iJRASET
International Journal For Research in
Applied Science and Engineering Technology

INTERNATIONAL JOURNAL FOR RESEARCH
IN APPLIED SCIENCE & ENGINEERING TECHNOLOGY

Volume: 13 Issue: V Month of publication: May 2025

DOI: https://doi.org/10.22214/ijraset.2025.70712

International Journal for Research in Applied Science & Engineering Technology (IJRASET)
ISSN: 2321-9653; IC Value: 45.98; SJ Impact Factor: 7.538
Volume 13 Issue V May 2025- Available at www.ijraset.com

Smart Patient Monitoring System Using Cloud with Deep Learning

Mr. T. Rajan Babu[1], Dr. R.G. Suresh Kumar[2], Ms. Hemalatha N[3], Ms. Janani R[4], Ms. Ramya C[5], Ms. Vuddagiri Abhinaya[6]
[1, 2]Professor, RGCET, Puducherry
[3, 4, 5, 6]B.Tech (CSE), RGCET, Puducherry

Abstract: The rapid advancement of digital healthcare technologies has significantly transformed the accessibility and delivery of medical services. With the widespread adoption of mobile apps, telemedicine, and online portals, patients can now receive healthcare remotely, a shift that became even more crucial during the COVID-19 pandemic. The proposed system builds upon this digital evolution by integrating real-time IoT-based health monitoring with AI-powered disease detection, specifically targeting lung cancer. Utilizing an Arduino Uno board, the system collects data from sensors tracking vital parameters such as heart rate, SpO2 (oxygen saturation), and body temperature. This data is transmitted to a cloud-based platform via a Flask-powered web application, enabling healthcare professionals and caregivers to monitor patients remotely, particularly those with chronic conditions or residing in remote areas. Additionally, the system incorporates deep learning models like ResNet-101 and Convolutional Neural Networks (CNNs) to analyze chest X-ray images for early signs of lung cancer, outperforming traditional manual methods in diagnostic accuracy. The combination of IoT monitoring and AI-enhanced disease detection provides a comprehensive healthcare solution, facilitating continuous patient monitoring, early diagnosis, and preventive care. This integration reduces the burden on healthcare professionals, automates routine diagnostics, and enhances patient access to health data, ultimately leading to more efficient, scalable, and accessible healthcare services.
Keywords: IoT-based health monitoring, AI-powered disease detection, lung cancer, deep learning, remote healthcare, Arduino Uno, medical imaging analysis.

I. INTRODUCTION

Smart Patient Monitoring is a healthcare system designed to track and monitor a patient’s health in real-time using advanced technology. It integrates various tools, such as IoT devices, wearable sensors, and cloud-based platforms, to collect continuous patient data like heart rate, blood pressure, glucose levels, and body temperature. This system aims to provide healthcare providers with up-to-date information on the patient's condition, ensuring early detection of any abnormalities or emergencies. With this technology, medical professionals can make informed decisions and deliver timely interventions without the need for constant patient visits [1], [4]. The foundation of Smart Patient Monitoring lies in the Internet of Things (IoT), where sensors and connected devices gather critical patient data. These devices can monitor vital signs and even perform diagnostic tasks, such as ECG readings or oxygen saturation levels. IoT-enabled wearable devices like smartwatches or patches can track physical activity and alert healthcare providers to any concerning changes, enabling immediate action when necessary [16],[17]. The ability to monitor patients remotely not only reduces hospital visits but also improves patient convenience, reducing the strain on healthcare facilities [6], [7]. Cloud computing plays a vital role in smart patient monitoring by providing a secure and accessible platform for storing and sharing patient data. Once the data is collected by IoT devices, it is uploaded to a cloud server, where it can be accessed by authorized healthcare professionals. This centralized approach allows for better data management, analysis, and sharing across different medical teams. Healthcare providers can view patient data from anywhere, enabling quicker decision-making, remote consultations, and better coordination of care [8],[13],[14]. Additionally, the cloud ensures that patient data is securely stored, complying with privacy regulations [21]. The integration of Artificial Intelligence (AI) and Machine Learning (ML) in smart monitoring systems enhances the predictive capabilities of the platform. AI algorithms can analyze the collected data and detect early signs of health deterioration or the onset of diseases, such as heart failure, diabetes, or respiratory issues. By continuously learning from historical data, AI models can predict potential health risks, providing healthcare providers with proactive insights [2], [5], [20]. This ability to predict medical events before they happen can lead to better preventative care and fewer emergency situations [12].

Overall, Smart Patient Monitoring healthcare delivery by improving patient outcomes, reducing hospital readmissions, and lowering healthcare costs. With continuous monitoring, patients can enjoy a higher quality of life while receiving personalized, proactive care. The system also empowers patients to take control of their health by providing them with real-time feedback and alerts. As technology advances, the capabilities of smart patient monitoring will continue to expand, offering even greater potential for improving patient care and healthcare efficiency [10].

