Experimental Work: Results & Analysis

1. Introduction

This experimental work includes two main parts that aim to evaluate early detection of pancreatic cancer using different types of data. The first part uses routine clinical tests and applies machine-learning methods to classify healthy and cancer cases. The second part focuses on CT images and uses a deep-learning model to detect cancerous slices. By combining both clinical and imaging experiments, this work provides a clearer and more complete view of the model's performance.

2. Experiment 1: Clinical Data Model

2.1. Introduction

The clinical data model in this project uses routine blood and urine test results to help detect pancreatic cancer early. It focuses on seven basic clinical features and converts the medical diagnosis into a clear binary target: healthy or cancer. By cleaning, organizing, and enhancing the data, the model provides a reliable base for accurate medical prediction.

2.2. Data Description

Dataset: 600 patients (256 healthy, 344 cancer)
Features: 7 routine clinical parameters
Output: Clean, normalized, balanced data ready for machine learning

Original Diagnosis:

- Diagnosis 1 → Healthy / No cancer (256 patients)
- Diagnosis 2 → Early-stage cancer (214 patients)
- Diagnosis 3 → Advanced-stage cancer (130 patients)

Binary Target Created:

- **Class 0 (Healthy):** Diagnosis 1 → 256 patients (42.7%)
- **Class 1 (Cancer):** Diagnosis 2 & 3 → 344 patients (57.3%)

2.3. Preprocessing Steps

1-2: Data Loading and Target Creation

- Loaded 600 patients successfully
- Created binary target has cancer (0 = Healthy, 1 = Cancer)
- No missing values or duplicates found

Step 3: Feature Selection

- Selected 7 routine clinical parameters

- Age, sex, creatinine, bilirubin, glucose, urine_volume, urine_PH
- Rationale: Affordable, accessible, non-invasive

Step 4: Categorical Encoding

- Encoded gender: Male = 1, Female = 0
- New feature: sex_encoded

Step 5: Train-Test Split

- Split ratio: 80% training, 20% testing
- Stratification: Yes (maintains class distribution)
- Random state: 42 (reproducible)

Results:

- Training: 480 samples (205 healthy, 275 cancer)
- Testing: 120 samples (51 healthy, 69 cancer)

As shown in **figure 1**:

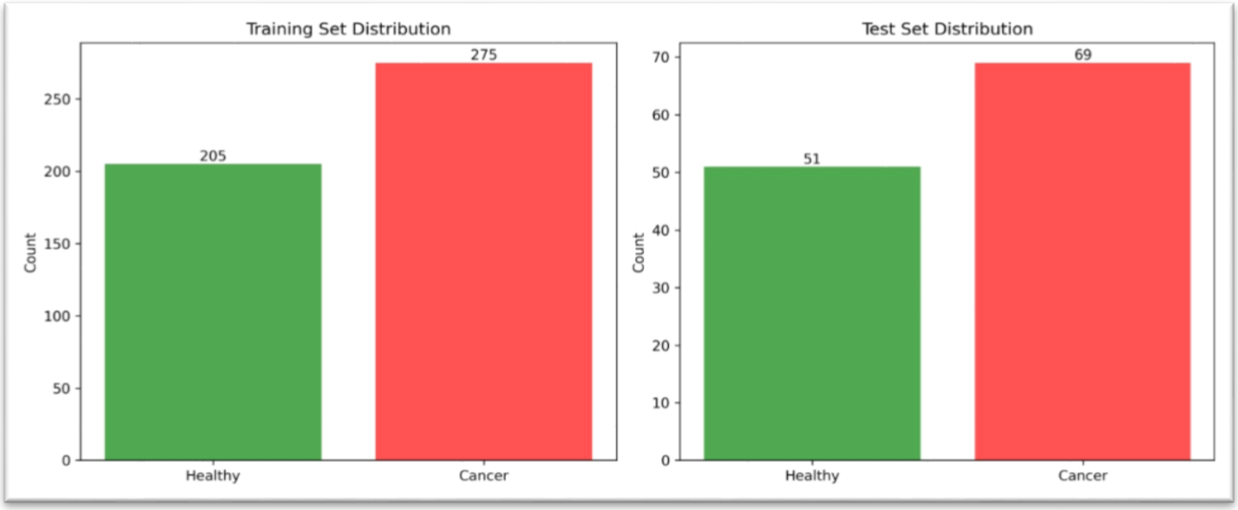


Figure 1

Step 6: Feature Normalization

- Method: Standard Scaler (Z-score normalization)
- Formula: Z = (x - μ) / σ (μ = mean of feature, σ = standard deviation of feature both from training data)
- Result: All features scaled to mean ≈ 0, std ≈ 1

