

A Novel Application for Real-time Arrhythmia Detection using YOLOv8

Guang Jun Nicholas Ang^{a,1}, Aritejh Kr Goil^{a,1}, Henryk Chan^{a,b}, Jieyi Jeric Lew^a, Xin Chun Lee^a, Raihan Bin Ahmad Mustaffa^a, Timotius Jason^a, Ze Ting Woon^a, and Bingquan Shen^{a,c}

^aNational University of Singapore, Singapore

^bUniversity of Sheffield, United Kingdom

^cDSO National Laboratories, Singapore

¹{angjunicholas, aritejh}@u.nus.edu

Abstract

In recent years, there has been an increasing need to reduce healthcare costs in remote monitoring of cardiovascular health. Detecting and classifying cardiac arrhythmia is critical to diagnosing patients with cardiac abnormalities. This paper shows that complex systems such as electrocardiograms (ECG) can be applicable for at-home monitoring. This paper proposes a novel application for arrhythmia detection using the state-of-the-art You-Only-Look-Once (YOLO)v8 algorithm to classify single-lead ECG signals. We proposed a loss-modified YOLOv8 model that was fine-tuned on the MIT-BIH arrhythmia dataset to detect to allow real-time continuous monitoring. Results show that our model can detect arrhythmia with an average accuracy of 99.5% and 0.992 mAP@50 with a detection time of 0.002s on an NVIDIA Tesla V100. Our study demonstrated the potential of real-time arrhythmia detection, where the model output can be visually interpreted for at-home users. Furthermore, this study could be extended into a real-time XAI model, deployed in the healthcare industry, and significantly advancing healthcare needs.

Keywords Healthcare · ECG · YOLO · Arrhythmia Detection · Computer Vision

1 Introduction

1.1 Background

Cardiac arrhythmia (or heart rhythm disorders) is a ubiquitous global ailment which occurs in 2.35% of adults, accounting for 60% of deaths caused by cardiovascular disease [1, 2, 3]. While some arrhythmias may be harmless or singular, persistent arrhythmias may cause organ dysfunction, increase the risk of stroke by five times and even cause sudden cardiac death [4]. Thus, early detection and classification of electrocardiogram (ECG) signals are crucial to diagnosing and treating arrhythmia to prevent the onset of life-threatening conditions [5]. ECG is a non-invasive approach to detecting arrhythmia by analysing the spatial map of the time-varying electrical potentials of the heart using electrodes placed at specific locations on the skin [3]. Interpretation of the ECG highlights structural and functional abnormalities of the heart to aid in diagnosing cardiovascular diseases [6]. However, interpretation requires expert knowledge and continuous monitoring over an extended period through ambulatory Holter devices [7, 8]. Furthermore, since Holter devices do not analyse the ECG, the high signal data makes manual interpretation time-consuming and prone to fatigue-induced error.

1.2 Machine learning

Machine learning has led to the development of computer-aided diagnostic systems (CADS) that can automatically classify ECGs to assist cardiologists in detecting arrhythmia from long-term ECG recordings. These systems employ feature engineering techniques such as Hermite functions and polynomials, wavelet-based features, and ECG morphology to extract features from ECG signals [9, 10, 11]. Subsequently, machine learning models, including the k-th

nearest-neighbours (KNN) algorithm, decision trees, and support vector machines (SVMs), are used to match these complex features to represent the preprocessed ECG signal as a sequence of stochastic patterns [12, 13, 14, 15]. Unfortunately, the combined use of feature engineering and dimensionality reduction algorithms significantly increased the computational complexity of the overall process, thereby limiting the usage of portable or wearable health monitoring devices [5].

1.3 Deep learning

In recent years, deep learning has further revolutionised arrhythmia detection for CADs. Deep learning (DL) uses artificial neural networks with multiple hidden layers to automatically learn complex and non-linear relationships [16]. DL addresses the computational burden caused by feature engineering and other preprocessing techniques. Various deep learning methods, including convolutional neural networks (CNN), recurrent neural networks (RNN), and long short-term memory (LSTM) [7, 17, 18, 19], have been applied to ECG signals for classification purposes. Among these models, CNN is the most common and effective model for arrhythmia detection today [20]. However, these methods require an intensive processing pipeline during application and require beat segmentation processes [21, 22]. Additionally, these processes require the data to be collected first and processed later for analysis, resulting in a large unprocessed signal volume, requiring high computational resources.

1.4 Objectives

Since its inception, ECG technology has undergone rapid advancement. As a result, home-based ECG monitoring using commercially available single-lead ECG devices with disposable sensor pads have recently gained popularity. However, interpreting ECG information still requires domain expertise in addition to the difficulties mentioned above in Sections 1.2 and 1.3. As such, a real-time explanation or visual interpretation of the ECG waveform is necessary for home-based users to analyse and caution users [8]. This paper proposes a novel application for arrhythmia detection through the object detection perspective using the state-of-the-art (SOTA) YOLOv8 model to address these issues. This model lets users see their detected ECG-beat classes and the prediction confidence. The model and results can then be stored using a cloud-based system for further diagnosis by experts.

2 Literature review

2.1 Arrhythmia and electrocardiograms

Any deviation from the normal sinus rhythm is considered a cardiac arrhythmia. Cardiac arrhythmia is broadly categorised by the origin of the abnormally occurring beat, medically known as an ectopic beat [23]. Supraventricular ectopic beats are arrhythmias with sinus, atrial or nodal origins, typically associated with abnormalities in the P wave. Ventricular ectopic beats are arrhythmias with ventricular origins, typically associated with abnormalities in the QRS complex. The associated waves' rate, regularity, presence, absence and morphology further classify specific arrhythmias within the category. Arrhythmias with atrial origin include atrial fibrillation, where P waves are absent, and atrial flutter, where P waves have a distinctive sawtooth appearance. Ventricular tachycardia is a ventricular ectopic beat where the QRS complex is distinctively wide, indicating a conduction delay in the ventricles, which may cause hypotension [7, 18, 24]. Most arrhythmia symptoms are non-fatal, including palpitations, dizziness, shortness of breath and fainting. However, if left untreated for prolonged periods, arrhythmia may become life-threatening, manifesting as heart failure and hypotension [2]. In addition, cardiac arrhythmias may be intermittent and absent in routine ECG measurements, making it difficult for in-clinic evaluations. Hence, long-term arrhythmia monitoring is crucial, as seen from the growing number of innovations and investigations for daily monitoring [25].

Therefore, practitioners use electrocardiograms (ECGs) to detect heart arrhythmias. The ECG is a non-invasive test that reflects the vector sum of the action potential in the heart throughout successive cardiac cycles as deflections on a voltage versus time graph [26, 27]. For example, on the ECG, the P wave is the vector sum of the action potential during atrial depolarisation. Likewise, the QRS complex reflects the action potential associated with the sequential depolarisation of the septum, ventricles and ventricular myocardium. Finally, the T wave occurs in conjunction with the ventricular myocardium's re-polarisation or relaxation. By analysing the temporal and morphological features of the waves, cardiologists identify deviations in the cardiac rhythm and make a cardiac diagnosis [28, 29, 30]. However, different heart conditions may present similarly on the ECG. Therefore, cardiologists refer to 12-lead ECGs and run further diagnostic tests.

2.2 You-Only-Look-Once (YOLO)

Object detection is a combination of image classification and object localisation tasks where a bounding box accompanies the predicted class of an object [31]. In general, object localisation algorithms are used to locate the presence of an object in the image and represent its location with a bounding box [32, 33]. Since Redmond et al. developed the first version of YOLO in 2015 [34], the object detection algorithm has revolutionised real-time object detection applications with its remarkable speed and accuracy. Researchers from [35, 36] deployed YOLO models to identify pedestrians, vehicles, and various obstacles for autonomous vehicle systems. Wu et al. deployed a face mask detector for COVID-19 prevention in public spaces [37]. In the application for real-time arrhythmia detection, Ji et al. proposed a Faster R-CNN to classify arrhythmia using bounding boxes [21]. Recently, Hwang et al. [22] proposed a 1D CNN YOLO-based model and replaced the bounding box with a bounding window.

