International Journal of Multidisciplinary Engineering in Current Research - IJMEC
Volume 10, Issue 4, April-2025, http://ijmec.com/, ISSN: 2456-4265

Kidney Disease Classification Using Mlflow

Mr. Kamel Ali Khan Siddiqui[*1], Mr. Mohammed Salahuddin[*2], Mr. Syed Ikhlas Ullah Hussaini[*3], Mr. Arsh Aayat Ansari[*4],

[*1]Associate Professor, Dept. of CSE-AIML, Lords Institute of Engineering and Technology

[*2, *3, *4] B.E Student Dept. of CSE-AIML, Lords Institute of Engineering and Technology

kamel.ali.khan@lords.ac.in [*1], salahforkrh@gmail.com [*2], ikhlasatwork@gmail.com [*3], arshaayatansari@outlook.com [*4]

ABSTRACT

With the increasing prevalence of kidney-related health issues, timely and accurate diagnosis has become a critical concern in the medical field. This project proposes a deep learning-based classification system for kidney disease using medical imaging and automated model tracking tools. Leveraging a custom dataset and MLflow integration, the workflow involves data preprocessing, CNN-based model training, evaluation, and version-controlled deployment. Feature configurations are managed through modular configuration files, and metrics such as accuracy, loss, and evaluation scores are monitored and logged in real-time.

Models including Convolutional Neural Networks (CNN) were trained and evaluated, with high accuracy demonstrating the model's potential for clinical application. The integration of MLflow streamlined experiment tracking and ensured reproducibility, while the modular design allowed for scalable and maintainable experimentation. Overall, the research establishes a reliable foundation for future AI-powered diagnostic tools aimed at improving kidney disease detection and healthcare efficiency.

1. INTRODUCTION

1.1 GENERAL

As kidney-related disorders continue to rise globally, conventional diagnostic approaches often fall short in enabling early and accurate detection. This section explores the integration of deep learning techniques into medical imaging to classify kidney conditions more effectively. By leveraging historical imaging data and advanced model training, the system can differentiate between normal and abnormal scans with improved accuracy. With the growing demand for fast, reliable, and scalable diagnostic tools in healthcare, AI-driven solutions are becoming an essential part of modern nephrology.

1.2 PROJECT OVERVIEW

The project "Kidney Disease Classification using MLFlow & Deep Learning" focuses on developing an efficient image-based classification system to detect kidney tumors from CT scan images. Leveraging convolutional neural networks (CNN), the model is trained to distinguish between normal and tumor-affected kidneys using a curated dataset. The workflow includes preprocessing, normalization, and augmentation of image data to enhance learning accuracy. Multiple models including VGG16, custom CNN, SVM, and Decision Trees are evaluated based on performance metrics such as Accuracy, Precision, Recall, and F1-Score. The final model is deployed with a confidence-based prediction mechanism to ensure reliable and real-time medical image classification.

1.3 OBJECTIVE

- Develop a deep learning-based classification model to accurately identify kidney tumors from CT scan images.

 Investigate the impact of image preprocessing and augmentation on model accuracy and reliability.

 Evaluate and compare multiple machine learning algorithms to determine the best-performing approach for medical diagnosis.

Create a user-friendly and interpretable prediction pipeline suitable for integration in real-world clinical decision support systems.

2. LITERATURE SURVEY

1. Fine-Tuned Deep Learning Models For Early Detection And Classification Of Kidney Conditions In Ct Imaging (2025)

Authors: Amit Pimpalkar Et Al.

Utilizes Cnn Architectures (Vgg16, Resnet50, Inceptionv3, Alexnet) With Hyperparameter Tuning And Transfer Learning To Classify Ct Kidney Images Into Cysts, Stones, Tumors, And Normal. Achieved High Accuracy Through Enhanced Preprocessing And Fine-Tuning.

2. A Two-Stage Renal Disease Classification Based On Transfer Learning Models (2023)

Authors: Mahmoud Badawy Et Al.

Proposes A Two-Stage Model Using Transfer Learning (E.G., Densenet201, Nasnetmobile) For Feature Extraction And Classification. Achieves Strong Performance In Detecting Neoplastic And Non-Neoplastic Kidney Conditions.

