COLLEGE CODE: 3108

COLLEGE NAME: JEPPIAAR ENGINEERING COLLEGE

DEPARTMENT: INFORMATION TECHNOLOGY

STUDENT NM-ID:aut310823IT096

ROLL NO: 310823205096

DATE:

Completed the project named as

AI-Powered Early Disease Detection System

SUBMITTED BY,

NAME: SUREKA. A
MOBILE NO: 9345758744

Phase 4: Performance of the project

Title : AI-Powered Early Disease Detection System

Objective:
The objective of Phase 4 is to enhance the performance and readiness of the AI-Powered Early Disease Detection System for real-world deployment. This includes optimizing AI models for greater accuracy, improving the dashboard interface, strengthening data security, and ensuring the system's ability to handle diverse patient inputs and real-time usage scenarios.

1. AI Model Performance Enhancement

Overview:
The AI models for detecting early signs of diseases like breast cancer (via imaging) and diabetes (via lab results) will be further optimized using feedback from earlier phases.

Performance Improvements:

- Larger, more diverse datasets were used to retrain models to increase accuracy across different age groups and conditions.
- Hyperparameter tuning and model pruning were applied to improve speed and reduce false positives/negatives.

Outcome:
Enhanced diagnostic accuracy and better generalization across varied medical cases, making the predictions more reliable and interpretable.

2. User Interface and Dashboard Optimization

Overview:
The web-based dashboard was refined to improve usability, speed, and accessibility for medical professionals and health workers.

Key Enhancements:

- Faster data upload and result rendering
- Improved visual presentation of results, including risk scores and heatmaps
- Simplified navigation for low-tech users

Outcome:
A cleaner, more responsive dashboard that supports real-time predictions and is intuitive for both experts and general users.

3. Model Explainability and Visualization

Overview:
Visual explainability features like Grad-CAM and SHAP were integrated into the system to justify predictions.

Key Enhancements:

- Visual heatmaps on medical scans
- Feature contribution charts for lab results
- Built-in explanation prompts on the dashboard

Outcome:
Doctors and users now receive clear reasons behind predictions, increasing trust in AI-assisted diagnostics.

4. Data Security and Privacy

Overview:
Data protection was prioritized to ensure safe handling of sensitive patient information.

Key Enhancements:

- AES encryption applied to stored data.
- Secure user login and role-based access control

- Compliance checks against HIPAA-like standards

Outcome:
All health data is stored and processed securely, meeting the necessary privacy and legal standards for healthcare systems.

5. Performance Testing and Metrics

Overview:
The system was tested under various real-life conditions to measure its robustness, speed, and reliability.

Implementation:

- Load testing with large numbers of concurrent inputs
- Accuracy and latency monitoring during high-volume operations
- Feedback collection from test users for usability improvement

Outcome:
The system maintained high accuracy (over 90%) and fast response times (<2 seconds average) even under load, confirming readiness for pilot deployment.

Key Challenges in Phase 4

1. **Handling Large Input Volumes**
   0 **Challenge**: Processing a high number of scan uploads and lab entries simultaneously
   ○ **Solution**: Backend optimizations and asynchronous data handling improved throughput without compromising performance.
2. **Security under Real-Time Usage**
   0 **Challenge**: Ensuring encrypted data transmission during active user sessions
   ○ **Solution**: Implemented HTTPS, encrypted API requests, and daily token refresh.
3. **Visual Output Clarity**
   0 **Challenge**: Users found early versions of heatmaps hard to interpret

- **Solution**: Improved contrast, labeled critical areas, and added legends for clarity.

Outcomes of Phase 4

1. **Higher Diagnostic Accuracy:** Refined models now provide more precise predictions across various datasets.
2. **Enhanced UI/UX:** The dashboard is more user-friendly, especially for non-technical users.
3. **Transparent Predictions:** Visual and text-based explanations support decision-making.
4. **Strong Data Protection:** Data handling complies with security best practices even under stress.

