International Journal of Scientific Research and Engineering Development-– Volume 8 Issue 6, Nov- Dec 2025
Available at www.ijsred.com

RESEARCH ARTICLE OPEN ACCESS

DETECTION OF ROAD IN SATELLITE IMAGERY

Rajeshwari S Hiremath*, Mrs. Geetha N B **
*(Computer Science & Engineering, University BDT College, Davanagere
Email: rajuhiremath738@gmail.com)
**(Computer Science & Engineering, University BDT College, Davanagere
Email: geethanb291979@gmail.com)

----------**********----------

Abstract:

The introduction of computer vision and deep learning technology has transformed the field of urban infrastructure analysis, especially in the field of automated roaddetection. This study presents a strong andefficient methodology for detecting road in urban environments using a mix of OpenCV, TensorFlow, and specialized deep learning models. Using convolutional neural networks (CNNs), the suggested system trained on large datasets of urban imagery to accurately segment road from other elements such as roads, buildings, and vegetation. By integrating TensorFlow's sophisticated deeplearning capabilities with OpenCV's image processing functions, precision of roaddetection but also optimizes computational efficiency, enabling real-time apps to use it. The utilization of normalization techniques, like those offered by TensorFlow Addons, further improves model performance by ensuring consistent input data quality, which is essential to preserving high precision. in diverse environmental conditions. The procedure incorporates several important steps the implementation process: image normalization, model prediction, and result visualization. Initially, The input pictures arepreprocessed to normalize RGB values, ensuring uniformity across the dataset. Subsequently, the normalized pictures are entered into the pre-trained CNN model, which outputs a probability map showing the existence of road

Keywords - Liver Disease Prediction, XGBoost, SHAP Analysis, Machine Learning, Interpretability, Django Web Application.

.

----------**********----------

I. INTRODUCTION

The rate of urbanization has increased recently. at an unprecedented pace, resulting in the requirement for efficient and scalable infrastructure management solutions. Road, as critical components of urban infrastructure, play a crucial part in ensuring transportation safety, accessibility, and overall mobility. Traditional methods of road detection and maintenance, which frequently use manual inspections and conventional surveying techniques, are time-consuming, labor-intensive, as well as subject to human mistake. The growing demands of modern

cities necessitate innovative approaches that can automate and streamline these processes. Advances within the fields of machine learning and computer vision provide intriguing approaches to these challenges. By leveraging the power of Specifically, ConvNets (Convolutional Neural Networks) in deeplearning, It is feasible to construct automated frameworks capable of accurately detecting and analyzing road from urban imagery. This document provides an extensive Methodology for road detection using a combination of Open CV for image processing and TensorFlow In order to apply deep learning models, aimed at enhancing the precision, effectiveness, and scalability of urban infrastructure analysis. The suggested system is designed To solve the shortcomings of existing road detection methods by the use of an robust deep learning framework. The procedure starts with the normalization of input images to standardize the data, guaranteeing steady performance across various environments. A pre-trained CNN model is then utilized to anticipate the occurrence of road in the images. This model, trained on extensive datasets of urban scenes, is capable of distinguishing road from other urban elements with high precision. The output regarding the model is a probability map, which is subsequently thresholder to produce a binary mask highlighting the detected road. This mask is overlaid onto the original image, providing a clear and intuitive visualization of the road regions. The integration of TensorFlow Add Ons further enhances the model's performance by incorporating advanced data normalization techniques that play a crucial role in normalization strategies that significantly contribute to handling the variability in urban imagery. The suggested approach not only enhances the detection accuracy but also optimizes the computational efficiency, enabling real-time apps to use it such as autonomous navigation, urban planning, and infrastructure maintenance. Through extensive experimentation and validation, this study demonstrates the efficiency of the system in various scenarios, showcasing its capacity to transform urban infrastructure management through automated, data driven approaches.