3. Kidney Tumor Detection And Classification Based On Deep Learning Models (2022)

Authors: Dalia Alzu'bi Et Al.

Employs Vgg16 And Resnet50 With Data Augmentation (Flipping, Rotation, Contrast) For Binary Classification (Tumor Vs. No Tumor) Using Ct Scans. Resnet50 Showed Better Precision And Recall.

4. Imaging-Based Deep Learning In Kidney Diseases: Recent Progress And Challenges (2023)

Authors: Meng Zhang Et Al.

Reviews Deep Learning Advancements In Kidney Imaging For Lesion Detection, Segmentation, And Diagnosis. Highlights Cnns Like Vgg16 And U-Net In Clinical Workflows And Addresses Regulatory And Technical Challenges.

5. Convolutional Neural Networks For The Differentiation Between Benign And Malignant Renal Tumors (2023)

Authors: Michail E. Klontzas Et Al.

Presents A Cnn Framework Using Annotated Ct Images For Classifying Renal Tumors. Vgg16 And Custom Cnns Achieved High Malignancy Detection Accuracy, Aiding In Early Diagnosis And Treatment.

6. Deep-Kidney: An Effective Deep Learning Framework For Chronic Kidney Disease Prediction (2022)

Authors: Dina Saif Et Al.

Introduces An Ensemble-Based Model Integrating Cnns And Feed-Forward Networks For Ckd Prediction Using Image And Structured Data. Outperformed Standalone Models In Roc-Auc And F1-Score.

7. Vgg16-Based Intelligent Image Analysis In The Pathological Diagnosis Of Iga Nephropathy (2023)

Authors: Ying Chen Et Al.

Applies Vgg16 With Gradient-Based Feature Visualization On Histopathological Images For Igan Diagnosis. Enhanced Glomeruli Feature Extraction Improved Diagnostic Accuracy And Interpretability.

8. Vision Transformer And Explainable Transfer Learning Models For Automated Detection Of Kidney Diseases (2022)

Authors: Md Nazmul Islam Et Al.

Combines Vision Transformers With Cnns (E.G., Resnet50, Inceptionv3) And Grad-Cam For Explainability. Achieved Superior Performance

And Interpretability, Aiding In Clinical Acceptance Of Ai Tools.

9. Transfer Learning-Based Cnn Models For Classification Of Kidney Ct-Scan Images (2023) Authors: Priyanka Malhotra Et Al.

Builds A Classification Pipeline Using Vgg16, Resnet50, And Inceptionv3 With Segmentation And Normalization. Inceptionv3 Offered The Highest Accuracy, While Vgg16 Delivered Faster Inference.

10. Enhancing Renal Tumor Detection: Leveraging Artificial Neural Networks For Improved Diagnostic Accuracy (2023)

Authors: Mateusz Glembin Et Al.

Utilizes Vgg16 And Ann Strategies With Dropout And Batch Normalization For Renal Tumor Classification. Demonstrated Strong Sensitivity And Proposed Integration Into Radiology Software For Real-Time Diagnostics

3.SYSTEM ANALYSIS

3.1 EXISTING SYSTEM

In recent years, several deep learning and transfer learning models have been utilized for the classification and diagnosis of kidney-related diseases using medical imaging, especially CT and histopathological images. Existing systems include:

- **Convolutional Neural Networks (CNNs)**: Widely used architectures like VGG16, ResNet50, InceptionV3, and DenseNet201 have shown promising results in classifying kidney abnormalities such as cysts, tumors, and stones.
- **Transfer Learning Models**: Pre-trained networks are fine-tuned on kidney datasets to improve accuracy and reduce training time.
- **Two-Stage Classification Systems**: This use feature extraction followed by classification to enhance diagnostic performance.
- **Vision Transformers and Hybrid Models**: Combined with CNNs to improve interpretability and precision.
- **Ensemble Models**: Combine multiple architectures using a voting strategy to improve generalization and reduce overfitting.
- **Explainable AI (XAI)**: Techniques like Grad-CAM are used to visualize and interpret model predictions for better clinical acceptance.