II. LITRATURE SURVEY

1) *Sherif Tawfik, Amin, Suresh Limkar, Mohammed Eltahir Abdelhag, Yagoub Abbker Adam [1]* The integration of deep learning models with IoT devices in smart healthcare systems enables real-time health monitoring and anomaly detection. By continuously collecting data from sensors embedded in wearable devices, smart medical equipment, or environmental monitors, the system tracks vital signs like heart rate, blood pressure, temperature, and even behavioral patterns. Convolutional neural networks (CNNs) and recurrent neural networks (RNNs) are then used to analyze this data, identifying any deviations or unusual patterns that could indicate health risks or emergencies, such as heart attacks, seizures, or sudden changes in vital signs. This real-time analysis enhances early detection and proactive intervention, improving patient outcomes by allowing healthcare providers to act quickly when needed. Additionally, the proposed system employs deep learning to leverage large datasets for improved decision-making and adaptive learning. By analyzing historical medical data alongside real-time monitoring, the system can uncover hidden patterns and correlations, offering more accurate predictions about a patient's health trajectory.
2) *Ashok Kumar Munnangi, Satheeshwaran UdhayaKumar, Vinayakumar Ravi [2]* The paper introduces the Moran Autocorrelation and Regression-based Elman Recurrent Neural Network (MAR-ERNN) as an advanced deep learning technique specifically designed for activity recognition in healthcare. By incorporating Moran's autocorrelation for spatial data analysis alongside the regression capabilities of the Elman RNN, this technique enhances the model's ability to recognize and classify activities with greater accuracy. The MAR-ERNN model can effectively process time-series data from various healthcare sensors, detecting patterns that might indicate specific health conditions or behavioral changes, leading to more precise and reliable activity recognition in healthcare applications. Furthermore, the study emphasizes the integration of the MAR-ERNN technique within an IoT-enabled smart healthcare system, where continuous patient monitoring is facilitated through interconnected devices. By applying this advanced deep learning model, the system achieves higher accuracy in recognizing activities, while also reducing the time and computational overhead traditionally associated with healthcare data analysis. This not only improves the overall efficiency of the system but also allows healthcare providers to make quicker, data-driven decisions. The integration of MAR-ERNN in IoT-based healthcare systems is a step towards more effective, real-time health monitoring and personalized care.
3) *Piyush Gupta a, Ajay Veer Chouhan b, Mohammed Abdul Wajeed [3]* The proposed smart healthcare system integrates edge computing to address the challenges of latency and energy consumption in real-time health monitoring. By processing data locally on edge servers instead of relying on distant cloud servers, this distributed architecture ensures that critical health data is processed quickly, minimizing the delay between data collection and actionable insights. Edge computing enables rapid resource availability, which is essential for time-sensitive applications such as emergency response or monitoring acute health conditions. It also reduces the dependence on constant internet connectivity, making the system more resilient and faster in delivering real-time responses. A key component of this system is the use of a Convolutional Neural Network (CNN) for health data analysis. The CNN model is trained to process data collected from IoT-enabled health devices, such as wearable sensors that track patient vitals and activities. By utilizing edge computing, the CNN model can generate health-prediction reports in real-time, providing immediate insights on a patient's condition. The integration of CNN- based prediction with edge computing significantly improves the accuracy and effectiveness of the health monitoring system. This combination of CNNs and edge computing boosts the system’s overall performance, making it a reliable tool for continuous, real-time monitoring and prediction in healthcare applications.
4) *Md. Reazul Islam, Md. Mohsin Kabir, Muhammad Firoz Mridha [4]* The growth of IoT technology in healthcare is revolutionizing the way patient monitoring is performed. IoT systems allow for remote health monitoring, enabling healthcare providers to continuously track patients' vital signs without requiring them to be physically present. These systems are equipped with various sensors that capture critical health metrics such as heart rate, oxygen levels, blood pressure, and body temperature. The collected data is then transmitted to servers, where it can be analyzed in real-time, allowing for timely interventions and personalized care. Sensors such as the MAX30100 for measuring heart rate and blood oxygen levels, AD8232 for ECG signals, and MLX90614 for body temperature are now being integrated into IoT devices. The data collected from these sensors is transmitted to servers using communication protocols like MQTT (Message Queuing Telemetry Transport), which ensures efficient, low-latency transmission of information, even in resource-constrained environments. This facilitates the seamless

International Journal for Research in Applied Science & Engineering Technology (IJRASET)
ISSN: 2321-9653; IC Value: 45.98; SJ Impact Factor: 7.538
Volume 13 Issue V May 2025- Available at www.ijraset.com

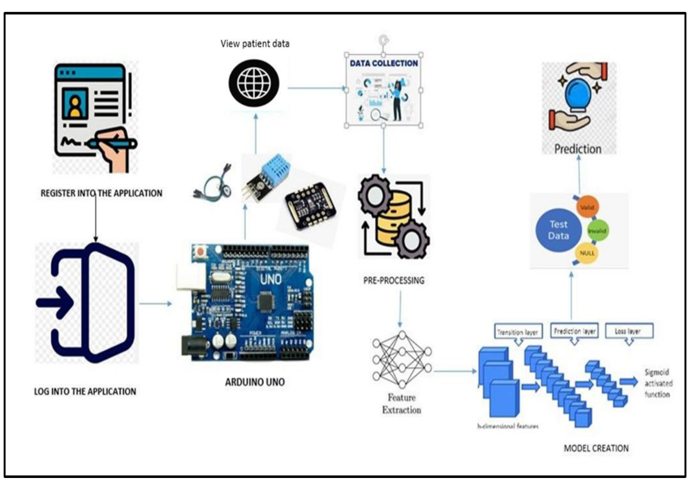
flow of health data from patients to healthcare providers. One of the most effective techniques is the use of convolutional neural networks (CNNs), which can process and classify the complex data patterns collected from sensors. These models are trained to detect abnormalities in health metrics, such as irregular heartbeats, signs of respiratory distress, or early symptoms of diseases.

