Deep Learning Architectures for Urolithiasis Classification: A Comparative Analysis of DNN, MLP, and Autoencoder-based Models

Ibtasam Ur Rehman[1], Abdulraqeb Alhammadi[2], and Jibran K. Yousafzai[3]

[1] [1] Ho Chi Minh City University of Technology (HCMUT), Ho Chi Minh City, Vietnam
ribtrasam.sdh231@hcmut.edu.vn
[2] [2] Faculty of Artificial Intelligence, Universiti Teknologi Malaysia, Kuala Lumpur, Malaysia
abdulraqeb.alhammadi@utm.my
[3] [3] College of Engineering and Applied Sciences, American University of Kuwait, Kuwait
jyousafzai@auk.edu.kw

Abstract. Kidney stones remain a frequent and painful urological condition, motivating the need for reliable and quick diagnostic tools. In this work, we compare three deep learning models for classifying kidney stone images: a Deep Neural Network (DNN), a Multi-Layer Perceptron (MLP), and an Autoencoder-based DNN (AE-DNN). Using 9,416 labeled images from an open-access dataset, each model was trained and evaluated under the same experimental setup. Among them, the MLP achieved the best overall accuracy of 99.67%, demonstrating strong precision (99.67%) and specificity (99.44%) while requiring less training time than the deeper networks. The AE-DNN performed comparably, achieving an accuracy of 99.47%. Although the findings are promising, they are based on a binary classification task with down-sampled images. Further testing on external datasets will be essential before clinical translation.

Keywords: Kidney Stone Classification · Deep Learning · Medical Imaging · Neural Networks · Computer-Aided Diagnosis

1 Introduction

1.1 Background and Motivation

Kidney stone disease is a significant global health issue, with incidence rates rising and impacting approximately 10-15% of the worldwide population [1,2]. Traditional diagnostic techniques, such as ultrasound scans, standardized clinical biomarker analysis, and manual interpretation by radiologists, can be time-consuming and prone to interobserver variability. Deep learning-based automated systems offer the opportunity to enhance diagnostic accuracy, thereby reducing interpretation time and increasing access to healthcare. Recent research

in medical imaging and computational techniques has shown great promise for enhancing kidney stone categorization and detection, potentially leading to the development of effective and precise diagnostic instruments.

1.2 Challenges and Research Contribution

Classical kidney stone diagnosis presents challenges, including stone variability, limited access to imaging, and subjective interpretation. Computational approaches to kidney stone diagnosis have been explored, but comprehensive comparisons of deep learning architectures for kidney stone classification remain limited. This research evaluates the three neural network architectures (DNN, MLP, and AE-DNN) and provides insight into their performance and efficiency for kidney stone detection.

1.3 Related Work in Kidney Stone Classification

Recent advancements have shown the effectiveness of machine learning and deep learning approaches for kidney stone classification across multiple data modalities and imaging techniques. Several studies have focused on traditional machine learning approaches using custom features. Asaye et al. [3] employed a machine learning approach to detect kidney stones in ultrasound images, utilizing 410 labeled samples. This approach incorporated Gabor filtering and threshold-based segmentation, along with GLCM texture features. Their approach achieved 98.4% accuracy with the KNN algorithm, illustrating the potential of traditional feature-based methods. Iparraguirre-Villanueva et al. [4] evaluated multiple ML algorithms using non-imaging clinical parameters, with Logistic Regression achieving 78% accuracy.

Furthermore, Panda et al. [5] reported 93% accuracy using urine analysis data, which included physical features such as pH and calcium concentration. Khan et al. [6] developed an automated detection system employing median filtering for noise reduction and adaptive thresholding for segmentation. By achieving 96.82% accuracy and 92.16% sensitivity, their method demonstrated computational efficiency while overcoming challenges such as speckle noise and low contrast in ultrasound images.

Deep learning approaches have shown great potential in addressing the challenges of kidney stone classification and diagnosis. Kumar et al. [7] proposed a hybrid CNN-ResNet model that achieved 90.9% accuracy by leveraging the capabilities of CNN feature extraction and the advantages of ResNet in deep learning. Further advancing deep learning applications, Liu et al. [8] developed an integrated radiomics and deep learning model for classifying kidney stone types on CT urography, achieving exceptional performance with an AUC of 0.95. Jacob et al. [9] compared three deep learning models for kidney stone identification from CT scans, finding that ResNet50 outperformed both VGG16 and standard CNN by 7.7% and 4.5%, respectively.

