A Major Project Report on

PANCREATIC TUMOR DETECTION USING ML

Master of Computer Applications

Submitted By:

Gaurav Singh Rawat,

Department of Computer Science,

University of Delhi

Examination Roll No: 21234757020

YEAR: 2025

Submitted To:

Dr. Mantosh Biswas

Associate Professor

UNDERTAKING

I declare that the work presented in this report titled “Pancreatic Tumor Detection” submitted to the Department of Computer Science, University of Delhi, New Delhi, for the award of the Master of Computer Application degree, is our original work. I have not plagiarized or submitted the same in part or full to any university or institution for the award of any degree or diploma. In case this undertaking is found incorrect, I accept that our degree may be unconditionally withdrawn.

Signature of Candidate

Date: 16/07/2025

Gaurav Singh Rawat

(Exam Roll. No. 21234757020)

Place: Delhi

DEPARTMENT OF COMPUTER SCIENCE

UNIVERSITY OF DELHI, DELHI – 110 007 (INDIA)

CERTIFICATE

I hereby certify that the work which is being presented in this MCA Internship Project report entitled “Pancreatic Tumor Detection”, in partial fulfilment of the requirements for the award of the Master of Computer Application is an authentic record of my own work carried out during a period from January, 2025 to May, 2025 under the supervision of mentor Dr. Mantosh Biswas, Associate Professor, Department of Computer Science, University of Delhi, The matter presented in this project report has not been submitted for the award of any other degree elsewhere.

Date: 16/07/2025 **Signature of Candidate**

Gaurav Singh Rawat

Place: Delhi (Exam Roll No. 21234757020)

This is to certify that the above statement made by the candidates is correct to the best of my knowledge.

Signature of Faculty Mentor

Assistant Professor

Acknowledgement

I would like to take this opportunity to sincerely thank everyone who supported and guided me throughout the completion of this project.

My deepest thanks go to **Dr. Mantosh Biswas**, Assistant Professor, Department of Computer Science, University of Delhi, for his valuable mentorship, clear direction, and continuous encouragement throughout this work. His support helped me stay focused and motivated from start to finish.

I also extend my gratitude to **Mr. Rajesh Kumar Yadav**, Ph.D. Scholar, Department of Computer Science, for his technical guidance, helpful insights, and readiness to assist whenever needed. His suggestions greatly improved the quality and clarity of the project.

I would like to place on record my deep sense of gratitude to **Professor Neelima Gupta, Head of Department** of the Department of Computer Science, University of Delhi for providing this opportunity.

Lastly, I would like to thank my **family and friends** for their constant encouragement, patience, and moral support. Their belief in me helped me remain determined and confident during all phases of this project.

Gaurav Singh Rawat

21234757020

Index

Chapter 1: Introduction

1. What Is Pancreatic Tumor?
2. What are CT Scan Images?
3. Why is Early Detection Important?
4. What is Machine Learning?
5. What is an SVM (Support Vector Machine)?
6. Why Did We Use SVM for This Project?
7. What Problem Does This Project Solve?
8. What is the Goal of This Project?

Chapter 2: Model Design / Proposed Solution

1. Preparing the Dataset for Training
2. Tools and Technologies Used
3. Enhancing CT Images with Image Processing
4. Algorithms and Techniques Used in This Project
5. Metrics Used to Measure Performance
6. Step-by-Step Process Overview (Flowchart)
7. Detailed Explanation of Each Step

Chapter 3: Project Requirements and Dataset Description

1. Development Environment
2. Data Required for the Project
3. Knowledge Requirements
4. Dataset Overview

Chapter 4: Implementation and Explanation

1. Programming Language and Libraries
2. Image Preprocessing

3. Feature Extraction
4. Label Encoding
5. Feature Standardization
6. Reducing Features with PCA
7. Splitting the Dataset
8. Building the SVM Classifier
9. Improving with Hyperparameter Tuning

Chapter 5: Simulation of the Model (Results/Output)