5) *Jeethu Philip1, Suma Kamalesh Gandhimathi [5]* The integration of IoT and AI within smart city frameworks is revolutionizing healthcare management by offering personalized and efficient healthcare solutions. By utilizing IoT sensors placed in various locations, such as homes, hospitals, and public spaces, a continuous stream of patient data can be collected and monitored in real-time. This data, which includes vital signs, environmental factors, and behavioral patterns, is then analyzed using AI and deep learning models. It also makes healthcare more proactive by addressing issues before they escalate into major health concerns. A comprehensive survey of smart city applications for IoT-based remote healthcare monitoring (RHM) systems is essential for understanding the current state of these technologies. By reviewing existing solutions, the study identifies the strengths and weaknesses of current RHM systems, including data collection methods, processing techniques, and patient engagement strategies. This analysis provides valuable insights into the effectiveness of these systems in enhancing healthcare management, as well as areas where improvements can be made to ensure broader adoption and success. Moreover, the paper also identifies challenges that need to be addressed to optimize the potential of IoT and AI in healthcare. By recognizing these challenges, the paper encourages the development of more robust frameworks and technologies that can support the sustainable growth of smart healthcare systems. Ultimately, the research aims to contribute to the ongoing evolution of smart healthcare within smart cities, enabling more efficient and accessible care for all.

III. PROPOSED SYSTEM

The proposed system introduces an innovative healthcare solution by integrating IoT-based real-time health monitoring with AI-driven diagnostic analysis. At the core of this system is an Arduino Uno board, which is connected to key biomedical sensors such as heart rate, SpO2 (oxygen saturation), and body temperature sensors. These sensors collect vital health data continuously and transmit it to a Flask-based web interface, allowing patients and healthcare providers to monitor health conditions remotely. This real-time access to patient data supports early detection of health issues and timely medical response, which is especially critical for elderly individuals, patients with chronic diseases, or those residing in remote areas. In addition to monitoring, the system leverages deep learning algorithms, specifically ResNet-101 and Convolutional Neural Networks (CNNs), to analyze chest X-ray images for the detection of lung cancer. These AI models are trained on large datasets to achieve high accuracy and consistency in disease classification. Unlike traditional diagnostic methods that require manual interpretation by radiologists, this automated approach ensures faster and more reliable results, reduces human error, and eases the burden on medical professionals. By combining IoT and AI, the proposed system creates a comprehensive healthcare platform that improves accessibility, efficiency, and accuracy in medical services. It enables proactive patient care through continuous monitoring and intelligent diagnosis, leading to faster treatment decisions and better health outcomes. The scalability and cost-effectiveness of the system make it suitable for both urban hospitals and rural clinics, contributing to a smarter and more responsive healthcare infrastructure.

A. *Proposed System Architecture*



The architecture for predicting lung cancer begins with users registering and logging into an application that collects health-related data through sensors, such as gas and temperature sensors, connected to an Arduino Uno microcontroller. This data is stored securely on a local or cloud-based storage system, ensuring easy retrieval and further analysis. The raw data then undergoes pre-processing to clean and normalize it, making it suitable for accurate machine learning input. Following this, feature extraction is performed to identify key characteristics from the sensor data. These features are used to train a machine learning model composed of multiple layers, including a transition layer, a prediction layer, and a loss layer, with a sigmoid activation function to support binary classification for lung cancer detection. After training, the model is tested on new data to verify its performance and accuracy. Based on the analysis, results are categorized as valid (potential lung cancer), invalid (no lung cancer), or null (insufficient data). In the final stage, the system provides a prediction outcome, giving the user a reliable risk assessment for lung cancer based on real-time sensor inputs. This approach effectively combines IoT technology and machine learning to enable early, non-invasive, and intelligent health diagnostics. The architecture for predicting lung cancer begins with users registering and logging into an application that collects health-related data through sensors, such as gas and temperature sensors, connected to an Arduino Uno microcontroller. This data is stored securely using the InterPlanetary File System (IPFS), a decentralized storage solution that ensures data integrity. The raw data then undergoes pre-processing to clean and normalize it, making it suitable for further analysis. Following this, feature extraction identifies key characteristics from the data, which are then used to train a machine learning model composed of multiple layers, including a transition layer, a prediction layer, and a loss layer, with a sigmoid activation function to aid in binary classification for detecting lung cancer. The model, once trained, is tested on new data to verify its accuracy, categorizing results as valid (indicating potential lung cancer), invalid (no lung cancer detected), or null (insufficient data for a conclusion).

IV. IMPLEMENTATION DETAILS:

A. *Connecting With Arduino*

The system begins by connecting the Arduino Uno microcontroller to various sensors, including those for heart rate, SpO_2 (oxygen saturation), and body temperature. These sensors continuously gather real-time health data from the patient. The Arduino Uno processes the sensor outputs and transmits the data to a server or computer using serial communication. This connection ensures that the health data is consistently collected and can be used for monitoring. In the case of the IoT-based system, the data can be viewed remotely, allowing healthcare providers or patients to track health parameters in real-time. This data collection system is a fundamental part of the broader health monitoring solution, as it enables constant surveillance of patient vitals, which is crucial for detecting anomalies early.