The YOLO framework uses convolutional layers to predict the bounding boxes and the class probabilities of all objects depicted in an image [34]. Since the YOLO algorithm is a single-shot detector, it only looks at the image once. YOLO computes a confidence score for each bounding box by multiplying the probability of containing an object in an underlying grid with the Intersection over Union (IoU) between the ground truth and the predicted bounding box. Subsequently, non-maximum suppression (NMS) discards overlapping bounding boxes surrounding the detected object by selecting the box with the highest confidence [38, 39].

In this paper, we used the SOTA YOLOv8 released by Ultralytics in January 2023 [40]. The YOLOv8 is an anchor-free model which optimises the number of box predictions, decreasing the time taken for NMS. Figure 1 shows the overall model architecture of YOLOv8 nano (YOLOv8n), which mainly comprises three parts: Backbone, Neck, and Head.

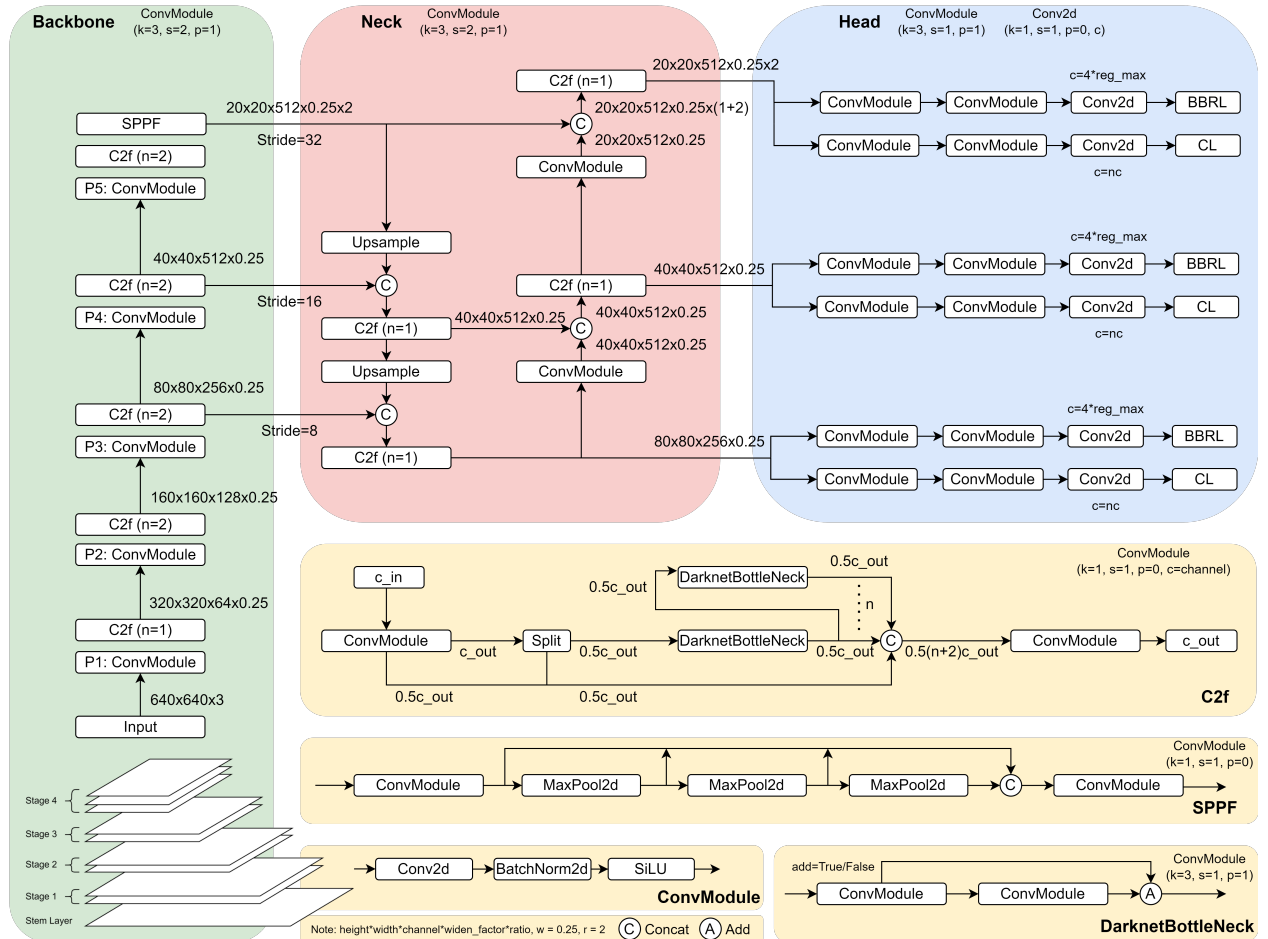


Figure 1: YOLOv8 nano (YOLOv8n) architecture, diagram based on [38, 40, 41].

2.2.1 Backbone

The YOLOv8 model uses a modified CSPDarknet53 backbone, which maintains the idea of Cross Stage Partial (CSP) modules but replaces C3 modules in YOLOv5 with C2f modules to keep the model lightweight for feature extraction. The C2f module (2 ConvModule and n BottleNeck) combines the C3 module (3 ConvModule and n BottleNeck) and ELAN [41], which was introduced in YOLOv7 [42]. As such, the C2f optimises using the DarknetBottleNeck structure to obtain richer gradient flow information. Each ConvModule is constructed by a convolutional layer, batch normalisation, and SiLU activation function, while n is a parameter of depth that controls the number of BottleNeck stacks by adding more layers, which varies between the model scales: YOLOv8n (nano) YOLOv8s (small), YOLOv8m (medium), YOLOv8l (large), and YOLOv8x (extra large). In Figure 1, we denote kernel, stride, and padding as k , s , and p , respectively.

2.2.2 Neck

The Backbone connects to the Neck at three different depths (Stage 2, Stage 3, Stage 4) to create a fusion of features obtained from the different layers of the network and passes that to the Head. The Neck comprises path aggregation network (PAN) [43] and feature pyramid network (FPN) [44] structures to prevent information loss due to multiple convolutions. First, the FPN structure upsamples the lower features from the top down, preventing them from losing less location information. In contrast, the PAN structure downsamples the features from the bottom up using ConvModule, allowing the top features to obtain more position information. Like the Backbone, the Neck’s FPN structure used the C2f module instead of the C3 module in YOLOv5 and removed ConvModule for direct upsampling. Using the PAN-FPN structure, the output channels of the Neck are equal to the output channels of the Backbone.

2.2.3 Head

The Head comprises three separate detection Heads, where each Head is decoupled to separate classification and regression tasks. The method of decoupling detection Heads was first presented in YOLOX and YOLOv6 [38]. As a result, the decoupled Head structure allows the YOLOv8 algorithm to be anchor-free by directly predicting the object’s centre and the bounding box offsets instead of predicting the centre offsets from the anchors. In addition, the anchor-free method uses task-aligned one-stage object detection (TOOD), which uses $t = s^\alpha \times u^\beta$, where s and u are the classifications and IoU scores, respectively; α and β are hyperparameter weights. With task-aligned positive sample matching, t aims to optimise the classification score and IoU for dynamic label assignment by selecting positive samples according to the weighted classification and regression scores.

2.2.4 Loss

The loss function is the weighted sum of classification and position-based regression losses. In the original implementation of YOLOv8, specifically the classification loss, the authors replaced Varifocal Focal Loss (VFL) with the standard Binary Cross Entropy (BCE) loss without explicitly incorporating any weighting scheme defined in Equation (1) below:

$$\text{BCE}_\ell = \ell_n(x, y) = y_n \cdot \log(\sigma(x_n)) + (1 - y_n) \cdot \log(1 - \sigma(x_n)), \quad (1)$$

where n is the number of samples, y_n is the ground truth value and x_n is the predicted value. The position-based regression losses comprise Distribution Focal Loss (DFL) and Complete IoU (CIoU) loss. Firstly, DFL aims to increase the probability around the target y by rapidly focusing on the target [45] and is defined as follows:

$$\text{DFL}(S_i, S_{i+1}) = -((y_{i+1} - y) \log(S_i) + (y - y_i) \log(S_{i+1})), \quad (2)$$

where y_i denotes the left side values of the label y , and y_{i+1} denotes the right side values of the label y , and $y = \sum_{i=0}^n P(y_i) y_i$ where $P(y_i)$ is implemented using softmax layer denoted by S_i . Finally, CIoU loss is defined as the aggregation of IoU, Distance IoU (DIoU) [46], and aspect ratio of the prediction and ground truth bounding boxes, as shown below:

$$\text{BBRL} = \text{CIoU}_\ell = 1 - \text{IoU} + \frac{d^2}{c^2} + \frac{v^2}{(1 - \text{IoU} + v)^2}, \quad (3)$$

$$v = \frac{4}{\pi^2} \left(\arctan \frac{w_{gt}}{h_{gt}} - \arctan \frac{w_p}{h_p} \right)^2, \quad (4)$$

where v is the measure of the consistency in aspect ratio with w_{gt} and h_{gt} are the width and height of the ground truth bounding box, w_p and h_p are the width and height of the predicted bounding box; d is the Euclidean distance between the centre point of the predicted and ground truth bounding boxes; c is the diagonal length of the smallest box enclosing both boxes [47]. Finally, the loss function in each decoupled head is expressed as:

$$f_\ell = \lambda_1 \text{BCE}_\ell + \lambda_2 \text{DFL} + \lambda_3 \text{BBRL}. \quad (5)$$

