NATIONAL INSTITUTE OF TECHNOLOGY KARNATAKA SURATHKAL

EC346 - FOUNDATIONS OF MACHINE LEARNING

REPORT ON TERM PROJECT
(To classify the given images as bleeding or non-bleeding using feature selection/extraction and ensemble methods of machine learning)

Submitted By
211EC240 Ravipati Varshith
211EC249 Sharath Balaji T

Submitted To
Dr. Aparna P.
Assistant Professor Dept. of Electronics and Communication Engineering

DEPARTMENT OF ELECTRONICS AND COMMUNICATION ENGINEERING
NATIONAL INSTITUTE OF TECHNOLOGY KARNATAKA,
SURATHKAL

NOVEMBER 2023

❑ INTRODUCTION:

Medical image classification plays a crucial role in modern healthcare by aiding in the early detection and diagnosis of various conditions. One application involves classifying medical images into categories such as "bleeding" and "non-bleeding". The objective of this report is to explore and implement machine learning techniques, specifically focusing on feature selection/extraction and ensemble methods, to classify given medical images as either "bleeding" or "non-bleeding." The classification task holds significant importance in medical imaging, where early detection of bleeding can lead to timely intervention and improved patient outcomes.

Machine learning, particularly feature selection/extraction and ensemble methods, provides an avenue for more accurate and robust classification. By leveraging advanced algorithms, these methods can discern subtle patterns and relationships in medical images, contributing to the early and accurate identification of bleeding conditions.

❑ OBJECTIVE:

To classify the given images as bleeding or non-bleeding using feature selection/extraction and ensemble methods of machine learning.

Image Preprocessing: Investigate preprocessing techniques to enhance image quality, address noise, and normalize pixel values.

Feature Selection/Extraction: Explore methods for extracting relevant features from medical images, such as texture analysis, shape descriptors, feature extraction.

Ensemble Methods: Implement ensemble methods like Random Forest, Bagging, or Boosting to enhance classification performance and mitigate overfitting.

Evaluation Metrics: Employ standard evaluation metrics such as accuracy, precision, recall, and F1 score to assess the performance of the models.

Interpretability and Explainability: Investigate the interpretability of selected features and the ensemble models to enhance the transparency of the classification process.

Clinical Relevance: Discuss the clinical implications of accurate bleeding detection, emphasizing the potential impact on patient care and treatment decisions**.**

❑ BLOCK DIAGRAM:

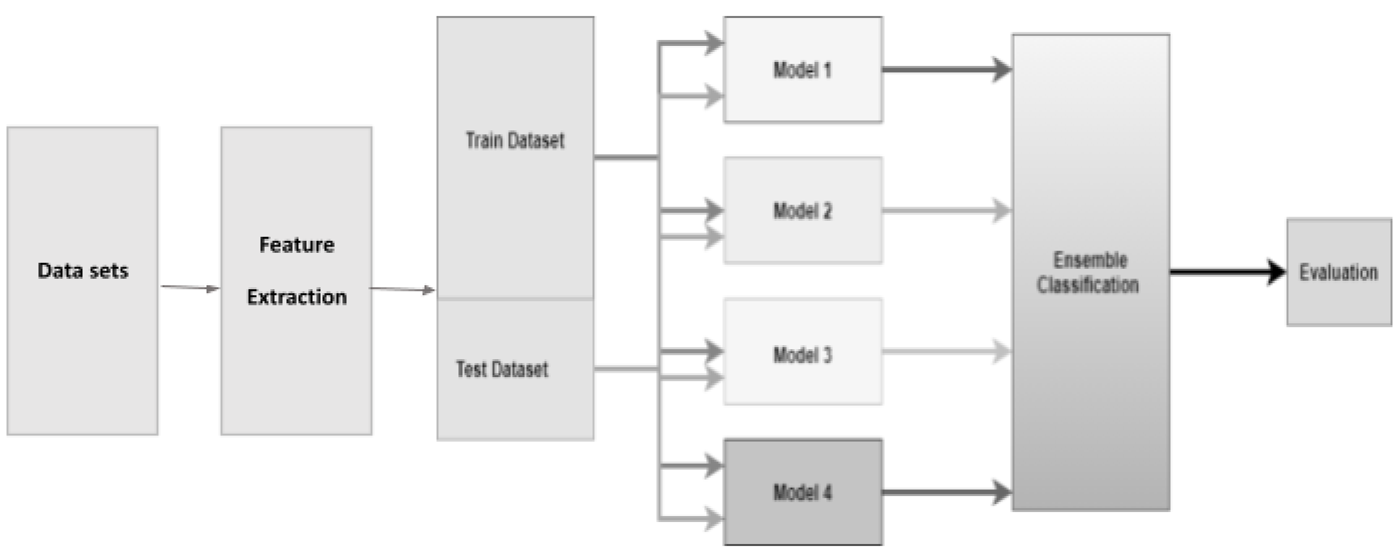

Fig1: Block Diagram of the entire model implemented in this Project.