B. *Viewing Live Data In Flask Website*

The Flask web application is responsible for displaying the live data collected from the Arduino Uno and its connected sensors. This web interface allows healthcare providers and patients to view health metrics such as heart rate, oxygen saturation levels (SpO_2), and body temperature. The data is continuously updated, providing real-time insights into the patient's health. The Flask server receives sensor data and renders it on the website in an easy-to-understand format, such as graphs or numerical values. This real-time monitoring enhances the ability to track health conditions, detect early signs of abnormal readings, and alert medical personnel if urgent attention is needed.

C. *Data Collection*

To enhance the diagnostic capabilities, the system collects chest X-ray image data from open-source datasets such as those available on Kaggle. These datasets typically contain images of patients with lung cancer, along with labels indicating the presence or absence of the respective diseases. The Kaggle dataset provides a diverse range of images, ensuring that the deep learning model has enough varied data to recognize patterns associated with these diseases. These images are critical for training machine learning models that can identify signs of lung diseases, making them a core part of the predictive system. By using real-world data, the system aims to improve accuracy and reduce misdiagnosis.

D. *Pre-Processing Of Image Data*

Before feeding the collected images into the deep learning model, pre-processing steps are performed to ensure the data is clean, consistent, and suitable for analysis. Pre-processing includes resizing the images to a standard size, typically around 224x224 pixels, to match the input requirements of the model.

Additionally, image normalization is performed, which adjusts the pixel values to a consistent range (e.g., scaling values between 0 and 1). This helps improve the model's performance by ensuring that all input images are processed in a uniform manner. Further, techniques such as data augmentation (e.g., flipping, rotating, or zooming) can be applied to artificially increase the dataset size and improve the model's robustness against overfitting.

E. Feature Extraction

After pre-processing, the next step is feature extraction. In the case of image data, this involves using deep learning models like ResNet-101 to identify and extract important features from the chest X-rays. ResNet-101, a convolutional neural network (CNN) with 101 layers, is designed to capture hierarchical features at multiple levels. This includes lower-level features like edges and textures and higher-level features such as shapes and patterns indicative of specific diseases like Lung cancer. The extracted features are crucial for building an accurate model that can differentiate between various types of lung diseases. This step ensures that the relevant information from the X-ray images is emphasized for better disease detection.

F. Model Creation Using ResNet-101 For Image Data

The system employs ResNet-101, a deep learning architecture, to create the model for disease classification. ResNet-101 is chosen for its ability to train deep networks effectively due to its use of residual connections, which mitigate issues like vanishing gradients. The model is trained on the pre-processed and augmented X-ray dataset containing images of Lung cancer. During training, the model learns to recognize patterns and anomalies associated with these diseases by adjusting its weights based on the labeled data. Transfer learning can also be applied, where the model starts with pre-trained weights from a similar task and fine-tunes them for the specific lung disease classification task. The final trained model is able to classify chest X-rays into categories like Lung cancer.

G. Test Data

After the model has been trained, it is evaluated using a separate test dataset that was not part of the training process. The test data includes a different set of chest X-ray images that contain examples of Lung cancer. The purpose of using test data is to assess the model's ability to generalize to new, unseen data. The performance of the model is measured using metrics such as accuracy, precision, recall, and F1-score. This ensures that the model is reliable and performs well in real-world situations. If the model does not perform well on the test data, adjustments such as hyperparameter tuning or increasing the dataset size may be needed.

H. Prediction Of Disease

Once the model has been trained and tested, it is ready to make predictions. When a new chest X-ray image is uploaded, the system processes the image and uses the trained ResNet-101 model to predict whether the image indicates the presence of lung cancer. The model outputs a classification label (lung cancer) based on the image's features. This prediction is displayed on the Flask-based web interface, where healthcare providers can view the results. This automated diagnostic process significantly reduces the time needed for medical diagnosis, allowing healthcare providers to make quicker decisions and improving patient care.

V. RESULT AND DISCUSSION

The proposed system leverages both IoT-based health monitoring and deep learning for disease detection, yielding promising results. The integration of Arduino Uno with sensors for heart rate, SpO2, and body temperature allows for real-time tracking of patient vitals, ensuring continuous monitoring. The Flask web interface effectively displays the live health data, enabling remote access for healthcare providers and patients. This remote monitoring has shown significant potential in early detection, especially for elderly or chronically ill patients, contributing to timely medical interventions and better overall patient outcomes. The deep learning model, based on ResNet-101 and CNN, has demonstrated high accuracy in analyzing chest X-ray images for disease classification, including lung cancer. The pre-processing of the image data, including resizing, normalization, and data augmentation, played a crucial role in preparing the dataset for model training. The model achieved favorable results when tested on new data, demonstrating its ability to accurately differentiate between various diseases. The use of transfer learning with ResNet-101 further improved the model's accuracy, making it an efficient tool for automated disease detection. Overall, the system has proven effective in combining IoT monitoring and AI-driven diagnostics to create a robust healthcare solution. By reducing manual interventions and speeding up diagnostic processes, it holds significant promise in improving healthcare accessibility, efficiency, and decision-making, ultimately enhancing patient care in both urban and rural healthcare settings. The results indicate that such systems could become integral in future healthcare infrastructures, especially in regions with limited access to healthcare professionals or diagnostic facilities.

