Efficient Arrhythmia Detection Using Progressive Resolution Shrinking

Tavonput Luangphasy Xinghui Zhao
School of Engineering and Computer Science
Washington State University Vancouver
Email: {tavonput.luangphasy, x.zhao}@wsu.edu

2024 IEEE International Conference on Big Data (BigData) | 979-8-3503-6248-0/24/$31.00 ©2024 IEEE | DOI: 10.1109/BigData62323.2024.10825678

***Abstract*—Cardiovascular diseases, such as heart attack and congestive heart failure, are the leading cause of death in the United States and worldwide. The current medical practice for diagnosing cardiovascular diseases is not suitable for long-term, out-of-hospital use. A key to long-term, at-home cardiac care is the ability to provide continuous monitoring, and detect abnormal cardiac rhythms, i.e., arrhythmia, in real-time. Various big data and deep learning based approaches have been developed to analyze electrocardiogram data to identify arrhythmia conditions. However, most existing studies only focus on the accuracy of arrhythmia classification, instead of runtime performance of the workflow, which is critical for real-time detection. In this paper, we propose progressive resolution shrinking, a new method for supporting efficient execution of deep learning models for arrhythmia detection, without compromising the detection accuracy. Specifically, we explored multidimensional methods in reducing the amount of information needed for the learning task, and developed a new training method to leverage the advantage of reduced resolution. We have evaluated this approach using real electrocardiogram data, and the experimental results show that it effectively improves the efficiency of arrhythmia detection while preserving high accuracy. We expect this approach will pave the way for real-time arrhythmia detection on resource-constrained wearable devices.**

***Index Terms*—Arrhythmia Detection; ECG Data Analytics; Big Data; Deep Learning; Convolutional Neural Networks**

I. Introduction

Arrhythmia, defined by the National Heart, Lung and Blood Institute (NHLBI) as irregular rate or rhythm of the heart, e.g., the heart beat is too fast, too slow, or out of rhythm. According to a study carried out by the American Heart Association [1], arrhythmia is listed as the primary cause of death for over 50,000 people, and it is also listed as one of the reasons leading to death for over 550,000 people. In addition, atrial fibrillation, a type of arrhythmia that is linked to strokes, is expected to be diagnosed in 2.6 million Americans in 2030. The most common way to diagnose arrhythmia is through examining electrocardiogram (ECG) recordings generated by traditional ECG machines [2], which are located in a hospital setting. Examining ECG recordings are usually done by medical professionals. However, the ever-increasing cost of in-hospital care has become a significant challenge. It is estimated that the total cost of diagnosing arrhythmia in 2012 is about $20 billion and is expect to raise to $53 billion in 2030 [3]. Patients with less-severe, chronic conditions would prefer an in-home, on-going monitoring system, which requires continuous, near real-time arrhythmia detection.

Many heart conditions exhibit biophysical signals that can be detected before acute, irreversible damage is sustained by the heart or before more extensive damage is incurred, thereby reducing adverse health events. For example, various types of arrhythmia, can be monitored using home-based/mobile health (m-Health) monitoring platforms based on single-mode sensing (i.e., electrocardiogram (ECG) [4]–[6]). In our prior work [7], we designed a rechargeable, compact, and wearable heart health monitor that acquires real-time ECG from the human body. Recorded information can be transferred wirelessly to the user's phone or computer, where a machine learning model can be used to monitor biophysical data and identify anomalies. The relatively inexpensive cost of this device enables home monitoring in many cases, if real-time ECG data processing can be implemented.

Existing studies on applying big data and machine learning technologies on ECG data for arrhythmic detection often focus on the learning performance, such as accuracy, precision, etc., instead of the potentials of supporting real-time processing. Some early work on real-time analysis of ECG data uses 1D representation coupled with time-series based processing to achieve higher computational performance [8]–[10]. However, representing ECG data in 1D loses the rich 2D features, and may hinder the potential of integrating data analytics technologies with traditional diagnostic approaches. In this paper, we present a different approach in supporting efficient arrhythmia detection using convolutional neural networks, aiming to preserve the 2D features and also satisfy the real-time processing needs. Specifically, we have investigated strategies for reducing the amount of data needed to perform the arrhythmia classification, and developed a novel training method to support the learning task with progressive resolution reduction. We have carried out extensive experiments to evaluate the proposed approach, and the results show that it demonstrates stable performance with much less demand on computational resources. This research provides potentials for faster diagnostic of arrhythmia and can be deployed in a medical environment without powerful hardware.

II. Related Work

Cardiovascular diseases are the leading cause of death both in the U.S. and worldwide. The direct domestic medical costs associated with congestive heart failure (CHF) are expected to reach $53 billion by 2030 [11], with the majority of costs

979-8-3503-6248-0/24/$31.00 ©2024 IEEE

related to hospitalization. However, hospitalization may be avertible provided patients and clinicians are cued to intervene prior to significant deterioration in cardiac functions. Long-term and realiable in-home monitoring is needed to address these challenges. Cardiac monitoring using ECG electrodes and bedside monitors has been implemented in the medical field for over 70 years. The standard 12-lead ECG, along with other reduced-lead (5- or 3-electrode) configurations, can accurately measure signals and help diagnose complex heart conditions [12]–[15]. Various machine learning approaches have been applied for predicting cardiovascular diseases [16]–[21]. One of the most well-known and popular methods used to classify ECG data is a Support Vector Machine (SVM) [22]–[24] with various kernels, feature extraction methods, and categories of arrhythmia.

For neural network approaches, a common method is to use a convolutional neural network. Güler and Übeyl propose a method for data engineering for ECG signals, by using discrete wavelet transform (DWT) to extract additional information about the signal in the form of wavelets [25]. These wavelets, in addition to a few statistical features derived from the signal, are used in a modular neural network, where each input of the network has its own neural network and work independently of each other. This network works because DWT can be broken down multiple times and each wavelet can be learned by the network. A 34-layer deep residual neural network is proposed in [26]. Beside the 1D-convolutional approaches, 2D-convolutional approaches exist [27], [28]. For example, [29] extracts an image from an ECG sample and uses a 2D-convolutional neural network to learn patterns from images. The network architecture of their proposed work is similar to that found in existing deep learning image models such as VGGNet [30].