❑ DATASET:

1. Dataset Overview:

The dataset employed in this project originates from a curated collection of medical images obtained from it. The dataset's main objective is to facilitate the accurate classification of images as either exhibiting signs of bleeding or non-bleeding.

2. Data Size and Distribution:

The dataset comprises a total of [specify the number] images, with an equal distribution between bleeding and non-bleeding instances to prevent class imbalance issues during model training. This ensures a robust model capable of handling various bleeding scenarios.

3. Data Loading Process:

To facilitate the training and evaluation of machine learning models for the classification of medical images, a meticulous data loading process was implemented. The following steps outline the procedure adopted.

4. Importing Libraries:

To leverage the capabilities of TensorFlow and its image preprocessing tools, the necessary libraries were imported in the Python script. (In our case it is os.path)

5. Defining Data Paths:

The dataset, consisting of medical images, was organized into separate directories for training and testing.

❑ DATA PREPROCESSING STEPS AND FEATURE EXTRACTION:

Feature extraction involves transforming the raw pixel values of medical images into a suitable format for training a machine learning model. Here is a guide on how we can perform feature extraction using resizing, normalization, and label encoding:

1. **Image Resizing**: Resizing ensures that all images are in a consistent format, which is essential for creating a uniform input for the machine learning model.

2**. Normalization:** Normalization is performed to scale pixel values to a standard range, typically [0, 1]. This ensures that the model learns more effectively.

3. Label Encoding: Label encoding is applied to convert categorical labels (bleeding, non-bleeding) into numerical format.

4. Flattening: The flattening process is commonly used as part of feature extraction, particularly when dealing with convolutional neural networks (CNNs) or other models that expect one-dimensional input. In the context of image data, this often means converting a 2D array (such as a grayscale image) or a 3D array (such as a color image) into a 1D array.

These preprocessing steps collectively prepare the dataset for model training, ensuring uniformity and compatibility with machine learning algorithms. The exploration and manipulation of the dataset laid the foundation for subsequent stages in the bleeding detection project.

❑ DATA SPLITTING INTO TRAIN-TEST:

The dataset is typically divided into two subsets: the training set and the testing set.

The training set is used to train the machine learning model, while the testing set is used to evaluate the model's performance.

The typical split ratio is 70-30 for training and testing, respectively. However, the split ratio may vary based on the size of your dataset and specific requirements.

❑ BASE MODELS:

For a project focused on classifying medical images as bleeding or non-bleeding, several machine learning models can be considered as base models. The choice of base models depends on the characteristics of your dataset, the complexity of the task, and the computational resources available. For classification tasks, models like Decision Trees, Random Forests, Support Vector Machines, and Gradient Boosted Trees are commonly used.

1. Random Forest Classifier:

Random Forest is an ensemble learning method that can be used for both classification and regression tasks. It is a powerful and versatile machine learning algorithm that operates by constructing a multitude of decision trees during training and outputting the class that is the mode of the classes (classification) or mean prediction (regression) of the individual trees. Here is a breakdown of how a Random Forest classifier works:

Decision Trees: At the core of a Random Forest are decision trees. A decision tree is a flowchart-like structure where each internal node represents a test on an attribute, each branch represents the outcome of the test, and each leaf node represents a class label (in the case of classification) or a numerical value (in the case of regression).

Ensemble Learning: Random Forest builds multiple decision trees during training. The "random" part comes from two key sources of randomness: a. Random Sampling of Data: Each tree is trained on a random subset of the training data, selected with replacement (this is known as bootstrapping). Random Subset of Features: At each node in the decision tree, only a random subset of features is considered for making a split.