1. Dataset Visualization and Statistics
2. Example Pancreatic Tumor Images
3. Model Predictions (Output)
4. Accuracy and Evaluation Metrics
5. ROC Curve and Model Reliability

Chapter 6: Discussion

1. Learning Experience and Growth
2. Technical Skill Development
3. Domain Knowledge
4. Problem Solving and Critical Thinking

Chapter 7: Conclusion and Future Work

1. Conclusion
2. Challenges Faced
3. Future Scope and Improvements
4. Final Summary

Chapter 8: References

1. Books, Research Papers, and Articles

Table of Abbreviations

Abbreviation	Full Form
CT	Computed Tomography
SVM	Support Vector Machine
RBF	Radial Basis Function
GLCM	Gray Level Co-occurrence Matrix
PCA	Principal Component Analysis
SMOTE	Synthetic Minority Over-sampling Technique
FCC	Freeman Chain Code
MCC	Matthews Correlation Coefficient
AUC	Area Under the Curve
ROI	Region of Interest
CPU	Central Processing Unit
GPU	Graphics Processing Unit
AI	Artificial Intelligence
ML	Machine Learning
DL	Deep Learning
CV	Cross Validation
XGBoost	Extreme Gradient Boosting

Abbreviation	**Full Form**
CLAHE	Contrast Limited Adaptive Histogram Equalization
VS Code	Visual Studio Code
OpenCV	Open Source Computer Vision Library
.npy	NumPy File Format (used for saving NumPy arrays)
sklearn	scikit-learn (Python ML library)
pywt	PyWavelets (Python Wavelet Transform library)
joblib	Job Library (for saving/loading Python objects like models and scalers)

Chapter 1. Introduction

1.1 What Is Pancreatic Tumor?

A pancreatic tumor is a lump or abnormal growth that forms inside the pancreas. The pancreas is an important organ that helps in digestion and in controlling blood sugar. Some tumors are harmless (called benign), but others can be dangerous (called malignant or cancerous). Pancreatic cancer is one of the most serious types of cancer because it grows quickly and is often not found until it is too late. That is why it is very important to find it as early as possible.

1.2 What are CT Scan Images?

CT stands for Computed Tomography. A CT scan is a special type of X-ray machine that takes many pictures of the inside of the body. These pictures are then combined to create detailed images of internal organs. In this project, we use CT scan images of the pancreas to check if a tumor is present. These images are in black and white (grayscale) and show the shapes and structures inside the body.

1.3 Why is Early Detection Important?

Pancreatic cancer is usually not found until it is in the final stage. At that point, treatment becomes very difficult and the chances of survival are low. If we can detect the tumor early, doctors can begin treatment sooner, and the patient has a better chance of recovery. But manually checking hundreds or thousands of CT images takes a lot of time and effort. So, using a computer to do this job can help doctors save time and avoid mistakes.

1.4 What is Machine Learning?

Machine Learning is a way to teach computers to make decisions by learning from data. Just like humans learn by looking at examples, computers can learn by looking at many images that are already labelled (like “tumor” or “normal”). After learning from these examples, the machine can look at new images and decide whether they show a tumor or not. In this project, we use machine learning to build a system that can identify tumors in CT scan images.

1.5 What is an SVM (Support Vector Machine)?

SVM is a supervised machine learning algorithm used for classification. SVM tries to find the best possible boundary (hyperplane) that separates classes with the maximum margin. In your tumor detection project, it's used to classify images as either normal or tumor.

If the data is simple and clearly separable, SVM can draw a straight line between the two groups. But medical image data is often complex and not easy to separate with a straight line. This is where we use kernel function in SVM to handle such cases. A kernel helps the SVM convert the data into a higher-dimensional space where it becomes easier to draw a clear boundary between the two classes. For this project, we use the RBF (Radial Basis Function) kernel, which is very good at handling curved and complex patterns in image data.