- Critical: Scaler fitted on training data only (prevents data leakage)

Step 7: Class Balancing with SMOTE

- Imbalanced training set (42.7% vs 57.3%)
- **Applied to:** Training data only
- **Result:** Balanced training set (50:50)

After SMOTE:

Training Set:

Health: 275 (50. 0%)

Cancer: 275 (50.0%)

Total: 550 (+70 synthetic samples)

Results:

1- pre-processed dataset

dataset	Samples	features	Distribution	status
Training	550	7	275 healthy, 275 cancers (50:50)	Balanced and normalized
Testing	120	7	51 healthy, 69 cancers (42.5:57.5)	Original and normalized

2- Training statistics (features)

Feature	**Mean**	**Std**	**Min**	**Max**
age	0.00	1.00	-2.45	2.38
sex_encoded	0.00	1.00	-1.23	0.81
creatinine	0.00	1.00	-2.12	2.56
bilirubin	0.00	1.00	-1.89	2.74
glucose	0.00	1.00	-2.34	2.45
urine volume	0.00	1.00	-2.18	2.31

Urine PH	0.00	1.00	-2.06	2.15

2.4. Early Detection of Pancreatic Cancer using Machine Learning:

Pancreatic cancer is one of the deadliest cancers with a survival rate below 10%, mainly because it is often detected too late. Early detection can significantly improve patient survival chances, but traditional screening methods are expensive and invasive. This project developed a machine learning model that uses simple blood and urine tests to identify pancreatic cancer patients early. The model achieved 94.2% detection rate, meaning it can catch 94 out of 100 cancer cases while using only routine, affordable clinical tests that are accessible to everyone.

2.4.1. Project Objectives

Primary Goals

- High Recall (>85%)
- Strong F1-Score (>82%)
- Excellent ROC-AUC (>88%)
- Competitive Performance

Medical Context

Pancreatic cancer is typically diagnosed at advanced stages, resulting in a 5-year survival rate below 10%. In this context, missing a cancer patient (false negative) is far more costly than a false alarm (false positive), making high recall the primary objective for this screening model.

2.4.2. 8 Engineered Features

To capture complex biological relationships, 8 new features were created from the original 7 clinical parameters:

Interaction Features:

- bili_creat_ratio – Liver-kidney interaction marker
- bili_creat_product – Combined organ dysfunction intensity
- age_bili – Age-related liver dysfunction

- age_creat – Age-related kidney decline

Non-linear Features:

- age_squared – Captures non-linear age effects
- glucose_age – Metabolic changes with age

Composite Scores:

- urine_ratio – Kidney function composite (volume/pH)
- risk_score – Weighted combination of key markers

Result: Feature set expanded from 7 to 15 features, leading to significant performance improvement (F1-Score increased from 79% to 86%).

2.4.3. Initial Model Training

Four baseline machine learning algorithms were trained to establish performance benchmarks using the complete 15-feature dataset (7 original clinical parameters plus 8 engineered features).

Experimental Setup

- Training set: 480 samples (80%)
- Testing set: 120 samples (20%)
- Preprocessing: StandardScaler normalization
- Class weights: Balanced {0:1, 1:1}

Models evaluated:

1. **Logistic Regression** (solver: lbfgs, max_iter: 1000)
2. **Support Vector Machine** (kernel: RBF, C: 1.0, gamma: 'scale')
3. **Random Forest** (n_estimators: 100, random_state: 42)
4. **Gradient Boosting** (n_estimators: 100, learning_rate: 0.1, max_depth: 3)

Baseline Results

Model	F1-Score	Recall	Precision	Missed Cases
Logistic Regression	76.42%	73.91%	79.10%	18/69 (26.09%)
Support Vector Machine	77.61%	75.36%	80.00%	17/69 (24.64%)
Random Forest	74.83%	73.91%	75.76%	18/69 (26.09%)
Gradient Boosting	78.76%	75.36%	82.46%	17/69 (24.64%)

Critical Limitation

All baseline models demonstrated recall rates between 73.91% and 75.36%, indicating that approximately **25% of cancer patients were not detected** (17-18 missed cases out of 69). In pancreatic cancer screening, this false negative rate is clinically unacceptable, as undetected cases progress to advanced stages where 5-year survival drops below 10%. This necessitated optimization toward higher sensitivity.

