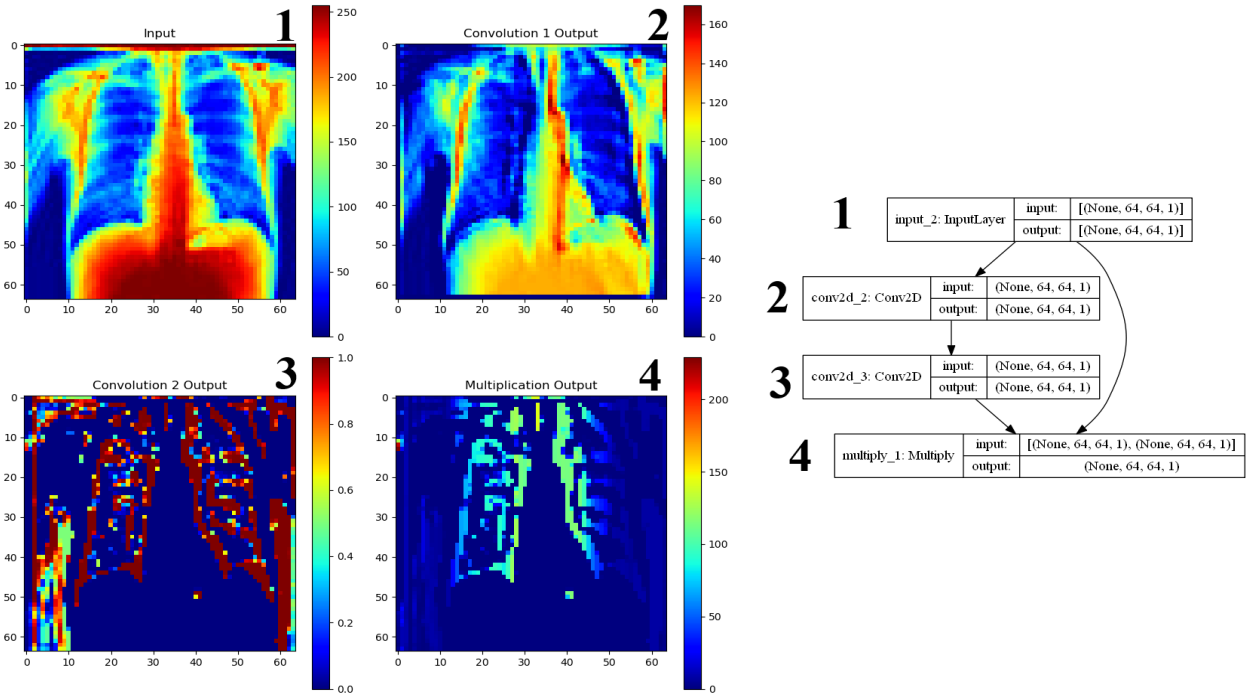

Figure 3: Illustration of the attention mechanism

• Residual Block: Skip connections and attention modules are included in the residual block, which is the fundamental unit of the network. The attended features from the previous layer are taken, two convolutional layers with batch normalization are applied, and the attention module is used to further refine the features. The purpose of adding the skip connection is to enhance feature representation and make information flow easier. The size of the skip connection features is adjusted using a 1x1 convolutional layer if they are different from the existing features. The attention module's output is combined with the skip connection features, and the resulting mixture is then run through an activation function (ReLU) to produce the final features. If the spatial dimensions of the features are greater than or equal to 2x2, a max pooling layer is applied to downsample the features. These connections enable the model to retain and propagate important features from earlier layers to later layers, mitigating the vanishing gradient problem and promoting better gradient flow during training. The residual connections also aid in feature reuse, allowing the model to learn more efficiently and effectively.

• Main Architecture: The overall model starts with an input tensor that represents the input image. The input tensor is passed through a 2D convolutional layer with 32 filters and a ReLU (Rectified Linear Unit) activation function. Max pooling is then applied to downsample the features. The attention module is applied twice consecutively to enhance the features. The features are then passed through a residual block with 64 filters and a kernel size of 3x3. Another 2D convolutional layer with 64 filters and a ReLU activation function is applied, followed by the attention module. Subsequently, a 2D convolutional layer with 256 filters and a ReLU activation function is applied, and max pooling is used for downsampling. The attention module is applied four times consecutively to further refine the features. Finally, a 1x1 convolutional layer with 125 filters and a ReLU activation function is applied to obtain the output features.

• Dense Layers: The output features are flattened and passed through a fully connected dense layer with 256 units and a GELU activation function. Dropout with a rate of 0.45 is applied for regularization. Another dense layer with the number of units equal to the class count is applied, followed by a softmax activation function to obtain the final class probabilities.

Figure 4 shows the activation function used in the ARN_CNN model.

The following formula represents the GELU (Gaussian Error Linear Unit) activation function. [15] It combines linear and non-linear components to transform an input value, x. (Equation1, 2)

$$GELU(x) = x\Phi(x) = x \cdot \frac{1}{2}[1 + erf(\frac{x}{\sqrt{2}})] \quad (1)$$

$$\text{erf}(x) = \frac{2}{\sqrt{\pi}} \int_0^x e^{-t^2} dt \quad (2)$$

In this equation:

- "x" represents the input value to the GELU function.
- Φ(x) represents the cumulative distribution function of the standard normal distribution (also known as the Gaussian function).
- The "erf" function is the error function, which is calculated as the integral of the Gaussian function. In this case, "erf" is used to compute the cumulative distribution function of the standard normal distribution.