1.6 Why Did We Use SVM for This Project?

We chose SVM for this project because:

- It works well with small to medium-sized datasets.
- It gives very good results even when the data is complex.
- It can separate images into tumor and normal groups very clearly.
- It does not get confused easily, even if there are some errors in the data.
- It works well when combined with other techniques like PCA (to reduce extra information) and SMOTE (to balance the data).

1.7 What Problem Does This Project Solve?

In many hospitals, doctors look at CT images one by one to check for tumors. This takes a lot of time and effort. Sometimes, small tumors can be missed. This project creates an automatic system that can quickly and correctly detect pancreatic tumors from CT images. It helps doctors by giving them fast and accurate results so that they can make better decisions for patients.

1.8 What is the Goal of This Project?

The main goal of this project is to build a smart computer program that can:

- Read and process CT images
- Improve the image quality using image processing
- Pick out important information (features) from each image
- Use PCA to reduce the number of features and keep only the useful ones
- Use SMOTE to balance the number of tumor and non-tumor samples
- Train a model using SVM and XGBoost to classify the images
- Show results like accuracy, sensitivity, specificity, and more to prove how well the system works

Chapter 2: Model Design / Proposed Solution

2.1 Preparing the Dataset for Training

To train a machine learning model, we first need to collect and organize the data. In this project, we use CT scan images of the pancreas. These images are divided into two categories:

- **Normal images** (no tumor)
- **Tumor images** (pancreatic tumor present)

Each image is labelled either 0 (normal) or 1 (tumor). These labels help the model learn what features belong to each class. We load the images from two separate folders and convert them into grayscale. This reduces the data complexity and makes processing faster.

After loading, the images are resized to a uniform size (128x128) so that the model receives inputs of the same shape. This step is important for consistency.

2.2 Tools and Technologies Used

We use the following technologies in this project:

- **Python** – The main programming language.
- **OpenCV (cv2)** – For image loading, resizing, enhancement, and processing.
- **NumPy** – For handling arrays and numerical operations.
- **scikit-learn (sklearn)** – For machine learning models and evaluation metrics.
- **XGBoost** – A powerful gradient boosting model used for classification.
- **SMOTE** – To handle class imbalance by generating synthetic samples.
- **PyWavelets (pywt)** – For extracting wavelet-based features from images.
- **Joblib** – To save models and important objects like PCA and scalers.

2.3 Enhancing CT Images with Image Processing

Medical images often contain noise and lack contrast. To improve image quality and extract better features, we use:

- **CLAHE (Contrast Limited Adaptive Histogram Equalization)**: Enhances local contrast of the image.
- **Morphological Operations**: These include denoising techniques like closing (to fill gaps or holes).
- **Binary Thresholding**: Converts grayscale images into black and white to help in detecting shapes and boundaries.
- **Contour Detection**: Identifies the edges of shapes (like tumors) within the image.

These techniques help highlight the parts of the image that are important for identifying tumors.

2.4 Algorithms and Techniques Used in This Project

We use several machine learning techniques:

- **Feature Extraction**:
    - **Shape Features**: Area, perimeter, and chain code length from the tumor boundary.
    - **Texture Features**: GLCM properties like contrast, homogeneity, and energy.
    - **Wavelet Features**: Extracted using wavelet transform to capture image details at different frequencies.
- **Feature Reduction**:
    - **PCA (Principal Component Analysis)**: Used to reduce the number of features while keeping the most important information.
- **Balancing Data**:

    - **SMOTE**: Creates synthetic tumor samples to balance the number of normal and tumor images.

- **Model Building**:
    - **SVM with RBF kernel**: Helps detect complex patterns in the data.
    - **XGBoost**: An advanced tree-based model that improves performance.
    - **BaggingClassifier**: Trains multiple SVMs to make the model more stable.
    - **VotingClassifier**: Combines the results of SVM and XGBoost to make the final decision.