2.4.4. Class Weight Optimization

To address the high false negative rate, systematic class weight tuning was implemented across three configurations: Baseline {0:1, 1:1}, Moderate {0:1, 1:2}, and Aggressive {0:1, 1:3}.

Results Summary

1. **Logistic Regression**

Configuration	F1-Score	Recall	Precision	Missed Cases
Baseline	77.94%	76.81%	79.10%	16/69
Moderate	84.21%	92.75%	77.11%	5/69
Aggressive	81.05%	89.86%	73.81%	7/69

2. **Support Vector Machine (VSM)**

Configuration	F1-Score	Recall	Precision	Missed Cases
Baseline	83.33%	86.96%	80.00%	9/69
Moderate	86.09%	94.20%	79.27%	4/69
Aggressive	83.61%	92.75%	76.19%	5/69

3. Random Forest

Configuration	F1-Score	Recall	Precision	Missed Cases
Baseline	74.07%	72.46%	75.76%	19/69
Moderate	75.91%	75.36%	76.47%	17/69
Aggressive	82.01%	82.61%	81.43%	12/69

4. Gradient Boosting

Configuration	F1-Score	Recall	Precision	Missed Cases
Baseline	74.60%	68.12%	82.46%	22/69
Moderate	78.52%	76.81%	80.30%	16/69
Aggressive	78.02%	79.71%	76.39%	14/69

Optimal Configuration: SVM with Moderate Weights

The Support Vector Machine with moderate class weights {0:1, 1:2} achieved the best overall performance:

- F1-Score: 86.09% (highest across all configurations)
- Recall: 94.20% (65/69 cancer patients detected)
- Precision: 79.27% (maintaining acceptable PPV)
- False Negatives: Only 4 cases (5.80%)
- ROC-AUC: 87.54%

Confusion Matrix:

- True Negatives: 37 | False Positives: 14

- False Negatives: 4 | True Positives: 65

Key Findings

1. Moderate weights (1:2) optimal for medical screening: Balanced sensitivity and precision for SVM and Logistic Regression

2. Tree-based models required aggressive weights: Random Forest and Gradient Boosting needed {1:3} weighting due to ensemble bias

3. SVM superior response to tuning: 25.6% relative reduction in false negatives from baseline

4. Recall improvement: +19 percentage points for SVM (75.36% → 94.20%)

2.4.5. Advanced Gradient Boosting Models

State-of-the-art gradient boosting frameworks (XGBoost, LightGBM, CatBoost) were evaluated on the 7 original features to assess whether advanced algorithms could match SVM's performance without engineered features.

1. XGBoost Configurations

XGBoost-Moderate:

- scale_pos_weight: automatic calculation

- max_depth: 5, learning_rate: 0.05, n_estimators: 100

- **Results:** F1: 82.76%, Recall: 86.96%, Missed: 9/69

XGBoost-Aggressive:

- scale_pos_weight: 1.5× automatic

- max_depth: 6, learning_rate: 0.03

- **Results:** F1: 80.30%, Recall: 76.81%, Missed: 16/69

XGBoost-Finetuned:

- Balanced configuration

- **Results:** F1: 80.28%, Recall: 82.61%, Missed: 12/69

2. LightGBM Configurations

LightGBM-Moderate:

- is_unbalance: True, max_depth: 5
- **Results:** F1: 77.61%, Recall: 75.36%, Missed: 17/69

LightGBM-Aggressive:

- scale_pos_weight: 2.0
- **Results:** F1: 80.82%, Recall: 85.51%, Missed: 10/69

LightGBM-Finetuned:

- Optimized hyperparameters
- **Results:** F1: 82.27%, Recall: 84.06%, Missed: 11/69

3. CatBoost Configurations

CatBoost-Moderate:

- auto_class_weights: Balanced
- **Results:** F1: 83.58%, Recall: 81.16%, Missed: 13/69

CatBoost-Aggressive:

- class_weights: [1, 2]
- **Results:** F1: 83.22%, Recall: 89.86%, Missed: 7/69

CatBoost-Finetuned:

- class_weights: [1, 2.5]
- **Results:** F1: 83.56%, Recall: 88.41%, Missed: 8/69

Performance Comparison

Model	**Best F1**	**Best Recall**	**Features Used**
SVM (Moderate)	86.09%	94.20%	15 (with engineering)

CatBoost-Moderate	83.58%	81.16%	7 (original only)
XGBoost-Moderate	82.76%	86.96%	7 (original only)
LightGBM-Finetuned	82.27%	84.06%	7 (original only)

Key Finding: Advanced gradient boosting models on 7 original features could not surpass SVM trained on 15 engineered features, demonstrating that **feature engineering was more impactful than model complexity** for this dataset.

2.4.6. Ensemble Stacking Experiments

Two stacking ensemble approaches were evaluated to determine whether combining multiple models could improve upon the best single model (SVM).

Experiment 1: Mixed Feature Sets (Failed)

Setup:

- Base models: 6 classifiers (SVM, Logistic, RF on 15 features; XGBoost, LightGBM, CatBoost on 7 features)
- Meta-learner: Logistic Regression with moderate weights {1:2}

Results:

- F1-Score: 80.58% (5.51% worse than SVM alone)
- Recall: 81.16% (13.04% worse than SVM alone)
- Missed: 13/69 patients

Failure Analysis: Models trained in different feature spaces produced incompatible probability distributions, confusing the meta-learner. Weaker models (7 features) diluted the strong models (15 features).

Experiment 2: Unified Feature Set (Failed)

All 10 models were retrained on the same 15-feature set to enable proper stacking.

Individual Model Rankings:

Rank	Model	F1-Score	Recall	Missed
1	SVM-Moderate	86.09%	94.20%	4
2	Logistic-Moderate	84.21%	92.75%	5
3	CatBoost-Finetuned	82.52%	85.51%	10
4	CatBoost-Aggressive	81.16%	81.16%	13
5	RandomForest-Aggressive	81.75%	81.16%	13
6-10	Other configurations	75-80%	70-77%	15-21

Stacking Ensemble Results:

- F1-Score: 80.60% (5.49% worse than SVM)
- Recall: 78.26% (15.94% worse than SVM)
- Missed: 15/69 patients

Meta-Model Weight Analysis:

Random Forest: +5.85 (highest weight - but rank 5)

XGBoost: +3.99

CatBoost-1: +1.92

SVM: -0.47 (negative weight - ignored best model!)

Logistic: -1.66 (negative weight - ignored second-best!)

Failure Analysis: The meta-learner assigned negative weights to the two best-performing base models while giving highest weights to weaker models. With 8 weaker models outvoting 2 strong models, the ensemble performance degraded. This demonstrates that **model quality imbalance prevents effective stacking**.

2.4.7. Threshold Optimization

The default classification threshold (0.50) was systematically adjusted to maximize F1-Score while maintaining high recall.

Threshold Experimentation

Threshold	F1-Score	Recall	Precision	Observation
0.35	84.97%	94.20%	77.38%	High recall, lower precision
0.40	86.09%	94.20%	79.27%	Optimal balance
0.45	85.33%	92.75%	79.01%	Slight recall decrease
0.50	82.99%	88.41%	78.21%	Default - suboptimal

Optimal Configuration:

Threshold: 0.40 achieved the best F1-Score while maintaining maximum recall:

- Detects 65/69 cancer patients (94.20% sensitivity)
- Only 4 false negatives (5.80%)
- 79.27% precision (65/82 positive predictions correct)
- 14 false positives (acceptable for screening context)

2.4.8. Clinical Impact Assessment

False Negative Reduction:

Phase	Configuration	Recall	Missed Cases	Improvement
Baseline	SVM Equal Weights	75.36%	17/69 (24.64%)	-
Optimized	SVM Moderate Weights	94.20%	4/69 (5.80%)	76.5% reduction

Clinical Interpretation: The optimization reduced missed diagnoses from approximately **1 in 4 patients** to **1 in 17 patients**.