Real-time decision making is the key requirement of many emerging applications that pose a set of new challenges to the deployment of machine learning models [31]. Specifically, machine learning models must be able to operate with low latency [32] and high throughput [33], [34], in order to satisfy the real-time requirement of these applications. In the area of cardiovascular medicine, initial efforts have been made towards real-time processing of ECG signals to diagnose relevant diseases [35], [36]. To this end, processing and analyzing ECG data as time series attracts increasing attention, and long short-term memory networks [10] have been used to achieve higher performance.

The existing studies on real-time processing have not used CNN based 2D approaches, due to its deep architecture which presents challenges for meeting the real-time processing needs. However, preserving 2D features offers the potential of integrating machine learning based approaches with the traditional diagnostic approaches, which opens up a broad range of opportunities to enhance the at-home monitoring. In this paper, we bridge this gap by exploring the potentials of using CNN based approaches to support efficient arrhythmic detection. Specifically, we have investigated various methods to reduce the amount of data needed for the learning task, and developed a new training method to leverage the efficiency enabled by data reduction without sacrificing the accuracy needed for arrhythmia detection. This approach is validated using an open-source dataset. We expect this approach will pave the way for continuous, real-time heart monitoring using small wearable devices.

Existing studies on 2D CNN based approaches focus mainly on either architectural development for the neural network or developing the preprocessing stage. Our exploration into data reduction along with our proposed training method takes a different approach for improving performance that can be applied independently to preprocessing or model choice under certain assumptions. Specifically, our methods can build upon previous workflows as long as the data resolution can be dynamically controlled and the choice of neural network can handle dynamic input sizes which most modern 2D CNNs can.

III. Background

The ECG machine processes the signals picked up from the skin by electrodes and produces a graphic representation of the electrical activity of the patient's heart. The impulses of the heart follow a specific pattern as follows: electrical activity towards a lead causes an upward deflection; electrical activity away from a lead causes a downward deflection; depolarization and repolarization deflections occur in opposite directions. This basic pattern of the electrical activity, when captured by ECG, show a common visual structure consisting of three waves, which have been named P, QRS (a wave complex), and T [37], as shown in Figure 1. The P wave is a small deflection wave that represents atrial depolarization, the QRS complex represent ventricular depolarization, and the T wave represents ventricular repolarization.

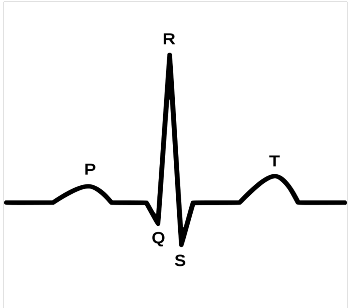

Fig. 1. QRS Complex

Arrhythmia conditions can be diagnosed by visually examining ECG images, as they show abnormal patterns. Figure 2 shows the ECG signals of nine different types of heartbeats, with the first being a normal beat while the remaining are arrhythmia. As the figure shows, the overall pattern of any arrhythmia differs from the normal beat, making it possible for a classification algorithm to recognize and classify different ECG patterns.

In this paper, we used the MIT-BIH Arrhythmia Database [38] to train and evaluate our machine learning

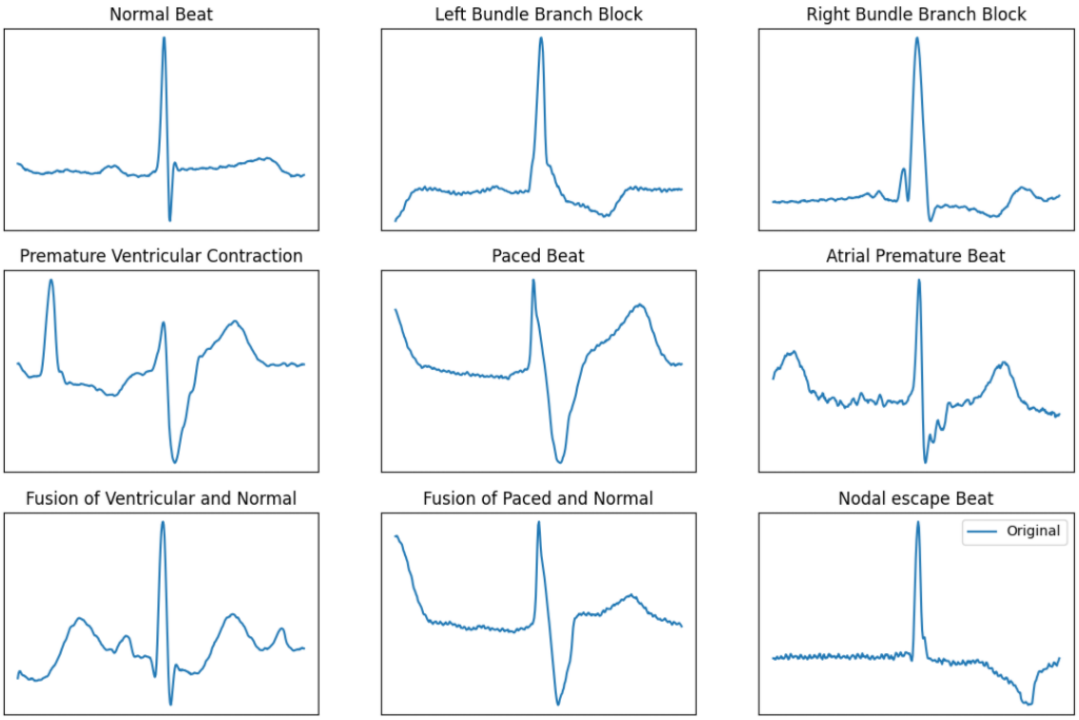

Fig. 2. ECG Signals of Nine Different Types of Heartbeat

TABLE I
MIT-BIH Dataset Class Description

Class	Annotation
Normal	Normal Left bundle branch block Right bundle branch block Atrial escape Nodal escape
SVEB	Atrial premature Aberrant atrial premature Nodal premature Supraventricular premature
VEB	Premature ventricular contraction Ventricular escape
Fusion	Fusion of ventricular and normal
Unknown	Paced Fusion of paced and normal Unclassifiable

TABLE II
MIT-BIH Dataset Class Distribution

Class	Samples	Train 80%	Test 20%
Normal	90593	72477	18116
SVEB	2781	2207	574
VEB	7235	5782	1453
Fusion	802	663	139
Unknown	8040	6431	1609
Total	109451	87560	21891

models. The database provides 48 records of different individuals, with varying age and medical conditions. Each record includes a 30-minute ECG recording recorded in two channels at a rate of 360 Hz (samples per second). The database provides a list of annotations that describes what conditions have been diagnosed in the nearby ECG region, as well as where the conditions are located. The full list of annotations contains various annotations that are labeled as single characters. These labels are divided into two categories, beat and non-beat annotations. Beat annotations describe the heartbeat and non-beat may describe the start/end to a region, peaks, and comments.