To prevent overfitting, several techniques were employed:

- Dropout[16]: Dropout regularization was applied to the fully connected dense layers. Dropout randomly sets a fraction of the input units to zero during training, which helps prevent the model from relying too heavily on specific features and encourages the network to learn more generalizable representations.

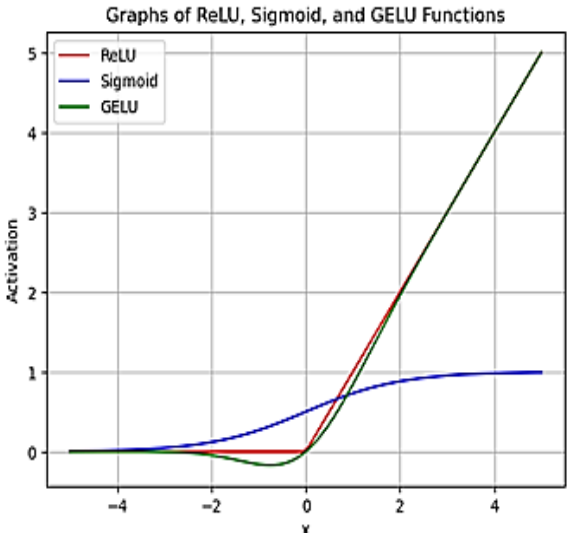

Figure 4: Activation functions used in ARN_CNN model (ReLU for Conv2d layers, sigmoid for attention and GELU in the dense layer)

- Regularization[17]: L2 (ridge) regularization and L1 (lasso) regularization were added to the dense layers' kernel and bias weights. These regularization terms impose a penalty on the model's complexity, discouraging large weight values and promoting sparsity in the learned representations. This regularization helps prevent overfitting by reducing the model's ability to fit noise into the training data.
- Batch Normalization: batch normalization layers were inserted after the convolutional layers. Batch normalization normalizes the activations of each layer, making the model more robust to changes in the input distribution. It helps stabilize the training process and reduces the reliance on specific input distributions, thereby preventing overfitting.
- Early Stopping[18]: During training, an early stopping mechanism was implemented. The training process was monitored based on the validation loss, and if the loss did not improve for a certain number of epochs, the training was stopped early. Early stopping helps prevent overfitting by stopping the training process before the model starts to overfit on the training data.

Results

Dataset Description

To test and evaluate the proposed architecture, we required a large dataset. However, since dentistry does not currently have standardized large datasets, we utilized the Medical MNIST dataset to assess its performance. The MedNIST dataset[10], also known as Medical MNIST (Figure 5), consists of 58,954 64x64 medical images. It includes six classes: abdominal CT, head CT, breast MRI, chest CT, CXR, and hand. Each class has a specific number of images for diagnostic and monitoring purposes.

- AbdomenCT: 10,000 CT scan images for abdominal organ-related conditions.
- Head CT: 10,000 CT scan images for brain-related ailments.
- BreastMRI: 8,954 MRI scan images for breast cancer and other breast-related conditions.
- Chest CT: 10,000 CT scan images for lung-related conditions like pneumonia and lung cancer.
- CXR: 10,000 chest X-ray images for lung and chest-related conditions.
- Hand: 10,000 X-ray images for bone and joint conditions in the hand.

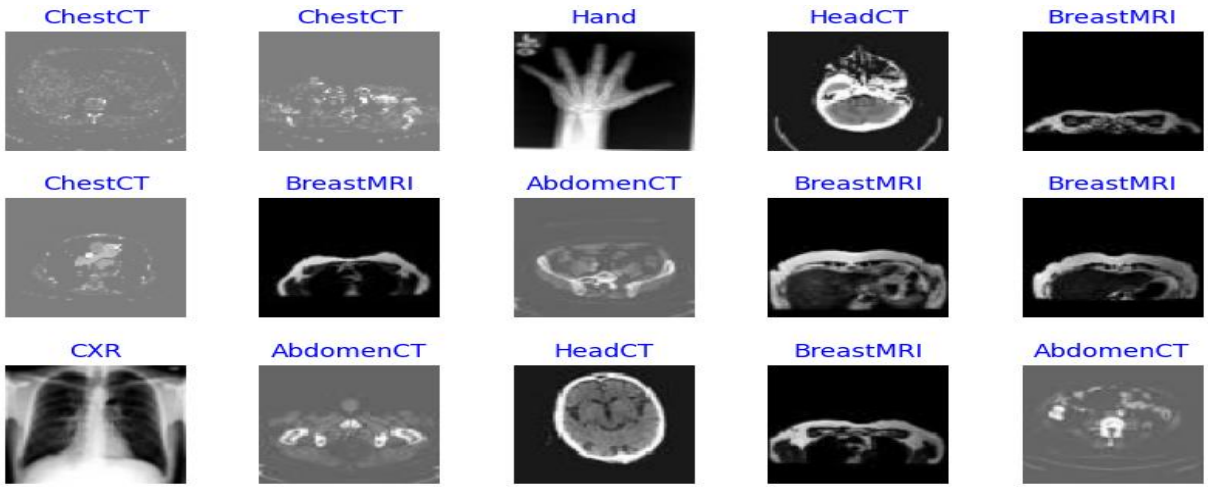

Figure 5: Some of the images in the dataset

The dataset is balanced, with a bar diagram representation of the number of images in each class. Sample images from the dataset are shown in Figure 6.