Despite these advancements, a need remains for comparative studies of different neural network architectures specifically optimized for kidney stone classification using clinical biomarkers and imaging data. Our research addresses this gap by providing an evaluation of DNN, MLP, and AE-DNN architectures on a dataset of kidney images, offering information on their relative strengths, limitations, and clinical applicability for kidney stone detection.

2 Methodology

This study investigates deep learning techniques for the binary classification of kidney stone images, employing a methodology that encompasses data preprocessing, model development, and a comprehensive evaluation strategy. Three neural network architectures are implemented and evaluated: DNN, MLP, and AE-DNN. The experimental workflow, illustrating the data pipeline and model training process, is shown in Figure 1. To guarantee reproducibility, all random seeds were set to 123 at the beginning of the execution.

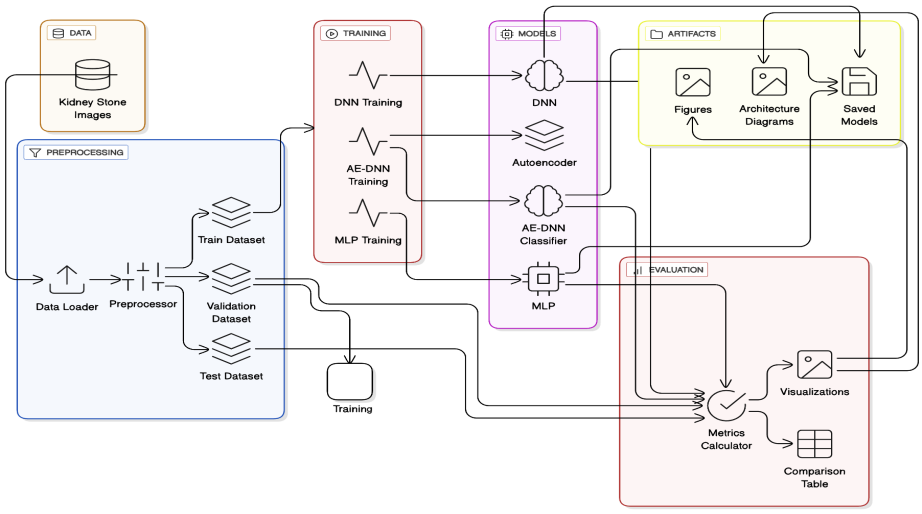

Fig. 1. Methodological Framework for Kidney Stone Classification using Deep Learning Approaches

2.1 Dataset Description and Preprocessing

Data Collection and Characteristics The study utilized the publicly available Kidney Stone Classification dataset from Kaggle [10], comprising 9,416 high-quality labeled images categorized into two distinct classes: **Normal** (4,708 images) and **Stone** (4,708 images). The dataset exhibits a balanced class distribution, hence mitigating class imbalance bias during model training and ensuring

robust performance evaluation. Moreover, images are systematically collected from multiple clinical sources under uniform acquisition conditions. All images are standardized to a resolution of 64×64 pixels with RGB color channels, resulting in an input dimension of $12,288$ features per sample ($64 \times 64 \times 3$). The resolution reduction was incorporated to ensure computational efficiency while maintaining diagnostic relevance. Preliminary analysis confirmed that stone characteristics remained discernible at this scale. The dataset splitting followed a structured approach with stratified sampling to maintain class distribution integrity across all subsets:

$$\begin{aligned} \text{Training Set} \quad & 6,026 \text{ images} \approx 64\% \\ \text{Validation Set} \quad & 1,507 \text{ images} \approx 16\% \\ \text{Test Set} \quad & 1,883 \text{ images} \approx 20\% \end{aligned}$$

$$\text{Total} = 6,026 + 1,507 + 1,883 = \mathbf{9,416}$$

Data Cleaning, Validation and Preprocessing Comprehensive data validation confirmed that there were no missing values or corrupted files within the dataset. Image preprocessing included pixel normalization to scale intensity values to the range [0,1] by the following transformation:

$$I_{\text{normalized}} = \frac{I_{\text{original}}}{255} \tag{1}$$

where I_{original} signifies original pixel intensity values. This normalization ensured stable gradient computation during neural network training and accelerated convergence. For model training, images were flattened into one-dimensional vectors to suit fully connected neural network architectures.

$$X_{\text{flat}} = \text{flatten}(I_{\text{normalized}}) \in \mathbb{R}^{12288} \tag{2}$$

Data Augmentation and Training Strategy Given the size and inherent balance of the dataset, explicit data augmentation techniques were not employed. This decision was made to allow the models to learn from authentic clinical representations without synthetic alterations. The dataset was processed using TensorFlow's data pipeline, which includes automatic caching and prefetching, to optimize training efficiency. The dataset splits were created with a fixed random seed (123) to ensure consistency across experiments. The validation set was used for model selection and hyperparameter tuning, while the final held-out test set provided an unbiased estimate of model performance.