2. LITERATURE REVIEW

Many scholars have investigated machine learning techniques to forecast and diagnose liver diseases by utilizing the Indian Liver Patient Dataset (ILPD) and other related clinical records. Early work by Rajeswari and Reena [1] applied Naive Bayes and Decision Tree classifiers to the ILPD, achieving close to 70% accuracy but showing poor specificity, which indicated that such basic models were unsuitable for dependable clinical use. Expanding on this work, Singh and Kaur [2] examined many classifiers, such as Support Vector Machines (SVM), Random forest, and k-Nearest Neighbors (k-NN). Their study's conclusions showed that while SVM achieved strong sensitivity, its specificity was not so strong, particularly when handling with the issue of imbalanced datasets—a frequent challenge in medical research. Kalra et al. [3] implemented Logistic Regression and Random Forest techniques for liver disease prediction. While Random Forest achieved around 85% accuracy, its lack of interpretability limited its acceptance in clinical decision-making, where transparency of the model's reasoning is crucial. To address generalization challenges, Choubey and Paul [4] proposed an ensemble-based model that combined Gradient Boosting and AdaBoost algorithms. Their findings demonstrated improved predictive performance when multiple classifiers were integrated, though the study did not apply modern explainability tools such as SHAP to make the predictions more transparent for healthcare professionals. More recently, Vaidya and Patil [5] applied the XGBoost algorithm on a preprocessed liver dataset and reported higher accuracy compared to conventional classifiers. However, the absence of real-time deployment or web-based implementation restricted the practical impact of their work, indicating that further research is still needed in developing deployable systems.

3. METHODOLOGY

3.1 System Implementation

The proposed Interpretable Liver Disease Prediction System integrates advanced machine learning techniques with a secure, web-based interface to assist in clinical decision-making. The system enables healthcare professionals or users to input patient parameters, obtain real-time predictions on liver disease risk, and manage previous prediction records in an organized and confidential manner.

The implementation consists of two major components:

3.1.1 Machine Learning Component

The predictive backbone of the system is developed using the **XGBoost (Extreme Gradient Boosting)** algorithm, well-known for its robustness and superior performance in classification tasks.

Before training, missing values were treated using a **hybrid K-Nearest Neighbour (KNN) imputation** method, ensuring data completeness without compromising data quality.

All numerical features were normalized to maintain uniform scale and enhance convergence during model training.

The trained model was evaluated through multiple performance metrics—**accuracy, sensitivity, specificity, precision, F1- score,** and **confusion matrix**—to ensure its reliability.To improve interpretability, **SHAP (SHapley Additive exPlanations)** analysis was employed to identify and rank the features contributing most to prediction outcomes.

3.1.2 Web Application Component

The finalized XGBoost model was integrated intoa **Django-based web application**, providing a simple yet secure interface for users. The platform allows users to **sign up, log in,** and securely interact with the prediction module. A **data entry form** facilitates input of clinical parameters, triggering the model to generate and display prediction results instantly.

The application includes **history management features**, allowing users to view or delete prior prediction records while ensuring **data privacy** through controlled access and secure storage mechanisms.

3.2 Dataset Preparation

3.2.1 Source of Dataset

The study utilized a clinical liver disorder dataset comprising 2,500 patient records. Each record includes demographic and biochemical attributes crucial for liver function assessment, such as Age, Gender, Total Bilirubin, Direct Bilirubin, Alkaline Phosphatase (ALP), Alanine Aminotransferase (ALT), Aspartate Aminotransferase (AST), Total Proteins, Albumin, and Albumin-to-Globulin Ratio.

3.2.2 Handling Missing Values

Approximately 10% of the dataset contained missing entries. Instead of discarding incomplete records, a hybrid KNN-based imputation approach was applied. This technique identifies the most similar records and estimates the missing values using feature averages, thereby preserving the natural relationships between variables and maintaining dataset integrity.

3.2.3 Feature Selection and Encoding

To ensure compatibility with the XGBoost algorithm, categorical attributes were converted into numeric format. The Gender feature was encoded as 0 for Female and 1 for Male. All clinical features were retained, as each had direct clinical

relevance to liver health prediction, avoiding unnecessary feature elimination.

3.2.4 Data Splitting

To validate the model's predictive performance, the dataset was partitioned into:

- **Training Set (80%)** – used for learning model parameters.
- **Testing Set (20%)** – used for performance evaluation on unseen data.