The original MIT-BIH Arrhythmia Database contains 40 different annotations for heartbeats. However, many of the annotations are not useful in classifying arrhythmic conditions. To improve the learning process, we grouped the labelled annotations into 5 classes as partitioned in Table I. The dataset is broken into a 80/20 train-test split through random sampling. Detailed information about data distributions are in Table II.

IV. Methodology

Memory and computation efficiency are two key factors for optimized deep learning. With 2D convolutional neural networks, it is observed that reducing the resolution of the input image will result in less compute and less memory consumption. In a preliminary study, we have measured the CPU latency on three different models, ResNet18, ResNet50, and MobileNetV3 Large, when given different input resolutions. The results are shown in Figure 3. For a given resolution, the latency throughput is measured for each model by computing the latency for a single forward pass with a dummy sample of batch size 1, averaged across 1,000 iterations. This process is done 10 times in total, then averaged to find the final measurement. There is a clear trend that indicates reduced latency with smaller resolutions. For memory consumption, consider the simple example of comparing 256×256 and 128×128 resolution images. The 256 resolution images are 4 times larger than the 128 resolution images. Therefore, we are motivated to reduce the input resolution as much as possible.

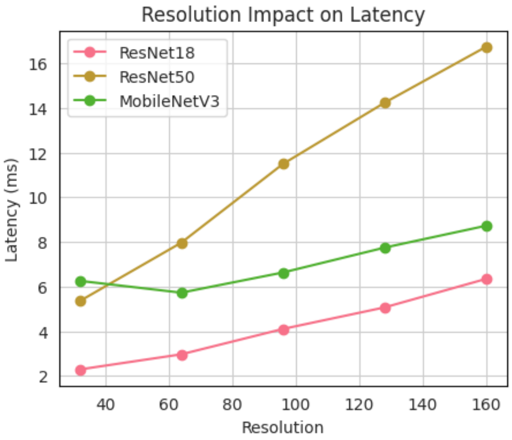

Fig. 3. Resolution Impact on Latency (CPU: Intel Core i9-10900X)

However, these gains in efficiency generally have a trade-off with accuracy. The reduction in accuracy can be correlated to a loss of information, as lowering the input resolution will decrease the information quality and the amount of information. In this study, we explore two different methods for reducing the input resolution. One approach is a reduction in information range, while the other is a reduction in information quality. Additionally, we propose a novel training method to improve accuracy recover for low resolution inputs while sacrificing no inference time penalties, namely *Progressive Resolution Shrinking*.

A. Preprocessing For Resolution Reduction

For image-based ECG classification, the data preprocessing can be divided into two stages. First, the raw ECG signal data is segmented into heartbeat samples, then each sample is subsequently transformed into an image.

During heartbeat segmentation, we begin by locating each r-peak. For each r-peak we select some number of samples before and after the peak to represent the heartbeat segments. This process of sample selection is what will define our two distinct resolution reduction methods. The first approach, which will be referred to as window segmentation, simply selects some window centered around the r-peak to use as the heartbeat segment. For example, consider selecting 64 samples before and after the r-peak. In this case, the resulting heartbeat segment will be 128 samples in total. The second approach follows the same initial step as window segmentation by first selecting some window centered around the r-peak. Although this time, we will first sample at a higher resolution (increased window size), then follow up with a resample down to the desired resolution. For example, consider selecting 128 samples before and after the r-peak. The resulting heartbeat segment will be 256 samples wide, but we will then apply a resample on this segment to reduce it down to a resolution of 128. In the first approach, we maintain information quality but the overall range is limited. In the second approach, we maintain the information range but sacrifice quality due to the resampling process. A visualization of these two preprocessing approaches is shown in Figure 4.

After ECG signals have been separated into heartbeat segments, each segment can be transformed into an image. In this study, we follow the transformation method outlined in [39], [40]. Specifically, for each heartbeat segment, three unique transformations are applied individually, Recurrence Plot (RP), Gramian Angular Field (GAF), and Markov Transition Field (MTF), to produce three separate images. RP extracts the periodicity and recurring patterns in the ECG signal over time. GAF encodes the signal by representing its angular perspective through transformation to polar coordinates. Angular differences between the encoded points are then extracted, leading to the detection of possible higher-order correlations. MTF encodes the probabilistic transitions between different states in the signal over time, essentially capturing the transition dynamics between different signal values. These three images, which capture unique aspects of the heartbeat signal's temporal dynamics, are then stacked in the channel dimension to create a three channel image. This multi-transformation stacking method has shown to produce strong results when compared to single transformation methods, but increases the overall computational cost. Thus, optimizing this particular method is in our best interest.