Preprocessing

During the preprocessing phase of the MedNIST dataset, the images were resized to 64x64, normalized, and converted to grayscale. This ensured uniformity and prepared the dataset for further analysis. The MedNIST dataset was split into three subsets: train_df (90% of the data), valid_df (6% of the data), and test_df (4% of the data). The splits were performed using the train_test_split function with shuffling and a random state of 123 for consistency. This division allows for training, validation, and evaluation of the model. (Figure 6).

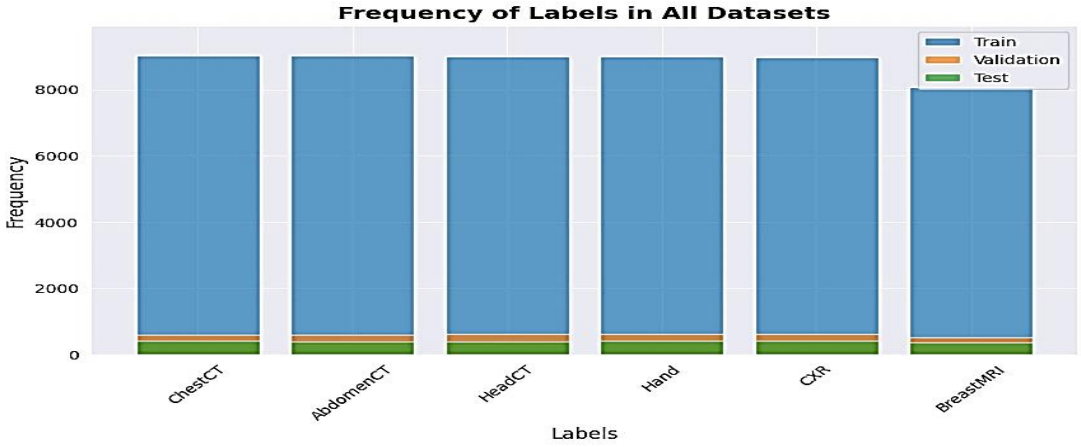

Figure 6: Frequency of labels in the dataset

Training and evaluation

During the training of the ARN-CNN model, the following data was recorded (Table 1):

- Epoch: The current epoch number out of 100.
- Loss: The training loss achieved during that epoch.
- Accuracy: The training accuracy achieved during that epoch.
- V_loss: The validation loss obtained during that epoch.
- V_acc: The validation accuracy obtained during that epoch.
- LR: The learning rate used during that epoch.
- Next LR: The learning rate scheduled for the next epoch.
- Monitor: The metric being monitored for early stopping (in this case, validation loss).
- % Improv: The percentage improvement in the monitored metric compared to the previous epoch.
- Duration: The duration of the epoch in seconds.

Table 1: Training performance metrics over epochs

Epoch	Loss	Accuracy	V_loss	V_acc	LR	Next LR	Monitor	% Improv	Duration
1	0.455	99.129	0.12981	99.859	0.001	0.001	val_loss	0.00	104.09 seconds
2	0.117	99.945	0.08788	99.943	0.001	0.001	val_loss	32.30	93.52 seconds
3	0.086	99.972	0.06450	99.972	0.001	0.001	val_loss	26.61	92.83 seconds
4	0.067	99.987	0.05427	99.972	0.001	0.001	val_loss	15.86	90.68 seconds
5	0.059	99.932	0.04867	99.972	0.001	0.001	val_loss	10.32	83.18 seconds

Training Procedure

The ARN-CNN model was trained on the MNIST medical dataset. The training process involved optimizing the model using the MedNIST dataset with the Adamax optimizer[19] and categorical cross-entropy loss function.[20] The following equations describe the update process in the Adamax optimization algorithm, where θ represents the parameters, η is the learning rate, u_t is the exponentially weighted infinity norm of the gradients, m^_t is the exponentially decaying average of past gradients, β_1 and β_2 are hyperparameters controlling the decay rates, g_t is the current gradient, and ε is a small value for numerical stability. (Equation 3-6)

$$\theta_{t+1} = \theta_t - \frac{\eta}{u_t}\hat{m}_t \quad (3)$$

$$u_t = \max(\beta_2 u_{t-1}, |\, g_t \,|, \varepsilon) \quad (4)$$

$$\hat{m}_t = \frac{m_t}{1-\beta_1^t} \quad (5)$$

$$m_t = \beta_1 m_{t-1} + (1-\beta_1) g_t \quad (6)$$

The categorical cross-entropy loss (L) for multi-class classification problems is represented by the following equation: For the true labels (y), it determines the average negative logarithm of the anticipated probabilities (y^). The reciprocal of the number of samples (N) is used to scale the loss. Reducing this loss is intended in order to increase the precision of the estimated probability. (Equation 7)

$$L(y, \hat{y}) = -\frac{1}{N}\sum_{i=1}^{N}\sum_{j=1}^{C} y_{ij}\, log(\hat{y}_{ij}) \quad (7)$$

