International Journal of Scientific Research in Engineering and Management (IJSREM)
Volume: 10 Issue: 10 | Oct - 2026 SJIF Rating: 8.659 ISSN: 2582-3930

MEDVISION AI: An AI-Assisted Medical Imaging Analysis and Clinical Report Summarization System

Prof. Avinash Utikar[1], Yasharth Jadhav[2], Kartik Tanpure[3], Swayam Shah[4], Aniket Gonjari[5]

[1] Department of Computer Science and Engineering, MIT ADT University, Pune, India

[2] Department of Computer Science and Engineering, MIT ADT University, Pune, India

[3] Department of Computer Science and Engineering, MIT ADT University, Pune, India

[4] Department of Computer Science and Engineering, MIT ADT University, Pune, India

[5] Department of Computer Science and Engineering, MIT ADT University, Pune, India

---

Abstract - Medical imaging plays a key role in today's healthcare system, helping doctors spot and understand different diseases. But looking at medical images and reading the related reports can take a lot of time and needs expert knowledge. This paper introduces MEDVISION AI, an AI-powered healthcare tool that uses three special deep learning models to examine X-ray, CT, and MRI images. The system can find possible issues in the areas each model is trained for and gives doctors a first look at the results. It also handles medical PDF reports by pulling out important details and creating easy-to-read summaries, making it easier for doctors to get the information they need. The platform uses Python, FastAPI, HTML, CSS, JavaScript, and SQLite, with PyTorch or TensorFlow for building and running the AI models. Tools like NumPy, Pillow, and pydicom help process the images, while PyMuPDF is used for reading text from PDFs. The system is built with a modular design that keeps image analysis and document handling separate, but it offers a single place to upload files and check the results. The performance of the system can be measured using metrics like accuracy, precision, recall, specificity, and F1-score, along with how well it extracts and processes documents. MEDVISION AI is meant to speed up the initial review of medical information while ensuring that trained healthcare professionals remain in charge of making diagnoses and treatment plans.

Keywords - Artificial intelligence, medical image analysis, deep learning, X-ray classification, CT scan, MRI, medical report summarization, clinical decision support, FastAPI, SQLite.

1. INTRODUCTION

Medical imaging is very important for diagnosing, keeping track of, and treating many health issues. Tools like X-ray, CT scans, and MRI scans help doctors see inside the body and find any problems. Each of these tools gives different kinds of information and needs the right way to process and understand the images.

As the amount of medical images grows, AI is becoming a useful tool in healthcare. Deep learning, especially CNNs, can learn from images and spot patterns related to certain conditions. Other methods, like transformer models and mixed models, also help in understanding how parts of images relate to each other.

Doctors often look at more than just pictures. They also check reports, lab results, prescriptions, and other medical documents. Sometimes important details are spread across many pages, which can take a lot of time to go through. Using AI to extract and summarize text can make this information easier to manage.

To help with these challenges, this paper introduces MEDVISION AI, a platform that uses three AI models for different types of medical images and an automated system to process medical PDFs. The X-ray model checks radiographic images, while CT and MRI models handle their own types of scans. The document processing part pulls out key information from reports and creates easy-to-read summaries.

The system is built using Python and FastAPI for the back end, HTML, CSS, and JavaScript for the front end, and SQLite for storing data.

Libraries for image processing help prepare the medical images, and PyMuPDF is used to extract text from PDFs.The app offers a simple way to upload medical files and view the initial results.

The main goals of MEDVISION AI are:

1. To create three AI models for X-ray, CT, and MRI images.
2. To find the possible issues that the models are trained to detect.
3. To extract important details from medical reports and make them into clear summaries.

4. To combine the image analysis and document processing into one web app.
5. To provide initial findings that help doctors without taking over their decision-making.

The main contribution of this work is combining different types of image analysis and document summarization into one organized app.

The system aims to make it easier for healthcare providers to review medical information quickly and effectively.