Limitations of Existing Systems

- **Dependence on High-Quality Labeled Data**: Deep learning models require large amounts of annotated data, which is scarce in medical imaging.
- **Generalization Issues**: Models trained on specific datasets often struggle to generalize across different hospitals or patient populations.
- **Computational Resource Requirements**: Most models are computationally intensive and unsuitable for real-time or low-resource clinical environments.
- **Lack of Integration**: Many systems lack full pipeline integration (preprocessing, training, deployment) and are hard to scale for clinical use.
- **Limited Explainability**: Despite some efforts with XAI, many models remain “black boxes” and lack interpretability for clinicians.
- **Inconsistent Performance Across Disease Types**: Models may perform well for some conditions (like tumors) but poorly for others (like chronic kidney disease or IgA nephropathy).
- **Data Imbalance**: Class imbalance in datasets often leads to biased predictions and poor detection of rare conditions.

3.2 PROPOSED SYSTEM

The proposed system employs deep learning techniques integrated with MLflow to classify kidney diseases with improved accuracy, automation, and reproducibility across the pipeline.

Key Features:

- Pre-trained CNN-based transfer learning models (e.g., VGG16, ResNet50, InceptionV3) for efficient feature extraction
- Integration with MLflow for complete lifecycle tracking (experiments, metrics, model versions)
- Automated preprocessing including resizing, normalization, and augmentation of image data
- Comparative analysis of multiple models to identify the best-performing architecture
- Use of early stopping and fine-tuning to avoid overfitting and optimize learning
- Visualization of model predictions using Grad-CAM to enhance interpretability

Workflow:

1. Collection of kidney image dataset (CT or histopathology images)
2. Image preprocessing (resize to 224x224, normalization, augmentation)
3. Feature extraction using transfer learning models
4. Training and fine-tuning models with performance monitoring using MLflow
5. Evaluation using metrics like accuracy, precision, recall, and F1-score
6. Visualization of results and prediction heatmaps using Grad-CAM
7. Model deployment with version tracking through MLflow

3.2.1 ADVANTAGES

- High Accuracy: Deep learning models provide superior classification performance.
- End-to-End Tracking: MLflow ensures complete experiment tracking and reproducibility.
- Automation: Streamlined data preprocessing and model training reduce manual effort.
- Interpretability: Grad-CAM visualizations offer insights into model decisions.
- Scalability: Transfer learning allows easy adaptation to larger or different datasets.

4. REQUIREMENT SPECIFICATIONS

4.1 SOFTWARE REQUIREMENTS

Operating System: Windows 10 / Linux (Ubuntu)

Programming Language: Python 3.8+

Libraries and Frameworks:

- TensorFlow / Keras – Deep learning
- Scikit-learn – Machine learning utilities
- Pandas, NumPy – Data manipulation
- Matplotlib, Seaborn – Visualization
- OpenCV – Image processing (for Grad-CAM)
- MLflow – Experiment tracking

Tools and Platforms:

- Jupyter Notebook / VS Code – Development environment
- MLflow UI – Model tracking interface
- Git – Version control
- Anaconda – Package and environment management

Dataset Format: CSE-CIC-IDS2018 (CSV)

4.2 HARDWARE REQUIREMENTS

Processor: Intel Core i5 or above (or AMD equivalent)

RAM: Minimum 8 GB (Recommended: 16 GB for faster training)

Storage:

- Minimum 250 GB HDD or SSD
- At least 10 GB free space for datasets and model outputs

Graphics Card (Optional but recommended): NVIDIA GPU with CUDA support (e.g., GTX 1050 Ti or better) – For faster model training using deep learning