2.2 Deep Learning Architectures

Three neural network architectures were designed with varying complexities to facilitate robust comparison. All models were implemented using TensorFlow/Keras and shared the same experimental setup: an input dimension of

12,288 (flattened 64×64×3 images), ReLU activation functions for hidden layers, a softmax output layer for the 2-class classification, Adam optimizer, a fixed training schedule of 10 epochs, a batch size of 32, and a random seed of 123 for reproducibility.

Deep Neural Network (DNN) Architecture The DNN was designed as a high-capacity model with two hidden layers comprising 512 and 256 neurons. Each layer employed ReLU activation, followed by batch normalization for stable training, and utilized Dropout regularization ($p = 0.3$) to mitigate overfitting. L2 weight regularization ($\lambda = 10^{-4}$) was also applied to all layers. The model was compiled with Adam optimizer and sparse categorical cross-entropy loss, $\mathcal{L} = \mathcal{L}_{\text{CE}} + \lambda\|\mathbf{W}\|_2^2$, resulting in a total of approximately 6.5 million trainable parameters. This architecture aimed to use its depth and breadth to learn complex, hierarchical features directly from the pixel data.

Multi-Layer Perceptron (MLP) Architecture The MLP was designed as a more efficient and compact architecture featuring two hidden layers with 128 and 64 neurons. It maintained use of ReLU, batch normalization, and L2 regularization ($\lambda = 10^{-4}$), but utilized a lower Dropout rate ($p = 0.2$), reflective of its reduced capacity, and had approximately 1.6 million parameters. The objective of this model was to determine whether a more basic network could match or surpass the performance, aiming to achieve a favorable balance between computational cost and performance.

Autoencoder-based DNN (AE-DNN) Architecture The AE-DNN employed a hybrid two-stage training process. A symmetric autoencoder was trained in an unsupervised manner to reconstruct its input, learning a compressed latent representation. The encoder consisted of layers with 512, 256, and 128 neurons (ReLU activation), with the 128-neuron layer serving as the latent space. The decoder mirrored this structure. The autoencoder was trained using Mean Squared Error (MSE) loss: $\mathcal{L}_{\text{recon}} = \text{MSE}(\mathbf{x}, \hat{\mathbf{x}})$. Subsequently, a classifier, which shared the encoder weights, was attached to the latent space and trained using the labeled data with sparse categorical cross-entropy loss, $\mathcal{L}_{\text{class}} = \mathcal{L}_{\text{CE}}$. The total parameter count for the full AE-DNN system was approximately 7.2 million. This model allowed for an investigation into whether features learned through unsupervised reconstruction could enhance the performance of the downstream classifier.

2.3 Experimental Setup

Training Configuration and Hyperparameters All models are trained for 10 epochs using a batch size of 32, and the Adam optimizer was used with its default parameters (learning rate $\alpha = 0.001$, $\beta_1 = 0.9$, $\beta_2 = 0.999$). The primary

loss function for classification tasks (DNN, MLP, and the second stage of AE-DNN) was sparse categorical cross-entropy:

$$\mathcal{L}_{\text{CE}} = -\frac{1}{N}\sum_{i=1}^{N}\sum_{c=1}^{2} y_{i,c}\log(\hat{y}_{i,c}) \tag{3}$$

Mean squared error loss was used during the autoencoder pretraining phase of the AE-DNN model:

$$\mathcal{L}_{\text{MSE}} = \frac{1}{N}\sum_{i=1}^{N} \|\mathbf{x}_i - \hat{\mathbf{x}}_i\|_2^2 \tag{4}$$

To optimize convergence, learning rate scheduling was implemented with a reduction on plateau (factor = 0.5, tolerance = 5 epochs).