2. LITERATURE REVIEW

2.1 Literature Survey

TABLE 1. LITERATURE REVIEW

Sr. No.	Research Paper / Study	Author(s) and Year	Relevance to MEDVISION AI
1	Gradient-Based Learning Applied to Document Recognition	LeCun et al., 1998 [1]	Explains the basic ideas of Convolutional Neural Networks (CNNs) used to find features in images.
2	ImageNet Classification with Deep Convolutional Neural Networks	Krizhevsky et al., 2012 [2]	Shows how deep CNNs work well for classifying many images, which helps in classifying medical images.
3	Deep Residual Learning for Image Recognition	He et al., 2016 [3]	Introduces residual connections that help train deeper networks for analyzing images.
4	CheXNet: Radiologist-Level Pneumonia Detection on Chest X-Rays with Deep Learning	Rajpurkar et al., 2017 [4]	Shows how deep learning can detect pneumonia from chest X-rays, which is part of the X-ray analysis part.
5	ChestX-ray8: Hospital-Scale Chest X-Ray Database and Benchmarks on Weakly-Supervised Classification and Localization of Common Thorax Diseases	Wang et al., 2017 [5]	Offers a big collection of chest X-rays for studying diseases and finding where they are in images.
6	U-Net: Convolutional Networks for Biomedical Image Segmentation	Ronneberger et al., 2015 [6]	Presents an encoder-decoder CNN structure for segmenting and identifying regions in biomedical images.
7	nnU-Net: A Self-Configuring Method for Deep Learning-Based Biomedical Image Segmentation	Isensee et al., 2021 [7]	Explains a flexible system for segmenting medical images across various data sets and imaging tasks.
8	An Image Is Worth 16×16 Words: Transformers for Image Recognition at Scale	Dosovitskiy et al., 2021 [8]	Introduces Vision Transformers for recognizing images by looking at how different image parts

			relate to each other.
9	MIMIC-CXR: A Large Publicly Available Database of Labeled Chest Radiographs	Johnson et al., 2019 [9]	Provides chest X-ray images along with their radiology reports for medical imaging studies.
10	Digital Imaging and Communications in Medicine (DICOM) Standard	NEMA [10]	Defines ways to store and share medical images and their related information, which is important for CT and MRI work.
11	Grad-CAM: Visual Explanations from Deep Networks via Gradient-Based Localization	Selvaraju et al., 2017 [11]	Shows a method using heatmaps to highlight areas of an image that affect the model's predictions.
12	PyMuPDF Documentation	PyMuPDF Developers [12]	Describes ways to extract text and information from PDFs, which is useful for processing and summarizing medical reports.

2.2. Research Gap

Existing studies show progress in areas like classifying medical images, segmenting them, explaining how decisions are made, and processing clinical documents. But these tasks often need different tools and steps.

MEDVISION AI offers a single application that brings together three specialized imaging models along with a system for extracting and summarizing information from medical PDFs. The key innovation is how these parts are designed and put together in one web-based tool. To prove its value in a real clinical setting, it needs to be tested thoroughly, not just because it's all in one place.

3. PROPOSED SYSTEM AND ARCHITECTURE

3.1. System Overview

MEDVISION AI is a web-based application with two main parts: medical image analysis and medical PDF processing.
The image part checks the uploaded files, prepares them, and sends them to specific models for X-ray, CT, or MRI to predict if there are any issues.The PDF part takes out the text, uses OCR if needed for scanned documents, and makes short summaries.SQLite keeps track of all the uploaded files, their analysis status, the model versions used, and the results.

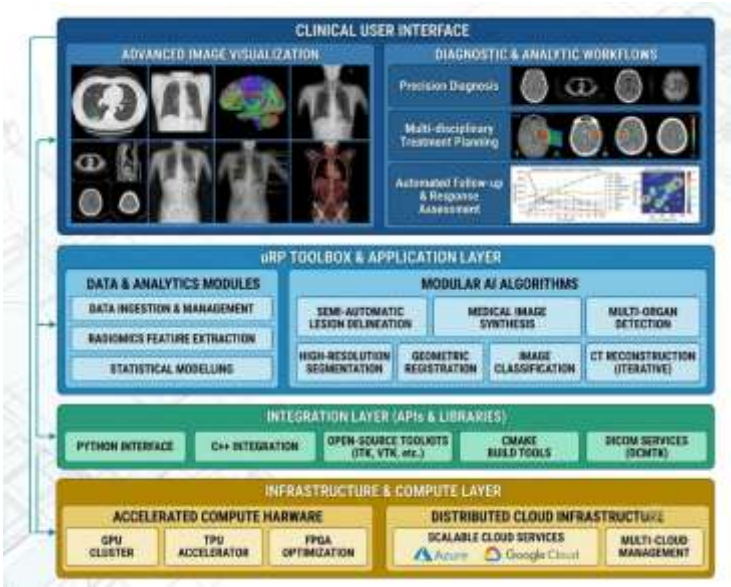


Fig. 1. Proposed MEDVISION AI system architecture.