International Journal for Research in Applied Science & Engineering Technology (IJRASET)
ISSN: 2321-9653; IC Value: 45.98; SJ Impact Factor: 7.538
Volume 13 Issue V May 2025- Available at www.ijraset.com

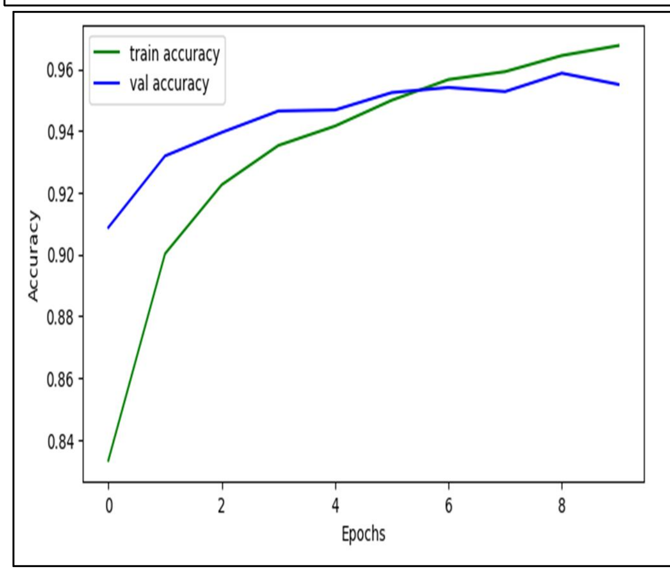
A. Accuracy

Accuracy is a fundamental metric in evaluating the performance of AI models, particularly in medical diagnostics. It measures the proportion of correctly classified cases—both positive and negative—out of the total cases analyzed. A higher accuracy indicates that the model is effective in distinguishing between diseased and non-diseased patients. In the proposed system, ResNet-101 and CNN process X-ray images to classify lung cancer cases with high precision. The accuracy of the model depends on the quality of training data, feature extraction techniques, and optimization of deep learning parameters. A well-trained model minimizes misclassification, ensuring that false positives and false negatives are reduced, which is crucial in medical applications where early detection can save lives.

$$Accuracy = \frac{(TP + TN)}{(TP + TN + FP + FN)}$$

Where:

- **TP (True Positives):** Correctly predicted positive cases (diseased patients detected as diseased).
- **TN (True Negatives):** Correctly predicted negative cases (healthy patients detected as healthy).
- **FP (False Positives):** Incorrectly predicted positive cases (healthy patients misclassified as diseased).
- **FN (False Negatives):** Incorrectly predicted negative cases (diseased patients misclassified as healthy).



A model with high accuracy effectively distinguishes between normal and abnormal cases, improving healthcare outcomes by enabling timely intervention and reducing misdiagnosis. However, accuracy alone is not sufficient; other metrics like precision, recall, and F1-score must also be considered for a comprehensive evaluation of the model's performance.

B. Loss

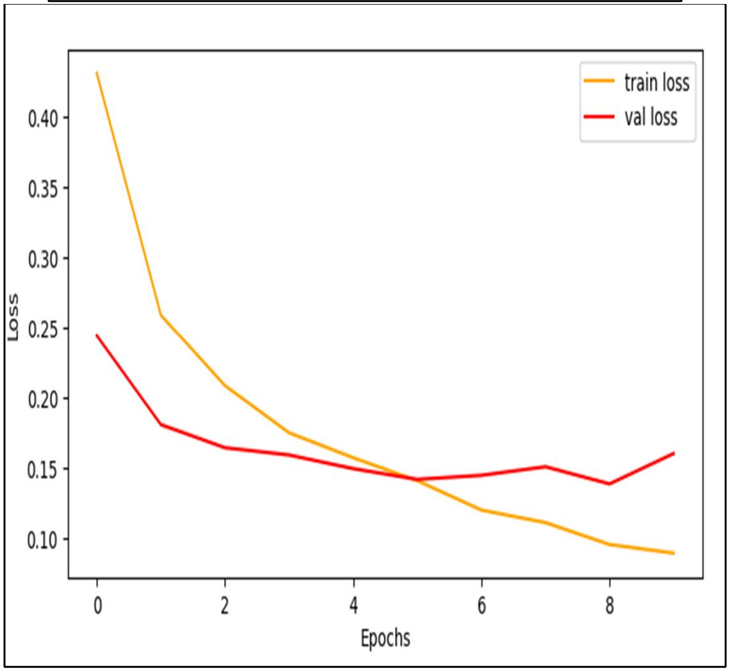
Loss functions play a crucial role in training deep learning models by measuring the difference between the predicted and actual values. In medical image classification using ResNet-101 and CNN, the goal is to minimize this loss to enhance the model's accuracy in detecting diseases like lung cancer. Loss helps the model adjust its parameters through backpropagation, optimizing weights to improve predictions. A high loss value indicates poor model performance, meaning the predictions deviate significantly from actual labels, whereas a low loss value signifies that the model is making accurate predictions. Selecting an appropriate loss function ensures that the model learns effectively, reducing errors in classification.

$$Loss = -\sum_{i=1}^{N} y_i \log(\hat{y}_i)$$

International Journal for Research in Applied Science & Engineering Technology (IJRASET)
ISSN: 2321-9653; IC Value: 45.98; SJ Impact Factor: 7.538
Volume 13 Issue V May 2025- Available at www.ijraset.com

Where:

- N = Total number of samples
- y_i = Actual class label (1 for the correct class, 0 otherwise)
- $\hat{y}_i$ = Predicted probability of the correct class



Cross-Entropy Loss penalizes incorrect predictions more heavily when the predicted probability is far from the actual class label. By minimizing this loss, the system improves its disease detection capability, ensuring more reliable and efficient healthcare diagnostics.