Training occurred over multiple epochs and batches, with a batch size of 40 determined by the dataset size. Dropout regularization with a rate of 0.45 prevented overfitting. Early stopping with a patience of 1 adjusted the learning rate, and training was halted if the metric didn't improve for three consecutive epochs. The learning rate was reduced by a factor of 0.5 when the metric failed to improve. The learning rate adjustment, reducing the learning rate by a factor of 0.5 when the monitored metric fails to improve, helps fine-tune the model's performance. This adaptive adjustment allows the model to make smaller updates when it is close to convergence, leading to more stable training. An experimental "dwell" features reverted weights to the previous epoch if the metric didn't improve. This procedure optimized the model's performance, ensuring accurate classification on the MedNIST dataset. Figure 7 shows the learning curve of the model.

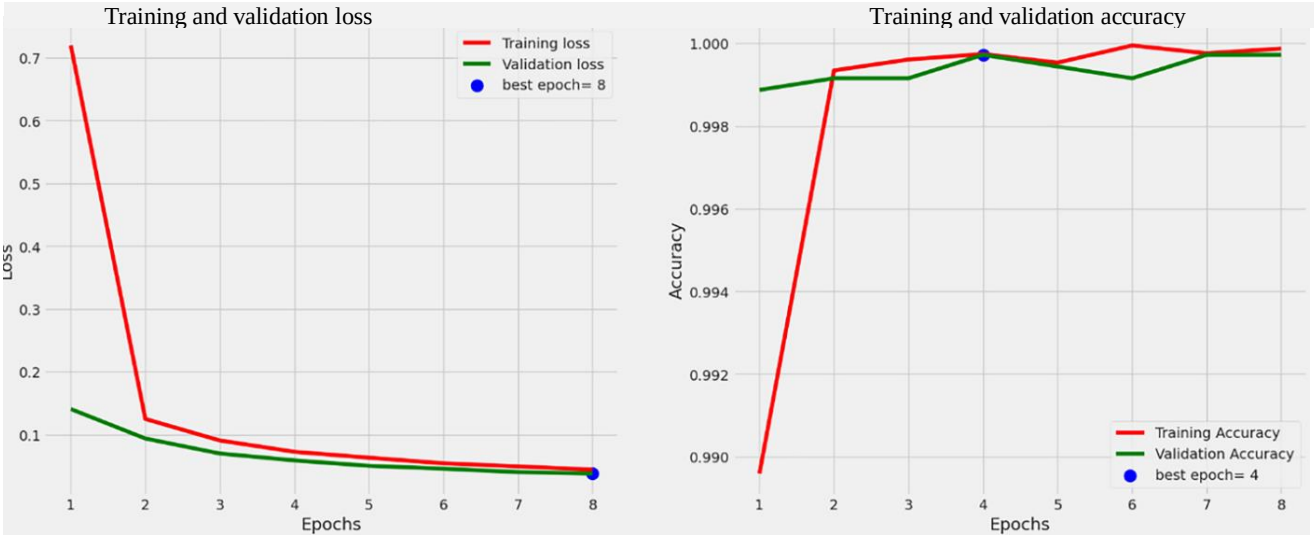

Figure 7: Learning curve of the proposed model

Evaluation Metrics

To assess the performance of the ARN-CNN model on the MNIST medical dataset, we used a separate test set consisting of unseen medical images. We employed several evaluation metrics, including accuracy, precision, recall, F1 score, and log loss. Each metric provides valuable insights into the model's performance and its ability to accurately classify medical images.

These evaluation metrics collectively demonstrate the outstanding performance of the ARN-CNN model on the MNIST medical dataset, showcasing its ability to accurately classify medical images with high precision, recall, and overall accuracy on previously unseen data.

In addition to the previously mentioned evaluation metrics, we also analyzed the uncertainty of the ARN-CNN model's predictions for each class in the MNIST medical dataset. The following information provides insights into the uncertainty of the model's predictions.

• Calculating Maximum Probability in Prediction: For each test data, we consider the model's predicted probabilities as a vector called y_pred. The maximum probability of prediction for each test data can be calculated using the following formula: (Equation 8)

$$y_{\text{pred\_prob}} = max(y_{\text{pred}}) \quad (8)$$

where $y_{\text{pred\_prob}}$ is a vector containing the maximum probability of prediction for each test data.

• Calculating Normalized Probabilities: We first calculate the total of the model's projected probability for every test set of data before calculating the normalized probabilities. The normalized probabilities for each test set of data are then determined by dividing the greatest probability in the prediction by this total. The following formula may be used to get normalized probabilities: (Equation 9)

$$y_{pred\_prob\_normalized} = \frac{y_{pred\_prob}}{\sum y_{pred}} \quad (9)$$

where $y_{\text{pred\_prob\_normalized}}$ is a vector containing the normalized probabilities for each test data.