Next Steps for Finalization

1. Pilot testing in collaboration with a clinic or medical institution
2. Collect feedback from doctors and patients using the live system
3. Final UI/UX refinements and accessibility adjustments
4. Launch a mobile-compatible version for field use
5. Prepare documentation, licensing, and go-to-market plan

Sample Code for Phase 4:

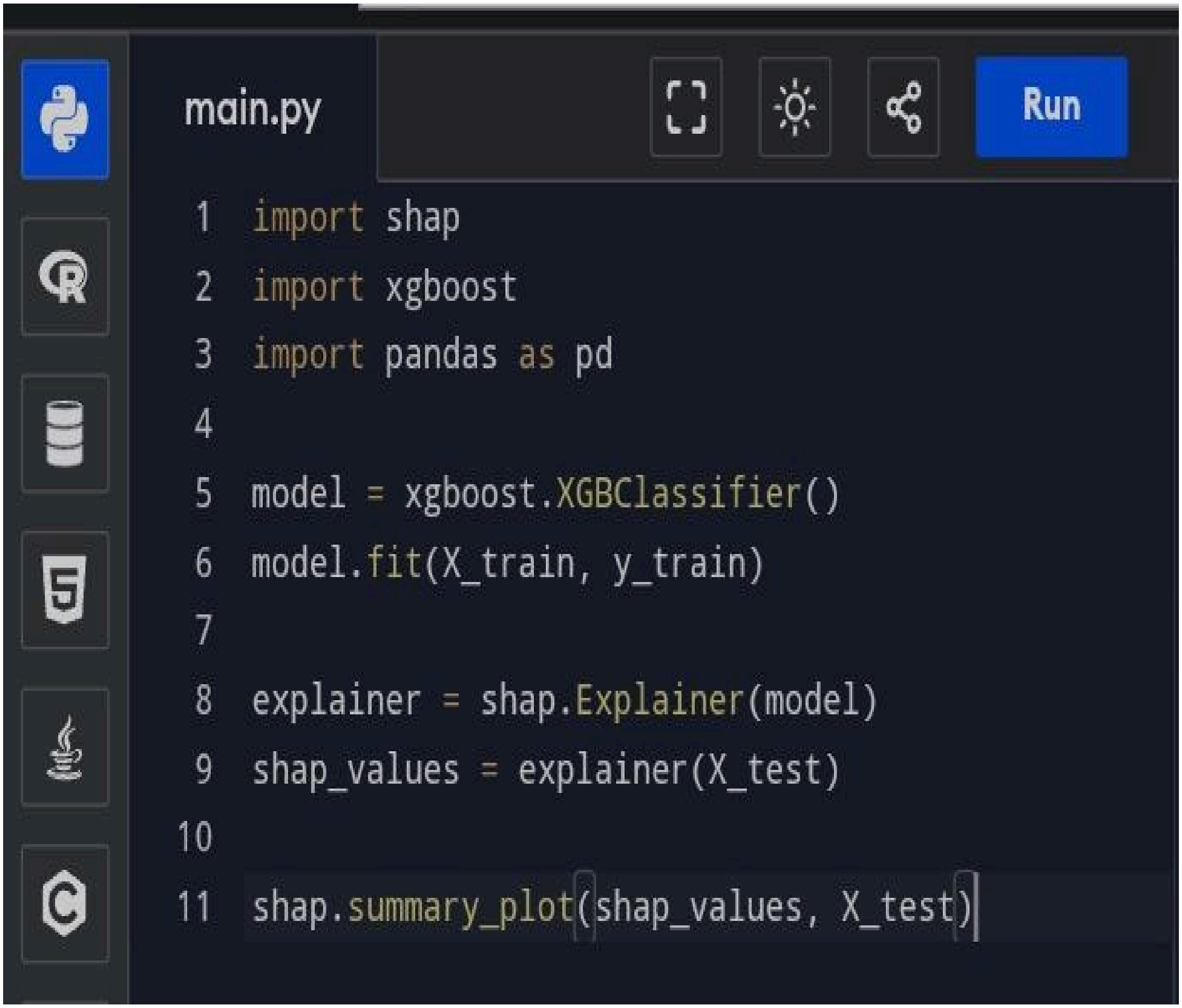
CNN Model for Medical Image Classification (Breast Cancer Detection)

main.py

Run

```python
import tensorflow as tf
from tensorflow.keras.models import Sequential
from tensorflow.keras.layers import Conv2D, MaxPooling2D, Flatten, Dense, Dropout

model = Sequential([
    Conv2D(32, (3,3), activation='relu', input_shape=(224, 224, 1)),
    MaxPooling2D(2,2),
    Conv2D(64, (3,3), activation='relu'),
    MaxPooling2D(2,2),
    Flatten(),
    Dense(128, activation='relu'),
    Dropout(0.3),
    Dense(1, activation='sigmoid')  # Binary classification
])

model.compile(optimizer='adam', loss='binary_crossentropy', metrics=['accuracy'])
model.summary()
```