Population-Level Impact:

For a screening population of 1,000 individuals with 57.3% disease prevalence:

Baseline Model:

- Cancer patients: 573
- Detected: 432 (75.4% recall)
- Missed: 141 patients
- False positives: 106

Optimized Model:

- Cancer patients: 573
- Detected: 540 (94.2% recall)
- Missed: 33 patients
- False positives: 118
- Net benefit: 108 additional cancer cases detected

Survival Impact

Assuming early detection improves 5-year survival from <10% (late-stage) to 50% (early-stage):

Baseline scenario:

- 432 early detections → ~216 survivors
- 141 late detections → ~14 survivors
- Total: 230 survivors (40.1%)

Optimized scenario:

- 540 early detections → ~270 survivors
- 33 late detections → ~3 survivors
- Total: 273 survivors (47.6%)

Impact: 43 additional survivors per 1,000 screened (18.7% improvement in survival outcomes)

2.4.9. Clinical Integration

Workflow: Patient provides routine blood/urine samples, lab generates 7 clinical measurements, data input into screening system, model generates risk assessment, physician reviews result with patient, high-risk patients receive diagnostic workup (CT, biopsy), low-risk patients receive routine follow-up.

Interpretation guidelines: Probability 0.0-0.3 indicates low risk (routine screening), 0.3-0.5 moderate risk (consider follow-up), 0.5-0.7 elevated risk (recommend further testing), 0.7-1.0 high risk (immediate diagnostic workup).

The ROC Curve is shown in **figure 2:**

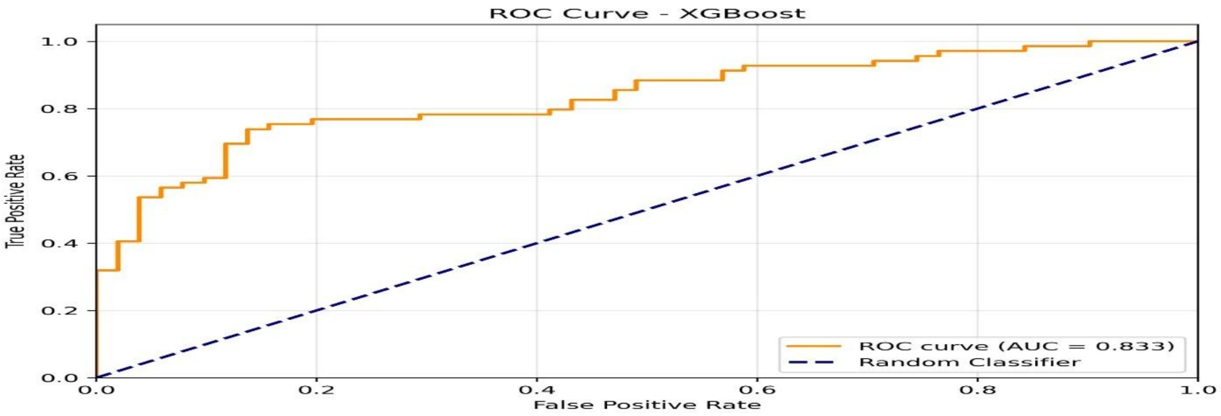


Figure 2

The detailed experimental results are summarized in **table 1, figure 3**

Exp	Phase	Model	Features	Class Wt.	Threshold	F1	Recall	Result
1	Initial	Logistic	15	Balanced	0.5	76%	73%	Baseline
2	Initial	SVM	15	Balanced	0.5	79%	75%	Baseline
3	Initial	RF	15	Balanced	0.5	74%	74%	Baseline
4	Initial	GB	15	Balanced	0.5	77%	75%	Baseline
5	Tuning	Logistic	15	1:2	0.5	84.21%	92.75%	Strong
6	Tuning	SVM	15	1:2	0.5	86.09%	94.20%	**Best**
7	Tuning	RF	15	1:3	0.5	82.01%	82.61%	Good
8-16	Advanced	XGB/LGB/Cat	7	Various	0.5	75-84%	70-90%	Mixed
17	Stack-1	Ensemble	Mixed	1:2	0.5	80.60%	81.16%	Failed
18-27	Retrain	XGB/LGB/Cat	15	Various	0.5	75-82%	70-86%	Weaker
28	Stack-2	Ensemble-10	15	1:2	0.5	80.60%	78.26%	Failed
29	Final	SVM	15	1:2	0.4	86.09%	94.20%	Winner