2.5 Metrics Used to Measure Performance

To understand how well our model is performing, we use the following evaluation metrics:

- **Accuracy** – How many images were correctly classified.
- **Sensitivity (Recall)** – How well the model detects tumor images.
- **Specificity** – How well the model avoids false alarms on normal images.
- **Precision** – How many of the predicted tumor images were actually tumors.
- **F1 Score** – A balanced score of precision and recall.
- **MCC (Matthews Correlation Coefficient)** – A strong performance measure for binary classification.
- **AUC (Area Under the Curve)** – Shows how well the model can separate tumor from normal across different thresholds.

These metrics help doctors and researchers understand whether the system is safe and reliable to use.

2.6 Step-by-Step Process Overview (Flowchart)

Below is a simplified flow of the system from start to end:

CT Image Input

↓

Image Preprocessing (CLAHE, Denoising)

↓

Feature Extraction (Shape, GLCM, Wavelet)

↓

Combining Features

↓

PCA (Reduce Dimensions)

↓

SMOTE (Balance Dataset)

↓

Train-Test Split

↓

Model Training (SVM + XGBoost + Ensemble)

↓

Prediction & Evaluation

2.7 Detailed Explanation of Each Step

- **Image Input**: Load CT scan images from the dataset.
- **Preprocessing**: Use CLAHE and morphological filters to improve image clarity.
- **Feature Extraction**: Calculate important values from images that help detect tumors.
- **Combine Features**: Combine all extracted features into a single array.
- **Apply PCA**: Reduce extra or noisy features while keeping the most useful ones.
- **SMOTE**: Fix imbalance in data so that the model is not biased.
- **Split Data**: Divide data into training and testing sets.
- **Model Training**: Train multiple classifiers and combine them using voting.
- **Evaluation**: Predict on test data and measure the results using various metrics.

Chapter 3 :Project Requirements And Data Set Description

3.1 Development Environment

The entire system is built using Python in an offline environment. For development, a code editor like Visual Studio Code or Jupyter Notebook was used to write, test, and debug the code. These platforms provide an easy interface for running Python scripts and viewing results interactively, especially when working with images and machine learning workflows.

The setup also includes support for external libraries needed for image processing, machine learning, and evaluation. Proper configuration and library installation were done before starting development.

3.2 Data Required for the Project

The success of any machine learning project depends on good-quality data. For this project, we need:

- **CT scan images of the pancreas**

  Some images should be from healthy individuals (normal).

  Some images should show visible pancreatic tumors (tumor).

- **Labelled images**

  Each image must be labelled as either “normal” or “tumor” so the model can learn.

- **Balanced data**

  We use SMOTE to make sure there is a good balance between tumor and normal images.

3.3 Knowledge Requirements

To carry out this project effectively, some background understanding was helpful in the following areas:

- **Medical Imaging**: Understanding how CT scans represent internal organs and tumors.
- **Machine Learning Concepts**: Knowing how classification, training/testing, and model evaluation works.
- **Image Processing**: Awareness of methods to clean, enhance, and extract meaningful patterns from images.
- **Data Imbalance**: Understanding the challenge of having fewer tumor images than normal ones and how synthetic sampling (SMOTE) can address this.

3.4 Dataset Overview

The dataset is split into two main folders:

- **Normal** – Images without any signs of tumors (17,927 images)
- **Tumor** – Images showing pancreatic tumor areas (8,792 images)

Each image is resized to **128×128 pixels** during preprocessing. This fixed size ensures uniformity and reduces computational load.

The total number of samples after combining both folders is approximately **27,000**. Since the tumor class is underrepresented, **SMOTE** is later applied to balance the training data.

These images are real medical scans, making the dataset suitable for building a reliable and practical tumor detection system.

Chapter 4: Implementation and Explanation

4.1 Programming Language and Libraries

This project is developed using **Python**, chosen for its readability and strong ecosystem in machine learning and image processing. Libraries like OpenCV, scikit-learn, NumPy, PyWavelets, and XGBoost offer pre-built functions to process images, extract features, build models, and evaluate performance efficiently. These tools reduce the time required to build a system from scratch and allow focus on the logic and performance of the tumor detection pipeline.