B. Progressive Resolution Shrinking

In addition to exploring the impact of resolution reduction methods, we propose a new training approach, namely progressive resolution shrinking, which establishes a mechanism to leverage the resolution reduction in the learning process. When reducing the resolution, information will always be lost. Thus we take inspiration from the idea of transfer learning to minimize the impact of this information loss. In typical transfer learning, a neural network is first trained on a generalized dataset and task to help the model establish broad data understanding which can then be transferred over to a new dataset or task through finetuning. For example, training a model first for ImageNet classification then finetuning for CIFAR classification. With progressive resolution shrinking, instead of learning general understanding from a different dataset, we choose to learn from the same dataset, as the ECG data preprocessing stage allows for varying approaches to generation using the same data. In this case, we will create each heartbeat sample at various different resolutions, where a higher resolution results in more information. Then, we can first train our models using higher resolution samples before finalizing them on a lower resolution version. In addition, we propose to apply a progressive shrinking approach to this multi-resolution reduction training to better distill information. In other words, we begin training at the highest sample resolution and gradually lower the resolution as training progresses. For example, consider training a model first with 256

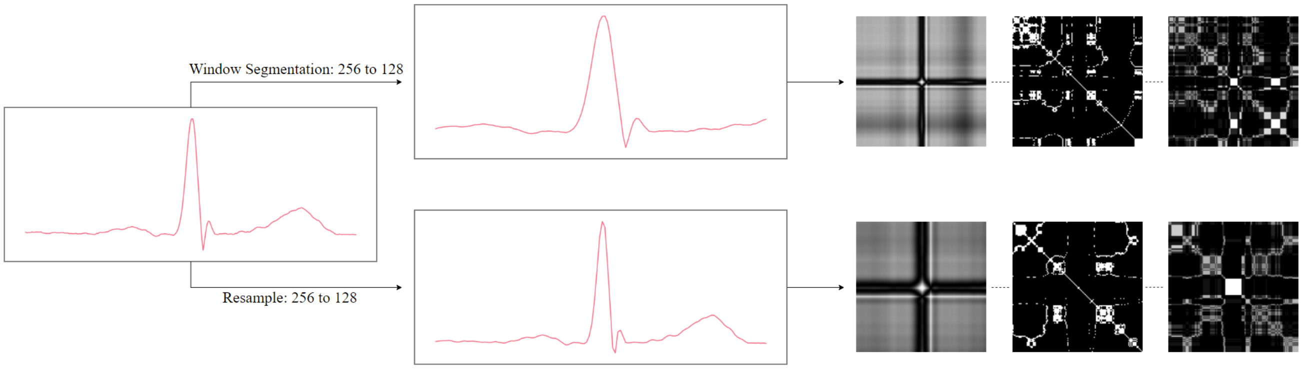

Fig. 4. Preprocessing For Resolution Reduction. The top path is with window segmentation and the bottom path is with resampling. Note that the resampled signal is still slightly different than its base despite being barely viable.

resolution samples and decreasing that resolution by 16 every 10 epochs. This approach is illustrated in Algorithm 1.

Algorithm 1 Progressive Resolution Shrinking

```
for Each $input\_resolution$ do
    Generate dataset using current $input\_resolution$
end for
Initialize model and training parameters
for Each $(input\_resolution, resolution\_epochs)$ in descending resolution order do
    Initialize training dataset on current $input\_resolution$
    for Each $epoch$ in $resolution\_epochs$ do
        Train model with current dataset
    end for
end for
```

To apply progressive resolution shrinking on the two preprocessing methods used in this study, window segmentation and resampling, we can simply generate different resolution samples during the heartbeat segmentation stage. With window segmentation, varying the time-step width on each sample will result in different resolutions per sample. For resampling, after a base resolution is selected, different resolutions can be generated by varying the resampling rate. A visualization of progressive resolution shrinking is provided in Figure 5.

V. Evaluation

A. Experiment Design

We train ResNet18 models under seven different resolutions, starting from 64 and incrementing by 32 up to a max resolution of 256. For each resolution, we test the two previously mentioned preprocessing approaches, window segmentation and resampling. Resampling is done by first creating each heartbeat segment at a resolution of 256 and then subsequently downsampling to the desired resolutions. Additionally, we train another set of ResNet18 models on resolutions of 64 and 96 while applying the progressive resolution shrinking framework for both window segmentation and resampling. For progressive resolution shrinking models, they begin training at a resolution of 128 with decrements of 8 until the final desired resolution is reached.

All baseline window segmentation and resampling models are trained for 50 epochs. Progressive resolution shrinking models are trained for 30 epochs during the high resolutions stages and 50 epochs for the final stage. The training data batch size is 128. The optimizer is Stochastic Gradient Decent with a learning rate of 0.01, a momentum of 0.9, and a weight decay of 1×10^{-4}. A cosine annealing scheduler is used for the learning rate [41]. Note that for progressive resolution shrinking, the optimizer and scheduler states do not reset during the resolution switches, only the dataset is swapped out. For hardware, two NVIDIA RTX 3080 are used in a distributed data parallel environment. Note that for distributed evaluation on the test set, the DistributedSampler from PyTorch prefers that the data be evenly divisible by the number of devices. If the data is not evenly divisible, then the last bit of data may be dropped. Although, because we are performing distributed evaluation on the test set, we do not want the drop any of the samples. In this case, PyTorch will add additional samples to make the data evenly divisible. For our experiments, we have two GPUs and the test set consists of 21,891 samples, thus an additional sample is added automatically to make the data evenly divisible. Because the difference between 21,891 and 21,892 is minuscule in terms of relative scale, we consider this factor negligible as whether or not the model correctly predicts the extra sample will have almost no overall impact.

B. Results and Discussion

For comparing the performance of each trained model, we consider the overall accuracy. Each baseline model is trained for 50 epochs and each progressive resolution shrinking model trained for 50 epochs during the final resolution stage. To make the accuracy measurements more reliable, every model is evaluated with the test set after each epoch to keep record of a running performance. The first 30 accuracy measurements are discarded as to allow the model to converge. The remaining final 20 accuracy measurements are then averaged. This mean accuracy will be used as the comparison metric. Results for all mean accuracy measurements are presented in Figure 6. Note that extreme outliers are removed to keep the results

