# Project Report: Pancreatic Cancer Prediction Web Application

## 1. Introduction

Pancreatic cancer is one of the most aggressive forms of cancer, and early detection is critical for effective treatment. This project leverages a Convolutional Neural Network (CNN) model integrated with a Flask-based web application to provide predictions on pancreatic cancer presence using medical images.

## 2. Project Objective

- To develop an AI-driven application for predicting pancreatic cancer.  
- To integrate the trained CNN model into a user-friendly web interface.  
- To visualize prediction results, including accuracy, image visualization, and confusion matrix.

## 3. Dataset

- Source: Dataset organized into three directories:  
 - Training Directory: `P:/Anusha/Images/validation`  
 - Validation Directory: `P:/Anusha/Images/Test`  
 - Test Directory: `P:/Anusha/Images/Test\_set`  
- Preprocessing:  
 - Images resized to `128 x 128` pixels.  
 - Pixel values normalized to the range [0, 1].

## 4. Model Architecture

The Convolutional Neural Network (CNN) model used in this project comprises:  
- Convolutional Layers: Extract spatial features from the images.  
- Pooling Layers: Downsample feature maps using max-pooling.  
- Flatten Layer: Converts the 2D feature maps into a 1D vector.  
- Fully Connected Layers:  
 - Dense layer with 128 units (ReLU activation).  
 - Dropout layer with 50% rate to prevent overfitting.  
 - Output layer with 1 unit (Sigmoid activation) for binary classification.

## 5. Web Application Details

- Frameworks Used:  
 - Flask for backend development.  
 - TensorFlow/Keras for model integration.  
 - Bootstrap for frontend design.  
- Key Features:  
 - Users can upload medical images for prediction.  
 - Prediction results displayed with accuracy.  
 - Confusion matrix and image visualization provided for better understanding.  
- Upload Directory: `./uploads` for storing uploaded images.

## 6. Prediction Workflow

1. User Uploads Image:  
 - The image is uploaded through the web interface and stored in the `./uploads` folder.  
2. Preprocessing:  
 - Image resized to `128 x 128` pixels and normalized.  
3. Prediction:  
 - CNN model predicts the likelihood of pancreatic cancer.  
 - Binary classification: Positive (Cancerous) or Negative (Non-Cancerous).  
4. Results Displayed:  
 - Prediction result with accuracy.  
 - Visualized confusion matrix.  
 - Image with prediction overlay.

## 7. Evaluation Metrics

- Confusion Matrix:  
 - Visual representation of predictions.  
 - Includes True Positives (TP), True Negatives (TN), False Positives (FP), and False Negatives (FN).  
- Accuracy:  
 - Measures the percentage of correct predictions.

## 8. Results

- Prediction Accuracy: Achieved \*\*(insert accuracy)\*\* on test images.  
- Confusion Matrix Visualization:  
 - Provides insight into model performance.  
- Image Visualization: Displays uploaded image with overlayed prediction result.

## 9. User Interface

- Design:  
 - Responsive and user-friendly interface built using Bootstrap.  
 - Color-coded labels for clear visualization.  
- Features:  
 - File upload with error handling.  
 - Display of prediction results.

## 10. Future Work

- Data Augmentation:  
 - Enhance training dataset with augmented images to improve generalization.  
- Model Optimization:  
 - Use transfer learning with pre-trained models for better accuracy.  
- Real-Time Deployment:  
 - Deploy application on cloud platforms for real-time usage.

## 11. Conclusion

This project demonstrates the use of a CNN model integrated into a Flask web application to predict pancreatic cancer. While the results show promising accuracy, further improvements can be made through advanced model tuning and dataset expansion. The web application provides an accessible tool for medical professionals to assist in early diagnosis.

## Appendix:

- Source Code: Includes Python scripts for the Flask app, model loading, and predictions.  
- Dependencies:  
 - Python packages: Flask, TensorFlow, Keras, NumPy, Matplotlib, scikit-learn, h5py.  
 - Frontend: Bootstrap, Font Awesome.