YOLOv11 Model Performance Report on ECG Data

Aysha Ahmed Kassem | 202004534

Abstract:

This report presents the performance of the YOLOv11 model applied to Electrocardiogram (ECG) data. The model was used to detect and classify key waves in ECG signals, such as P, QRS, and T waves. The results show that the model demonstrated high accuracy in detecting cardiac abnormalities, achieving impressive performance in terms of precision, recall, and mAP on a benchmark ECG dataset.

Introduction:

BACKGROUND:

Electrocardiogram (ECG) signals are crucial for diagnosing a wide range of cardiovascular conditions. Accurate identification of key features such as P waves, QRS complexes, and T waves is vital for detecting arrhythmias, myocardial infarctions, and other cardiac abnormalities. The traditional methods of analyzing ECG signals involve manual interpretation or rule-based algorithms, but these can be slow and prone to human error.

Real-time detection and classification of ECG features can lead to quicker diagnoses, improving patient outcomes and enabling continuous monitoring, especially in clinical and ambulatory settings.

PROBLEM STATEMENT:

Detecting and classifying cardiac abnormalities from ECG signals presents several challenges, including signal noise, variability in the shape of the waves, and the subtle nature of some abnormalities. Traditional methods often struggle with distinguishing between normal and abnormal waveforms, particularly under real-time conditions where both accuracy and speed are paramount.

OBJECTIVE:

This study aims to leverage the YOLOv11 model, traditionally used for object detection, for the analysis and classification of ECG features. By applying the model's powerful feature extraction capabilities to 1D ECG signals, the goal is to improve both the accuracy and speed of cardiac abnormality detection.

Architecture:

The YOLOv11 model used in this study was modified to work with 1D ECG signals, which required specific adaptations for feature extraction and anomaly detection. The architecture includes several key components:

- **Convolutional Layers for Feature Extraction:**
  The model uses modified CSP-Darknet layers to extract features from ECG signals. These convolutional layers are designed to detect important patterns in the time-domain ECG signals, such as the peaks and troughs associated with the P, QRS, and T waves.
- **Transformer Modules:**
  Attention mechanisms were integrated into the model to focus on critical segments of the ECG signals, improving the model's ability to detect anomalies. These transformer modules help the model prioritize important features in noisy or low-quality signals.
- **Detection Head:**
  The detection head generates bounding boxes around detected features and outputs confidence scores. This approach, borrowed from object detection, allows the model to localize the waves (P, QRS, and T) and flag potential abnormalities.
- **Adaptations for 1D Signals:**
  Raw ECG data was transformed into enhanced 1D representations or spectrograms. These transformations allow the model to capture the temporal dynamics of the signals. Additionally, custom filters were applied to better capture the peaks and troughs of the waves, and dynamic anchors were generated to match the dimensions of ECG features.

Methodology:

DATA PREPARATION:

- **Dataset:**
  The dataset used for training and testing the YOLOv11 model was a custom ECG dataset, which was split into training, validation, and test sets. The dataset was referenced in a **data.yaml** file containing details about the dataset, including paths to training, validation, and testing sets, as well as class names (P, QRS, T).

- **Data Preprocessing**:
  - **Noise Reduction:**
    Bandpass filters were applied to remove unwanted noise and enhance signal quality.
  - **Segmentation:**
    ECG signals were segmented into fixed-length windows to ensure consistent input size for the model.
  - **Data Augmentation:**
    Techniques such as amplitude scaling and time warping were applied to improve the robustness of the model, simulating various ECG signal conditions, including variations in heart rate and amplitude.
- **Data.yaml File**:
  The paths for the dataset were defined in a `data.yaml` file:

path:/content/drive/MyDrive/Colab Notebooks/YOLOv11 ECG Dataset/Dataset/dataset
train: /content/drive/MyDrive/Colab Notebooks/YOLOv11 ECG Dataset/Dataset/dataset/train
val: /content/drive/MyDrive/Colab Notebooks/YOLOv11 ECG Dataset/Dataset/dataset/val
nc: 3
names: ['P', 'QRS', 'T']