3.3 Pre-Processing Pipeline

The preprocessing phase transformed raw clinical data into a clean and consistent format suitable for machine learning. The following steps were systematically applied:

1. **Handling Missing Values** – A hybrid KNN-based imputation strategy filled missing entries using the mean of the closest feature vectors, preserving key statistical patterns.
2. **Feature Scaling / Normalization** – Numerical attributes with differing ranges (e.g., enzyme levels and protein concentrations) were normalized to ensure equal contribution during model training.
3. **Encoding Categorical Variables** – The categorical Gender variable was numerically encoded to make it compatible with the XGBoost model

3.4 Model Construction

The predictive framework was constructed using the **XGBoost** algorithm, a scalable and efficient implementation of gradient boosting that builds an ensemble of decision trees.

3.4.1 Working Principle of XGBoost

XGBoost constructs trees sequentially, where each new tree corrects the residual errors of the previous one.

- The process starts with a weak learner that makes initial predictions.
- Residuals (errors) are calculated as the difference between predicted and actual outcomes.
- Subsequent trees are trained to minimize these residuals using gradient descent.
- A **regularization term** is incorporated into the objective function to prevent overfitting and control model complexity.
- The final prediction is produced by aggregating the weighted outputs of all trees.

This ensemble approach allows XGBoost to achieve high predictive accuracy and handle missing data efficiently.

3.4.2 Model Configuration and Application

Key hyperparameters such as learning rate, maximum tree depth, and number of estimators were fine-tuned using GridSearchCV. The preprocessed dataset was supplied as input for training, and the resulting optimized model was serialized and saved as best_xgb_model.pkl for integration into the Django web framework.

3.5 Training Procedure

The training phase transformed pre-processed data into a functional predictive model:

1. **Input Data Feeding:** The 80% training data containing demographic and biochemical attributes was provided to the XGBoost classifier.
2. **Sequential Tree Building:** Each tree minimized the residuals from its predecessors, progressively enhancing prediction accuracy.
3. **Optimization via Gradient Descent:** The loss function was iteratively minimized by

updating model parameters in the direction of the steepest descent.

4. **Regularization:** Hyperparameters controlling model complexity were fine-tuned to prevent overfitting and enhance generalization.

Once training stabilized, the final model was saved for deployment in the web interface, enabling real-time predictions.

3.6 Performance Evaluation

The trained model was evaluated using the 20% testing subset to verify its predictive capability and reliability.

3.6.1 Evaluation Metrics

The following metrics were computed:

- **Accuracy:** Overall proportion of correct classifications.
- **Sensitivity (Recall):** Ability to correctly detect patients with liver disease.
- **Specificity:** Ability to correctly identify healthy individuals.
- **Precision:** Proportion of true positives among predicted positives.
- **F1-Score:** Harmonic mean of precision and recall, providing a balanced measure of model performance.

All metrics were derived from the confusion matrix, which summarizes true positives (TP), true negatives (TN), false positives (FP), and false negatives (FN).

3.6.2 Confusion Matrix and ROC-AUC Analysis

The confusion matrix helped assess classification balance between diseased and healthy cases. The Receiver Operating Characteristic (ROC) curve was plotted to evaluate discriminative power, and the Area Under Curve (AUC = 0.9296) confirmed the high accuracy and reliability of the XGBoost classifier.

3.6.3 SHAP-Based Feature Interpretability

To enhance transparency, SHAP analysis was employed to interpret the influence of each feature on model predictions. The analysis revealed that Direct Bilirubin, Total Bilirubin, Aspartate Aminotransferase (AST), and Alanine Aminotransferase (ALT) were the most influential predictors. Features such as Age, Albumin, and Albumin-to-Globulin Ratio also contributed significantly, aligning with medical understanding of liver health indicators. This interpretability ensures that the model's predictions are both clinically relevant and trustworthy.

4. MATERIALS AND METHODS

The proposed liver disease prediction framework was developed through a structured workflow consisting of dataset collection, preprocessing, model training, and deployment. The Indian Liver Patient Dataset (ILPD) [11], containing ten clinical.

attributes such as age, gender, bilirubin concentration, liver enzyme levels, protein values, and the albumin-to-globulin ratio, was utilized for this study.

To ensure data quality and consistency, preprocessing steps were applied, including the imputation of missing values, encoding of categorical variables, and feature normalization. For classification, the Extreme Gradient Boosting (XGBoost) algorithm [6] was selected due to its strong accuracy and robustness in handling imbalanced datasets.