3 Materials and methods

3.1 Preprocessing

We acquired our dataset from the MIT/Beth Israel Hospital (MIT-BIH) Arrhythmia Database and extracted the verified QRS detection points as our ground truth annotation labels [48]. Subsequently, we transformed the dataset into full-length signals for each patient without resampling and denoising. Following the Association for the Advancement of Medical Instrumentation (AAMI) recommended practice, we removed four-paced beats and focused on distinguishing ventricular ectopic beats (VEBs) from non-ventricular ectopic beats [49, 50].

We included a ground-truth PhysioBank annotation symbol above each R-peak in the QRS complex for each patient’s full-length signal to annotate the dataset. We extracted ± 5 seconds around non-N waves to rebalance the MIT-BIH dataset, which is heavily skewed towards N-type heartbeats based on the AAMI EC57 standard shown in Table 1. Thus, our preprocessed dataset comprises images with multiple waves of varying numbers, and the model would not be biased towards detecting arrhythmia based on single QRS complexes but rather learn from the patterns in the continuous real-time data.

Table 1: MIT-BIH Arrhythmia dataset based on AAMI EC57 standard [50].

AAMI Classes	MIT-BIH Arrhythmia Beat Types
Normal (N)	Normal beat (NOR)
	Left bundle branch block (LBBB)
	Right bundle branch block (RBBB)
	Atrial escape beat (AE)
	Nodal escape beat (NE)
Supraventricular (S)	Atrial premature beat (AP)
	Aberrant atrial premature beat (aAP)
	Nodal premature beat (NP)
	Supraventricular premature beat (SP)
Ventricular (V)	Premature ventricular contraction (PVC)
	Ventricular escape beat (VE)
Fusion (F)	Fusion of normal & ventricular beat (FVN)
Unknown (Q)	Paced beat (/)
	Fusion of paced & normal (FPN)
	Unclassified (U)

3.2 Generating datasets

We annotated the preprocessed images with bounding boxes from the base of each R-peak. Each bounding box coordinate is saved in the standard normalised YOLO $xywh$ format, where x and y are the box’s centre coordinates. Following that, w and h are the width and height of the box encompassing the signal, respectively. Previous research by [51] showed this method effectively extracts RR intervals without filtering or making signal morphology assumptions. We resized our dataset to the YOLO benchmark resolution of 640 by 640 pixels, applied image-level augmentation of 70% grayscale, and bounding box level augmentations with $\pm 1^\circ$ rotation and noise of up to 1% of pixels [52]. Table 2 shows the final dataset comprising 42,417 images with an average of 4.343 annotations per image.

Table 2: Dataset.

AAMI Label	Number of Annotations	Composition
N	166556	45.352%
V	88322	24.049%
S	86861	23.652%
F	15182	4.134%
Q	10331	2.813%
Total	184205	100.000%

3.3 Loss modifications

3.3.1 Classification

To address the class imbalance shown in Table 2, we introduced dynamic inverse class frequency (DICF) in Equation (6) and (7) to emphasise underrepresented classes. This enhances our model’s ability to effectively learn from imbalanced data per batch and improve overall recall (sensitivity) results.

$$\text{DICF-BCE}_\ell = \ell_{n,c}(x, y) = -w_{n,c}[p_c y_{n,c} \cdot \log(\sigma(x_{n,c}) + (1 - y_{n,c})) \cdot \log(1 - \sigma(x_{n,c}))], \quad (6)$$

$$p_c = \log\left(\frac{n}{f(c)}\right), \quad (7)$$

where n is the total number of samples in the batch, c is the class labels of the annotations in the batch, $f(c)$ denotes the frequency of class c , and p_c is the dynamic logarithmic, inverse class frequency weights per batch. In Equation (7), $f(c)$ is also updated with respect to mosaic augmentation and albumentations. Finally, we applied a logarithmic transformation to reduce extreme values and prevent any single class from dominating the updated weights.

3.3.2 Bounding box regression

To improve the precision confidence in the case of low-quality sample data, we modify the BBRL in Equation (3). We accomplish this by utilising the SOTA Wise-IoU (WIoU) v3 loss to enhance precision for classes with less prominent features or low-quality annotation samples [53]. WIoU v3 is a two-layer attention-based (WIoU v1) with a dynamic nonmonotonic focusing mechanism (FM) that employs a wise gradient gain allocation approach using an outlier degree β of the predicted box. The WIoU v3 can be defined as the WIoU v1 with a nonmonotonic focal number expressed as follows:

$$\text{BBRL} = \text{WIoU v3}_\ell = \gamma \text{WIoU v1}_\ell, \quad (8)$$

$$\gamma = \frac{\beta}{\delta \alpha^{\beta-\delta}}, \quad (9)$$

$$\text{WIoU v1}_\ell = \text{IoU}_\ell \exp \frac{(x_p - x_{gt})^2 + (y_p - y_{gt})^2}{(W_g^2 + H_g^2)*}, \quad (10)$$

$$\text{IoU}_\ell = 1 - \text{IoU} = 1 - \frac{W_i H_i}{w_p h_p + w_{gt} h_{gt} - W_i H_i}, \quad (11)$$

where α and δ are hyperparameters with 1.9 and 3, respectively; β is the outlier degree; x_p and y_p are coordinates of the predicted bounding box; x_{gt} and y_{gt} are coordinates of the ground truth bounding box; W_i and H_i are the intersecting width and intersecting height between the predicted and ground truth bounding boxes; W_g and H_g are the width and height of the smallest enclosing bounding box; and $(*)$ denotes detaching from computational graph. During the early stages of training, to prevent low-quality anchor boxes from being dropped, we set a small momentum given by $m = 1 - \sqrt[n]{0.05}$, where t is the epoch when the lifting speed of AP slows significantly, and n is the number of batches. During later stages of training, when β is large, a small gradient gain is assigned to the low-quality anchor boxes to minimise harmful gradients. Hence, WIoU v3 allows the model to mask the influence of low-quality samples while focusing on normal-quality anchor boxes.

3.4 Training

In this study, we trained a YOLOv8 nano (YOLOv8n) model without using pretrained weights. The YOLOv8n model comprises only 3.2M parameters with 8.7B FLOPS. We chose this model for its prediction speeds for home-based real-time arrhythmia detection [40]. Furthermore, we can easily deploy our model on edge deployment devices such as the Raspberry Pi, on cloud services such as AWS SageMaker [54], and as an app on mobile phones [41].

Our model was trained on Tesla V100-SXM2-16GB on Google Colab. The model training parameters are as follows: (1) epochs: 200, (2) optimiser: SGD, (3) batch size: 64, (4) initial learning rate: 0.01, (5) momentum: 0.937, (6) weight decay: 0.0005, (7) IoU threshold: 0.7, (8) warmup epochs: 3.0, (9) warmup momentum: 0.8, (10) NMS: False, (11) cls gain: 2, (12) patience: 50, (13) WIoU_t: 15, (14) WIoU_n: 515. In addition, we applied mosaic augmentation during the first 50 epochs in training, as shown in Figure 2. We kept the class balance the same between the train, validation and test set, with 70% of the dataset used for training, 20% for validation and 10% for testing for holdout training.