Evaluation Metrics and Validation Strategy A comprehensive set of metrics was used to assess model performance from multiple perspectives. The primary metrics included accuracy, precision, recall, and F1 score, calculated as follows:

1. **Performance Metrics:**
   - **Accuracy:** The proportion of correctly classified samples
   $$\text{Accuracy} = \frac{TP + TN}{TP + TN + FP + FN} \tag{5}$$
   - **Area Under the ROC Curve (AUC):** The area under the Receiver Operating Characteristic curve, measuring the model's ability to distinguish between classes across all classification thresholds.
2. **Clinical Metrics:**
   - **Sensitivity (Recall):** The true positive rate, measuring the model's ability to identify positive cases correctly
   $$\text{Sensitivity} = \frac{TP}{TP + FN} \tag{6}$$
   - **Specificity:** The true negative rate, measuring the model's ability to identify negative cases correctly
   $$\text{Specificity} = \frac{TN}{TN + FP} \tag{7}$$
   - **Precision:** The positive predictive value, measuring the proportion of true positives among all predicted positives
   $$\text{Precision} = \frac{TP}{TP + FP} \tag{8}$$

– **F1-Score:** The harmonic mean of precision and recall, providing a balanced measure

$$\text{F1-Score} = 2 \times \frac{\text{Precision} \times \text{Recall}}{\text{Precision} + \text{Recall}} \tag{9}$$

3. **Error Analysis:**

– **Error Rate:** The proportion of incorrectly classified samples

$$\text{Error Rate} = \frac{FP + FN}{TP + TN + FP + FN} \tag{10}$$

– **False Negative Rate (FNR):** The proportion of actual positives incorrectly classified as negative

$$\text{FNR} = \frac{FN}{TP + FN} \tag{11}$$

– **False Positive Rate (FPR):** The proportion of actual negatives incorrectly classified as positive

$$\text{FPR} = \frac{FP}{TN + FP} \tag{12}$$

Supplementary metrics such as Cohen's Kappa and Matthews Correlation Coefficient were also computed, and Cohen's Kappa (κ) measures inter-rater agreement for categorical items, accounting for chance agreement:

$$\kappa = \frac{p_o - p_e}{1 - p_e} \tag{13}$$

where p_o is observed agreement and p_e is expected agreement by chance. Matthews Correlation Coefficient (MCC) provides a balanced measure even with imbalanced classes:

$$\text{MCC} = \frac{TP \times TN - FP \times FN}{\sqrt{(TP + FP)(TP + FN)(TN + FP)(TN + FN)}} \tag{14}$$

The validation strategy followed a standard supervised learning protocol. Models were trained on the designated training set, while model selection and hyperparameter tuning were guided by performance on the validation set. Final comparative results were reported based on the validation outcomes, consistent with established practice. An independent test set was subsequently employed as an unbiased evaluation, with results (e.g., Test Accuracy: MLP = 99.31%) showing the model's generalizability within the dataset.

Table 1. Comprehensive Kidney Stone Classification Performance Analysis and Clinical Evaluation Metrics

Model	Performance Metrics		Clinical Metrics				Error Analysis	
	Acc (%)	Time (s/epoch)	Sen (%)	Spe (%)	Pre (%)	F1 (%)	FN (%)	FP (%)
AE-DNN	99.47	4.2	99.47	98.87	99.48	99.47	0.53	1.13
MLP	**99.67**	**1.5**	**99.67**	**99.44**	**99.67**	**99.67**	**0.33**	**0.56**
DNN	98.95	5.3	98.95	97.75	98.97	98.94	1.05	2.25

3 Results and Analysis

3.1 Overall Performance and Computational Efficiency

The results show that all three deep learning models achieved high performance on the kidney stone classification task. A comprehensive summary of results, including performance, clinical, and error metrics calculated on the validation set, is presented in Table 1. The MLP emerged as the model with the best performance, achieving a validation accuracy of 99.67%, closely followed by AE-DNN at 99.47% accuracy. Similarly, the DNN established a strong baseline with 98.95% accuracy. This pattern was consistent across all primary clinical metrics, with MLP also achieving the highest sensitivity (99.67%), specificity (99.44%), precision (99.67%), and F1-score (99.67%). A critical differentiator was computational efficiency. The MLP's compact architecture resulted in a dramatically faster training time of 1.5 s/epoch, which is significantly more efficient than DNN (5.3 s/epoch) and AE-DNN (4.2 s/epoch).

3.2 Comprehensive Model Evaluation and Clinical Relevance

The confusion matrix in Figure 2 provides a combined classification performance across all models, confirming high accuracy and low error rates, as reported in Table 1. MLP achieved the lowest error rates (4 false positives, 1 false negative), followed by AE-DNN (8 false positives, 0 false negatives) and DNN (16 false positives, 0 false negatives). High sensitivity (98.95-99.67%) minimized critical false negatives, while strong specificity (97.75-99.44%) reduced unnecessary follow-ups. MLP's balanced performance with 99.67% sensitivity and 99.44% specificity established it as the most clinically reliable model.