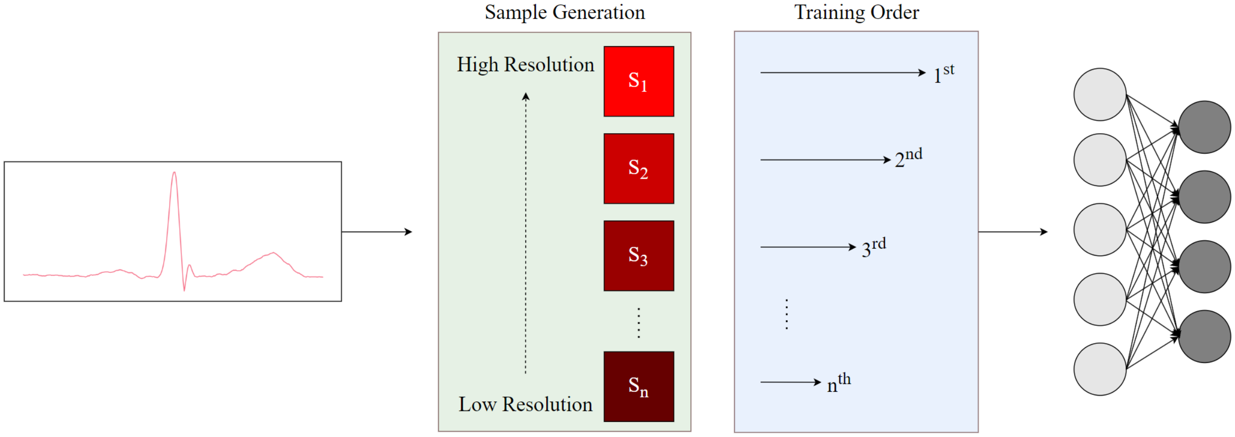

Fig. 5. Progressive Resolution Shrinking. Each heartbeat signal generates training samples of varying resolutions. The model is trained on descending sample resolution order, meaning the model trains on the highest resolution first and the lowest resolution last.

stable. Specifically we remove any accuracy measurement below 90%. In this case, there is only one iteration during the baseline resampling model trained under a resolution of 64 that hit 82.42% accuracy. All other accuracy measurements are greater than 90%.

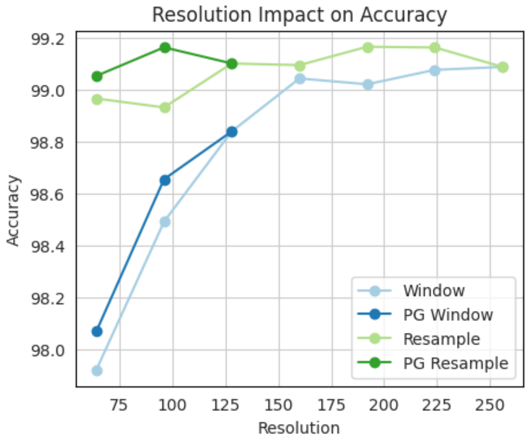

Fig. 6. Progressive Resolution Shrinking Accuracy. PG (the dark lines) marks the progressive resolution shrinking model variant.

When comparing baseline resampling against window segmentation, it is clear that resampling maintains stronger accuracy when decreasing the resolution. Notably, the resampling accuracy from resolutions 256 to 128 did not see any negative change with the 256 resolution accuracy at 99.09% and the 128 resolution accuracy at 99.10%. The accuracy only starts to slightly drop off at resolutions below 128, with the 96 resolution accuracy at 98.93% and the 64 resolution accuracy at 98.97%. For window segmentation, the accuracy differences between resolutions are much more substantial. From resolution 256 to 160, the accuracy is relatively stable, only seeing up to a 0.07% absolute decrease. When the resolution reaches below 160, the accuracy degradation begins to be more severe, with the 64 resolution model reaching a global low of 97.92%. Therefore, we can conclude that resampling results in far greater performance for resolution reduction.

With the addition of progressive resolution shrinking, improvements to the low resolution models (96 and 64) are apparent for both resampling and window segmentation. For the 64 resolution models, resampling and window segmentation accuracies increased from 98.97% to 99.05% and from 97.92% to 98.07% respectively. For the 96 resolution models, accuracies increased from 98.93% to 99.16% and from 98.49% to 98.66% respectively. To better analyze progressive resolution shrinking, we can explore the running metrics during the training process. Specifically, we will consider the running accuracy and accumulated loss, comparing the final stage of progressive resolution shrinking against the baseline counterparts (Figure 7). For resampling-based progressive resolution shrinking the running accuracy and loss are consistently ahead of their baseline counterparts. With window segmentation progressive resolution shrinking, the 96 resolution model also outperforms the baseline. Additionally, the progressive resolution shrinking methods generally show more stability. Therefore, progressive resolution shrinking shows improvements to overall accuracy while sacrificing no deployment penalties.

To understand the computational complexity and run-time performance, we measure the MACs (Multiply-Accumulate Operations) and latencies for the resampling-based models. Results are provided in Table III. Without much accuracy loss, we see the low resolution models with progressive resolution shrinking hit upto a 16× reduction in MACs and a 9.9× reduction in latency.

VI. Analysis and Discussion

One observation with progressive resolution shrinking is that it often performs better with resampling when compared to window segmentation. To verify this observation, we define a metric representing the relative accuracy recovery as follows:

$$\text{relative\_accuracy\_recover} = \frac{\alpha - \beta_{lower}}{\beta_{upper} - \beta_{lower}} \quad (1)$$