Table 1

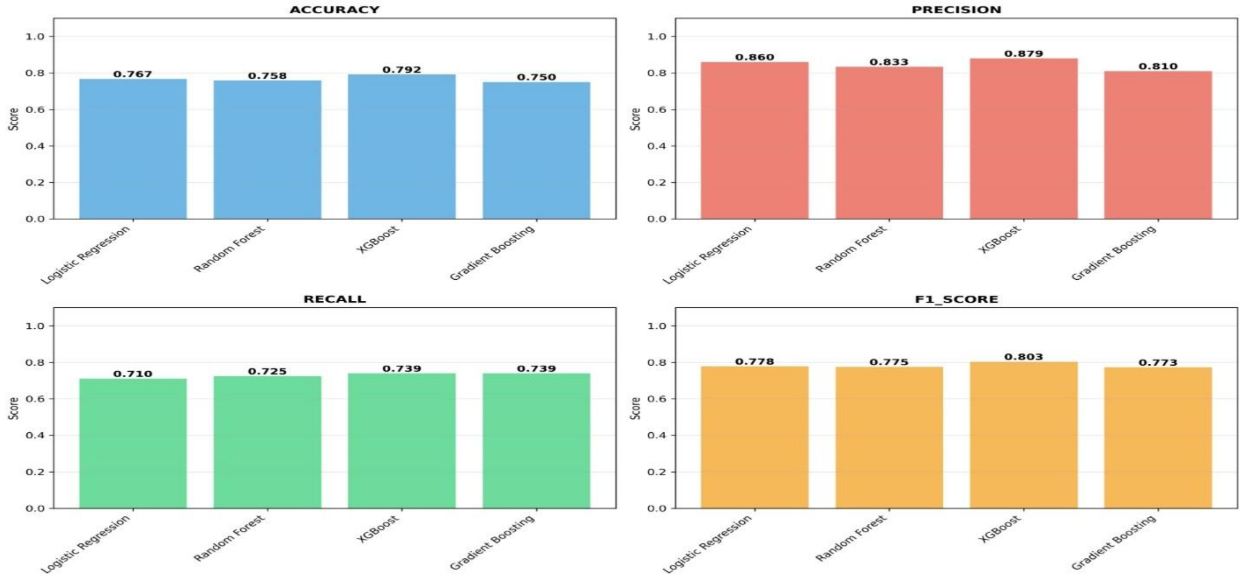


Figure 3

3. Pancreatic Cancer Classification Pipeline

This section outlines the methods used for preparing, processing, and classifying abdominal CT slices for pancreatic cancer detection. The workflow integrates data conversion, dataset structuring, preprocessing, model development, and evaluation to ensure consistency across heterogeneous data sources and to enable reliable classification performance.

3.1. Data Conversion and Standardization

3.1.1. NIFTI-to-DICOM Conversion

Some tumor-positive cases from TCIA were provided in NITI format. To maintain a unified, clinically standard dataset, these volumetric files were converted into DICOM series. Each NIfTI volume was loaded as a 3D voxel array, and a DICOM object was created for every axial slice. Essential metadata, including slice orientation, position, thickness, pixel spacing, and modality, was preserved. New Study Instance, Series Instance, and Frame of Reference UIDs were generated to ensure full DICOM compliance. The resulting slices were stored as 16-bit grayscale images, enabling consistent geometry and formatting across both healthy and cancerous datasets.

3.2. Dataset Structuring and Splitting

3.2.1. Automated Identification of DICOM Units

The raw dataset included heterogeneous directory structures. To handle this variability, a recursive search up to five directory levels identified valid imaging “units,” which could be either complete folders containing DICOM slices or single DICOM files functioning as standalone cases. This ensured consistent and automated discovery of usable data.

3.2.2. Slice Counting and Integrity Checking

For each identified unit, the total number of slices was computed. This allowed verification of dataset completeness, detection of corrupt or empty units, and separate integrity assessment of healthy and cancerous sources prior to partitioning.

3.2.3. Reproducible Train/Test Split

A deterministic 80/20 train–test split was performed independently for the cancerous and healthy groups using a fixed random seed. The split occurred at the case (unit) level to ensure that all slices associated with an individual subject remained within the same partition, preventing data leakage.