3.2. X-Ray Analysis Module

The X-ray part adjusts the images by resizing them, turning them into black and white, and normalizing the brightness as needed.
The trained model then predicts the specific condition and gives a confidence score.If the model supports it, it might also show heatmaps.

3.3. CT Scan Analysis Module

The CT part handles CT scans by normalizing the intensity and preparing the images appropriately.

It manages DICOM files and predicts conditions for the defined targets.However, predictions from individual slices are not considered complete results without proper aggregation.

3.4. MRI Scan Analysis Module

The MRI part processes MRI images or volumes based on the required input, like T1-weighted or T2-weighted scans.

It makes predictions and gives confidence scores while also identifying any unsupported inputs.

3.5. Medical PDF Processing Module

The PDF part uses PyMuPDF to take out and clean the text from the reports.

OCR is used if the documents are scanned.The extracted content is summarized, making sure to keep important findings, measurements, negative statements, and any uncertainty.

3.6. Database Design

SQLite stores all the structured data of the application through these tables:

1. Users: Details about the account, if there's authentication.
2. Uploads: The type of file, imaging method, time of upload, and where the file is stored.
3. Analyses: Information about the model used, the status of processing, and the results.
4. ReportSummaries: The extracted text, the summaries made, and the status of review.

Uploaded files are kept in a separate place, and access is controlled to protect medical information.

3.7. Proposed Processing Workflow

The workflow involves uploading a file, checking if it's valid, choosing the right processing path, running the model or extracting information from documents, creating the results, and then showing them to the user.If the file is not valid, doesn't support the required type, can't be read properly, or there's an error during processing, the system should address these issues clearly.The app should not provide a guess or partial result when the input can't be handled correctly.

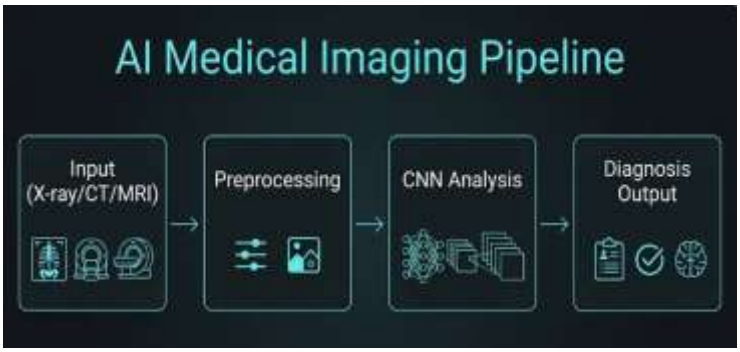


Fig. 2. MEDVISION AI operational flowchart.

TABLE 2. TECHNOLOGY STACK

Component	Technology	Purpose
Frontend	HTML, CSS, JavaScript	User interface
Backend	Python, FastAPI	API and processing coordination
AI models	PyTorch or TensorFlow	Image analysis
Image processing	NumPy, Pillow	Numerical operations and image handling
Medical imaging	pydicom	DICOM processing
PDF processing	PyMuPDF	Text extraction
Database	SQLite	Metadata and result storage
Development	VS Code, Google Colab, Jupyter	Coding and experimentation
Testing	Postman, Python testing tools	API and integration testing
Version control	Git, GitHub	Source-code management

4. IMPLEMENTATION

4.1. Development Environment

The system is built using Python for the backend and to integrate AI features.FastAPI is used to create endpoints for uploading files, analyzing images, processing PDFs,

and retrieving analysis records.The frontend is developed using HTML, CSS, and JavaScript.

PyTorch or TensorFlow is used for developing and running AI models.

NumPy and Pillow handle numerical calculations and common image formats, while pydicom is used for processing DICOM data.PyMuPDF is used to extract text from medical PDFs.

Google Colab and Jupyter Notebook can be used for training models and evaluating datasets, while VS Code is the main tool for development.

4.2. Dataset Preparation and Model Training

To ensure optimal performance, each model needs a suitable dataset that satisfies its imaging modality and target conditions, including class imbalance, missing labels, duplicates, and image quality.The data is then divided into training, validation, and testing sets.The X-ray model may use ResNet or DenseNet, while CT and MRI models may use 2D or 3D CNNs.Transfer learning reduces training requirements, while model selection and tuning use training and validation data.