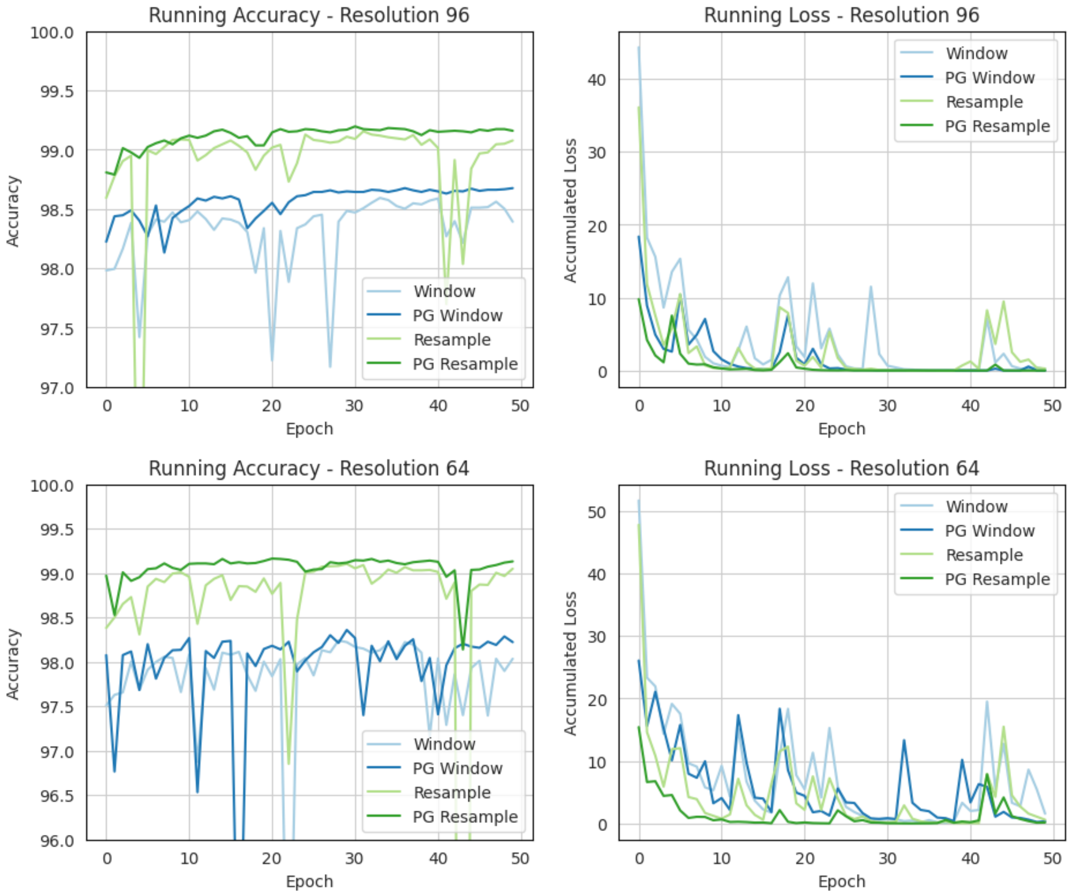

Fig. 7. Progressive Resolution Shrinking Running Statistics. PG (the dark lines) marks the progressive resolution shrinking model variants.

TABLE III
RESNET-18 RESOLUTION PERFORMANCE
RESAMPLING-BASED MODELS ARE USED FOR THE ACCURACY MEASUREMENTS. LATENCY CPU IS MEASURED ON AN INTEL CORE I9-10900X, LATENCY GPU IS MEASURED ON A NVIDIA RTX 3080, AND LATENCY EDGE IS MEASURED ON A JETSON ORIN NANO.

Resolution	Acc %	MACs		Latency	
			CPU Batch 1	GPU Batch 64	Edge Batch 16
256	99.09	2382 M	10.21 ms	19.99 ms	64.42 ms
224	99.16	1824 M	8.53 ms	15.71 ms	53.34 ms
192	99.17	1340 M	7.80 ms	12.56 ms	39.14 ms
160	99.10	930 M	6.17 ms	8.65 ms	29.40 ms
128	99.10	595 M	5.01 ms	5.47 ms	20.40 ms
96 w/ PRS	**99.16**	**335 M**	**4.01 ms**	**3.95 ms**	**16.00 ms**
64 w/ PRS	**99.05**	**149 M**	**2.95 ms**	**2.02 ms**	**9.91 ms**

where α is the accuracy after applying progressive resolution shrinking, β_{lower} is the baseline lower-bound accuracy, and β_{upper} is the baseline upper-bound accuracy. For example, consider that we have a baseline upper-bound accuracy is 90% and a baseline lower-bound accuracy of 80%. Then if we consider an α of 85%, the relative accuracy recover is 0.5, and for an α of 86%, the relative accuracy recover is 0.6.

We can now apply this metric to the previously ran experiment to understand the effectiveness of progressive resolution shrinking for resampling and window segmentation. For each method, we let β_{upper} be the baseline mean accuracy at the starting resolution (both methods started at a resolution of 128) and let β_{lower} be the baseline mean accuracy at the final resolution (96 and 64). Then, let α be the respective mean accuracy when applying progressive resolution shrinking. To better understand this, let us examine one concrete example. For resampling, the baseline mean accuracy without progressive resolution shrinking at a resolution of 128 is 99.1017%. The baseline mean accuracy at a resolution of 64 is 98.9667%. When progressive resolution shrinking is applied from 128 to 64, the resulting mean accuracy is 99.0538%. Using these values, we find the relative accuracy recover for this case to be 0.65. Continuing this, we find the relative accuracy recover for progressive resampling at a resolution of 96 to be 1.37 and progressive window segmentation at 96 and 64 to be 0.47 and 0.16 respectively. We see that progressive resolution shrinking with resampling is more effective than with window segmentation as the relative accuracy recover for resampling is far greater.

To better understand why progressive resolution shrinking performs better with resampling when compared to window segmentation, we will explore the information distillation during resolution reduction. Specifically, how similar are the images produced at each resolution stage to each other. The intuition is that greater similarity between resolutions will result in a stronger distillation of information. To test this idea, we compare multiple different resolution samples against each other. Each data preprocessing method, resampling and window segmentation, has five different resolutions generated per sample, from 256 to 128 in decrements for 32. We then measure the pairwise similarity between the resolutions. This is performed across 100 random samples per resolution pair to gather averages. For similarity measurements, two metrics will be used, Mean Square Error (MSE) and Mean Structural Similarity Index Measure (MSSIM) [42].

MSE is a computationally simple and intuitive similarity metric that measures the average of the squared difference between two values. It is expressed as

$$\text{MSE} = \frac{1}{n}\sum_{i=1}^{n}(y_i - \hat{y_i})^2 \tag{2}$$

where n is the total number of elements, y and $\hat{y}$ are the two values to compare. A MSE of zero represent two identical values.