4.2 Image Preprocessing

Before extracting any meaningful information, each image must be cleaned and standardized:

- **Grayscale Conversion**: CT images are already grayscale, but we ensure this using OpenCV.
- **Resizing**: Every image is resized to 128×128 pixels to maintain uniformity.
- **CLAHE (Contrast Limited Adaptive Histogram Equalization)**: Enhances contrast locally in different image regions, helping in revealing details that might indicate a tumor.

- **Morphological Operations**: Techniques like closing are used to remove small holes or noise, especially helpful in highlighting boundaries for shape-based features.
- **Thresholding**: Binary images are created to separate the foreground (tumor region) from the background for further shape analysis.

These preprocessing steps ensure the images are clean and consistent for feature extraction.

4.3 Feature Extraction

To make the images understandable to a machine learning model, we extract the following features:

a. Shape-Based Features

Using contours and boundaries:

- **Area**: Measures the region of the object (tumor).
- **Perimeter**: Length of the boundary of the object.
- **Freeman Chain Code Length**: Describes the shape's outline as a sequence of directions. Helps the model understand the complexity of the tumor's boundary.

b. Texture-Based Features (GLCM)

The Gray-Level Co-occurrence Matrix helps extract texture information:

- **Contrast**: Difference between neighboring pixel values.
- **Homogeneity**: Uniformity of the image.
- **Energy**: Sum of squared elements, representing smoothness.

c. Wavelet-Based Features

Wavelets capture both frequency and location information from an image:

- Coefficients like **cA, cH, cV, cD** are extracted using sym4 wavelet.
- From these, the **mean** and **variance** are calculated to represent textures at different scales and directions.

These features are later saved in .npy files for fast access during training.

4.4 Label Encoding

The dataset is organized such that:

- All images from the normal folder are labeled **0**
- All images from the tumor folder are labeled **1**

This binary classification labeling allows the model to learn which patterns represent healthy tissue and which indicate the presence of a tumor.

4.5 Feature Standardization

Before training the model, features must be scaled:

- **StandardScaler** is used to normalize all values.
- This helps algorithms like SVM and XGBoost treat all features equally.
- Mean is set to 0 and standard deviation to 1, which speeds up learning and improves model performance.

The scaler object is saved using joblib to apply the same transformation to future test data or real-time input.

4.6 Reducing Features with PCA

Since we extract many features from shape, texture, and wavelets, there may be redundancy. To avoid overfitting and reduce computational time:

- **Principal Component Analysis (PCA)** reduces the number of dimensions.
- It keeps the most important information while discarding noise and repetition.
- We chose n_components=10 after analyzing explained variance ratios. These 10 features captured most of the useful information.
- PCA helps speed up training and makes the model more generalizable.

4.7 Splitting the Dataset

The entire dataset is divided into:

- **Training Set**: 74% of the data
- **Testing Set**: 26% of the data

This split ensures the model is trained on a wide variety of examples and then tested on unseen images to check real-world performance.

We also use **stratified sampling** to make sure both tumor and non-tumor images are proportionally present in both training and testing datasets.

4.8 Building the SVM Classifier

We use the **Support Vector Machine (SVM)** algorithm for classification due to its ability to handle complex, high-dimensional data.

- The **RBF (Radial Basis Function)** kernel is used because it can handle non-linear relationships in image data.
- SVM with RBF draws flexible boundaries that can separate tumor from non-tumor data even when the decision boundary is not straight.
- The model is trained using balanced class weights to deal with the dataset imbalance.
- Multiple SVMs are also trained using **BaggingClassifier** to make the model more stable and less sensitive to specific training samples.

4.9 Improving with Hyperparameter Tuning

Rather than guessing the best values for SVM's parameters (C and gamma), we use **RandomizedSearchCV**:

- Tests different combinations of values in multiple folds (3-fold cross-validation)
- Finds the best combination that gives the highest accuracy

- Makes the model more optimized and less prone to overfitting

Once the best parameters are found, they are used to re-train the final model.