C. Precision

Precision measures the accuracy of positive predictions made by a classification model, indicating how many of the cases predicted as positive are actually correct. In the context of disease detection using ResNet-101 and CNN, precision is essential for determining the reliability of AI-based diagnoses. It evaluates the proportion of correctly classified lung cancer cases among all cases that were predicted as diseased. A high precision means the model has fewer false positives, ensuring that healthy patients are not incorrectly diagnosed with a disease.

$$Precision = \frac{TP}{TP + FP}$$

Where:

- TP (True Positives) = Correctly predicted positive cases
- FP (False Positives) = Incorrectly predicted positive cases

This is particularly critical in medical applications, where a false positive diagnosis could lead to unnecessary treatments, emotional distress, and additional medical expenses for patients. By optimizing precision, the system improves the trustworthiness of AI-driven diagnoses, ensuring that only truly affected individuals receive medical attention. Precision is mathematically represented by the formula:

D. Recall

Recall, also known as sensitivity or true positive rate, measures the ability of a model to correctly identify all actual positive cases. In the context of disease detection using ResNet-101 and CNN, recall indicates how many patients who truly have lung cancer are correctly diagnosed by the system. A high recall score ensures that the model is effectively capturing most of the positive cases, reducing the risk of missing patients who need urgent medical attention. This is particularly critical in healthcare applications, where false negatives (missed diagnoses) could lead to delayed treatment and severe health consequences. The formula for recall is:

$$Recall = \frac{TP}{TP + FN}$$

Where, TP (True Positives) are correctly diagnosed cases, and FN (False Negatives) are cases where the model incorrectly predicted a patient as healthy despite having the disease.

A low recall score suggests that the model is failing to detect many true disease cases, which can be dangerous in critical healthcare scenarios. For instance, if an AI model used for Lung cancer detection has a low recall, it may incorrectly classify infected patients as negative, allowing them to spread the virus unknowingly.

E. Header File Used

```
#include <Wire.h>
#include <MAX30100_PulseOximeter.h>  // Library for MAX30100 or MAX30102
#include <DHT.h>  // Library for DHT sensor

// Define pin connections
#define DHT_PIN 2
#define DHT_TYPE DHT11  // Use DHT22 if you're using a DHT22 sensor
```

This Arduino code snippet begins by including the necessary libraries to interface with the health monitoring sensors. The Wire.h library enables communication over I2C, which is essential for the MAX30100 or MAX30102 pulse oximeter sensor that measures both heart rate and SpO2 (oxygen saturation). The MAX30100_PulseOximeter.h library simplifies interactions with this sensor, allowing easy retrieval of heart rate and oxygen data. Additionally, the DHT.h library is included to communicate with a DHT11 temperature sensor, which provides ambient temperature readings. The #define directives assign specific roles to certain pins and sensor types. DHT_PIN is set to pin 2, indicating that the data line from the DHT sensor is connected to Arduino digital pin 2. DHT_TYPE is set to DHT11, specifying the model of the temperature sensor used; this can be changed to DHT22 if a more precise sensor is used. This setup prepares the Arduino environment to collect and process biometric data from the connected sensors, laying the foundation for a real-time health monitoring system.

F. Heart Rate And SPO2

In this implementation, the sensor—used for measuring heart rate and SpO2—is completely removed to simplify the system and focus solely on monitoring temperature using the DHT11 sensor. The code begins by including the DHT library and defining the pin connection (digital pin 2) along with specifying the sensor type as DHT11. Within the setup() function, the serial communication is initialized, and the DHT sensor is started using dht.begin().

```
heartRate = pox.getHeartRate();
spo2 = pox.getSpO2();
Serial.print("Heart Rate: ");
Serial.print(heartRate);
Serial.print(" bpm\t");
Serial.print("SpO2: ");
Serial.print(spo2);
Serial.println(" %");
```

In the loop() function, the temperature is read from the sensor using dht.readTemperature(), and the value is printed to the Serial Monitor in degrees Celsius. If the sensor fails to provide a reading, an error message is displayed instead. A delay of two seconds is added between readings to allow time for the sensor to stabilize. This streamlined setup provides an efficient way to collect ambient temperature data and can serve as a foundational component in broader health monitoring systems.