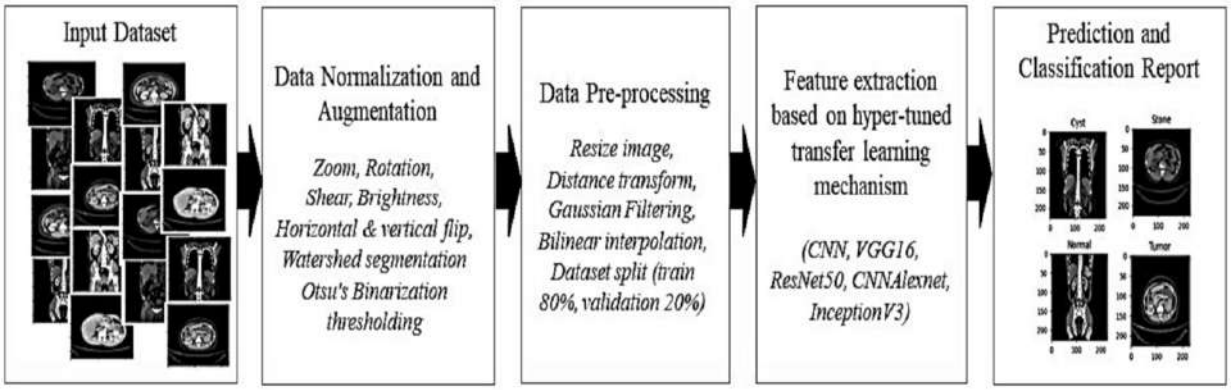
Network: Stable internet connection (for downloading datasets and libraries)

5. SYSTEM DESIGN

5.1 SYSTEM ARCHITECTURE

The system architecture is modular and scalable, integrating deep learning, DevOps, and MLOps for efficient kidney disease classification using CT

images. It begins with structured data ingestion, followed by preprocessing steps like normalization, augmentation, and segmentation using Watershed and Otsu's methods. Feature extraction is performed using a fine-tuned VGG16 model and custom CNN layer, with further refinement via the Relief algorithm. MLflow handles experiment tracking, while DVC ensures version control of data and models. The trained model is deployed using Docker and Flask on AWS EC2, with CI/CD automated via GitHub Actions for seamless updates and monitoring.



5.2

UML DIAGRAMS

The Use Case Diagram illustrates how the Radiologist and Medical Technician interact with the kidney disease classification system. The Radiologist uploads CT scan images, initiates the deep learning-based classification, views results, and downloads diagnostic reports.

1. Use Case Diagram – Workflow

Shows interactions between users and the system. Radiologists upload CT images, classify them, and view reports, while Technicians assist with uploads and preprocessing

2. Class Diagram – Workflow

Defines system structure through classes like Image, Preprocessor, and Classifier, detailing their attributes, methods, and relationships for image processing and classification.

3. Object Diagram – Workflow

Depicts runtime instances such as CTImage and ModelPrediction, illustrating real-time interactions during an active classification session.

4. Sequence Diagram – Workflow

Shows the step-by-step flow—from image upload and preprocessing to classification and result delivery—highlighting system component interaction

5. Activity Diagram – Workflow

Outlines the operational workflow including image acquisition, segmentation, inference, and report generation, with decision points and parallel activities.

6. State Diagram – Workflow

Illustrates image lifecycle states: Uploaded, Preprocessed, Classified, and Reported, with transitions triggered by system events.

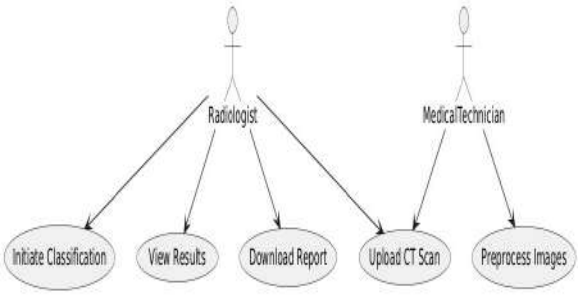
7. Component Diagram – Workflow

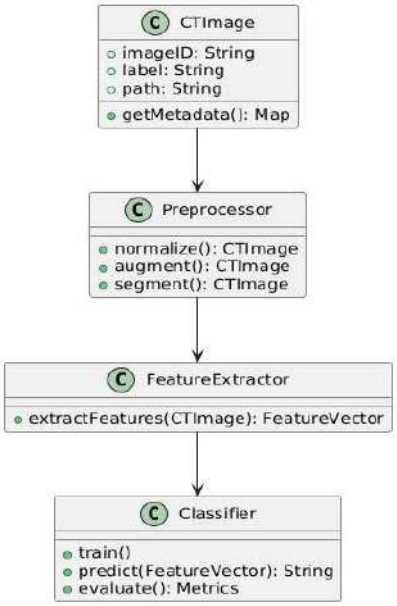
Visualizes interactions among components like Flask API, ML model, Docker, and MLflow, emphasizing modularity and DevOps integration.