• Calculating Prediction Uncertainty Standard Deviation: The prediction uncertainty of the model's predictions is calculated for each class. The standard deviation of the normalized probabilities connected to each class is used to compute this uncertainty. The following formula may be used to get the prediction uncertainty standard deviation: (Equation 10)

$$\text{prediction\_uncertainty} = \text{std\_dev}\left(y_{\text{pred\_prob\_normalized}}\right) \quad (10)$$

where $y_{\text{pred\_prob\_normalized}}$ is a vector containing the normalized probabilities associated with each class.

• Calculating Entropy for Each Class Prediction: For every class prediction, the entropy is calculated. The entropy function in the Python programming language's scipy.stats package may be used to calculate entropy, which is a measure of uncertainty. The following is the formula to determine entropy: (Equation 11)

$$H(p) = -\sum_{x \in X} p(x) \log_b p(x)$$

$$entropies = [entropy(predictions)\ for\ predictions\ in\ ypred] \quad (11)$$

Shannon's entropy[21], denoted as H(p), is a measure in information theory that quantifies the uncertainty or information content in a probability distribution. It is computed using the formula H(p) = -∑_(x∈X) p(x) log_b p(x), where X represents a set of events and p(x) is the probability of event x. The entropy increases as the distribution becomes more unpredictable and decreases as it becomes more certain. The base of the logarithm, denoted as b, determines the unit of measurement for entropy.

By calculating the entropy of each probability distribution using the `entropy` function, the function aims to measure the uncertainty or information content of the model's predictions for each class.

Results

Based on the evaluation results obtained using the ARN-CNN model on the MNIST medical dataset's test set, the metrics are summarized in Table 2.

Table 2- Performance metrics for ARN-CNN model predictions on the test data

Metric	Value	Interpretation
Precision	1.0000	Perfect precision for all classes
Recall	1.0000	Correct identification of positive samples
F1 Score	1.0000	Perfect balance between precision and recall
Accuracy	0.9996	Correct classification of 99.96% of the samples
Cohen's Kappa	0.99	Excellent agreement between model's predictions and labels
Log Loss	0.0037	Highly confident and accurate probability estimates

The mean entropy across the predictions for each class is also calculated to provide an overall measure of uncertainty. (Table 3 and Table 4)

Table 3 - Uncertainty of model predictions (Entropy)

Class	Uncertainty (Entropy)	Uncertainty Level
AbdomenCT	0.04	Moderate uncertainty
BreastMRI	0.01	Low uncertainty
CXR	0.02	Low uncertainty
ChestCT	0.03	Moderate uncertainty
Hand	0.02	Low uncertainty
HeadCT	0.02	Low uncertainty

Table 4 - Uncertainty of model predictions (Standard Deviation):

Class	Standard Deviation	Uncertainty Level
AbdomenCT	0.00	Low uncertainty
BreastMRI	0.00	Low uncertainty
CXR	0.01	Low uncertainty
ChestCT	0.03	Moderate uncertainty
Hand	0.01	Low uncertainty
HeadCT	0.00	Low uncertainty

These uncertainty measures provide insights into the model's confidence in its predictions for each class. Overall, the model demonstrates low to moderate uncertainty, indicating a high level of confidence in its predictions for the different classes in the MNIST medical dataset. In general, lower uncertainty values indicate higher confidence and certainty in the model's predictions. Therefore, classes with lower uncertainty values (such as breast MRI, CXR, hand, and head CT) are associated with more accurate and certain predictions. On the other hand, classes with slightly higher uncertainty values (such as AbdomenCT and ChestCT) suggest a relatively higher level of uncertainty in the model's predictions for those classes. Figure 8 shows the confusion matrix and model prediction confidence (based on normalized SoftMax probabilities) on the test data.

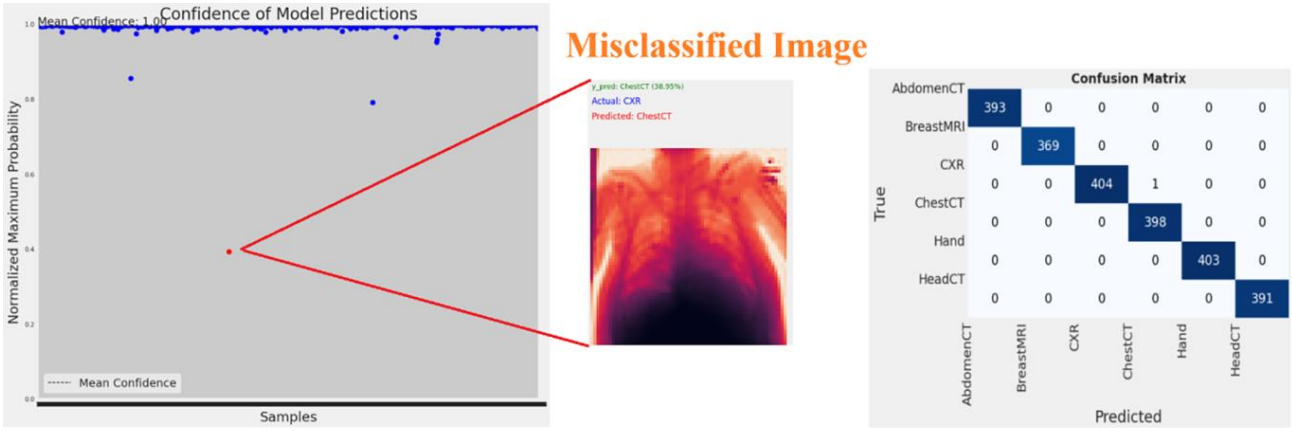

Figure 8: Confusion matrix and model prediction confidence on test data

Figure 9 presents a specific focus on the last Conv2D layer within the ARN-CNN model. This layer plays a crucial role in extracting high-level features from the input data. Through the utilization of Grad-CAM, the visualization showcased in Figure 9 generates heatmap-like images, effectively highlighting the significant regions within the input image that contribute to the model's prediction.