MSSIM measures the similarity between two images taking into account human perception. MSSIM in particular is an extension of its base metric SSIM. SSIM is a weighted combination of three comparison metrics defined as:

$$\text{SSIM}(x,y) = l(x,y)^{\alpha} \cdot c(x,y)^{\beta} \cdot s(x,y)^{\gamma} \tag{3}$$

where x and y are the images, $\alpha, \beta, \gamma > 0$, and the functions l, c, s are comparison metrics on luminance, contrast, and structure respectively. These comparison functions are defined as:

$$l(x,y) = \frac{2\mu_x\mu_y + C_1}{\mu_x^2 + \mu_y^2 + C1} \tag{4}$$

$$c(x,y) = \frac{2\sigma_x\sigma_y + C_2}{\sigma_x^2 + \sigma_y^2 + C2} \tag{5}$$

$$s(x,y) = \frac{\sigma_{xy} + C_3}{\sigma_x\sigma_y + C3} \tag{6}$$

where μ_x, μ_y are the pixel sample means, σ_x, σ_y are the standard deviations of x and y, σ_x^2, σ_y^2 are the variance of x and y, σ_{xy} is the covariance of x and y, and C_1, C_2, C_3 are constants. When $C_3 = C_2/2$ and $\alpha = \beta = \gamma = 1$, SSIM can be simplified to:

$$\text{SSIM}(x,y) = \frac{(2\mu_x\mu_y + C_1)(2\sigma_{xy} + C_2)}{(\mu_x^2 + \mu_y^2 + C_1)(\sigma_x^2 + \sigma_y^2 + C_2)} \tag{7}$$

In practice, computing the SSIM on the entire images can lead to undesirable results, thus MSSIM is used instead. MSSIM takes the mean SSIM of local windows across the images and is defined as:

$$\text{MSSIM}(X,Y) = \frac{1}{M}\sum_{j=1}^{M}\text{SSIM}(x_j, y_y) \tag{8}$$

where X and Y are the images, x_j and y_j are the image contents of the jth local window, and M is the number of local windows. The windows used in this analysis follow the original paper with a circular-symmetric Guassian weighting function. MSSIM is a measurement from $[-1, 1]$ where 0 represents complete dissimilarity and larger magnitudes represent increased correlation or anti-correlation. Though, MSSIM can be normalized to $[0, 1]$.

Note that MSE and MSSIM require the two images in question to be of the same dimensions, thus for each lower resolution image we downscale the higher resolution image to match in size. Empirical results of the proposed analysis are presented in Figure 8. For both methods, the similarity decreases as the resolution gap increases. The overall similarity for resampling across resolutions is far greater than window segmentation, with resampling seeing a lower MSE and a higher MSSIM. Therefore, we may conclude that sample similarity across resolutions is one potential factor in understanding the performance of progressive resolution shrinking.

When resampling, the design space can be large. In this work, we chose to resample from a base resolution of 256 down to the desired resolution, but the choice for the base resolution could have been anything. For example, if our desired resolution is 128, should we resample from 224, 256, or 288. The choice for the base resolution could potentially be impacted by the magnitude of the desired resolution, the sampling rate of the ECG signal, or other variable factors. As of this study, it is still unclear on how to efficiently choose the best resampling parameters.

Progressive resolution shrinking requires that the models be initially trained on higher resolution samples. This increases training time and requires that the system be able to efficiently process potentially large data samples. For the experiment ran in this study, progressive resolution shrinking on the 96 and 64 resolution models extended training times by 4.6× and 8.8× respectively. Note though that total training time does not always impact the accuracy convergence. On-device training may have difficulties implementing progressive resolution shrinking. Although, despite the increase to training time, it is important to state that the training time will not negatively impact the model during its deployment. Additionally, due to the nature of progressive resolution shrinking being a training method, its application to existing workflows will not impose any additional computational overhead post training.

VII. Conclusion

Arrhythmia such as atrial fibrillation is a common cause of death in the United States. While the most common way to diagnose arrhythmia is through an ECG reading, a challenge that comes from this method of diagnosis is that it requires a trained medical profession to evaluate the ECG reading. The development of a machine assisted method can speed up the diagnostic process and potentially reduces fatality. Both 1D and 2D based deep learning approaches have been studied in the context of arrhythmia detection. 2D based approaches provide the advantages of high accuracy and preserving features that can be used for visual verification, however, these approaches have higher requirements on computational resources, due to