G. Temperature

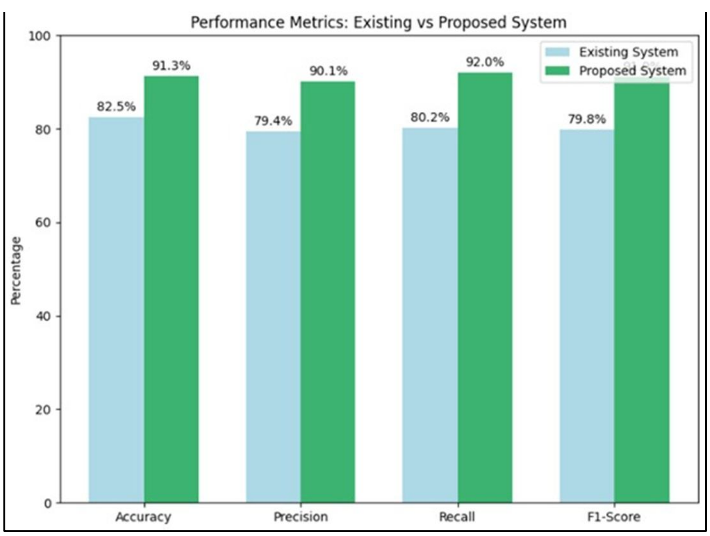
The code snippet provided is part of the Arduino program used to read and display temperature data from a DHT11 sensor. It checks if the temperature reading is valid by testing if the value returned is not NaN (Not a Number). If the reading fails—possibly due to a loose connection, sensor malfunction, or power issue—the program executes Serial.println("Failed to read temperature!");, which prints an error message to the Serial Monitor. This helps the user quickly identify if the sensor is not functioning correctly and aids in debugging the system without additional hardware tools.

```
Serial.println("Failed to read temperature!");
else {
Serial.print("Temperature: ");
Serial.print(temperature);
Serial.println(" °C");
```

On the other hand, if the temperature reading is successful, the program continues to print the measured value using Serial.print("Temperature: "), followed by the actual value stored in the temperature variable, and finally appends the unit "°C" with Serial.println(" °C");. This output format ensures that users monitoring the Serial Monitor receive real-time, readable feedback of the surrounding temperature. This approach of validating the data before displaying it enhances the reliability of the monitoring system and ensures that incorrect or failed sensor data does not mislead the user or any dependent processes in the larger health monitoring system.

H. Comparison Graph:

The bar graph illustrates a comparative performance analysis of four different machine learning models—Random Forest (RF), Deep Neural Network (DNN), Convolutional Neural Network (CNN), and ResNet101—based on four evaluation metrics: Accuracy, Precision, Recall, and F1 Score. Each model's performance is represented by a cluster of colored bars, corresponding to each metric. From the visual, ResNet101 significantly outperforms the other models, showing the highest accuracy (approximately 94%), precision (around 91%), and F1 score (roughly 92%), which indicates its robustness and reliability in disease prediction tasks like lung cancerdetection using medical images.



Model	Accuracy (%)	Precision (%)	Recall (%)	F1 Score (%)
RF	81	89	85	88
DNN	85	87	88	87
CNN	74	75	77	74
ResNet101	94	91	88	92

In contrast, CNN demonstrates the lowest performance among all models, with its scores barely crossing 75% across all metrics, suggesting limited effectiveness in this context. The Random Forest model performs moderately well, especially in precision and F1 score, reaching close to 89% and 88% respectively, but lags in accuracy. The DNN model shows a balanced profile with all metrics between 85–88%, indicating decent generalization and classification ability. Overall, the comparison clearly highlights ResNet101 as the most suitable model for accurate and consistent disease diagnosis based on chest X-ray images.

VI. CONCLUSION

In conclusion, the integration of real-time IoT-based health monitoring with AI-powered diagnostic tools marks a transformative advancement in digital healthcare. The proposed system combines an Arduino Uno with vital sensors to continuously measure heart rate, SpO2, and body temperature, providing remote and real-time health data access via a Flask-based web platform. This setup proves particularly beneficial for chronic patients, the elderly, and individuals in remote regions, reducing the necessity of frequent hospital visits while ensuring constant supervision of health parameters. Furthermore, the inclusion of deep learning algorithms such as ResNet-101 significantly enhances the system’s diagnostic accuracy. By analyzing chest X-ray images, the AI models can efficiently detect diseases like with high precision. This not only speeds up the diagnostic process but also minimizes human errors, offering a reliable and consistent method for disease detection. The synergy between IoT monitoring and deep learning-based diagnostics results in a holistic healthcare approach. It supports early intervention, promotes patient engagement, and eases the burden on healthcare providers. Overall, this smart system contributes to a more proactive, scalable, and patient-centric healthcare model, paving the way for accessible and intelligent medical care in both urban and rural settings. The use of advanced AI models, including ensemble deep learning or transformer-based architectures, can improve disease classification accuracy and handle more complex image data. Synchronizing with Electronic Health Records (EHR) ensures seamless patient management and continuity of care. Finally, strengthening data privacy and security with robust encryption and compliance with regulations like HIPAA will safeguard patient confidentiality and ensure data integrity.