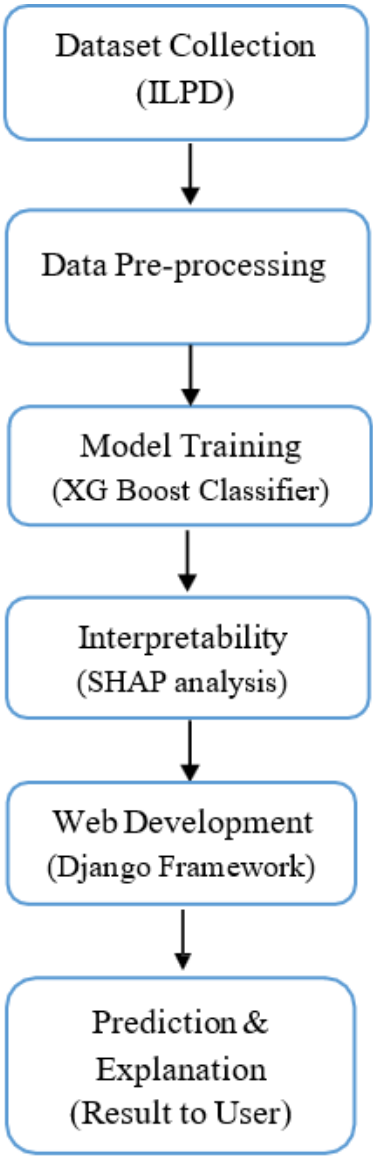

Figure: Flowchart of the System

5. RESULTS AND DISCUSSION

The performance of the proposed system was evaluated using the Indian Liver Patient Dataset (ILPD) [11]. After preprocessing steps such as feature normalization, missing value imputation, and categorical variable encoding, multiple machine learning algorithms were implemented and compared to assess their effectiveness in predicting liver disease. Model performance was evaluated through cross-validation and test accuracy, along with additional classification metrics including sensitivity, specificity, precision, recall, and F1-score.

Table 1 summarizes the comparative performance of the examined models. Among them, the Extreme Gradient Boosting (XGBoost) classifier achieved the highest test accuracy of 92.4%, marginally outperforming Random Forest (92.2%) and demonstrating clear superiority over Gradient Boosting (86.6%), Decision Tree (85.4%), and K- Nearest Neighbors (83.2%). In contrast, Logistic Regression, Support Vector Machine, and Naïve Bayes yielded considerably lower accuracy scores, underscoring their limited effectiveness on this dataset.

Table 1: Model Comparison (Cross-Validation and Test Accuracy)

Model	CV Accuracy (Mean)	Test Accuracy
XGBoost	0.9160	0.924
Random Forest	0.9170	0.922
Gradient Boosting	0.8570	0.866
Decision Tree	0.8565	0.854
K-Nearest Neighbors	0.8195	0.832
Logistic Regression	0.7120	0.712
Support Vector Machine	0.7140	0.708
Naive Bayes	0.5460	0.556

Table 2 presents the comparative results of sensitivity and specificity across the evaluated models. The XGBoost classifier attained a sensitivity of 95.48% and a specificity of 84.93%, indicating strong effectiveness in correctly identifying patients with liver disease while preserving balanced detection of healthy cases. Random Forest achieved a slightly higher sensitivity of 97.18%, but this improvement came at the cost of reduced specificity (80.14%), reflecting a greater tendency to misclassify healthy individuals. In contrast, Gradient Boosting and Decision Tree exhibited comparatively weaker performance on both metrics. Models such as Support Vector Machine and Logistic Regression,

which demonstrated particularly low specificity, were deemed less suitable for clinical application, as reliable discrimination between diseased and non-diseased cases is essential for medical decision-making.

Table 2: Model Comparison with Sensitivity and Specificity

Model	Test Accuracy	Sensitivity	Specificity
XGBoost	0.924	0.9548	0.8493
Random Forest	0.922	0.9718	0.8014
Gradient Boosting	0.866	0.9435	0.6781
Decision Tree	0.854	0.8955	0.7534
K-Nearest Neighbors	0.832	0.8870	0.6986
Logistic Regression	0.712	0.9294	0.1849
Support Vector Machine	0.708	1.0000	0.0000
Naïve Bayes	0.556	0.4124	0.9041

Further optimization of the XGBoost model through hyperparameter tuning resulted in an improved classification accuracy of 93.4%, as reported in the classification summary. For the liver disease class, the model achieved an F1-score of 0.95, reflecting a strong balance between precision and recall. Such equilibrium is particularly critical in clinical contexts, where minimizing false negatives—patients incorrectly classified as healthy—is essential to ensure timely diagnosis and safeguard patient outcomes.