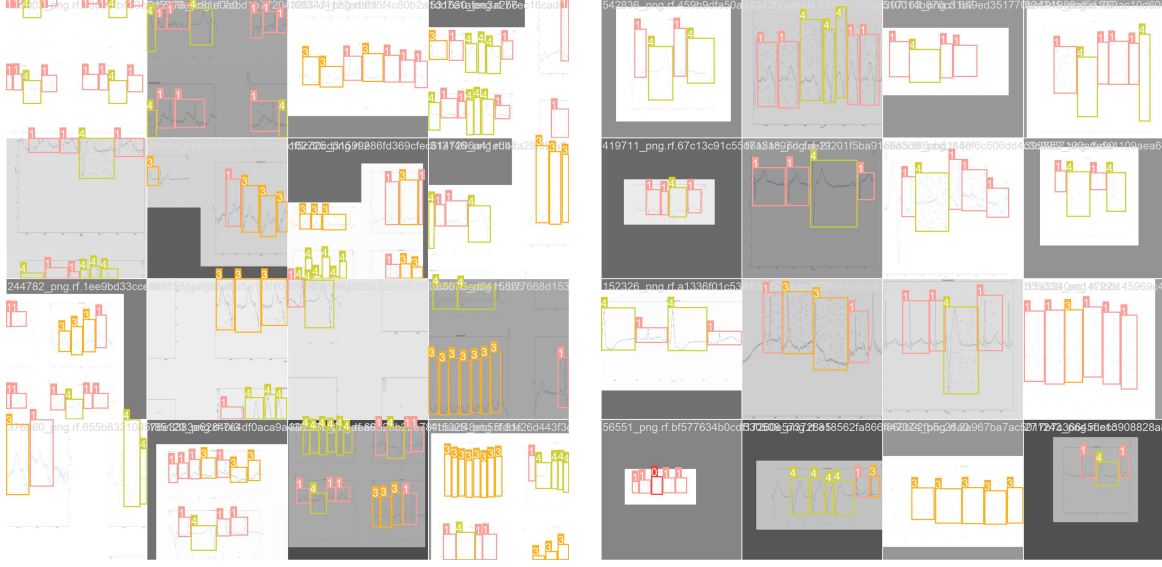


Figure 2: Mosaic augmentation for epochs 1-50 (left) and without mosaic augmentation for epochs 51-200 (right).

3.5 Evaluation methods

3.5.1 YOLO evaluation metrics

The performance of YOLO models is evaluated using the mean average precision (mAP) metrics. The mAP is commonly used as a measure of object detection performance, which considers both classification and localisation. In addition, the mAP also comprises the precision-recall (PR) area under the curve (AUC), multiple object categories (MOC), and IoU. To balance the PR trade-off, AUC is considered in calculating mAP. For each category, the PR curve is calculated by varying the confidence of the model’s prediction. Each class’s average precision (AP) is calculated individually by sampling the PR curve to classify and localise multiple ECG categories. Subsequently, the AP at different IoU thresholds is calculated (AP_{50-90}), and the mean of APs (mAP_{50-90}) for each IoU threshold is calculated across every class [38, 39]. Finally, the overall AP is computed by averaging the AP values calculated at each IoU threshold defined below:

$$mAP = \frac{1}{N} \sum_{i=1}^N AP_i, \quad (12)$$

where AP_i is the average precision of each class, and N is the total number of classes, including the background as a class type. During training, we set the IoU threshold as 0.7, where $IoU \geq 0.7$ is considered a true positive (TP), while an $IoU < 0.7$ is classified as a false positive (FP). When the model fails to detect an object where the ground truth exists, it is considered a false negative (FN). On the other hand, a true negative (TN) refers to the remaining image (background) where an object was not detected.

3.5.2 Performance metrics

To further evaluate our results, we defined our classification based on the four possible states: true positive (TP); true negative (TN); false positive (FP); false negative (FN). Following this, we reported our findings using accuracy, specificity, precision, and recall (sensitivity) and F_1 , which are standard metrics in arrhythmia classification models calculated as follows:

$$Accuracy = \frac{TP + TN}{TP + TN + FP + FN}, \quad (13)$$

$$Specificity = \frac{TN}{TN + FP}, \quad (14)$$

$$Precision = \frac{TP}{TP + FP}, \quad (15)$$

$$Recall = \frac{TP}{TP + FN}, \quad (16)$$

$$F_1 = 2 \frac{precision \cdot recall}{precision + recall}, \quad (17)$$

where accuracy represents the ratio of correct ECG beat classification to all beats; specificity quantifies our model’s ability to avoid false positives; precision represents the ratio of correct ECG classification out of all ECG beats predicted as the class; recall (sensitivity) quantifies our model’s ability to avoid false negatives by calculating the ratio of genuine arrhythmic ECG beats out of all beats that truly belong to that class, and F_1 is the traditional F-measure representing the harmonic mean of precision and recall to provide a balanced assessment of our model’s performance [21, 55].

In the results section, we presented our results based on ablation experiments using holdout training and 10-fold cross-validation. Firstly, we presented the ablation experimental model results for YOLOv8n, YOLOv8n + ICF, YOLOv8n + DICF, YOLOv8n + WIoU, and YOLOv8n + DICF + WIoU. ICF denotes the traditional inverse class frequency method where the weights are initialised once during training instead of updating weights per batch (DICF). Next, we select the best model and present the precision-recall and F_1 -confidence curves. Secondly, we conducted a 10-fold cross-validation test to ensure the legitimacy of the best model’s results, which resamples the dataset into ten equal parts for training and validation. We calculated the performance values of each fold and presented their means and standard deviations. In doing so, we could assess the variation in model performance. Finally, we tabulated the results metrics mentioned above in Equations (12)–(17).

4 Results

4.1 Ablation experiments

Table 3 shows the ablation experimental results using YOLO evaluation metrics for modification comparison and selecting the best model suited for our application.

Table 3: Ablation experimental results using YOLO evaluation metrics (3.5.1).

Model	Dataset	P	R	F1-Confidence	mAP ₅₀	mAP ₇₅	mAP _{50–90}
YOLOv8n	Val	0.981	0.974	0.977@0.455	0.991	0.978	0.874
	Test	0.980	0.972	0.976@0.447	0.991	0.978	0.874
YOLOv8n + ICF	Val	0.966	0.981	0.973@0.422	0.991	0.968	0.871
	Test	0.957	0.973	0.965@0.591	0.989	0.965	0.846
YOLOv8n + DICF	Val	0.971	0.981	0.976@0.595	0.991	0.978	0.867
	Test	0.969	0.978	0.973@0.606	0.991	0.978	0.867
YOLOv8n + WIoU	Val	0.983	0.975	0.979@0.443	0.992	0.979	0.874
	Test	0.982	0.972	0.977@0.447	0.992	0.979	0.874
YOLOv8n + DICF + WIoU	Val	0.974	0.981	0.977@0.603	0.992	0.979	0.868
	Test	0.971	0.977	0.977@0.589	0.992	0.979	0.869

We can observe the following:

1. When comparing the unmodified YOLOv8n with YOLOv8 + ICF, there is a significant improvement in recall score but poorer precision, resulting in a poorer mAP@50 score. However, this is expected since $p_c > 1$ for the minority classes (F and Q) to improve recall as a trade-off for precision;
2. There is a 0.518% increase in precision when using DICF instead of ICF, which suggests that DICF generally improves the model performance since a varying number of object classes exist in each iteration during training. When using ICF, extreme values exist due to the dataset imbalance in Table 2;
3. When comparing YOLOv8n with YOLOv8 + WIoU, there is an improvement in mAP@50 and mAP@75 scores, which are consistent to [53];
4. When comparing YOLOv8n + DICF + WIoU with the other models, it achieved the best scores for Recall and F1-Confidence. Moreover, WIoU improved the precision by 0.309% against YOLOv8n + DICF and improved confidence among the tested models. Therefore, we select YOLOv8n + DICF + WIoU for further evaluation.

Figure 3 shows the model training curve for 200 epochs, where we can see a significant drop across the three losses after closing mosaic augmentations at epoch 51. Figure 4 shows the Precision-Recall and F1-Confidence curves, respectively. F and Q achieved the best scores with 0.993 at mAP@0.5 and the best F1 scores with the highest confidence. Figure 5 is the normalised confusion matrix. The per-class performance metrics results for the validation and test sets are shown in Table 4. The average classification accuracy for validation and test is $99.5\% \pm 0.4\%$ and $99.4\% \pm 0.5\%$, respectively.