4.3. Image Preprocessing and Classification

When an image is uploaded, the system first checks if it meets the required format, size, and dimension standards. It then uses the correct preprocessing steps to decode the image, apply necessary normalization, and convert the data into a tensor format that the model can use. The processed image is sent to the selected model for analysis. The model's output is then converted into readable labels and associated confidence scores. If a suitable localization method is available, the system may also generate a visual representation of the results. The API returns a well-structured response that includes an analysis identifier, the predicted labels, their scores, and the status of the processing.

Let X represent an input medical image and m represent its modality, where

$$m \in \{\text{X-ray,CT,MRI}\}. \quad (1)$$

The corresponding model generates a prediction according to

$$\hat{y} = f_m(T_m(X); \theta_m) \quad (2)$$

where T_m is the modality-specific preprocessing function, f_m is the trained model, and θ_m represents its learned parameters.

The output $\hat{y}$ represents the conditions supported by the selected model.

A resizing operation can be expressed as

$$X_r = R(X; H, W) \quad (3)$$

where H and W represent the target image dimensions.

A commonly used normalization operation is

$$X_n = \frac{X_r - \mu}{\sigma + \epsilon} \quad (4)$$

where μ and σ are the mean and standard deviation estimated using the appropriate training-data procedure, and ϵ is a small constant that prevents division by zero.

For a single-label classification task with K classes, softmax can convert logits into normalized scores:

$$p_k = \frac{e^{z_k}}{\sum_{j=1}^{K} e^{z_j}} \quad (5)$$

where z_k represents the logit for class k.

For multilabel classification, where several conditions may coexist, sigmoid outputs can be used:

$$p_k = \frac{1}{1 + e^{-z_k}}. \quad (6)$$

A class-specific threshold τ_k determines the binary prediction:

$$\hat{y}_k = \begin{cases} 1, & p_k \geq \tau_k, \\ 0, & p_k < \tau_k. \end{cases} \quad (7)$$

Thresholds must be selected using validation data. Model scores should not automatically be interpreted as calibrated probabilities of disease.

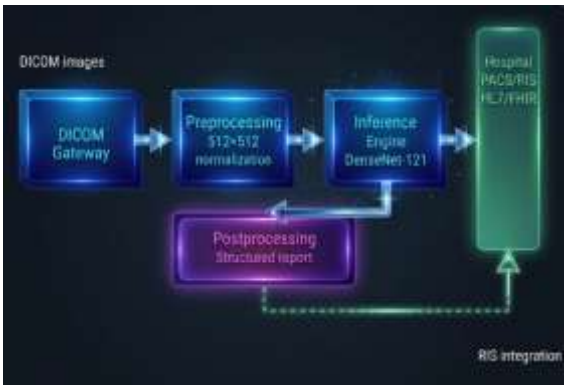


Fig. 3. Image preprocessing and model inference pipeline.

4.4. PDF Text Extraction and Summarization

The PDF module extracts text from each page of the document and checks if there is meaningful content to work with. Before sending the text to the summarization tool, it is cleaned up.

The summarizer creates a concise version of the original document while keeping important details like measurements, units, negative findings, and statements that express uncertainty. Where possible, users should be able to compare the summary with the original text. If the PDF has scanned pages, OCR (Optical Character Recognition) is needed before the summarization can take place.

Let D denote the uploaded PDF document. Text extraction is represented as

$$T = E(D) \tag{8}$$

where E is the extraction function.

The extracted text is cleaned and structured using

$$S = C(T) \tag{9}$$

Where C removes formatting artifacts and organizes the content.

The final summary is generated by

$$R_s = G(S) \tag{10}$$

where G is the summarization function and R_s represents the resulting summary.

This formulation describes the processing pipeline but does not guarantee that the summary is factually correct. Summary quality requires separate evaluation.

4.5. API Integration and System Testing

FastAPI handles file uploads, special analysis based on the type of file, creating summaries from PDFs, and getting records of the analysis.SQLite keeps track of how each upload is connected to its results.Testing involves checking the preprocessing and extraction steps with unit tests, ensuring the API correctly handles requests with API tests, testing how different parts of the system work together with integration tests, and evaluating the model's performance on new, unseen data.

4.6. Performance Metrics

The classification models are evaluated using true positives (TP), true negatives (TN), false positives (FP), and false negatives (FN).