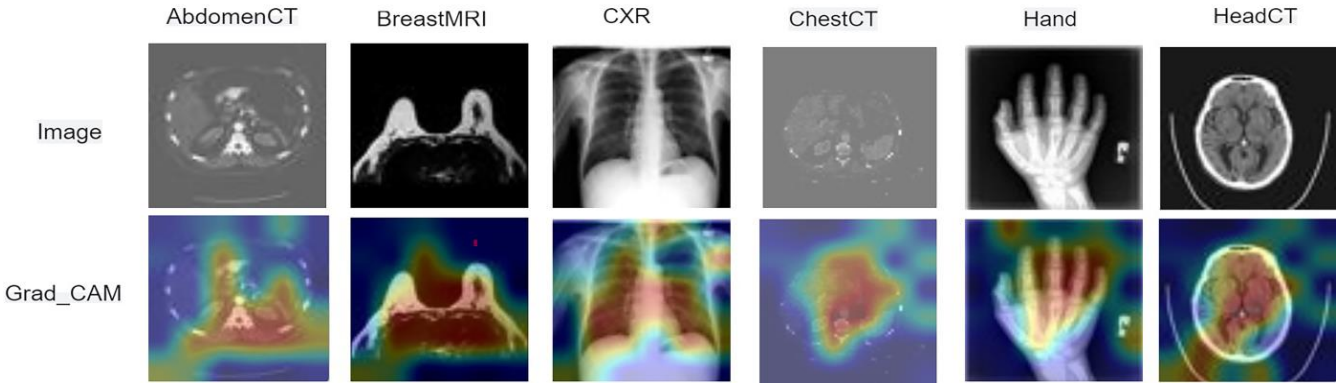

Figure 9: Enhanced visualization of Grad_CAM for test data samples

Discussion

Numerous studies have suggested the attention-based method as an effective approach for enhancing convolutional neural networks (CNNs).[16] Ran Gu et al.[22] introduced CA-Net, a Comprehensive Attention Convolutional Neural Network, for explainable medical image segmentation. This network leverages multiple attention mechanisms to enhance segmentation accuracy and explainability by focusing on important spatial positions, channels, and scales simultaneously. CA-Net includes a joint spatial attention module to emphasize the foreground region, a channel attention module to recalibrate feature responses, and a scale attention module to adapt to object sizes. Experimental results showed significant improvements in segmentation Dice scores for various medical imaging tasks compared to existing networks like U-Net. Additionally, CA-Net offers better explainability through visualizing attention weight maps. Anwar et al. [23] reported on the use of deep convolutional neural networks for medical image analysis, specifically focusing on tasks such as medical image retrieval, brain tumor segmentation, Alzheimer's disease classification, and optic disc localization. Their research highlights the potential of deep learning techniques for improving the accuracy and efficiency of various medical image analysis tasks. Zhang et al. [24] introduced an innovative strategy for medical image analysis through the utilization of AutoML. Their methodology combines automated optimization techniques for data augmentation with neural architecture to achieve better performance. Several datasets were used in extensive experiments to demonstrate the effectiveness of their suggested technique in comparison to current approaches. Jiang et al. [25] proposed DSLSA-Net, a Deep Neural Network (DNN) for medical image classification. DSLSA-Net selectively attends to informative layers, enhancing learning efficiency. Their approach achieves superior performance with fewer labeled samples compared to other methods, as demonstrated in experiments on public datasets. Rajaraman et al. [26] systematically analyzed model calibration's impact on performance using chest X-rays and fundus images. They evaluated classifiers (VGG-16, DenseNet-121, Inception-V3, and EfficientNet-B0) with varied training dataset imbalances. Three calibration methods (Platt scaling, beta, and spline) based on the ECE metric were employed, along with classification using default thresholds. Zheng et al. [27] proposed an approach called Implicit Distribution Representation (IDR) for label distribution learning. IDR leverages an implicit representation of label distribution, leading to more efficient learning and improved generalization performance. Valliani et al. [28]proposed a general technique using Generative Adversarial Networks (GANs) to address the issue of dataset shift. They applied the DenseNet architecture for classifying opacities in chest radiographs and the LeNet architecture for handwritten digit recognition. Nawaz et al. [29] developed a deep learning approach using the EfficientNet model to detect and classify chest abnormalities in X-ray images. They achieved high accuracy in disease localization and categorization, with an AUC score of 0.9080 and an IOU of 0.834. Awad et al. [30] proposed a robust approach for classifying and detecting big medical data. They used logistic regression and YOLOv4 algorithms but enhanced their performance by incorporating advanced parallel k-means pre-processing. Their approach aimed to accurately classify large amounts of medical data and detect medical images, making these algorithms more reliable for medical applications. Esraa Hassan et al. [11] proposed a novel architecture called MQCNN for the classification of medical images in the MNIST dataset. The architecture combines a quantum convolutional layer with a modified ResNet pre-trained model. Experimental results showed that MQCNN outperformed other comparable works, achieving 99.6% accuracy. The authors concluded that MQCNN can be a promising approach for improving the performance of biomedical image classification.