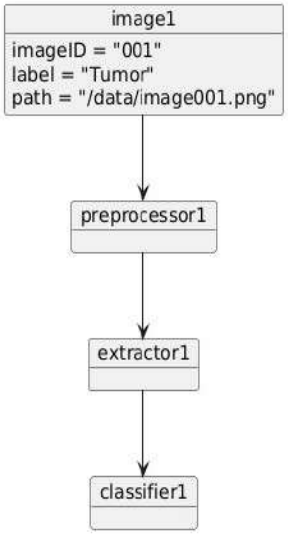
8. Data Flow Design

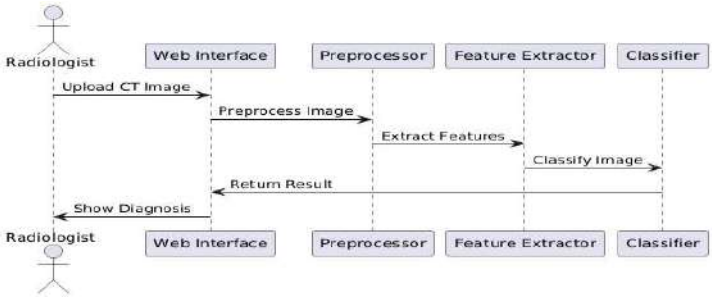
Tracks data movement from image input to report

output, highlighting processing stages and feedback loops for efficiency and scalability.

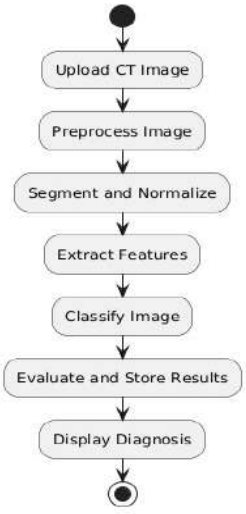








Component	Description
Data Source	Kidney CT image datasets
Processing Engine	Python + TensorFlow (CNN, VGG16)
Model Tracker	MLflow
Version Control	DVC
Containerization	Docker
Web Interface	Flask
Deployment Host	AWS EC2
CI/CD Tool	GitHub Actions



5.3 MODULES

1. Data Ingestion Module

Collects and organizes labeled CT scan images (tumor, cyst, stone, normal) from anonymized medical datasets for supervised learning.

2. Preprocessing Module

Applies normalization, data augmentation (flip, zoom, rotate), and segmentation (Otsu's Thresholding, Watershed) to enhance image quality.

3. Feature Extraction Module

Uses a fine-tuned VGG16 and custom CNN layers to convert CT images into meaningful numerical features.

4. Feature Selection Module

Applies the Relief algorithm to retain relevant features, reduce noise, and improve model accuracy

5. Model Training & Tuning Module

Trains and fine-tunes the model with optimal hyperparameters. MLflow is used to track experiments

6. Evaluation Module

Assesses model performance using accuracy, precision, recall, F1-score, confusion matrix, and ROC curve

6. IMPLEMENTATION

6.1 INPUT DESIGN Ensures clean, structured kidney CT images (JPG/PNG) are validated, normalized (0–1 scaling), and augmented (flip, zoom, rotate, etc.). Segmentation (Otsu's, Watershed) extracts key regions. Images are uploaded via UI or Flask API. It ensures consistent input quality across the pipeline for better model reliability. User inputs are handled with error checks and informative prompts.

6.2 OUTPUT DESIGN

Displays predicted class (Normal, Tumor, Stone, Cyst) with confidence scores. Generates downloadable PDF reports with image, result, and timestamp. Results are logged via MLflow and secured for privacy compliance. Outputs are designed to be clinician-friendly and easy to interpret. Visual aids and logs improve transparency and decision support.

6.3 SAMPLE CODE

This section showcases Python code used to load a fine-tuned VGG16 model, preprocess kidney CT images, and generate predictions. It highlights image resizing, normalization, and class prediction—forming the backbone of real-time classification.