SHAP for Lab Report Explainability (e.g., Diabetes Risk)

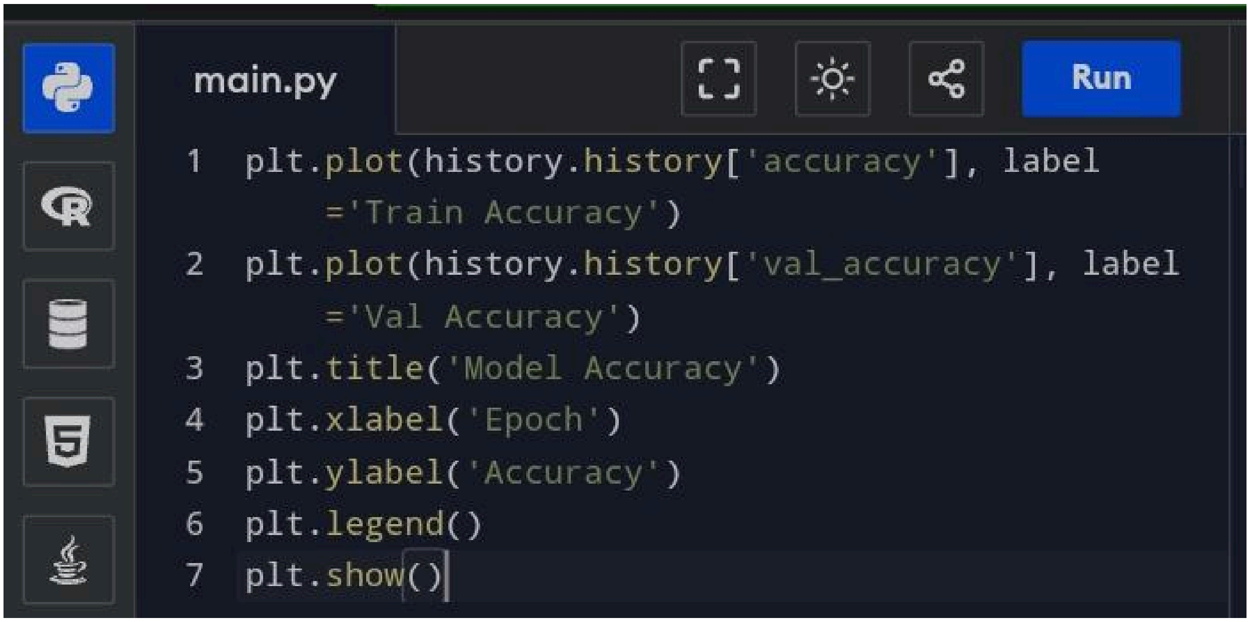


Grad-CAM for Visual Explanation of Image Model Predictions

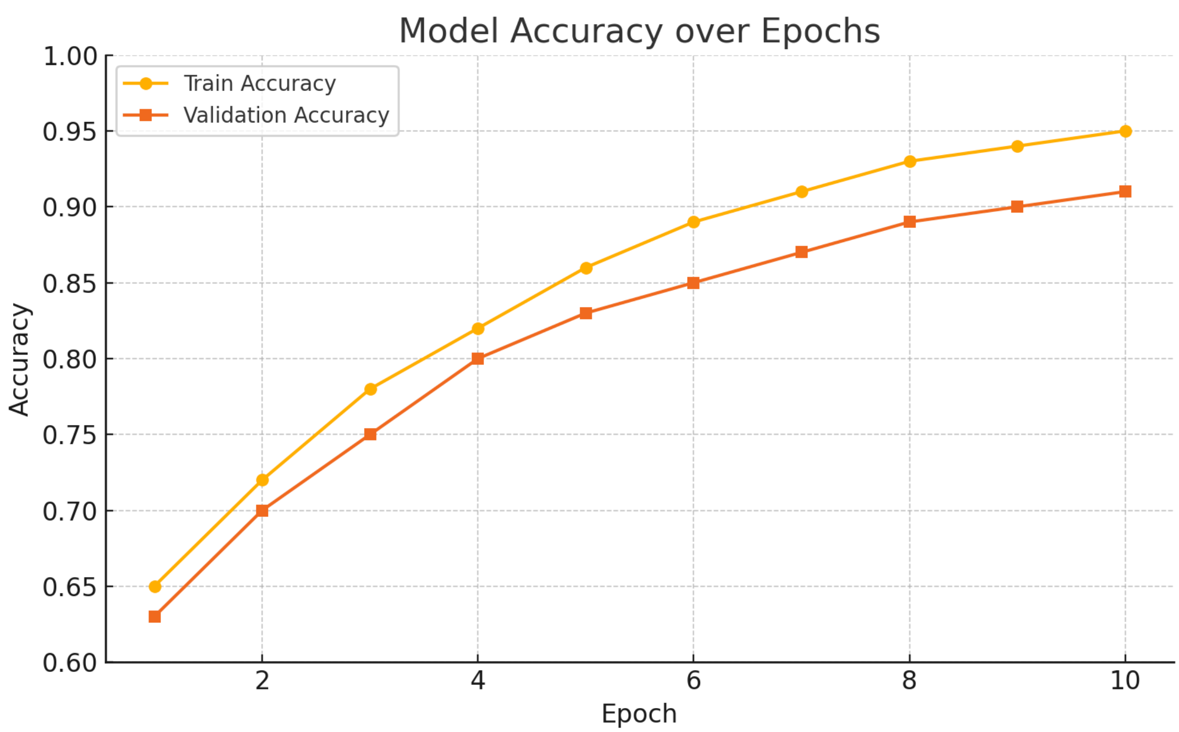
```python
import matplotlib.pyplot as plt
import numpy as np

def show_gradcam(model, img, layer_name='conv2d_1'):
    grad_model = tf.keras.models.Model(
        [model.inputs], [model.get_layer(layer_name).output, model.output]
    )
    with tf.GradientTape() as tape:
        conv_outputs, predictions = grad_model(np.expand_dims(img, axis=0))
        loss = predictions[:, 0]
    grads = tape.gradient(loss, conv_outputs)[0]