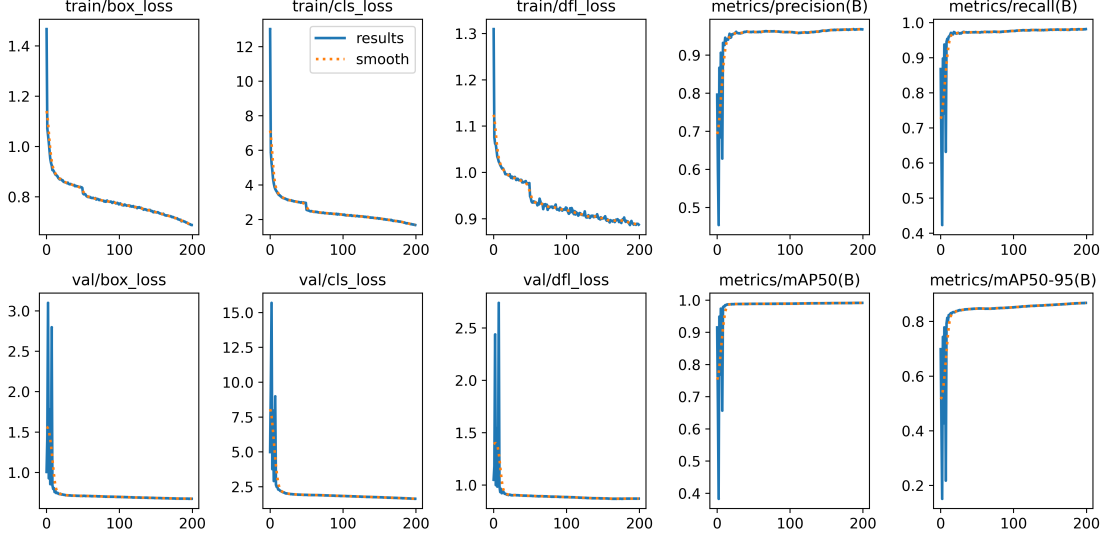


Figure 3: Training/validation results.

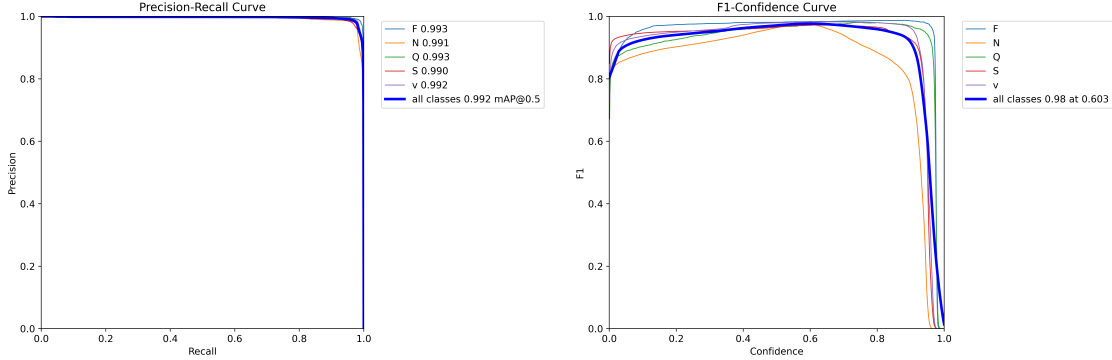


Figure 4: Precision-Recall curve (left) and F1-Confidence curve (right).

From Figure 3, both classification and regression losses when mosaic augmentation was used were shown to have prematurely converged. However, when we closed mosaic augmentation after 50 epochs, the losses continued to improve even beyond the 200th epoch and patience was never activated until we stopped training. This suggests that our model could be further optimised beyond the 200th epoch with WIoU v3. Nonetheless, mosaic augmentation seems to be appropriate for warming up in the early stages of training. A sample prediction is shown in Figure 6, where our model correctly predicts the beat classes with more than 90% confidence. Furthermore, we can see that the bounding boxes are precisely drawn without major overlapping. Therefore, our loss-modified YOLOv8n model can effectively segment between different heartbeats without the need for computationally expensive beat-segmentation processes.

4.2 K-fold cross-validation results

To ensure the legitimacy of our training results, we present our 10-fold cross-validation results for YOLOv8n + DICEF + WIoU in Tables 5 and 6. We observe that the model performs amicably with both YOLO evaluation metrics (3.5.1) and performance metrics (3.5.2). Based on YOLO evaluation metrics in Table 5, our model achieved averages similar to the holdout training test set, where (a) Precision: 0.969 ± 0.002 , (b) Recall: 0.979 ± 0.002 , (c) F1-Confidence:

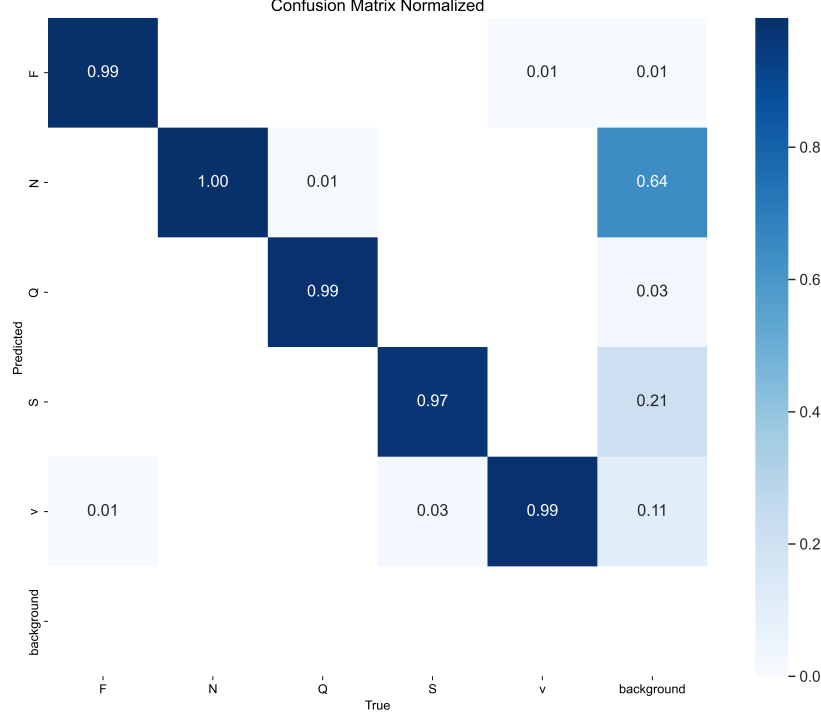


Figure 5: Normalised confusion matrix (YOLOv8n + DICE + WIoU v3).

Table 4: Performance metrics without background classes (3.5.2).

Classes	Dataset	Performance metrics (± 0.001)				
		Accuracy	Specificity	Precision	Recall	F1
N	Val	0.998	0.999	0.998	0.998	0.998
	Test	0.998	0.998	0.998	0.998	0.998
V	Val	0.989	0.989	0.966	0.988	0.977
	Test	0.988	0.988	0.963	0.987	0.975
S	Val	0.991	0.998	0.994	0.970	0.982
	Test	0.990	0.998	0.993	0.965	0.978
F	Val	0.998	0.999	0.975	0.984	0.979
	Test	0.998	0.999	0.972	0.978	0.975
Q	Val	0.999	0.999	0.990	0.995	0.992
	Test	0.998	0.999	0.972	0.978	0.975
Average	Val	0.995	0.997	0.985	0.987	0.986
	Test	0.994	0.996	0.980	0.981	0.980
Standard deviation	Val	0.004	0.004	0.012	0.010	0.008
	Test	0.005	0.004	0.014	0.012	0.009

0.974@0.601 \pm 0.002@0.004, (d) mAP_{50} : 0.992 \pm 0.001, and (e) mAP_{50-90} : 0.870 \pm 0.003. Each fold has a low standard deviation, indicating consistent model performance.

Regarding the classification performance metrics, the averaged 10-fold cross-validation results are consistent to the holdout validation results presented in Table 4. Our model achieved an average of (a) 99.5% \pm 0.1% accuracy, (b) 99.7% \pm 0.1% specificity, (c) 98.5% \pm 0.3% precision, (d) 98.7% \pm 0.2% recall, and (e) 98.6% \pm 0.2% F1 score. Overall, given the low standard deviations in both results, our model has a generalised understanding of different ECG waveforms and could adapt to new data.