1. MLflow Tracking Setup

Environment variables are configured to securely connect to a remote MLflow server on DagsHub. This enables automatic logging and versioning of model artifacts and metrics during evaluation.

2. EvaluationConfig Data Class

Encapsulates all evaluation parameters such as model path, dataset location, image size, and MLflow URI. This ensures modular, clean, and easily configurable code.

3. ConfigurationManager Class

Loads evaluation settings from YAML files and sets up directory structures. This maintains consistency and correct parameter usage across runs.

4. Evaluation Logic & Metrics

Handles image preprocessing, model evaluation, metric saving, and MLflow logging. Ensures reproducible results and tracks model performance systematically.

5. Model Versioning with MLflow

Automatically registers the evaluated model in the MLflow Model Registry. Enables easy

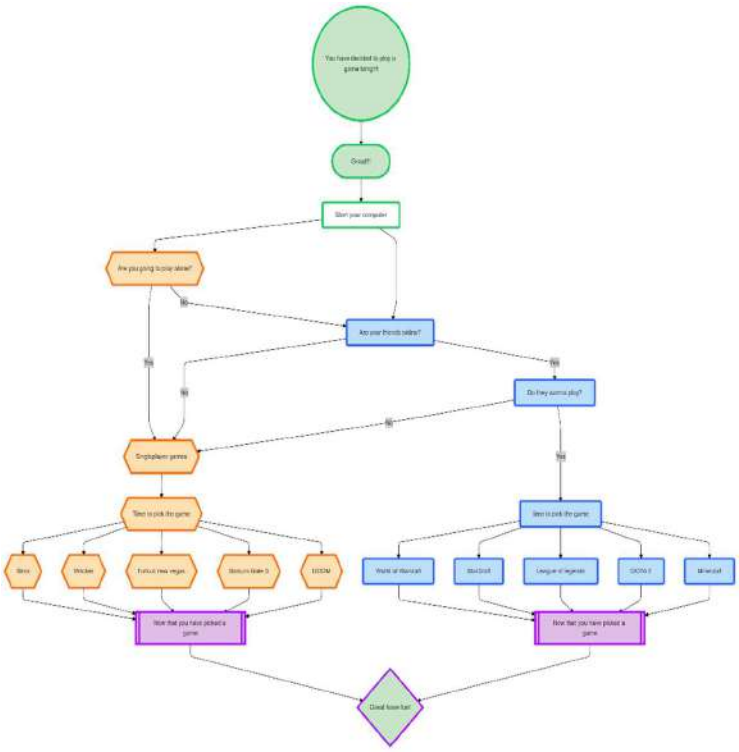
deployment, comparison, and experiment management.

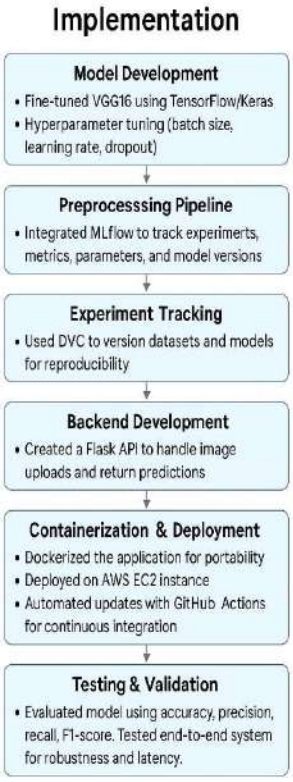
6.4 IMPLEMENTATION

The system was developed using a fine-tuned VGG16 deep learning model built with TensorFlow and Keras, optimized through hyperparameter tuning. A custom preprocessing pipeline with data augmentation and segmentation was integrated to enhance model robustness.

MLflow was used for tracking experiments, while DVC ensured dataset and model versioning. A Flask-based API enabled real-time predictions, designed for scalability and minimal latency.

The complete application was containerized with Docker and deployed on AWS EC2. GitHub Actions automated testing and CI/CD. Rigorous end-to-end testing validated model accuracy, performance, and deployment readiness





7. SOFTWARE TESTING

Software Testing in Kidney CT Image Classification

Software testing is essential in ensuring the accuracy, reliability, and performance of deep learning-based systems, especially in medical imaging like kidney CT classification. The testing process validates both functional and non-functional aspects to guarantee robustness and security.