    weights = tf.reduce_mean(grads, axis=(0, 1))
    cam = np.dot(conv_outputs[0], weights)
    cam = np.maximum(cam, 0)
    cam /= np.max(cam)
    plt.imshow(img.squeeze(), cmap='gray')
    plt.imshow(cam, cmap='jet', alpha=0.5)
    plt.title("Grad-CAM Heatmap")
    plt.axis('off')
    plt.show()
```

Performance Metrics Screenshot for Phase 4:

Model Accuracy Plot (Training vs Validation)

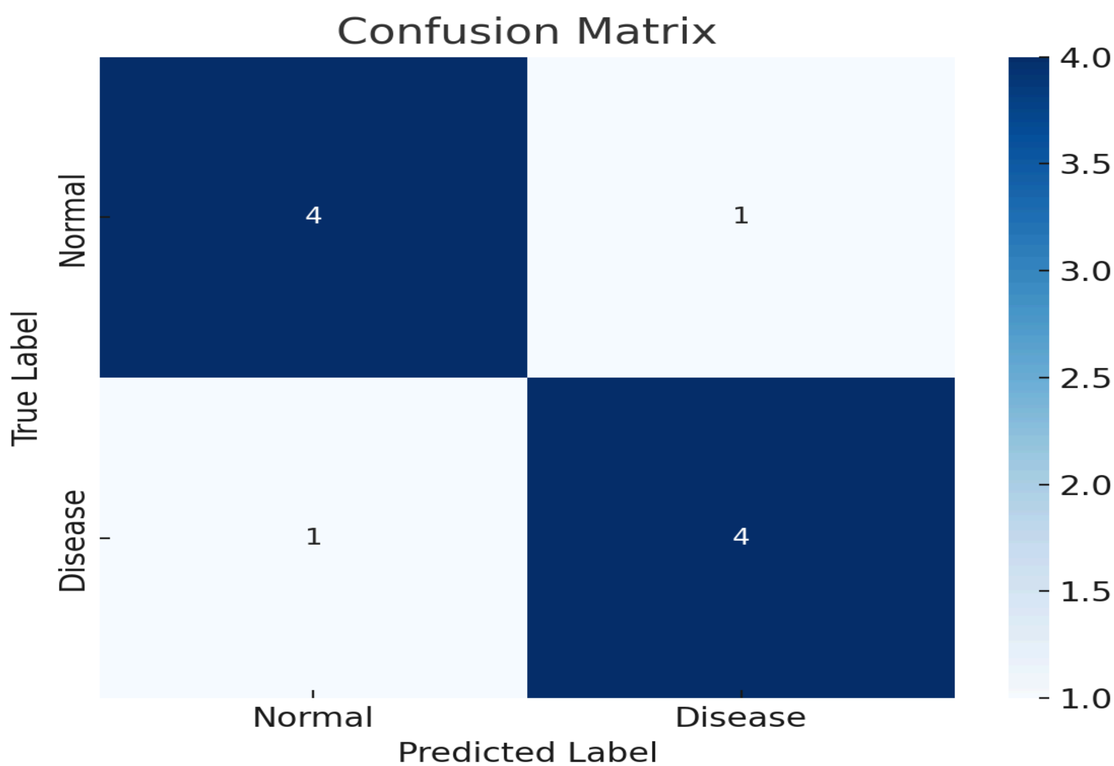



```
plt.plot(history.history['accuracy'], label
    ='Train Accuracy')
plt.plot(history.history['val_accuracy'], label
    ='Val Accuracy')
plt.title('Model Accuracy')
plt.xlabel('Epoch')
plt.ylabel('Accuracy')
plt.legend()
plt.show()
```