Chapter 5: Simulation of the Model (Results/Output)

5.1 Dataset Visualization and Statistics

Before training, it's important to understand how the dataset looks. Our dataset contains approximately **27,000 grayscale CT images**:

- **Normal class (label 0)**: ~17,927 images
- **Tumor class (label 1)**: ~8,792 images

This imbalance can negatively affect training. Hence, we use **SMOTE (Synthetic Minority Over-sampling Technique)** to create artificial tumor samples and balance the training set.

After SMOTE, both classes have approximately the same number of samples in the training set.

5.2 Example Pancreatic Tumor Images

The images below are representative of the types of data used:

- **Normal Image**:
  - Clear structure of the pancreas
  - No visible mass or growth
- **Tumor Image**:
  - Irregular shape or bright region indicating growth
  - Lower contrast or blurry regions may hide small tumors

Image preprocessing (CLAHE + thresholding + denoising) improves the visibility of tumor boundaries before feature extraction.

5.3 Model Predictions (Output)

After preprocessing, feature extraction, PCA, and training, the final model makes predictions:

- **Input**: Processed image features

- **Output**:

  0 = Normal

  1 = Tumor

Internally, the model combines results from two different classifiers:

- A **Bagged SVM** using the RBF kernel
- An **XGBoost classifier** trained on the same features

These models are combined in a **VotingClassifier**, where both models contribute to the final decision (soft voting based on predicted probabilities).

5.4 Accuracy and Evaluation Metrics

After training and testing the model, the following results were obtained:

Metric	**Value**
Training Time	121.36 seconds
Testing Time	53.01 seconds
Train Accuracy	96.69%
Test Accuracy	89.64%
Sensitivity (Recall)	87.52%
Specificity	91.43%
Precision	81.90%
F1 Score	84%
MCC (Matthews Corr.)	77%
AUC (ROC Curve)	0.95%

These results show that the model performs well on both training and unseen test data, with a high **AUC of 0.95**, indicating strong classification ability.

5.5 ROC Curve and Model Reliability

A **Receiver Operating Characteristic (ROC) curve** was plotted to visualize the model's performance across different thresholds. The curve rises sharply and covers most of the top-left region, showing strong separation between classes.

The **AUC (Area Under Curve)** score of **0.95** confirms the model is reliable and generalizes well across unseen data.

Summary of Outputs: -

- The system successfully distinguishes between normal and tumor CT scans.
- It achieves high accuracy and F1-score using **shape, texture, and wavelet features**.
- The combination of **PCA + SMOTE + Bagging + XGBoost** enhances stability and robustness.
- **Training time remains low (<2 minutes)** despite a large dataset of 27,000 images.
- The trained model is saved (ensemble_model.pkl) and can be reused for new predictions.

Chapter 6: Discussion

6.1 Learning Experience and Growth: -

Working on this project has provided deep hands-on learning in multiple areas:

- Image Processing: Understanding how CT images are processed, enhanced, and prepared using methods like CLAHE, resizing, thresholding, and morphological operations.
- Machine Learning Pipeline: End-to-end knowledge of how to convert raw images into features, train models, and evaluate performance.
- Practical Problem Solving: Realization of how real-world data often comes with challenges like imbalance, noise, and variability.
- Research and Application: Connecting theoretical machine learning concepts with actual implementation in Python to build a working medical diagnostic tool.

6.2 Technical Skill Development

This project led to growth in many technical areas:

Area	**What was Learned**
Python Programming	Writing optimized, modular, and scalable code
OpenCV	Preprocessing images for contrast enhancement, resizing, binary segmentation
Feature Engineering	Extracting shape, texture (GLCM), and frequency-based (wavelet) features
Dimensionality Reduction	Using PCA to keep essential information and speed up training
Handling Imbalanced Data	Applying SMOTE to create synthetic tumor samples and balance classes

Area	What was Learned
Modeling	Training, tuning, and combining SVM and XGBoost classifiers
Evaluation	Interpreting sensitivity, specificity, precision, F1 score, MCC, and AUC
Model Persistence	Saving and loading models and preprocessing tools using joblib

6.3 Domain Knowledge

This project strengthened understanding of how machine learning can support healthcare:

- **Medical Imaging Basics**: Learned how CT images represent internal anatomy and how tumors might appear.
- **Tumor Characteristics**: Recognized how shape irregularities and contrast changes help in detecting tumors.
- **Data Interpretation**: Gained appreciation for how subtle image differences can indicate critical health issues.