Applications in bleeding: Random Forests can be employed to classify medical images into bleeding and non-bleeding categories. The algorithm's ability to handle complex and high-dimensional data makes it suitable for image classification tasks**.** It can help identify which image features (such as pixel intensities, textures, or other image characteristics) contribute the most to the classification of bleeding or non-bleeding cases. The ensemble nature of Random Forests helps mitigate the risk of overfitting and enhances generalization to new, unseen data. This is crucial in medical imaging where the dataset may be limited, and the goal is to make accurate predictions on new patient cases.

Parameters Tunning: Random Forests have several hyperparameters that can be tuned to optimize performance. This includes the number of trees in the forest, the depth of each tree, and the number of features considered at each split. (**WE HAVE GIVEN n_estimators=50, randomstate=42**)

2. AdaBoost Classifier:

AdaBoost, short for Adaptive Boosting, is another ensemble learning method widely used for classification tasks. Like Random Forest, it is versatile and powerful, aiming to improve the performance of weak learners by combining them into a strong learner. Here is a breakdown of how an AdaBoost classifier works:

Weak Learners (Base Classifiers): At the core of AdaBoost are weak learners, typically simple models like decision trees with limited depth. These weak learners perform slightly better than random chance.

Weighted Data Points: During training, each data point is assigned a weight, and initially, all weights are set equally. The algorithm focuses on the misclassified data points by assigning higher weights to them in subsequent rounds.

Sequential Training: AdaBoost trains weak learners sequentially, and at each iteration, it gives more emphasis to the misclassified samples from the previous round. This adaptability allows the algorithm to concentrate on the "hard-to-classify" instances.

Combining Weak Learners: After each iteration, the weak learner's performance is evaluated, and a weight is assigned based on its accuracy. The combined model gives more weight to the accurate classifiers, effectively creating a strong learner.

Final Classification: The final classification is a weighted sum of the individual weak learners' predictions. Weak learners that perform well on the training data receive higher weights in the final decision.

Parameter Tunning: AdaBoost also involves tuning hyperparameters, such as the number of weak learners, to optimize its performance for a specific dataset. (**WE HAVE GIVEN n_estimators=25, randomstate=42**)

3. Bagging Classifier:

Bagging, short for Bootstrap Aggregating, is an ensemble machine learning technique primarily used for classification but can also be applied to regression tasks. It operates by constructing multiple base models, often decision trees, on random subsets of the training data and then aggregating their predictions.

Ensemble Learning: Like Random Forest, Bagging works on the principle of ensemble learning. It builds multiple base models by randomly sampling the training dataset with replacement.

Bootstrap Aggregation: Each base model (e.g., decision tree) is trained on a different subset of the training data, randomly selected with replacement from the original dataset. This process creates diverse subsets for each base model.

Decision Trees or Base Models: At its core, Bagging uses a collection of base models, often decision trees, to make predictions. Each base model is trained independently on a subset of the data.

Feature Importance and Generalization: Bagging helps in identifying relevant image features contributing to bleeding or non-bleeding categorization. Its ensemble approach aids in preventing overfitting and improving generalization, crucial in scenarios with limited medical image datasets. (**WE HAVE GIVEN n_estimators=25, randomstate=42**)

Parameters Tuning: Bagging classifiers offer various hyperparameters to fine-tune for optimal performance. Parameters such as the number of base models, their complexity, and the subset size for training can be adjusted to enhance the model's accuracy and generalization.

Both Random Forest and Bagging classifiers belong to the ensemble learning family, leveraging multiple models to achieve robust predictions and are particularly useful in handling complex data like medical images.

❑ Ensemble Model:

An ensemble model combines the predictions of multiple individual classifiers to achieve a more robust and accurate result than any individual model. The Voting Classifier, a popular ensemble method, facilitates this combination by aggregating the predictions through a majority vote or averaging probabilities.

● **Diversity of Models**: The strength of an ensemble model lies in the diversity of its constituent classifiers. Each individual classifier may excel in various aspects of the data or capture distinct patterns.

● **Voting Strategies:** The Voting Classifier supports different voting strategies, including 'hard' and 'soft.' In 'hard' voting, the majority class is selected, while in 'soft' voting, the class with the highest average probability is chosen.

Applications of Ensemble method: In our bleeding detection project, the ensemble model is created using the Voting Classifier, incorporating predictions from Random Forest, AdaBoost and Bagging. This diverse set of classifiers contributes to a more reliable and accurate bleeding detection algorithm. By leveraging the strengths of different models, the ensemble approach enhances the overall performance, particularly in scenarios where individual models may exhibit limitations.