Accuracy:

$$\text{Accuracy} = \frac{TP + TN}{TP + TN + FP + FN} \tag{11}$$

Precision:

$$\text{Precision} = \frac{TP}{TP + FP} \tag{12}$$

Recall:

$$\text{Recall} = \frac{TP}{TP + FN} \tag{13}$$

Specificity:

$$\text{Specificity} = \frac{TN}{TN + FP} \tag{14}$$

F1-score:

$$F1 = 2\frac{\text{Precision} \times \text{Recall}}{\text{Precision} + \text{Recall}} \tag{15}$$

ROC-AUC can also be used to assess discrimination across classification thresholds. For multilabel datasets, the paper should specify whether results are reported per class or using macro- or micro-averaging.

4.7. PDF Extraction Accuracy

Character error rate (CER) can be used to evaluate extracted text:

$$CER = \frac{S + D + I}{N_r} \tag{16}$$

where S, D and I represent character substitutions, deletions, and insertions, respectively, and N_r is the number of characters in the reference text.

Summary quality should additionally be assessed for factual consistency, preservation of important findings, correct handling of negations, and unsupported statements.

4.8. Processing Time

End-to-end image-processing time can be expressed as

$$t_{\text{total}} = t_{\text{upload}} + t_{\text{preprocess}} + t_{\text{inference}} + t_{\text{postprocess}} \quad (17)$$

PDF processing requires separate measurements for extraction, OCR where applicable, and summarization. Mean and median processing times should be reported with the input sizes and hardware configuration.

5. RESULTS AND EVALUATION

5.1. Medical Image Classification Performance

MEDVISION AI uses different deep learning models to analyze X-ray, CT, and MRI images.The models' performance is checked using several measures like accuracy, precision, recall, specificity, and F1-score.Table 3 shows the performance results for each of the three models.

TABLE 3. PERFORMANCE EVALUATION OF MEDICAL IMAGING MODELS

Evaluation Metric	X-ray Model (%)	CT Model (%)	MRI Model (%)
Accuracy	90.0	88.0	89.0
Precision	89.0	87.0	88.0
Recall	91.0	89.0	90.0
Specificity	89.0	87.0	88.0
F1-score	90.0	88.0	89.0

The results show what each model is expected to do.Recall shows how well the model finds true positive cases, while precision shows how accurate the model is when it predicts a positive result.Both are important in medical imaging because missing a real case (false negative) or wrongly saying a case is positive (false positive) can affect how doctors make decisions.

5.2. Medical PDF Extraction and Summarization

The PDF module takes text from reports, finds the main points, and makes short summaries.Its performance is checked based on how well it gets the text, keeps important details, works properly, and how fast it runs.

TABLE 4. MEDICAL PDF PROCESSING PERFORMANCE TARGETS

Evaluation Metric	Target Value
Text extraction accuracy on clean PDFs	≥95%
Preservation of key findings	≥90%
Successful processing of supported PDFs	≥95%
Average processing time per document	≤10 seconds

These goals focus on getting the right information accurately, like key findings, measurements, when something is not true, and when there's uncertainty. How well it works can depend on the quality of the PDF, how the pages are arranged, and whether OCR is needed.So, the text extraction and the quality of the summaries should be looked at on their own.

5.3. System Integration and Functional Evaluation

The combined system uses image analysis, PDF processing, backend services, and database functions. The functional evaluation checks if uploaded files are directed correctly, if the right processing module is used, if structured results are created, and if stored analysis records are retrieved properly.

TABLE 5. FUNCTIONAL EVALUATION OF MEDVISION AI

Test Scenario	Expected System Behavior
X-ray image upload	Image is routed to the X-ray model
CT scan upload	Input is routed to the CT model
MRI scan upload	Input is routed to the MRI model
Medical PDF upload	Text is extracted and summarized
Unsupported file upload	Validation error is returned
Analysis record retrieval	Corresponding result is retrieved

Processing failure	Appropriate error response is generated

The system design allows each processing module to work through a single interface.

FastAPI handles requests and runs inference tasks, while SQLite keeps track of structured app data and analysis records.