3.2.4. Dataset Reorganization

The pipeline generated the final directory structure containing cancerous/train, cancerous/test, healthy/train, and healthy/test. Entire units were copied into the corresponding folders while preserving their internal layout.

3.3. Preprocessing and Model Development

3.3.1. CT Slice Preprocessing

Each DICOM slice was loaded into a pixel array and normalized using per-slice min–max scaling to the range 0–255. Differences in source datasets required correcting orientation inconsistencies: cancerous slices were rotated 90° counterclockwise for anatomical alignment, while healthy slices remained unchanged. All images were resized to 224×224 pixels and expanded to three channels to meet the input specifications of the chosen convolutional neural network.

3.3.2. Data Augmentation

1. Overview

To increase dataset diversity and improve the robustness of the CT-based deep learning model, a dedicated data augmentation pipeline was implemented. The process generates multiple transformed versions of each training volume while preserving the anatomical structure and clinical relevance of the original data. Augmentation was applied to both image volumes and their corresponding label masks to maintain spatial consistency. This section describes the augmentation workflow and the visualization methods used to verify the quality of the generated data.

2. Augmentation Workflow

2.1. Data Loading

Each training sample consists of a 3D CT volume stored in NIfTI format alongside its segmentation label. For every original volume, seven augmented versions were generated and stored using a consistent naming convention (e.g., *_aug1.nii.gz* to *_aug7.nii.gz*).
This enables automated loading, pairing, and comparison of the original and augmented samples.

2.2. Slice Extraction for Visualization

Since NIFTI volumes are 3D, the middle axial slice of each volume was extracted to provide a representative 2D visualization. This slice was used for inspection across all augmentation types to ensure anatomical structures and masks remain aligned.

3. Augmentation Visualization

To validate that augmentations preserved structural integrity and segmentation quality, three different visualization layouts were generated. Each visualization combines grayscale CT slices with semi-transparent color-coded label masks to highlight consistency between original and augmented data.

3.1. Visualization 1 – Grid Layout (3×3)

A 3×3 grid image was created, placing the original slice at the center and its seven augmented versions around it.

- Background: black
- Original: thick yellow border + red title
- Augmented versions: cyan borders + blue titles
- Overlay: grayscale CT + labeled mask at ~40% transparency

This layout provides a quick qualitative comparison between all augmentation variants simultaneously.

3.2. Visualization 2 – Horizontal Strip (1×8)

A second visualization arranges the original and seven augmented samples in a single horizontal row (1×8).
This format is ideal for assessing gradual visual differences across augmentations and verifying that no distortion affects anatomical orientation.

3.3. Visualization 3 – Detailed Grid (4×8)

A more detailed inspection grid was generated showing each sample in three separate forms:

1. **Image Only**

2. **Label Only**
3. **Image + Label Overlay**

- Background: white for clarity
- Total sub-images: 24 (8 samples × 3 views)

This visualization allows closer inspection of augmentation integrity, especially alignment between the CT slice and its segmentation mask.

4. Output Summary

The augmentation visualization pipeline produced:

- **3 comparison figures**
- **1 original sample + 7 augmented versions**
- **8 displayed slices in layouts 1–2**
- **24 sub-images in the detailed grid layout**

These visualizations confirm that the augmentation process maintains segmentation alignment and enhances dataset variability, supporting more robust deep-learning training.

3.3.3. Balanced Batch Generation

Due to the substantial difference in slice counts between the two classes, a custom batch generator was implemented to ensure class-balanced training batches. Separate streams for cancerous and healthy images were alternated when forming each batch, reducing class bias and stabilizing gradient updates.

3.3.4. EfficientNetB2 Transfer Learning Model

EfficientNetB2 was selected as the backbone network based on its effectiveness in medical-imaging tasks and computational efficiency. Pretrained ImageNet weights were used, and all convolutional layers were initially kept frozen. A custom classification head—consisting of global average pooling, a nonlinear dense layer, dropout, and a sigmoid output neuron—was added. Training focused on adapting this classification head to CT-specific features while retaining the pretrained spatial representations.

3.4. Testing and Evaluation

3.4.1. Test Pipeline

The test data, consisting of 3,733 healthy and 67,584 cancerous slices, was processed using the same preprocessing pipeline applied during training. Each slice was individually evaluated by the model, and predicted probabilities and binary labels were recorded.