Parameter Tunning: Estimators: The individual classifiers contributing to the ensemble, including Random Forest, AdaBoost, Bagging. **(estimators= [('random_forest', random_forest_model), ('adaboost', adaboost_model), ('bagging', bagging_model)], voting='soft')**

Voting: In this project, 'soft' voting is utilized. The ensemble model in our project encapsulates the collective intelligence of diverse classifiers, resulting in a more resilient and adaptable bleeding detection algorithm. Through the combination of individual strengths, the ensemble model aims to mitigate weaknesses and provide a reliable solution for accurate bleeding identification in medical images.

❑ RESULTS AND DISCUSSIONS:

Results and discussions based on the confusion matrix and ROC-AUC curve are crucial in evaluating and interpreting the performance of a classification model.

Confusion Matrix Analysis:

A confusion matrix is a table that allows the visualization of a classifier's performance by comparing actual and predicted values. It comprises four terms:

True Positives (TP): The number of instances correctly predicted as the positive class.

True Negatives (TN): The number of instances correctly predicted as the negative class.

False Positives (FP): The number of instances incorrectly predicted as the positive class.

False Negatives (FN): The number of instances incorrectly predicted as the negative class.

Discussions based on Confusion Matrix; the calculation of evaluation matrices are given below:

Accuracy: The overall accuracy of the model is calculated as the ratio of correct predictions (TP + TN) to the total number of predictions. A high accuracy indicates the model's effectiveness in making correct predictions across both classes.

Accuracy - (TP+TN) / (TP+TN+PF+FN)

Precision and Recall: Precision measures the accuracy of positive predictions, while recall (sensitivity) measures the model's ability to capture positive instances. A balance between precision and recall is necessary depending on the specific application.

Recall - TP/ (TP+FN)

Precision - TP/ (TP+FP)

F1-Score: The F1-score combines precision and recall into a single metric, useful when there is an uneven class distribution.

F1 score = (2 * Precision * Recall) / (Precision + Recall) [In F-beta, beta=1]

Below table shows accuracy, precision, recall, F1 score of each ensemble model.

MODEL NAME	ACCURACY	PRECISION	RECALL	F1 SCORE
Random Forest	0.97583	0.96517	0.98728	0.97610
Adaboost Classifier	0.97710	0.96992	0.98473	0.97727
Bagging Classifier	0.97710	0.96526	0.98982	0.97739
Voting Classifier (Final Ensemble)	0.98601	0.97750	0.99491	0.98613

*Fig2: Table for **evaluation metrics** of each model.*

ROC-AUC Curve Analysis:

Receiver Operating Characteristic (ROC) curve visualizes the classifier's performance across various threshold settings. It plots the True Positive Rate (TPR) against the False Positive Rate (FPR) at various threshold values.

$$TPR = TP/(TP+FN)$$

$$FPR = FP/(FP+TN)$$

AUC (Area Under the Curve): AUC measures the area under the ROC curve. A higher AUC indicates better discrimination between classes. An AUC of 1 represents a perfect model, while an AUC of 0.5 suggests a model that performs no better than random.

Threshold Analysis: ROC curves help understand the trade-off between sensitivity and specificity. It assists in choosing the optimal threshold based on the specific requirements of the problem.

Comparative Analysis: Comparing multiple models' ROC-AUC curves can assist in determining which model performs better in distinguishing between classes.

PLOTS OF CONFUSION MATRIX AND ROC-AUC CURVE:

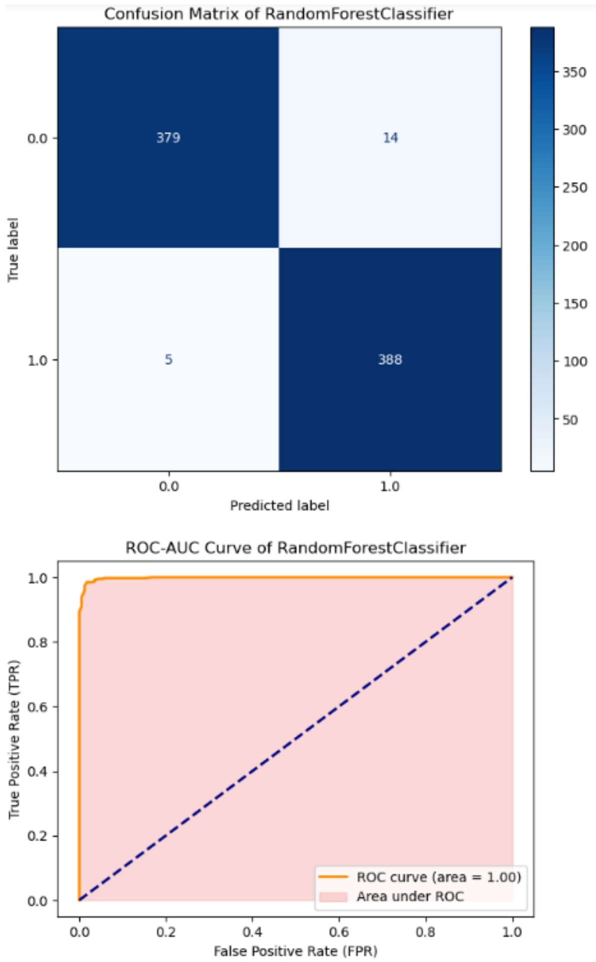

Fig3: Confusion matrix and ROC-AUC curve for Random Forest Classifier

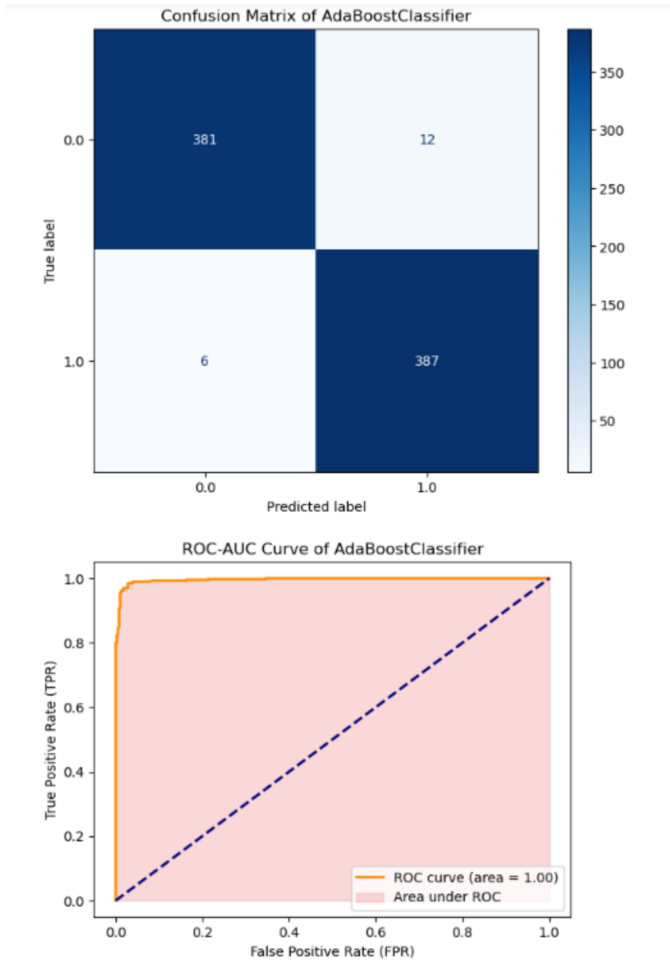

Fig4: Confusion matrix and ROC-AUC curve for Adaboost Classifier

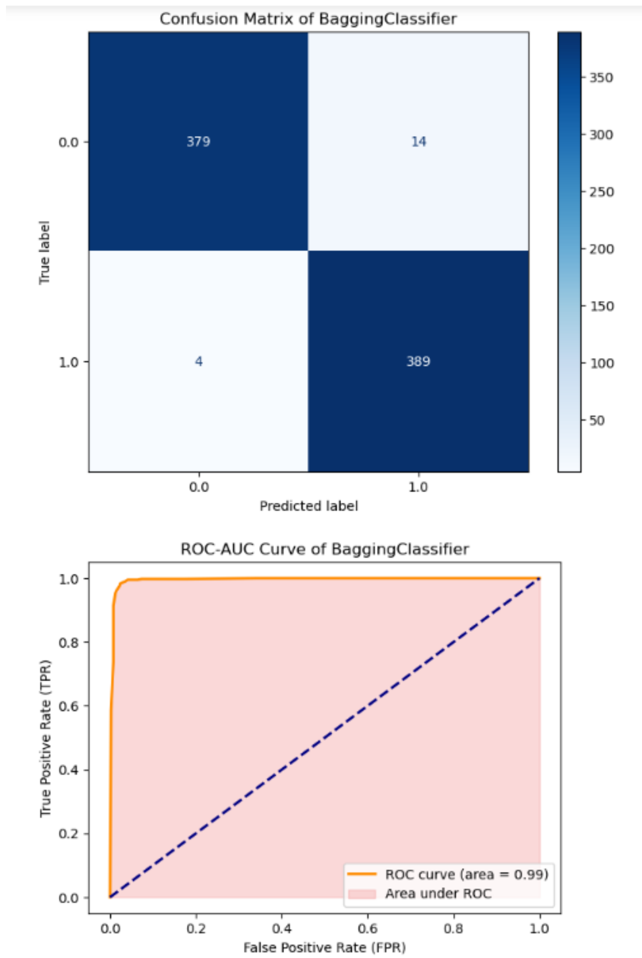

Fig5: Confusion matrix and ROC-AUC curve for Bagging Classifier

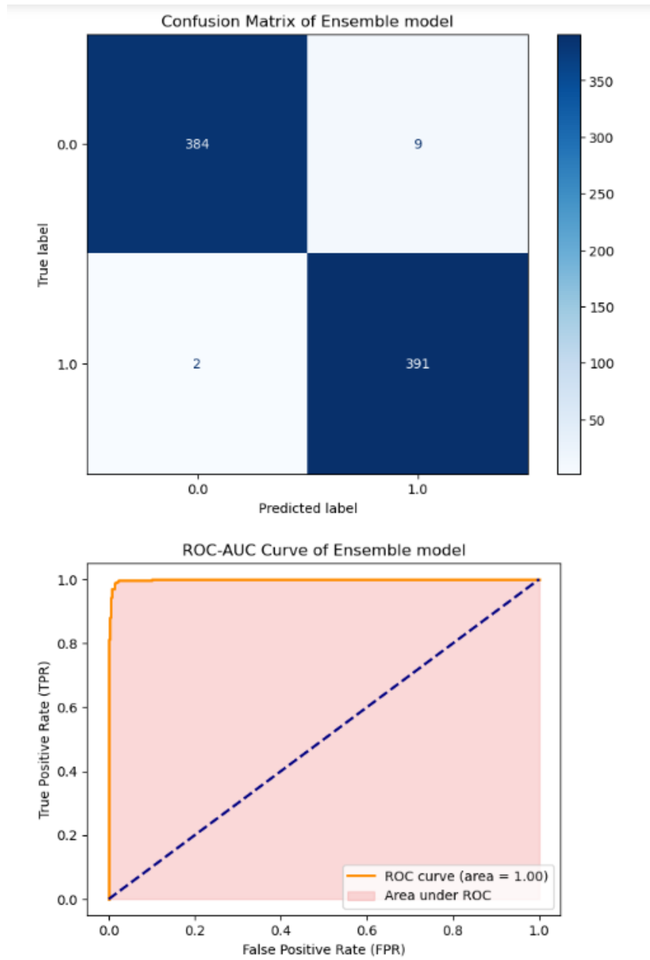

Fig6: Confusion matrix and ROC-AUC curve for Voting Classifier (Final ensemble model)

❑ CONCLUSION:

In conclusion, the application of ensemble methods, including Random Forest and AdaBoost, in bleeding detection within medical imaging presents a promising avenue for enhancing accuracy, robustness, and real-time decision support in healthcare. Leveraging the strengths of ensemble learning, these methods address the challenges inherent in complex and high-dimensional medical data, offering valuable insights and predictive capabilities.

Each individual classifier demonstrated strengths and weaknesses in bleeding detection, offering unique insights into their applicability. The adaptability of AdaBoost, with its emphasis on misclassified instances and sequential learning, makes it well-suited for detecting bleeding in medical images, especially in scenarios where early identification is crucial. Similarly, Random Forest, with its ensemble of decision trees and features like bootstrapping and random feature subset selection, provides a powerful tool for medical image analysis.

Ensemble Model Synergies: The ensemble model, combining predictions from diverse classifiers, presented a synergistic approach to bleeding detection. The combination of different learning strategies contributed to improved accuracy and robustness.

Performance Matrices: Accuracy, Precision, Recall, and F1-score were systematically evaluated for each model, providing a comprehensive understanding of their effectiveness in bleeding detection.

A comparative analysis facilitated the identification of scenarios where specific models excelled or faced challenges. Insights from this analysis guide model selection based on project requirements