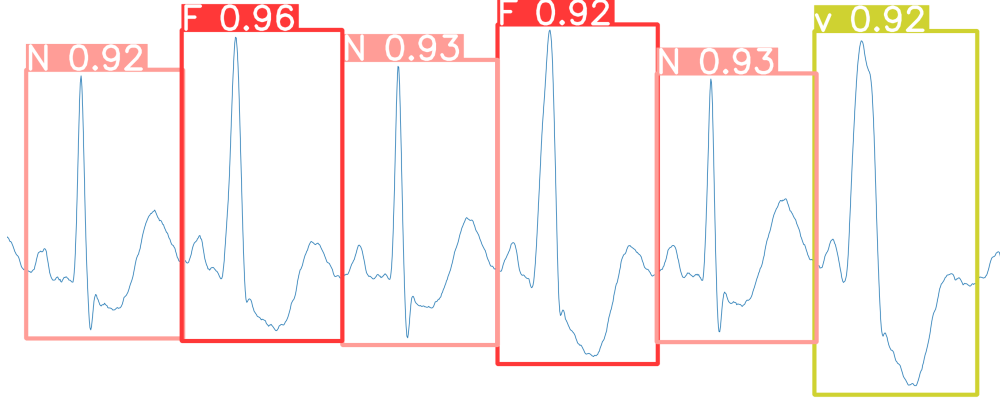


Figure 6: Test set prediction result.

Table 5: Results of 10-fold cross-validation (± 0.001) using YOLO evaluation metrics (3.5.1).

Fold	P	R	F1-Confidence	mAP ₅₀	mAP ₅₀₋₉₀
Fold 1	0.970	0.983	0.976@0.604	0.992	0.868
Fold 2	0.971	0.981	0.976@0.602	0.992	0.868
Fold 3	0.971	0.978	0.974@0.595	0.992	0.866
Fold 4	0.967	0.981	0.974@0.606	0.992	0.869
Fold 5	0.971	0.978	0.974@0.595	0.992	0.867
Fold 6	0.966	0.978	0.972@0.605	0.991	0.870
Fold 7	0.967	0.976	0.971@0.600	0.991	0.869
Fold 8	0.973	0.978	0.975@0.597	0.992	0.870
Fold 9	0.968	0.979	0.973@0.602	0.991	0.869
Fold 10	0.967	0.979	0.973@0.606	0.992	0.879
Average	0.969	0.979	0.974@0.601	0.992	0.870
Standard Deviation	0.002	0.002	0.002@0.004	0.001	0.003

Table 6: Results of 10-fold cross-validation (± 0.001) using performance metrics (3.5.2).

Fold	Accuracy	Specificity	Precision	Recall	F1
Fold 1	0.994	0.996	0.979	0.985	0.982
Fold 2	0.997	0.998	0.988	0.991	0.989
Fold 3	0.996	0.998	0.987	0.989	0.988
Fold 4	0.995	0.997	0.985	0.987	0.986
Fold 5	0.996	0.997	0.986	0.989	0.987
Fold 6	0.995	0.997	0.983	0.987	0.985
Fold 7	0.994	0.996	0.981	0.984	0.982
Fold 8	0.995	0.997	0.987	0.987	0.987
Fold 9	0.996	0.997	0.988	0.986	0.987
Fold 10	0.995	0.997	0.984	0.987	0.985
Average	0.995	0.997	0.985	0.987	0.986
Standard Deviation	0.001	0.001	0.003	0.002	0.002

5 Discussion

5.1 Results

Overall, our model achieved a state-of-the-art arrhythmia classification accuracy of 99.5% in the lens of object detection. Moreover, our model performs best in detecting minority classes Q and F, with the highest overall F1-Confidence scores. We improved the recall and F1-Confidence using DICF and improved the precision-confidence and mAP scores using

WIoU v3. Several class balancing techniques can be further explored in future studies, such as resampling the dataset to under-represent N beats and over-represent F and Q beats or dynamically updating the effective number of object class [56]. We could also apply various object detection augmentation techniques, but this has to be done with consideration and may require domain knowledge.

Interestingly, we observe that in Figure 5, there is a significant number of false predictions in the background class. In other words, our model falsely detected an arrhythmia class when it was the image’s background. This is expected since each beat is only annotated if the next R-peak exists to simulate continuous signals during real-time detection. Hence, the model could correctly detect an object class but it is penalised for detecting it prematurely. Furthermore, based on our dataset generation and annotation, wherein most of the ground truth box is occupied by the white background, our model could be led to draw a correlation, resulting in poorer YOLO evaluation metrics scores.

Table 7 shows our model compared with other related work classifying based on the AAMI convention using deep learning detector or CNN models. Our model achieved SOTA results using an object detection-based method to detect arrhythmia. Specifically, our model performed best across both YOLO evaluation metrics (3.5.1) in mAP₅₀ score and detection speeds and classification-based performance metrics (3.5.2). While Ji et al. [21] and Hwang et al. [22] did not elaborate on aspects comprising total detection time, we report their findings in Table 7. For our model, we define the total detection time per frame to include preprocessing, inference, loss, and post-processing, where the inference took 0.7 ms alone. In summary, our loss-modified arrhythmia detector (YOLOv8n + DICF + WIoU v3) achieved 0.992 mAP@50 with an input frame size of 640 pixels with a speed of 430 FPS on Tesla V100-SXM2-16GB.

Table 7: Summary and comparison with related work.

Work	Model	YOLO evaluation metrics (3.5.1) & performance metrics (3.5.2) \pm 0.001						
		Accuracy	Specificity	Precision	Recall	F1	mAP ₅₀	Time
<i>Kiranyaz et al.</i> [57]	1D CNN	0.964	0.995	0.688	0.651	0.669	-	- s
<i>Xiao et al.</i> [58]	2D CNN	0.984	0.978	0.659	0.721	0.681	-	- s
<i>Ji et al.</i> [21]	Faster RCNN	0.992	0.994	0.959	0.971	0.965	-	0.025 s
<i>Hwang et al.</i> [22]	1D YOLO	-	-	0.976	0.954	0.964	0.960	0.030 s
YOLOv8n (Val)	YOLOv8n	0.995	0.997	0.985	0.987	0.986	0.992	0.002 s
YOLOv8n (Test)	YOLOv8n	0.994	0.996	0.980	0.981	0.980	0.992	0.002 s

Similar to Ji et al. [21], our model also poses the same benefits where we did not require feature extraction and resampling of the original ECG signal. Given the large dataset, we agree with the authors that the model requires long training hours and GPU. However, our work differs in that no denoising and beat segmentation was implemented, thereby simplifying the deep learning pipeline. Notably, using the YOLO family of detectors, our ultimate goal is to balance real-time performance without compromising the detection results. We may improve the results (3.5.2 and 3.5.1) by using larger versions of the YOLOv8, but at a trade-off in real-time performance. We used the lightest architecture (YOLOv8n) to demonstrate deployability in Section 5.2. Future studies could include using YOLOv8s (small), YOLOv8m (medium), YOLOv8l (large), and YOLOv8x (extra large), different frame sizes, and exploring the trade-off between speed and accuracy.

5.2 Real-time detection

We implemented real-time arrhythmia detection with our trained model using commercial devices such as the Sparkfun AD8232. The AD8232 is a cardiograph hardware for biopotential measurement applications in noisy environments. Research from Prasad and Kavanashree [59] successfully implemented the AD8232 with IoT capabilities to monitor the ECG of patients remotely and extended their study to include 12-lead ECG acquisition. We plot the unfiltered serial data from the AD8232 onto a white canvas of 640 by 640 pixels to demonstrate real-time deployability. Without using any digital filters, we fed the continuously updating canvas as the source into the trained YOLOv8n model. Figure 7 shows a sampled signal captured in real-time using commercially available electrodes from the AD8232 through a NodeMCU module as the cost-effective Internet of Things (IoT) firmware. Here, we used 640 by 640 pixels to reduce preprocessing time.

We may require digital filtering for noisy at-home environments, and filtering may result in losing signal features, which may not be ideal for arrhythmia detection. Moreover, the additional noise removal computation may increase end-to-end inference time. Therefore, further research on preprocessing real-time ECG signals could be conducted to improve detection results. This paper could be extended to optimise and improve the latency of the real-time detection system.