In our study, we not only validated the efficacy of this method but also proposed a novel combination of attention and residual modules, which led to accelerated convergence of the neural network with a reduced number of epochs. Consequently, our model achieved higher accuracy and lower loss. The visual analysis of Grad_CAM images further highlighted the ability of this approach to swiftly identify meaningful pixels in the images. In comparison to the method reported in previous studies on medical MNIST datasets, like the study by Esraa Hassan et al. [11] (Quantum Convolutional Neural Networks (QCNN) with an accuracy of 99.6%), the proposed ARN-CNN method demonstrated higher accuracy (99.9%). This can be attributed to the combination of residual connections and attention modules in the ARN-CNN model with reduced epochs and training time. The innovative integration and combination of residual connections and attention modules has the potential to introduce new advancements in machine vision within the field of medical sciences. Table 6 illustrates the performance of different models based on their accuracy and loss values.

Table 6 - Comparison of model accuracy and loss		
Model	Accuracy	Loss
ResNet (50)[11]	0.986	-
CNN • Attention = 0 • Residual block = 1	0.967	0.106
CNN • Attention = 1 • Residual block = 0	0.975	0.012
ARN-CNN • Attention = 1 • Residual block = 1	0.996	0.0037

To mitigate the risk of overfitting the ARN-CNN model, we incorporated specific mechanisms into our model architecture. The experimental results demonstrated that our model likely does not exhibit signs of overfitting, affirming the effectiveness of our preventive measures. Moreover, we investigated the performance of different optimization algorithms and found that the Adamax optimizer outperformed others in terms of model optimization. [19] Therefore, we employed the Adamax optimizer in our study and observed comparable performance to Adam while outperforming the stochastic gradient descent (SGD) optimizer in terms of loss reduction.

In the dense layers of our model, we employed the GELU activation function. Previous studies have consistently shown that GELU activation tends to yield superior results compared to other activation functions. [15] Although we did not observe significant performance differences when employing the GELU activation function in the convolutional layers, we did notice a modest improvement in model performance compared to the standard ReLU activation when applied to the dense layers. (Figure 10)

The proposed architecture combines attention-based methods, residual connections, and regularization techniques to enhance the performance of the CNN in terms of accuracy, robustness, and generalization. The flexibility of the architecture allows for customization and adaptation to various tasks and datasets, making it a valuable tool for researchers and practitioners in the field of deep learning. In future directions, we can focus on optimizing the ARN-CNN model for broader datasets, going beyond MNIST to enhance its applicability. Additionally, exploring advanced architectural enhancements such as attention mechanisms and residual connections can improve the model's ability to capture intricate features in medical images. Transfer learning and domain adaptation techniques can be applied to fine-tune the model for specific medical imaging domains. Enhancing interpretability through attention visualization and feature attribution methods can improve the understanding of the model's predictions. Integrating the ARN-CNN model with clinical decision support systems can enable accurate diagnoses and treatment decisions. Improving robustness to variations in input data and enhancing uncertainty estimation techniques can enhance the model's reliability. Optimizing scalability and computational efficiency for large-scale medical imaging datasets will facilitate efficient analysis. Finally, conducting collaborative research and validation studies can validate the effectiveness of the ARN-CNN model in real-world medical scenarios. In the next study, we plan to experiment with different activation functions on various datasets using this model. We will explore the performance and behavior of the model when different activation functions are applied. The present study has uncovered an optimized architecture or structure which we intend to utilize in upcoming projects focusing on dental image classification and diagnostic tasks, particularly within the field of dentistry.

Figure 10: Performance comparison of train losses using different activation functions in dense layer of the proposed model.

Conclusion

In conclusion, the ARN-CNN model, integrating attention mechanisms and residual connections, has significantly improved the accurate classification of medical images on the MNIST dataset. Through rigorous experiments, it has shown enhanced feature extraction and precise predictions. The model's high-performance metrics and computational efficiency highlight its potential for real-world applications in medical image analysis. Future research can focus on optimizing and extending the model for broader datasets in medical and dental images, advancing the field of biomedical computer vision.

Acknowledgement: None
Author Contributions: S.K. conceived and designed; conducted the experiments; analyzed the data; wrote the manuscript.