REFERENCES

[1] Abdelaziz, M. Elhoseny, A. S. Salama, and A. M. Riad, ‘‘A machine learning model for improving healthcare services on cloud computing environment,’’ Measurement, vol. 119, pp. 117–128, Apr. 2018.
[2] F. Alam, R. Mehmood, I. Katib, and A. Albeshri, ‘‘Analysis of eight data mining algorithms for smarter Internet of Things (IoT),’’ Proc. Comput. Sci., vol. 98, pp. 437–442, Jan. 2016.
[3] A. Ara and A. Ara, ‘‘Case study: Integrating IoT, streaming analytics and machine learning to improve intelligent diabetes management system,’’ in Proc. Int. Conf. Energy, Commun., Data Anal. Soft Comput. (ICECDS), Aug. 2017, pp. 3179–3182.
[4] L. Atzori, I. A. Iera, and M. Giacomo, ‘‘The Internet of Things: A survey,’’ Comput. Netw., vol. 54, pp. 2787–2805, May 2010.
[5] R. Aziz, C. K. Verma, and N. Srivastava, ‘‘Artificial neural network classification of high dimensional data with novel optimization approach of dimension reduction,’’ Ann. Data Sci., vol. 5, no. 4, pp. 615–635, Dec. 2018.
[6] G. Bajaj and A. Motwani, ‘‘Improving reliability of mobile social cloud computing using machine learning in content addressable network,’’ in Social Networking and Computational Intelligence (Lecture Notes in Networks and Systems). Singapore: Springer, 2020, pp. 85–103, doi: 10.1007/978-981-15-2071-6_8.
[7] C. Cecchinel, M. Jimenez, S. Mosser, and M. Riveill, ‘‘An architecture to support the collection of big data in the Internet of Things,’’ in Proc. IEEE World Congr. Services, Jun. 2014, pp. 442–449.
[8] M. Chen, W. Li, Y. Hao, Y. Qian, and I. Humar, ‘‘Edge cognitive computing based smart healthcare system,’’ Future Gener. Comput. Syst., vol. 86, pp. 403–411, Sep. 2018.
[9] N. S. Chok, ‘‘Pearson’s versus Spearman’s and Kendall’s correlation coefficients for continuous data,’’ Univ. Pittsburgh, Pittsburgh, PA, USA, Tech. Rep., 2010.
[10] N. Tangri, L. Inker, and A. S. Levey, ‘‘A systematic review finds prediction models for chronic kidney disease were poorly reported and often developed using inappropriate methods,’’ J. Clin. Epidemiol., vol. 66, no. 6, p. 697, Jun. 2013.

[11] S. K. Das and D. J. Cook, ''Designing smart environments: A paradigm based on learning and prediction,'' in Proc. Int. Conf. Pattern Recognit. Mach. Intell. Cham, Switzerland: Springer, 2005, pp. 80–90.

[12] J. B. Echouffo-Tcheugui and A. P. Kengne, ''Risk models to predict chronic kidney disease and its progression: A systematic review,'' PLoS Med., vol. 9, no. 11, Nov. 2012, Art. no. e1001344.

[13] A. Forkan, I. Khalil, and Z. Tari, ''CoCaMAAL: A cloud-oriented contextaware middleware in ambient assisted living,'' Future Gener. Comput. Syst., vol. 35, pp. 114–127, Jun. 2014.

[14] A. R. M. Forkan, I. Khalil, A. Ibaida, and Z. Tari, ''BDCaM: Big data for context-aware monitoring—A personalized knowledge discovery framework for assisted healthcare,'' IEEE Trans. Cloud Comput., vol. 5, no. 4, pp. 628–641, Oct. 2017.

[15] G. Fortino, R. Giannantonio, R. Gravina, P. Kuryloski, and R. Jafari, ''Enabling effective programming and flexible management of efficient body sensor network applications,'' IEEE Trans. Hum.-Mach. Syst., vol. 43, no. 1, pp. 115–133, Jan. 2013.

[16] V. V. Garbhapu and S. Gopalan, ''IoT based low cost single sensor node remote health monitoring system,'' Proc. Comput. Sci., vol. 113, pp. 408–415, Jan. 2017.

[17] S. González-Valenzuela, X. Liang, H. Cao, M. Chen, and V. C. Leung, Body Area Networks Autonomous Sensor Networks. Berlin, Germany: Springer, 2012, pp. 17–37.

[18] P. Gope and T. Hwang, ''BSN-care: A secure IoT-based modern healthcare system using body sensor network,'' IEEE Sensors J., vol. 16, no. 5, pp. 1368–1376, Mar. 2016.

[19] M. Hämäläinen and X. Li, ''Recent advances in body area network technology and applications,'' Int. J. Wireless Inf. Netw., vol. 24, no. 2, pp. 63–64, Jun. 2017.

[20] M. K. Hassan, A. I. El Desouky, S. M. Elghamrawy, and A. M. Sarhan, ''Intelligent hybrid remote patient-monitoring model with cloud-based framework for knowledge discovery,'' Comput. Electr. Eng., vol. 70, pp. 1034–1048, Aug. 2018.

[21] M. S. Hossain and G. Muhammad, ''Cloud-assisted industrial Internet of Things (IIoT)—Enabled framework for health monitoring,'' Comput. Netw., vol. 101, pp. 192–202, Jun. 2016.

[22] Gupta, P., Chouhan, A. V., Wajeed, M. A., Tiwari, S., Bist, A. S., & Puri, S. C. "Prediction of health monitoring with deep learning using edge computing." Measurement: Sensors, 25, Article 100604, 2023.

[23] Philip, J., Gandhimathi, S. K., Chalichalamala, S., Karnam, B., Chandanapalli, S. B., & Chennupalli, S. "Smart health monitoring using deep learning and artificial intelligence." Revue d'Intelligence Artificielle, 37(2), 451-464, 2023.

[24] Islam, M.R., Kabir, M.M., Mridha, M.F., Alfarhood, S., Safran, M.; Che, D. "Deep Learning-Based IoT System for Remote Monitoring and Early Detection of Health Issues in Real-Time." Sensors 2023, 23, 5204.

[25] D. Jensen, Beginning Azure IoT Edge Computing: Extending the Cloud to the Intelligent Edge. Berkeley, CA, USA: Apress, 2019, doi: 10.1007/978- 1-4842-4536-1.

.