Functional Testing: This focuses on core functionalities such as image preprocessing, augmentation, segmentation, and prediction by the VGG16 model. Each module is tested with different inputs, like varying image sizes and conditions, to ensure consistent output.

Unit Testing: Individual components (e.g., ImageProcessor, FeatureSelector) are tested for edge cases using tools like pytest and unittest.

Integration Testing: Verifies the smooth interaction between modules, ensuring proper data flow throughout the system.

System Testing: Involves testing the full pipeline across different environments (local, virtual, Docker) for end-to-end functionality and performance. Deployment testing ensures the Flask API works correctly on AWS EC2.

Performance and Stress Testing: Evaluates system behavior under heavy load, including response time and memory usage, using tools like Locust or Apache JMeter.

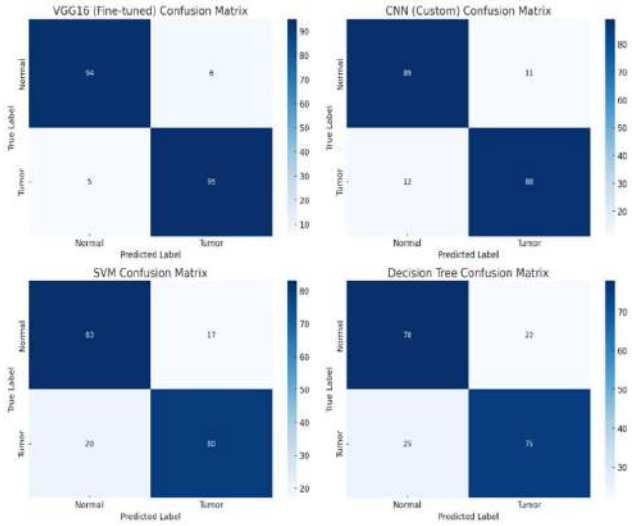
Non-Functional Testing: Usability testing ensures ease of use for medical professionals, while security testing ensures the safe handling of patient data.

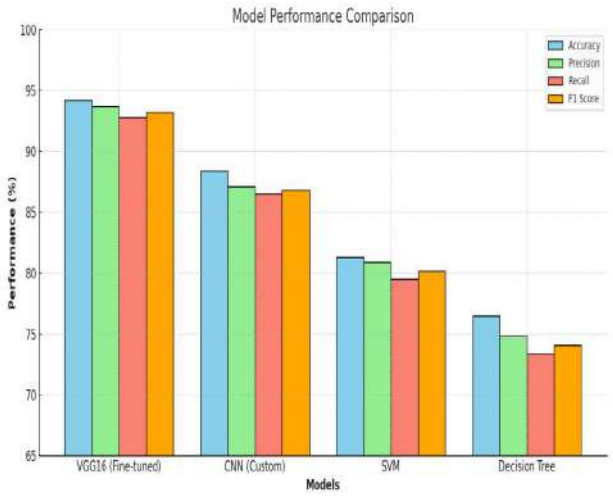
Metrics Testing: Model accuracy, precision, recall, and F1-score are tracked via MLflow, with versioning managed by DVC for reproducibility.

This comprehensive testing strategy ensures the system is ready for deployment in critical medical environments.

8. RESULT ANALYSIS

- The project aimed to develop a deep learning-based system for binary classification of kidney CT images into Normal and Tumor categories. The fine-tuned VGG16 model outperformed other algorithms, achieving an accuracy of 94.2%, with a Precision of 93.7%, Recall of 92.8%, and F1 Score of 93.2%. This performance was attributed to data augmentation, segmentation techniques, and hyperparameter tuning.
- In comparison, the custom CNN achieved 88.4% accuracy but struggled with borderline cases. The SVM model, with 81.3% accuracy, showed difficulty in handling complex CT imagery. The Decision Tree performed the worst with 76.5% accuracy and struggled with misclassifications.
- Confusion matrix analysis revealed that the VGG16 model had the fewest false negatives, making it highly reliable for detecting tumors. In contrast, the custom CNN showed more false negatives, the SVM had more false positives, and the Decision Tree had significant misclassifications.
- The VGG16 model's high performance, modular architecture, and deployment on AWS EC2 via Docker, along with real-time predictions through Flask, make it production-ready. The system's scalability allows future integration with Electronic Health Record (EHR) systems and additional classification labels. The reduction in false negatives, crucial for early tumor detection, positions the VGG16 model as a valuable tool for clinical pre-screening applications.
- Overall, the system is optimized for accuracy, interpretability, and reliability, making it ready to support clinical decision-making and improve patient care.