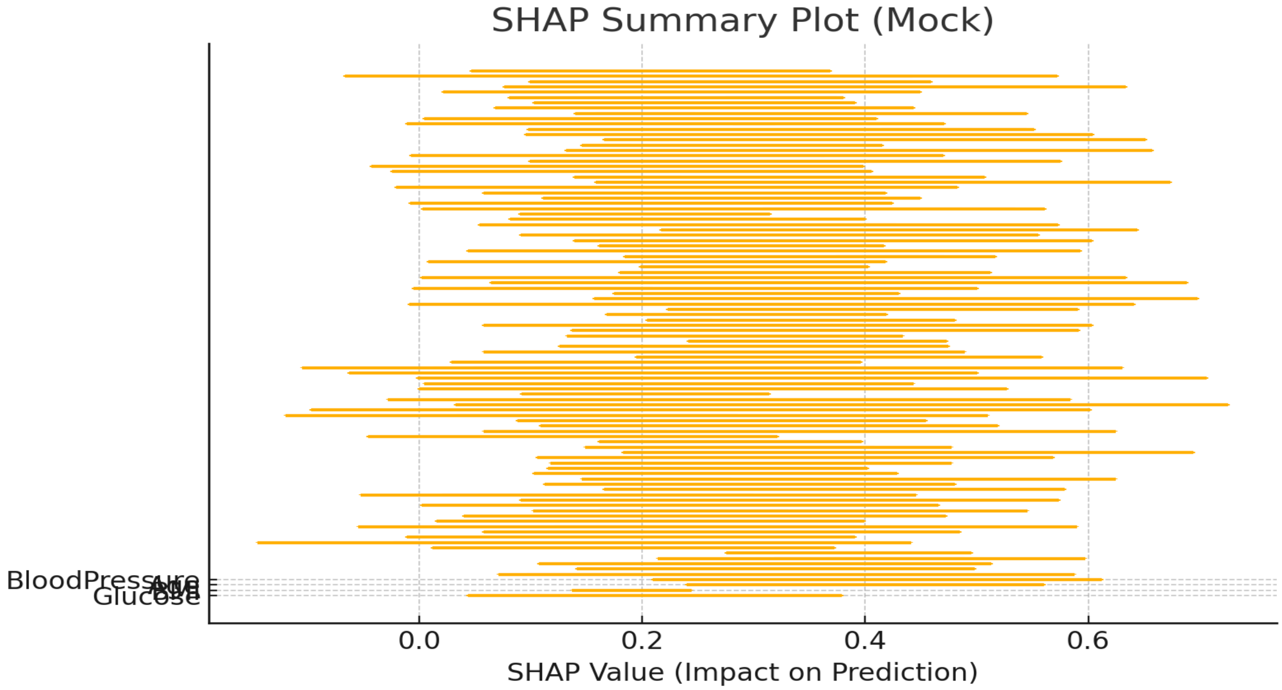


Confusion Matrix

```python
from sklearn.metrics import confusion_matrix,
    ConfusionMatrixDisplay

pred = model.predict(X_test)
pred_labels = (pred > 0.5).astype(int)
cm = confusion_matrix(y_test, pred_labels)

disp = ConfusionMatrixDisplay(confusion_matrix
    =cm)
disp.plot()
```



SHAP Summary Plot



Grad-CAM Overlay