Classification Report (Tuned XGBoost Model)

- Accuracy: **0.934**
- Precision (Liver Disease = 1): **0.94**
- Recall (Liver Disease = 1): **0.97**
- F1-score (Liver Disease = 1): **0.95**
- Precision (Non-Liver = 0): **0.91**
- Recall (Non-Liver = 0): **0.86**
- F1-score (Non-Liver = 0): **0.88**

The incorporation of SHapley Additive Explanations (SHAP) provided model interpretability by quantifying the relative contribution of each feature to predictive outcomes. The global SHAP analysis revealed that Alkaline Phosphatase, Direct Bilirubin, Total Bilirubin, and the Albumin-to-Globulin Ratio were the most influential variables in distinguishing patients with liver disease. In addition, local SHAP explanations delivered patient-specific insights, allowing clinicians to understand the rationale behind individual predictions and thereby facilitating more informed medical decision-making.

In contrast to earlier studies that focused predominantly on maximizing predictive accuracy [1–5], the proposed system demonstrates both high classification performance and interpretability. This combination strengthens clinical reliability, enhances practitioner confidence, and establishes the system as a viable decision-support tool for liver disease diagnosis.

6. CONCLUSIONS

This endeavour effectively created a robust and user-friendly system for the early identification of liver disease using a machine learning-based approach. This project uses the XGBoost method to create a robust model capable of forecasting liver illness with high precision using patient details and test values. The model was trained on a cleaned dataset, where values that were missing were filled in and the data was adjusted to improve performance. The model was then connected to a Django web application. The web app has a simple front page where users can enter medical details, get prediction results, check past records, and manage their data easily. SHAP is also included to make the results clearer. It shows how each patient detail affects the prediction, which helps doctors and users understand the outcome and trust the system more. Overall, the system meets its goals of being accurate, easy to

use, secure, and scalable. It gives fast and accessible liver disease screening,

REFERENCES

[1] S. M. Metev and V. P. Veiko, *Laser Assisted Microtechnology*, 2nd ed., R. M. Osgood, Jr., Ed. Berlin, Germany: Springer-Verlag, 1998.
[2] J. Breckling, Ed., *The Analysis of Directional Time Series: Applications to Wind Speed and Direction*, ser. Lecture Notes in Statistics. Berlin, Germany: Springer, 1989, vol. 61.
[3] S. Zhang, C. Zhu, J. K. O. Sin, and P. K. T. Mok, "A novel ultrathin elevated channel low-temperature poly-Si TFT," *IEEE Electron Device Lett.*, vol. 20, pp. 569–571, Nov. 1999.
[4] M. Wegmuller, J. P. von der Weid, P. Oberson, and N. Gisin, "High resolution fiber distributed measurements with coherent OFDR," in *Proc. ECOC'00*, 2000, paper 11.3.4, p. 109.
[5] R. E. Sorace, V. S. Reinhardt, and S. A. Vaughn, "High-speed digital-to-RF converter," U.S. Patent 5 668 842, Sept. 16, 1997.
[6] (2002) The IEEE website. [Online]. Available: http://www.ieee.org/
[7] M. Shell. (2002) IEEEtran homepage on CTAN. [Online]. Available: http://www.ctan.org/tex-archive/macros/latex/contrib/supported/IEEEtran/
[8] *FLEXChip Signal Processor (MC68175/D)*, Motorola, 1996.
[9] "PDCA12-70 data sheet," Opto Speed SA, Mezzovico, Switzerland.
[10] A. Karnik, "Performance of TCP congestion control with rate feedback: TCP/ABR and rate adaptive TCP/IP," M. Eng. thesis, Indian Institute of Science, Bangalore, India, Jan. 1999.
[11] J. Padhye, V. Firoiu, and D. Towsley, "A stochastic model of TCP Reno congestion avoidance and control," Univ. of Massachusetts, Amherst, MA, CMPSCI Tech. Rep. 99-02, 1999.
[12] *Wireless LAN Medium Access Control (MAC) and Physical Layer (PHY) Specification*, IEEE Std. 802.11, 1997.