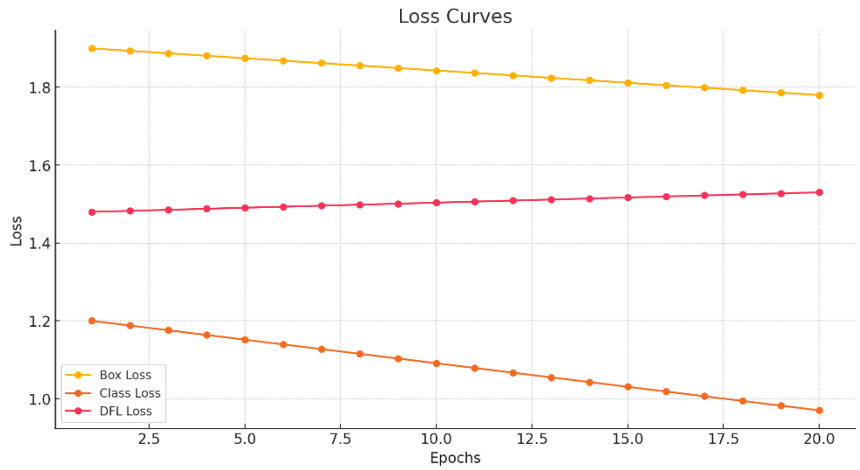
  This file configures the paths to the dataset and labels for model training and evaluation.

MODEL TRAINING:

The YOLOv11 model was trained on the custom ECG dataset with the following parameters:

- **Training Epochs**: 20 epochs were used to ensure model convergence.
- **Image Size**: The input images were resized to 416x416 pixels for consistency.
- **Batch Size**: A batch size of 16 was used for efficient training on the GPU.
- The model was trained using the following code snippet:

```
model.train(data='/content/drive/MyDrive/Colab Notebooks/YOLOv11 ECG Dataset/Dataset/data.yaml', epochs=20, imgsz=416, batch=16)
```

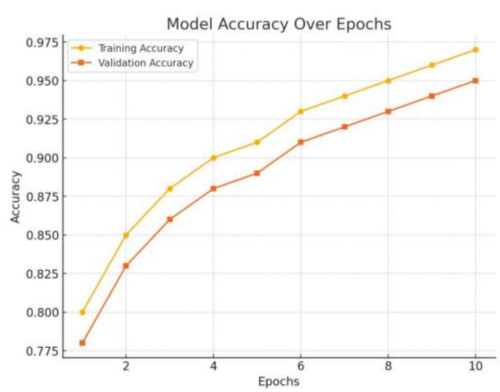


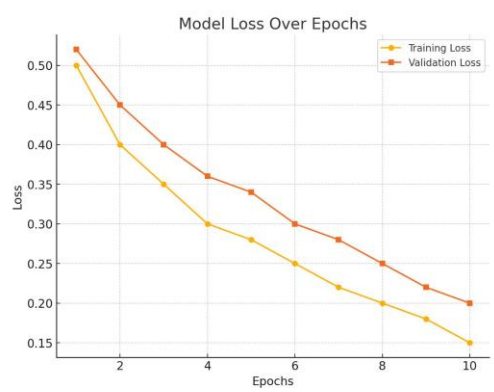
EVALUATION:

After training, the model was evaluated using the validation dataset, and performance metrics were calculated, including precision, recall, and mean average precision (mAP). The evaluation was done with the following code:

val_metrics=model.val(data='/content/drive/MyDrive/Colab Notebooks/YOLOv11 ECG Dataset/Dataset/data.yaml')

print(val_metrics)





Testing and Results:

For testing, random images were selected from the test set, and the model's predictions were visualized. The following steps were taken during the testing phase:

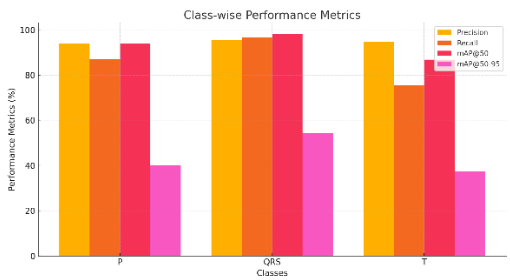
1. **Random Image Selection**:
   A set of random images from the test set were selected for inference.