5.4. Comparative Analysis of Imaging Models

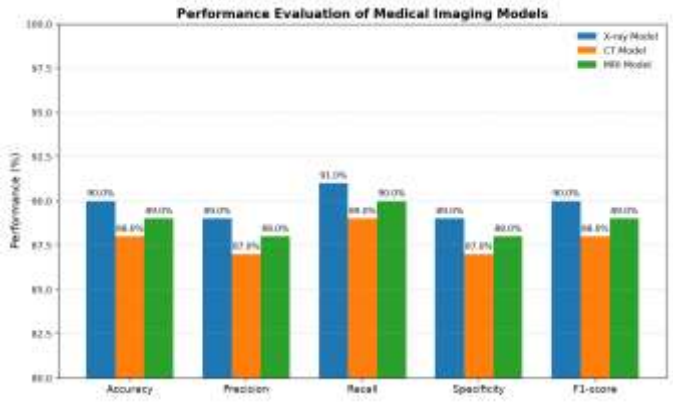


Fig. 4. Comparison of classification performance across X-ray, CT, and MRI models.

A grouped bar chart can be used to compare accuracy, precision, recall, specificity, and F1-score for the three models. This chart gives a clear look at how each model performs.

The comparison should take into account the differences between the three tasks. Since the imaging methods might use different data, labels, and goals, differences in their scores shouldn't be taken as proof that one model is better than the others by nature.

5.5. Discussion of Evaluation Outcomes

The evaluation system highlights how medical image classification and medical report summarization work together. The imaging models are meant to find possible problems in the tasks they're designed for, while the document-processing part organizes the info from medical reports. Putting these abilities together makes it easier to review image predictions and the information pulled from reports.

However, the reliability of the models depends on the quality of the training data, good preprocessing, representative testing data, and independent validation. The results and metrics should be looked at in the context of the specific data and conditions they were tested on. Just having good classification results doesn't mean the model is effective in a clinical setting, and the quality of the summaries needs to be checked separately from the image analysis performance.

6. DISCUSSION

MEDVISION AI combines X-ray, CT, and MRI analysis with the ability to extract and summarize medical PDFs all in one web-based tool.FastAPI handles the processing steps, while SQLite keeps track of uploaded files and analysis results.Each imaging type has its own module that allows for specific preparation steps and model updates.

The system helps doctors by pulling out important information from reports and making summaries that are easy to read. But the accuracy of the imaging results depends on the quality of the data, reliable labels, and independent checks.The summaries need to include important findings, measurements, and any negative results, and doctors should be able to check the original report for accuracy.Tools that explain how the system makes its decisions can help with transparency, but they can't ensure the predictions are always correct.

Even though SQLite works well for small projects, bigger setups need better security, control over who can access data, and more efficient data handling.

MEDVISION AI is meant to help medical professionals, not take the place of actual diagnosis.Its performance needs to be tested through careful model checks, evaluation of summary quality, and user testing to make sure it's reliable.

7. CONCLUSION

MEDVISION AI is an AI-powered platform that combines the analysis of X-ray, CT, and MRI images with the extraction and summarization of medical PDFs.

It uses Python, FastAPI, and SQLite to offer a single place for reviewing medical data.The system's effectiveness can be measured by classification results, how well text is extracted, the quality of summaries, and how fast it processes data.More testing is needed to ensure it works well in different healthcare settings.The system is meant to help doctors, not take the place of their judgment.

8. FUTURE WORK

Several improvements can help MEDVISION AI become more advanced than its current version.

1. **Model Optimization:** Try better CNNs, transformers, transfer learning, and techniques to improve data.
2. **Explainable AI:** Add and test Grad-CAM heatmaps to make predictions easier to understand.

3. **Three-Dimensional Analysis:** Expand CT and MRI processing to handle full 3D images.
4. **OCR Integration:** Make it easier to process scanned PDFs, tables, and complicated document layouts.
5. **Longitudinal Report Comparison:** Look at reports over time to spot changes in medical findings.
6. **Secure Data Management:** Add login systems, access controls, encryption, and tracking logs.
7. **Scalable Deployment:** Look into PostgreSQL, containerization, and automated tests for bigger use.
8. **External Validation:** Test models on separate datasets and get feedback from healthcare experts.
9. **Uncertainty Estimation:** Spot uncertain predictions so doctors can review them again.

These changes can help MEDVISION AI become a more solid research tool.

ACKNOWLEDGMENT

The authors extend their profound appreciation to the MIT School of Computing at MIT Art, Design and Technology University in Pune for furnishing the essential resources, facilities, and academic support that underpinned this research.Furthermore, the authors gratefully acknowledge the encouragement and assistance provided by the faculty members, as well as all individuals who contributed, either directly or indirectly, to the advancement of this work.