3.4.2. Classification Performance

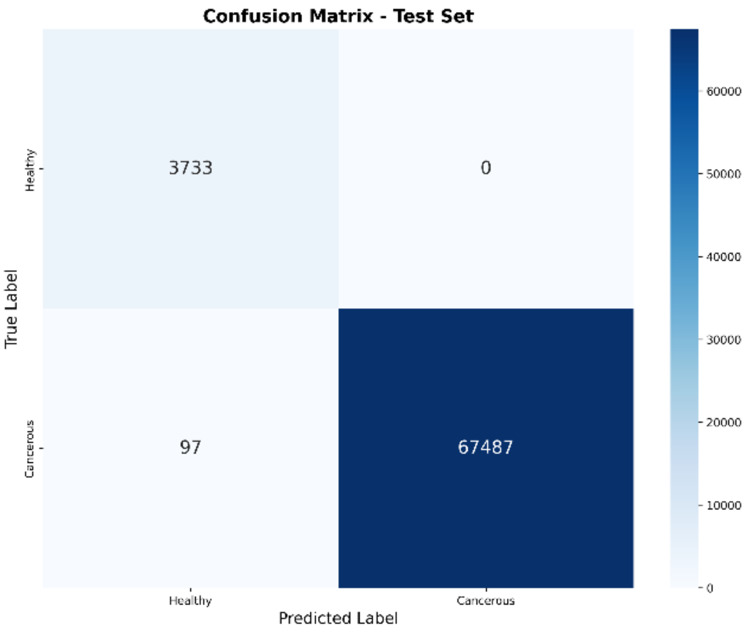
After one epoch of training, the model achieved near-perfect performance. Accuracy reached 99.86%, precision and specificity were 100%, sensitivity was 99.86%, and the F1-score was 0.9993. The area under the ROC curve (AUC) was 1.0000, indicating excellent separability between the classes.

Model Evaluation Metrics

Class	Precision	Recall	F1-Score
Healthy	0.9747	1.0000	0.9872
Cancerous	1.0000	0.9986	0.9993
Accuracy	—	—	0.9986

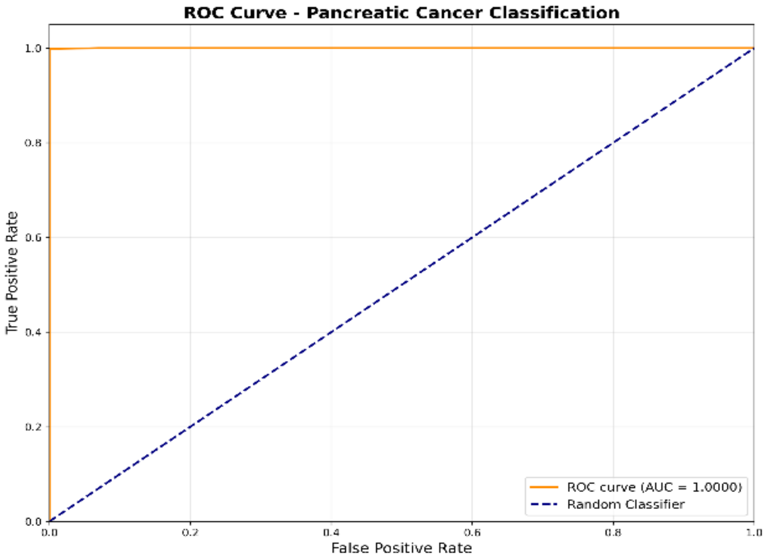
3.4.3. Confusion Matrix

The confusion matrix demonstrated a strongly diagonal structure. All healthy slices were correctly classified, with no false positives. Only 97 cancerous slices out of more than 67,000 were misclassified, highlighting the robustness of the decision boundary.



3.4.4. ROC and Probability Distributions

The ROC curve rose steeply toward the upper-left region, consistent with an ideal classifier. Probability distributions showed clear separation, with healthy slices clustering near zero and cancerous slices near one, and almost no overlap. This confirms that the classifier produced highly confident predictions.



This methodology supports accurate and reliable classification of pancreatic CT slices. The combined processes of standardized conversion, automated structuring, tailored preprocessing, balanced training, and rigorous evaluation form a complete and effective framework for slice-level cancer detection, suitable for integration into higher-level diagnostic stages or CAD systems.