3.3 Limitations and Statistical Considerations

Despite the encouraging outcomes, a few issues deserve attention. The dataset was perfectly balanced between normal and stone images. It made training stable, but it doesn't mirror the real-world clinical data. As a result, the models' predictive value in routine clinical use could differ. Another limitation arises from using flattened images instead of spatially aware convolutional layers; this

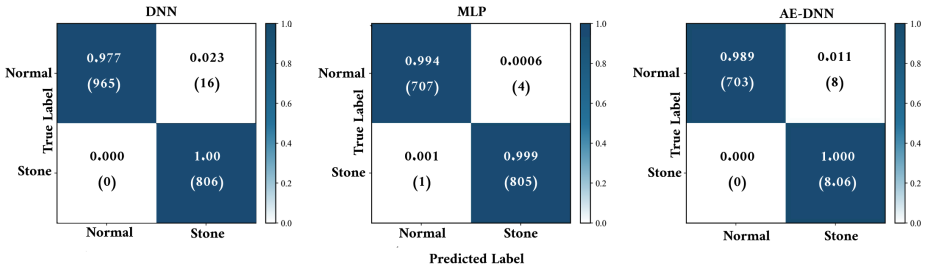

Fig. 2. Confusion Matrix for All Models

was intentional for comparison, but it inevitably removes some texture information. The almost flawless accuracy achieved by all models may therefore reflect the simplicity of the dataset rather than a truly difficult classification challenge. Testing on more diverse and unbalanced data would provide a clearer picture of how well these models generalize.

4 Conclusion and Future Work

The results of this study show that all three deep learning models can classify kidney stone images with high accuracy, with the MLP standing out for its strong performance and faster training time. Such consistency across models suggests that these methods could be useful in real-world diagnostic systems. Looking ahead, we intend to develop an integrated tool that utilizes explainable AI techniques, such as SHAP and LIME, to make model decisions more transparent to clinicians. We also plan to test the approach on data from multiple centers and expand it to distinguish between different types of stones. Together, these efforts could help translate AI-based analysis into more precise and accessible urological care.

References

1. W. Kittanamongkolchai, L. E. Vaughan, F. T. Enders, T. Dhondup, R. A. Mehta, A. E. Krambeck, C. H. McCollough, T. J. Vrtiska, J. C. Lieske, and A. D. Rule, "The changing incidence and presentation of urinary stones over 3 decades," in *Mayo Clinic Proceedings*, vol. 93, no. 3. Elsevier, 2018, pp. 291–299.
2. A. Chewcharat and G. Curhan, "Trends in the prevalence of kidney stones in the united states from 2007 to 2016," *Urolithiasis*, vol. 49, no. 1, pp. 27–39, 2021.
3. Y. A. Asaye, P. Annamalai, and L. G. Ayalew, "Detection of kidney stone from ultrasound images using machine learning algorithms," *Scientific African*, vol. 28, p. e02618, 2025.
4. O. Iparraguirre-Villanueva, G. Paucar-Palomino, and C. Paulino-Moreno, "From data to diagnosis: evaluation of machine learning models in predicting kidney stones," *Neural Computing and Applications*, pp. 1–14, 2025.

5. A. R. Panda, J. Tripathy, M. K. Mishra, L. Mohanty, J. J. Jena, and M. K. Gourisaria, "Kidney stone prediction based on urine analysis: A comprehensive study of machine learning models," in *2025 International Conference on Emerging Systems and Intelligent Computing (ESIC)*, 2025, pp. 748–753.
6. A. Khan, R. Das, and M. Parameshwara, "Detection of kidney stone using digital image processing: a holistic approach," *Engineering Research Express*, vol. 4, no. 3, p. 035040, 2022.
7. P. Kumar, D. Singh, and J. S. Samagh, "A hybrid model for kidney stone detection using deep learning," *IJSTM*, vol. 13, pp. 65–85, 2024.
8. Y. Liu, H. Song, D. Luo, R. Xu, Z. Xu, B. Wang, W. Hu, B. Xiao, G. Zhang, and J. Li, "Integrated radiomics and deep learning model for identifying medullary sponge kidney stones," *Frontiers in Medicine*, vol. 12, p. 1623850, 2025.
9. Y. J. Jacob, B. Janney *et al.*, "Optimised hybrid deep learning classification model for kidney stone diagnosis," *Results in Engineering*, p. 105221, 2025.
10. Kaggle, "Kidney stone classification dataset," https://www.kaggle.com/datasets/nazmul0087/ct-kidney-dataset-normal-cyst-tumor-stone, 2024, accessed: 2024-10-27.