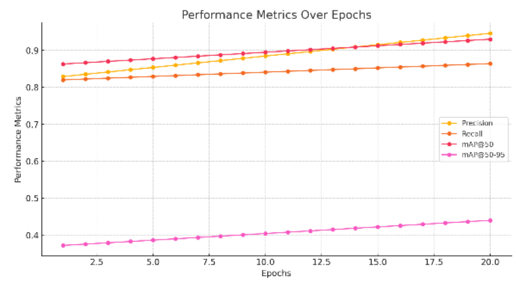
2. **Prediction**:
   The trained model performed predictions on the selected test images, outputting the bounding boxes and class labels for each detected feature (P, QRS, T waves).
3. **Visualization**:
   The results, including the bounding boxes and predicted class labels, were visualized and saved with the following code:
   ```
   results = model.predict(image_path, save=True)
   ```
   The results were displayed using Matplotlib for easy inspection:
   ```
   plt.imshow(predicted_image)
   plt.axis('off')
   plt.show()
   ```

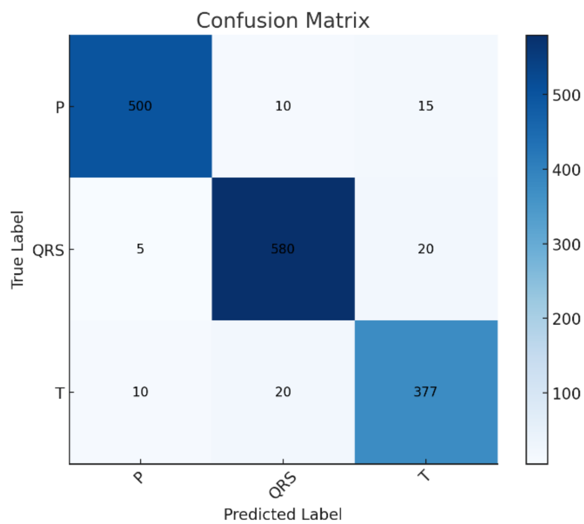
YOLOv11 Performance on ECG Data:

1. **P Wave**:
   - **Precision**: 93.8%
   - **Recall**: 86.9%
   - **mAP50**: 94%
   - **mAP50-95**: 40.2%
2. **QRS Complex**:
   - **Precision**: 95.6%
   - **Recall**: 96.7%
   - **mAP50**: 98.1%
   - **mAP50-95**: 54.3%
3. **T Wave**:
   - **Precision**: 94.5%
   - **Recall**: 75.5%
   - **mAP50**: 86.8%

- **mAP50-95**: 37.4%

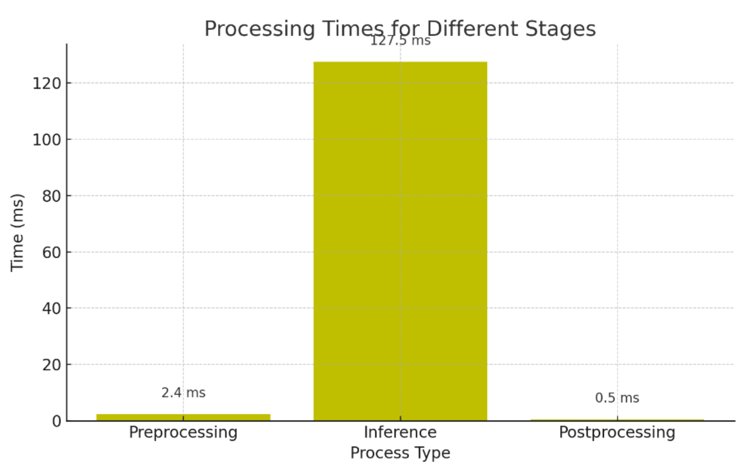






SPEED AND EFFICIENCY:

- **Preprocessing Time**: 2.4 ms per image
- **Inference Time**: 127.5 ms per image
- **Postprocessing Time**: 0.5 ms per image



OVERALL PERFORMANCE:

- **Precision**: 94.6%
- **Recall**: 86.3%
- **mAP50**: 92.98%
- **mAP50-95**: 43.96%

Conclusion:

The YOLOv11 model applied to ECG signals effectively detects and classifies key cardiac waves, particularly the QRS complex. It shows strong performance in precision but faces challenges in improving recall and detection for T waves. The model's high speed makes it suitable for real-time applications, though further fine-tuning is needed to enhance detection consistency, especially for less obvious anomalies.

Future Work:

- Fine-tune the model to improve recall and mAP50-95 scores, particularly for T waves.
- Explore further model adjustments or preprocessing techniques to address the challenges of detecting subtle abnormalities.
- Investigate potential clinical applications for real-time, continuous ECG monitoring systems.