6.4 Problem Solving and Critical Thinking

Several technical and logical challenges were encountered and resolved:

- **Imbalanced Dataset**: Initially, tumor images were significantly fewer than normal ones. SMOTE was explored and applied to solve this.
- **Feature Redundancy**: Extracting many features led to overfitting risks. PCA was applied to retain only meaningful data.
- **Model Overfitting**: Instead of using only a single model, ensemble learning (SVM + XGBoost) was implemented to improve generalization.
- **Evaluation Bias**: Rather than relying on accuracy alone, multiple metrics were used to get a fair picture of model reliability.

- **Runtime Efficiency**: Feature extraction and model training were optimized to complete within 25 minutes even for large datasets.

Chapter 7: Conclusion and Future Work: -

7.1 Conclusion

The goal of this project was to design a robust, accurate, and efficient system for detecting pancreatic tumors using CT images and machine learning techniques. Throughout the project, various stages were implemented successfully:

- Preprocessing techniques like **CLAHE** and **morphological operations** improved image quality.
- Feature extraction methods including **shape-based**, **texture (GLCM)**, and **wavelet-based** features helped in capturing useful patterns from images.
- **PCA** reduced feature dimensions, making the model faster and less prone to overfitting.
- **SMOTE** balanced the dataset and addressed class imbalance.
- An **ensemble model (SVM + XGBoost)** provided high accuracy and reliable results.

The model achieved:

- **Test Accuracy**: 89.64%
- **Sensitivity**: 87.52%
- **Specificity**: 91.43%
- **AUC**: 0.95

7.2 Challenges Faced

During the development of this project, several challenges were encountered:

- **Class Imbalance**: The dataset had more normal images than tumor images. This was resolved using SMOTE.
- **Feature Selection**: Too many features slowed down the model and increased risk of overfitting. PCA helped reduce unnecessary features.

- **Image Variability**: Some CT scans had low contrast or unclear boundaries. CLAHE and morphological filtering helped bring out hidden structures.
- **Model Overfitting**: Prevented by using ensemble methods and proper cross-validation.
- **Execution Time**: Initial implementations were slow. After optimizations, the complete pipeline (on 27,000 images) now runs in under 25 minutes.

7.3 Future Scope and Improvements

While the model performed well, My ongoing plan is: -

- Designing a **visually appealing and responsive UI** using animations, and vibrant color gradients.
- Enhancing the **HTML templates (index.html and result.html)** to display results in a more user-friendly, visually intuitive format.

7.4 Final Summary

This project successfully applied image processing and machine learning techniques to detect pancreatic tumors in CT images. By combining traditional feature engineering with powerful classifiers like SVM and XGBoost, and handling data issues with PCA and SMOTE, a fast and accurate diagnostic tool was developed.

Chapter 8: References

8.1 Research Papers and Articles

1. Gonzalez, R. C., & Woods, R. E. (2008). *Digital Image Processing*. Pearson Education.
2. **Automatic Detection of Pancreatic Tumors in CT scans Using Machine Learning** – IEEE Transactions on Medical Imaging.
3. National Cancer Institute. *Pancreatic Cancer Overview*. https://www.cancer.gov/types/pancreatic
4. **"Support Vector Machines for Pattern Recognition,"** C. Cortes and V. Vapnik, Machine Learning, 1995.
5. **"A Survey on SMOTE: Synthetic Minority Over-sampling Technique"**, He & Garcia, IEEE Transactions on Knowledge and Data Engineering.
6. **"Combining SVM and XGBoost for Medical Diagnosis,"** Springer Journal of Medical Systems, 2020.

Thank you