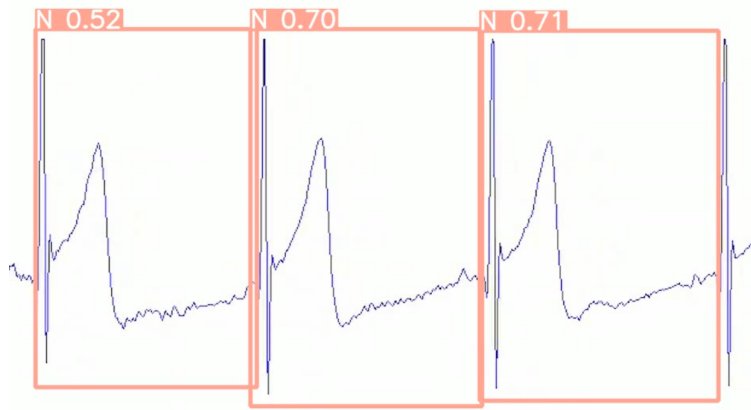


Figure 7: Example of real-time detection.

Additionally, we could deploy the model using Amazon S3 and Amazon SageMaker endpoints to store data and enhance scalability. Hence, we can further apply this research to deploy and develop a fully wireless and portable product. We could also actualise the YOLOv8 model into the cloud system for ambulatory applications, and it is a significant step towards developing real-time applications of Explainable Artificial Intelligence (XAI) models in this domain [8].

6 Conclusion

In this paper, we implemented a loss-modified YOLOv8 model to enable users to interpret ECG signals in home-based environments. We trained the original YOLOv8n model using the MIT-BIH dataset and improved the results of the minority classes using dynamic inverse class frequency and Wise IoU. Our model achieved state-of-the-art results for arrhythmia detection in the lens of object detection, with an average accuracy of 99.4% and 99.5% for validation and test sets, respectively. Moreover, these results are consistent with 10-fold cross-validation tests. Therefore, the YOLOv8 model is suitable for the application of real-time arrhythmia detection. While the field is relatively young, we have demonstrated the potential of YOLOv8 as a real-time application for analysing abnormal ECG signals to allow continuous monitoring.

Acknowledgements

We thank Associate Professor Chua Kian Jon Ernest and Mr Nelliyan Karuppiyah for guiding us and providing valuable advice. We extend our gratitude to Professor U Rajendra Acharya for his review of our work.

References

- [1] Centers for Disease Control and Prevention, National Center for Health Statistics. About multiple cause of death, 1999–2019, 2021. data retrieved from CDC WONDER Online Database website, <https://wonder.cdc.gov/mcd-icd10.html>.
- [2] Mohamed Hammad, Rajesh NVPS Kandala, Amira Abdelatey, Moloud Abdar, Mariam Zomorodi-Moghadam, Ru San Tan, U Rajendra Acharya, Joanna Pławiak, Ryszard Tadeusiewicz, Vladimir Makarenkov, et al. Automated detection of shockable ecg signals: A review. *Information Sciences*, 571:580–604, 2021.
- [3] Bartosz Grabowski, Przemysław Głomb, Wojciech Masarczyk, Paweł Pławiak, Özal Yıldırım, U Rajendra Acharya, and Ru-San Tan. Classification and self-supervised regression of arrhythmic ecg signals using convolutional neural networks. *arXiv preprint arXiv:2210.14253*, 2022.
- [4] Sherin M Mathews, Chandra Kambhamettu, and Kenneth E Barner. A novel application of deep learning for single-lead ecg classification. *Computers in biology and medicine*, 99:53–62, 2018.
- [5] Eduardo José da S Luz, William Robson Schwartz, Guillermo Cámara-Chávez, and David Menotti. Ecg-based heartbeat classification for arrhythmia detection: A survey. *Computer methods and programs in biomedicine*, 127:144–164, 2016.
- [6] Leon Glass. Cardiac oscillations and arrhythmia analysis. *Complex Systems Science in Biomedicine*, pages 409–422, 2006.
- [7] The-Hanh Pham, Vinitha Sree, John Mapes, Sumeet Dua, Oh Shu Lih, Joel EW Koh, Edward J Ciaccio, and U Rajendra Acharya. A novel machine learning framework for automated detection of arrhythmias in ecg segments. *Journal of Ambient Intelligence and Humanized Computing*, pages 1–18, 2021.
- [8] Hui Wen Loh, Chui Ping Ooi, Silvia Seoni, Prabal Datta Barua, Filippo Molinari, and U Rajendra Acharya. Application of explainable artificial intelligence for healthcare: A systematic review of the last decade (2011–2022). *Computer Methods and Programs in Biomedicine*, page 107161, 2022.
- [9] Martin Lagerholm, Carsten Peterson, Guido Braccini, Lars Edenbrandt, and Leif Sörnmo. Clustering ecg complexes using hermite functions and self-organizing maps. *IEEE Transactions on Biomedical Engineering*, 47(7):838–848, 2000.
- [10] Lotfi Senhadji, G Carrault, JJ Bellanger, and Gianfranco Passariello. Comparing wavelet transforms for recognizing cardiac patterns. *IEEE Engineering in Medicine and Biology Magazine*, 14(2):167–173, 1995.
- [11] Yu Hen Hu, Surekha Palreddy, and Willis J Tompkins. A patient-adaptable ecg beat classifier using a mixture of experts approach. *IEEE transactions on biomedical engineering*, 44(9):891–900, 1997.
- [12] J Millet-Roig, R Ventura-Galiano, FJ Chorro-Gasco, and A Cebrian. Support vector machine for arrhythmia discrimination with wavelet transform-based feature selection. In *Computers in Cardiology 2000. Vol. 27 (Cat. 00CH37163)*, pages 407–410. IEEE, 2000.
- [13] Yasin Kaya and Hüseyin Pehlivan. Classification of premature ventricular contraction in ecg. *International Journal of Advanced Computer Science and Applications*, 6(7), 2015.
- [14] I Christov, I Jekova, and G Bortolan. Premature ventricular contraction classification by the kth nearest-neighbours rule. *Physiological measurement*, 26(1):123, 2005.
- [15] Santanu Sahoo, Asit Subudhi, Manasa Dash, and Sukanta Sabut. Automatic classification of cardiac arrhythmias based on hybrid features and decision tree algorithm. *International Journal of Automation and Computing*, 17(4):551–561, 2020.
- [16] Yann LeCun, Yoshua Bengio, and Geoffrey Hinton. Deep learning. *nature*, 521(7553):436–444, 2015.
- [17] Shu Lih Oh, Eddie YK Ng, Ru San Tan, and U Rajendra Acharya. Automated diagnosis of arrhythmia using combination of cnn and lstm techniques with variable length heart beats. *Computers in biology and medicine*, 102:278–287, 2018.
- [18] Özal Yıldırım, Paweł Pławiak, Ru-San Tan, and U Rajendra Acharya. Arrhythmia detection using deep convolutional neural network with long duration ecg signals. *Computers in biology and medicine*, 102:411–420, 2018.
- [19] U Rajendra Acharya, Shu Lih Oh, Yuki Hagiwara, Jen Hong Tan, Muhammad Adam, Arkadiusz Gertych, and Ru San Tan. A deep convolutional neural network model to classify heartbeats. *Computers in biology and medicine*, 89:389–396, 2017.
- [20] Zahra Ebrahimi, Mohammad Loni, Masoud Daneshtalab, and Arash Gharehbaghi. A review on deep learning methods for ecg arrhythmia classification. *Expert Systems with Applications: X*, 7:100033, 2020.