Funding: No funding was received for this research.
Ethical Approval Code: Not applicable
Informed Consent Statement: None
Data Availability Statement: None
Conflict of Interest: No Conflict of Interest Declared■

References

1. Hassan E, Shams MY, Hikal N A,Elmougy E. A novel convolutional neural network model for malaria cell images classification. Comput Mater Contin. 2022; 72(3): 5889-907.
2. Kayalibay B, Jensen G, van der Smagt P. CNN-based segmentation of medical imaging data. arXiv preprint arXiv.2017;1701:03056.
3. Anand R, Sowmya V, Gopalakrishnan EA, Soman KP. Modified Vgg deep learning architecture for Covid-19 classification using bio-medical images. IOP Conf Ser Mater Sci Eng. 2021; 1084(1): 012001.
4. Tapaswi S, Joshi RC. Classification of bio-medical images using neuro fuzzy approach. International Conference on Database Systems for Advanced Applications. 2004; 568-81.
5. O'Shea K, Nash R. An introduction to convolutional neural networks. arXiv preprint arXiv. 2015;1511:08458.
6. Obeso AM, Benois-Pineau J, Vázquez MSJ, Acosta A Á R. Visual vs internal attention mechanisms in deep neural networks for image classification and object detection. Pattern Recognit. 2022; 123:108411.
7. Garcia ACP. Convolutional Neural Networks and Residual Connections for Cow Teat Image Classification." arXiv preprint arXiv.2014;1(1):1409.1556.
8. Xu L, Huang J, Nitanda A, Asaoka R, Yamanishi K. A novel global spatial attention mechanism in convolutional neural network for medical image classification. arXiv preprint arXiv.2020;2007.15897.
9. Abdi M, Nahavandi S. Multi-residual networks: Improving the speed and accuracy of residual networks. arXiv preprint arXiv. 2016;1609:05672.
10. MedicalMNIST. https://www.kaggle.com/ datasets/ andrewmvd / medical-mnist.
11. Hassan E, Hossain MS, Saber A, Elmougy S, Ghoneim A, Muhammad G. A quantum convolutional network and ResNet (50)-based classification architecture for the MNIST medical dataset. Biomed Signal Process Control. 2024;87:105560.
12. Sarvamangala DR, Kulkarni RV. Convolutional neural networks in medical image understanding: a survey. Evol. Intell. 2022;15(1):1-22.
13. Tajbakhsh N, Shin JY, Gurudu SR, Hurst RT, Kendall CB, Gotway MB, et al. Convolutional neural networks for medical image analysis: Full training or fine tuning? IEEE Trans Med Imaging.7;35(5):1299-312.
14. Schwendicke F, Golla T, Dreher M, Krois J. Convolutional neural networks for dental image diagnostics: A scoping review. J Dent. 2019;91:103226.
15. Lee M. Mathematical analysis and performance evaluation of the gelu activation function in deep learning. J. Math.2023; 2023:1-13.
16. Li J, Jin K, Zhou D, Kubota N, Ju Z. Attention mechanism-based CNN for facial expression recognition. Neurocomputing. 2020;411:340-50.
17. Murugan P, Durairaj S. Regularization and optimization strategies in deep convolutional neural network. arXiv preprint arXiv. 2017;1712:04711.
18. Song H, Kim M, Park D, Lee JG. Prestopping: How does early stopping help generalization against label noise?. 2019; P:1-17.
19. Vani S, Rao TM. An experimental approach towards the performance assessment of various optimizers on convolutional neural network. In2019 3rd international conference on trends in electronics and informatics (ICOEI). 2019 (pp. 331-336). IEEE.
20 Mao A, Mohri M, Zhong Y. Cross-entropy loss functions: Theoretical analysis and applications. arXiv preprint arXi. 2023;2304:07288.
21 S. Vajapeyam, "Understanding Shannon's entropy metric for information," arXiv preprint arXiv.2014;1405:2061.
22 Gu R, Wang G, Song T, Huang R, Aertsen M, Deprest J, Ourselin S, Vercauteren T, Zhang S. CA-Net: Comprehensive attention convolutional neural networks for explainable medical image segmentation. IEEE Trans Med Imaging. 2020;40(2):699-711.
23 Anwar SM, Majid M, Qayyum A, Awais M, Alnowami M, Khan MK. Medical image analysis using convolutional neural networks: a review. J Med Syst. 2018;42:1-3.
24 Zhang J, Li D, Wang L, Zhang L. Auto machine learning for medical image analysis by unifying the search on data augmentation and neural architecture. arXiv preprint arXiv. 2022;2207:10351.
25 Jiang P, Liu J, Wang L, Ynag Z, Dong H, Feng J. Deeply Supervised Layer Selective Attention Network: Towards Label-Efficient Learning for Medical Image Classification. arXiv preprint arXiv2022;2209:13844.
26 Rajaraman S, Ganesan P, Antani S. Deep learning model calibration for improving performance in class-imbalanced medical image classification tasks. PloS one. 2022;17(1):e0262838.
27 Zheng Z, Jia X. Label distribution learning via implicit distribution representation. arXiv preprint arXiv.2022;2209:13824.
28 Valliani AA, Gulamali FF, Kwon YJ, Martini ML, Wang C, Kondziolka D, Chen VJ, Wang W, Costa AB, Oermann EK. Deploying deep learning models on unseen medical imaging using adversarial domain adaptation. Plos one. 2022;17(10):e0273262.
29 Nawaz M, Nazir T, Baili J, Khan MA, Kim YJ, Cha JH. Cxray-effdet: chest disease detection and classification from x-ray images using the efficientdet model. Diagnostics. 2023;13(2):248.
30 Awad FH, Hamad MM, Alzubaidi L. Robust classification and detection of big medical data using advanced parallel k-means clustering, yolov4, and logistic regression. Life. 2023;13(3):691.