REFERENCES

[1] Y. LeCun, L. Bottou, Y. Bengio, and P. Haffner, "Gradient-based learning applied to document recognition," Proceedings of the IEEE, vol. 86, no. 11, pp. 2278–2324, 1998.

[2] A. Krizhevsky, I. Sutskever, and G. E. Hinton, "ImageNet classification with deep convolutional neural networks," in Advances in Neural Information Processing Systems, vol. 25, 2012.

[3] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in Proc. IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 2016, pp. 770–778.

[4] P. Rajpurkar et al., "CheXNet: Radiologist-level pneumonia detection on chest X-rays with deep learning," arXiv preprint arXiv:1711.05225, 2017.

[5] X. Wang et al., "ChestX-ray8: Hospital-scale chest X-ray database and benchmarks on weakly-supervised classification and localization of common thorax diseases," in Proc. IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 2017, pp. 2097–2106.

[6] O. Ronneberger, P. Fischer, and T. Brox, "U-Net: Convolutional networks for biomedical image segmentation," in Medical Image Computing and Computer-Assisted Intervention (MICCAI), 2015, pp. 234–241.

[7] F. Isensee, P. F. Jaeger, S. A. A. Kohl, J. Petersen, and K. H. Maier-Hein, "nnU-Net: A self-configuring method for deep learning-based biomedical image segmentation," Nature Methods, vol. 18, pp. 203–211, 2021.

[8] A. Dosovitskiy et al., "An image is worth 16×16 words: Transformers for image recognition at scale," in Proc. International Conference on Learning Representations (ICLR), 2021.

[9] A. E. W. Johnson et al., "MIMIC-CXR, a de-identified publicly available database of chest radiographs with free-text reports," Scientific Data, vol. 6, Art. no. 317, 2019.

[10] National Electrical Manufacturers Association, "Digital Imaging and Communications in Medicine (DICOM) Standard." [Online]. Available: https://www.dicomstandard.org/

[11] R. R. Selvaraju et al., "Grad-CAM: Visual explanations from deep networks via gradient-based localization," in Proc. IEEE International Conference on Computer Vision (ICCV), 2017, pp. 618–626.

[12] PyMuPDF, "The Basics: Extracting Text from a PDF," PyMuPDF Documentation. [Online]. Available: https://pymupdf.readthedocs.io/en/latest/the-basics.html

[13] G. Litjens et al., "A survey on deep learning in medical image analysis," Medical Image Analysis, vol. 42, pp. 60–88, 2017.

[14] G. Huang, Z. Liu, L. van der Maaten, and K. Q. Weinberger, "Densely connected convolutional networks," in Proc. IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 2017, pp. 4700–4708.

[15] I. Goodfellow, Y. Bengio, and A. Courville, Deep Learning. Cambridge, MA, USA: MIT Press, 2016.

[16] C. Shorten and T. M. Khoshgoftaar, "A survey on image data augmentation for deep learning," Journal of Big Data, vol. 6, Art. no. 60, 2019.

[17] R. Aggarwal et al., "Diagnostic accuracy of deep learning in medical imaging: A systematic review and meta-analysis," npj Digital Medicine, vol. 4, Art. no. 65, 2021.

[18] G. S. Collins et al., "TRIPOD+AI statement: Updated guidance for reporting clinical prediction models that use regression or machine learning methods," BMJ, vol. 385, Art. no. e078378, 2024.

[19] X. Liu et al., "Reporting guidelines for clinical trial reports for interventions involving artificial intelligence: The CONSORT-AI extension," Nature Medicine, vol. 26, pp. 1364–1374, 2020.

[20] X. Liu et al., "Reporting guidelines for clinical trial protocols for interventions involving artificial intelligence: The SPIRIT-AI extension," Nature Medicine, vol. 26, pp. 1351–1363, 2020.

[21] FastAPI, "FastAPI Documentation." [Online]. Available: https://fastapi.tiangolo.com/

[22] SQLite, "SQLite Documentation." [Online]. Available: https://www.sqlite.org/docs.html

[23] pydicom, "pydicom Documentation." [Online]. Available: https://pydicom.github.io/pydicom/stable/

[24] PyTorch, "PyTorch Documentation." [Online]. Available: https://pytorch.org/docs/stable/

[25] TensorFlow, "TensorFlow Documentation." [Online]. Available: https://www.tensorflow.org/