- [21] Yinsheng Ji, Sen Zhang, and Wendong Xiao. Electrocardiogram classification based on faster regions with convolutional neural network. *Sensors*, 19(11):2558, 2019.
- [22] Won Hee Hwang, Chan Hee Jeong, Dong Hyun Hwang, and Young Chang Jo. Automatic detection of arrhythmias using a yolo-based network with long-duration ecg signals. *Engineering Proceedings*, 2(1):84, 2020.
- [23] F.J. Dowd. Arrhythmias. In *Reference Module in Biomedical Sciences*. Elsevier, 2014.
- [24] Roshan Joy Martis, U Rajendra Acharya, Choo Min Lim, KM Mandana, Ajoy K Ray, and Chandan Chakraborty. Application of higher order cumulant features for cardiac health diagnosis using ecg signals. *International journal of neural systems*, 23(04):1350014, 2013.
- [25] Zengding Liu, Bin Zhou, Zhiming Jiang, Xi Chen, Ye Li, Min Tang, and Fen Miao. Multiclass arrhythmia detection and classification from photoplethysmography signals using a deep convolutional neural network. *Journal of the American Heart Association*, 11(7):e023555, 2022.
- [26] Adam Feather, David Randall, and Mona Waterhouse. *Kumar and clark's clinical medicine E-Book*. Elsevier Health Sciences, 2020.
- [27] Leonard S Lilly. *Pathophysiology of heart disease: a collaborative project of medical students and faculty*. Lippincott Williams & Wilkins, 2012.
- [28] Marius Reto Bigler, Patrick Zimmermann, Athanasios Papadis, and Christian Seiler. Accuracy of intracoronary ecg parameters for myocardial ischemia detection. *Journal of electrocardiology*, 64:50–57, 2021.
- [29] Eedara Prabhakararao and Samarendra Dandapat. Myocardial infarction severity stages classification from ecg signals using attentional recurrent neural network. *IEEE Sensors Journal*, 20(15):8711–8720, 2020.
- [30] Pavel Lyakhov, Mariya Kiladze, and Ulyana Lyakhova. System for neural network determination of atrial fibrillation on ecg signals with wavelet-based preprocessing. *Applied Sciences*, 11(16):7213, 2021.
- [31] Youzi Xiao, Zhiqiang Tian, Jiachen Yu, Yinshu Zhang, Shuai Liu, Shaoyi Du, and Xuguang Lan. A review of object detection based on deep learning. *Multimedia Tools and Applications*, 79:23729–23791, 2020.
- [32] Peiyuan Jiang, Daji Ergu, Fangyao Liu, Ying Cai, and Bo Ma. A review of yolo algorithm developments. *Procedia Computer Science*, 199:1066–1073, 2022.
- [33] Liquan Zhao and Shuaiyang Li. Object detection algorithm based on improved yolov3. *Electronics*, 9(3):537, 2020.
- [34] Joseph Redmon, Santosh Divvala, Ross Girshick, and Ali Farhadi. You only look once: Unified, real-time object detection. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 779–788, 2016.
- [35] Nurin Mirza Afiqah Andrie Dazlee, Syamimi Abdul Khalil, Shuzlina Abdul-Rahman, and Sofianita Mutalib. Object detection for autonomous vehicles with sensor-based technology using yolo. *International Journal of Intelligent Systems and Applications in Engineering*, 10(1):129–134, 2022.
- [36] Wenbo Lan, Jianwu Dang, Yangping Wang, and Song Wang. Pedestrian detection based on yolo network model. In *2018 IEEE international conference on mechatronics and automation (ICMA)*, pages 1547–1551. IEEE, 2018.
- [37] Peishu Wu, Han Li, Nianyin Zeng, and Fengping Li. Fmd-yolo: An efficient face mask detection method for covid-19 prevention and control in public. *Image and vision computing*, 117:104341, 2022.
- [38] Juan Terven and Diana Cordova-Esparza. A comprehensive review of yolo: From yolov1 to yolov8 and beyond. *arXiv preprint arXiv:2304.00501*, 2023.
- [39] Tausif Diwan, G Anirudh, and Jitendra V Tembhurne. Object detection using yolo: Challenges, architectural successors, datasets and applications. *Multimedia Tools and Applications*, pages 1–33, 2022.
- [40] Glenn Jocher, Ayush Chaurasia, and Jing Qiu. YOLO by Ultralytics, January 2023.
- [41] Rui-Yang Ju and Weiming Cai. Fracture detection in pediatric wrist trauma x-ray images using yolov8 algorithm. *arXiv preprint arXiv:2304.05071*, 2023.
- [42] Chien-Yao Wang, Alexey Bochkovskiy, and Hong-Yuan Mark Liao. YOLOv7: Trainable bag-of-freebies sets new state-of-the-art for real-time object detectors. *arXiv preprint arXiv:2207.02696*, 2022.
- [43] Shu Liu, Lu Qi, Haifang Qin, Jianping Shi, and Jiaya Jia. Path aggregation network for instance segmentation. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 8759–8768, 2018.
- [44] Tsung-Yi Lin, Piotr Dollár, Ross Girshick, Kaiming He, Bharath Hariharan, and Serge Belongie. Feature pyramid networks for object detection. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 2117–2125, 2017.

- [45] Xiang Li, Wenhai Wang, Lijun Wu, Shuo Chen, Xiaolin Hu, Jun Li, Jinhui Tang, and Jian Yang. Generalized focal loss: Learning qualified and distributed bounding boxes for dense object detection. *Advances in Neural Information Processing Systems*, 33:21002–21012, 2020.
- [46] Zhaohui Zheng, Ping Wang, Wei Liu, Jinze Li, Rongguang Ye, and Dongwei Ren. Distance-iou loss: Faster and better learning for bounding box regression. In *Proceedings of the AAAI conference on artificial intelligence*, volume 34, pages 12993–13000, 2020.
- [47] Zhaohui Zheng, Ping Wang, Dongwei Ren, Wei Liu, Rongguang Ye, Qinghua Hu, and Wangmeng Zuo. Enhancing geometric factors in model learning and inference for object detection and instance segmentation. *IEEE transactions on cybernetics*, 52(8):8574–8586, 2021.
- [48] George B Moody, Roger G Mark, and Ary L Goldberger. Physionet: a web-based resource for the study of physiologic signals. *IEEE Engineering in Medicine and Biology Magazine*, 20(3):70–75, 2001.
- [49] Jinyuan He, Jia Rong, Le Sun, Hua Wang, Yanchun Zhang, and Jiangang Ma. A framework for cardiac arrhythmia detection from iot-based ecgs. *World Wide Web*, 23:2835–2850, 2020.
- [50] AAMI ECAR. Recommended practice for testing and reporting performance results of ventricular arrhythmia detection algorithms. *Association for the Advancement of Medical Instrumentation*, 69, 1987.
- [51] Mohammad Kachuee, Shayan Fazeli, and Majid Sarrafzadeh. Ecg heartbeat classification: A deep transferable representation. In *2018 IEEE international conference on healthcare informatics (ICHI)*, pages 443–444. IEEE, 2018.
- [52] Barret Zoph, Ekin D Cubuk, Golnaz Ghiasi, Tsung-Yi Lin, Jonathon Shlens, and Quoc V Le. Learning data augmentation strategies for object detection. In *Computer Vision–ECCV 2020: 16th European Conference, Glasgow, UK, August 23–28, 2020, Proceedings, Part XXVII 16*, pages 566–583. Springer, 2020.
- [53] Zanjia Tong, Yuhang Chen, Zewei Xu, and Rong Yu. Wise-iou: Bounding box regression loss with dynamic focusing mechanism. *arXiv preprint arXiv:2301.10051*, 2023.
- [54] Shihao Zhang, Hekai Yang, Chunhua Yang, Wenxia Yuan, Xinghui Li, Xinghua Wang, Yinsong Zhang, Xiaobo Cai, Yubo Sheng, Xiujuan Deng, et al. Edge device detection of tea leaves with one bud and two leaves based on shufflenetv2-yolov5-lite-e. *Agronomy*, 13(2):577, 2023.
- [55] Zahra Ebrahimi, Mohammad Loni, Masoud Daneshtalab, and Arash Gharehbaghi. A review on deep learning methods for ecg arrhythmia classification. *Expert Systems with Applications: X*, 7:100033, 2020.
- [56] Trong Huy Phan and Kazuma Yamamoto. Resolving class imbalance in object detection with weighted cross entropy losses. *arXiv preprint arXiv:2006.01413*, 2020.
- [57] Serkan Kiranyaz, Turker Ince, and Moncef Gabbouj. Real-time patient-specific ecg classification by 1-d convolutional neural networks. *IEEE Transactions on Biomedical Engineering*, 63(3):664–675, 2015.
- [58] Xiaolong Zhai and Chung Tin. Automated ecg classification using dual heartbeat coupling based on convolutional neural network. *IEEE Access*, 6:27465–27472, 2018.
- [59] Anitha S Prasad and N Kavanashree. Ecg monitoring system using ad8232 sensor. In *2019 International Conference on Communication and Electronics Systems (ICCES)*, pages 976–980. IEEE, 2019.