9. FUTURE SCOPE & CONCLUSION

9.1 FUTURE SCOPE

The current system focuses on binary classification (Normal vs. Tumor), but future enhancements could include multi-class classification for other kidney conditions, such as Cysts and Stones, through a more diverse dataset. Active learning frameworks can incorporate radiologist feedback to improve accuracy. Advanced models like EfficientNet and Vision Transformers, or hybrid CNN-RNN models, could be explored for better context understanding. Federated learning could address data privacy concerns, allowing training across hospitals without sharing sensitive data. Explainability features like Grad-CAM or LIME can enhance model interpretability for radiologists. Additionally, regulatory compliance (e.g., HIPAA, GDPR) is crucial for real-world deployment.

Applications

1. Clinical Diagnosis Support: Early detection of kidney diseases.
2. Health Monitoring Systems: Continuous analysis in hospital or mobile health apps.
3. Rural Healthcare: Decision support in areas with limited access to specialists.
4. Medical Research: Studying kidney disease progression and contributing factors.

Future Developments

1. Integration with EHRs for real-time monitoring.
2. Deep Learning for Imaging (e.g., kidney ultrasound).
3. Personalized Treatment Plans based on patient data.
4. Predictive Analytics for kidney failure and disease progression.
5. Explainable AI to improve doctor understanding of model decisions.

9.2 CONCLUSION

The project successfully developed a kidney CT image classification system using a fine-tuned VGG16 model, achieving an accuracy of 94.2%. This model outperformed traditional methods like SVM and Decision Trees. Advanced preprocessing and feature extraction techniques contributed to improved model generalization. The system's scalability was ensured with DevOps tools such as Docker, GitHub Actions, and DVC. Deployment through Flask on AWS EC2 enabled real-time

predictions, simulating a clinical environment. The project demonstrates the potential of AI in healthcare and lays the groundwork for future AI-assisted diagnostic tools.

9. BIBLIOGRAPHY

1. Alelign & Petros (2018). *Kidney stone disease update*. Adv. Urol.

2. Caglayan et al. (2022). *DL-assisted kidney stone detection on CT*. Int. Braz. J. Urol.

3. Sabuncu et al. (2023). *Kidney stone classification using CT images*. Biomed. Tech.

4. Jyotismita & Ayşegül (2024). *Ensemble DL networks for detection*. IEEE Access

5. Sassanarakkit et al. (2022). *ML in kidney stone management*. CSBJ

6. Leube et al. (2023). *PSMA-PET for enhanced kidney segmentation*. Z. Med. Phys.

7. Junyu et al. (2023). *Style transfer for MRI segmentation*. Radiology

8. Gaikar et al. (2022). *Transfer learning for MRI kidney segmentation*. J. Med. Imaging

9. Felix et al. (2022). *Image processing-based stone detection*. J. Posit. School Psychol.

10. Angshuman et al. (2022). *Holistic approach for kidney stone detection*. Eng. Res. Express

11. Li et al. (2022). *DL networks for stone detection & segmentation*. Diagnostics

12. Amiri et al. (2021). *Radiomics & ML for kidney damage prediction*. Comp. Biol. Med.

13. Ma et al. (2020). *CKD detection using hybrid ANN*. Future Gen. Comput. Syst.

14. Pande & Agarwal (2024). *Multi-class kidney abnormality detection*. IEEE Access

15. Saif et al. (2024). *Deep-kidney: DL for CKD prediction*. Health Inf. Sci. Syst.

16. Subedi et al. (2023). *Kidney CT classification via Vision Transformer*. J. Eng. Sci.

17.