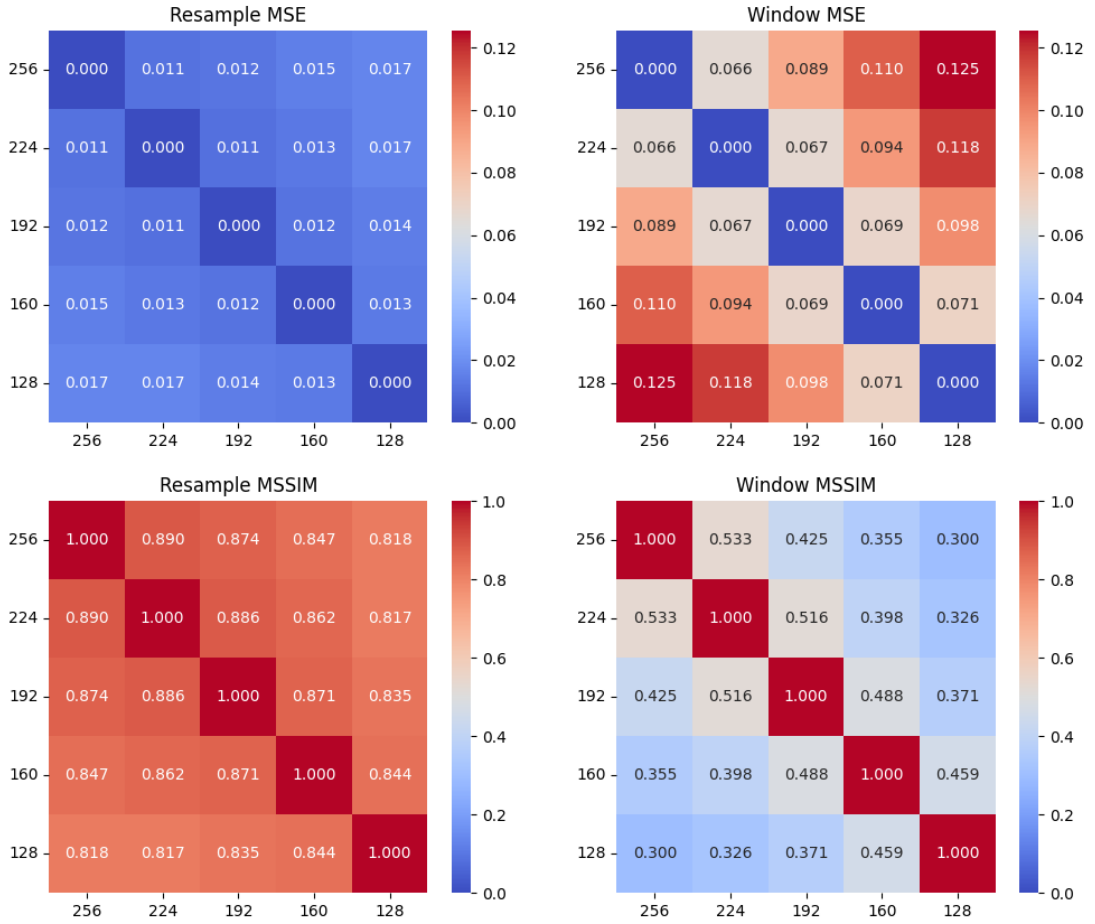

Fig. 8. Resampling and Window Segmentation Similarity Analysis. For MSE, lower magnitude (blue) represents higher similarity. For MSSIM, higher magnitude (red) represents higher similarity.

their complexity and amount of data required for learning. In this paper, we have investigated strategies for reducing the amount of data needed to perform the arrhythmia classification, and developed a novel training method to support the learning task with progressive resolution reduction. We have carried out extensive experiments to evaluate the proposed approach, and the results show that it demonstrates stable performance with much less demand on computational resources. This research provides potentials for faster diagnostic of arrhythmia and can be deployed in a medical environment without powerful hardware.

Work is ongoing in several directions. First, we will investigate mechanisms to improve the distillation capability of progressive resolution shrinking. Second, we will build on top of the classification results, and develop feature extraction methods and machine learning models for anomaly detection. Classification provides a good guidance on diagnosis, because it classifies data samples to a number of known conditions. However, in a real-time setting, it might not be necessary to diagnose the underlying problems. Instead, being able to quickly detect abnormal heart rhythms is more critical, i.e., anomaly detection. Therefore, a binary classifier which detects anomaly signals from the normal ones is the essential part of real-time monitoring and detection of arrhythmia. Finally, we will develop real-time sensing techniques to collect data at real-time, and evaluate our models using data collected from patients.

Acknowledgment

This research is supported in part by the National Science Foundation through awards CNS #2216108 and OAC #2243980.

References

[1] S. S. Virani, A. Alonso, H. J. Aparicio, E. J. Benjamin, M. S. Bittencourt, C. W. Callaway, A. P. Carson, A. M. Chamberlain, S. Cheng, F. N. Delling, M. S. Elkind, K. R. Evenson, J. F. Ferguson, D. K. Gupta, S. S. Khan, B. M. Kissela, K. L. Knutson, C. D. Lee, T. T. Lewis, J. Liu, M. S. Loop, P. L. Lutsey, J. Ma, J. Mackey, S. S. Martin, D. B. Matchar, M. E. Mussolino, S. D. Navaneethan, A. M. Perak, G. A. Roth, Z. Samad, G. M. Satou, E. B. Schroeder, S. H. Shah, C. M. Shay, A. Stokes, L. B. VanWagner, N.-Y. Wang, and C. W. Tsao, "Heart disease and stroke statistics update," *Circulation*, vol. 143, no. 8, pp. e254–e743, 2021.

[2] National Heart Lung and Blood Institute, "Arrhythmia," https://www.nhlbi.nih.gov/health-topics/arrhythmia.

[3] A. Heidenreich Paul et al., "Forecasting the Impact of Heart Failure in the United States," *Circulation: Heart Failure*, vol. 6, no. 3, pp. 606–619, 2013.

[4] P. Singh and A. Jasuja, "IoT based low-cost distant patient ECG monitoring system," in *2017 International Conference on Computing, Communication and Automation (ICCCA)*, 2017, pp. 1330–1334.

[5] G. Xu, "IoT-Assisted ECG Monitoring Framework With Secure Data Transmission for Health Care Applications," *IEEE Access*, vol. 8, pp. 74 586–74 594, 2020.

[6] S. Ahsanuzzaman, T. Ahmed, and M. A. Rahman, "Low Cost, Portable ECG Monitoring and Alarming System Based on Deep Learning," in *2020 IEEE Region 10 Symposium (TENSYMP)*, 2020, pp. 316–319.

[7] K. Yakut, M. Usman, W. Xue, F. M. Haas, R. A. Hirsh, J. Boothby, T. Petty, and X. Zhao, "Electro-Mechanical Data Fusion for Heart Health Monitoring," in *Proceedings of the 10th IEEE International Conference on Healthcare Informatics (ICHI 2022)*, 2022.

[8] D. Bertsimas, L. Mingardi, and B. Stellato, "Machine learning for real-time heart disease prediction," *IEEE Journal of Biomedical and Health Informatics*, vol. 25, no. 9, pp. 3627–3637, 2021.

[9] P. Zhou, B. Schwerin, B. Lauder, and S. So, "Deep learning for real-time ecg r-peak prediction," in *2020 14th International Conference on Signal Processing and Communication Systems (ICSPCS)*. IEEE, 2020, pp. 1–7.

[10] T. Petty, T. Vu, X. Zhao, R. A. Hirsh, G. Murray, F. M. Haas, and W. Xue, "Evaluating deep learning algorithms for real-time arrhythmia detection," in *2020 IEEE/ACM International Conference on Big Data Computing, Applications and Technologies (BDCAT)*, 2020, pp. 19–26.

[11] P. Heidenreich, N. Albert, L. Allen, D. Bluemke, J. Butler, G. Fonarow, J. Ikonomidis, O. Khavjou, M. Konstam, T. Maddox, G. Nichol, M. Pham, I. Pina, and J. Trogdon, "Forecasting the impact of heart failure in the united states: A policy statement from the american heart association," *Circulation: Heart Failure*, vol. 6, pp. 606–619, 05 2013.

[12] J. Francis, "ECG monitoring leads and special leads," *Indian Pacing and Electrophysiology Journal*, vol. 16, no. 3, pp. 92–95, 2016.

[13] B. J. Drew et al, "Accuracy of the EASI 12-lead electrocardiogram compared to the standard 12-lead electrocardiogram for diagnosing multiple cardiac abnormalities," *Journal of Electrocardiology*, vol. 32, pp. 38–47, 1999.

[14] J. K. Zègre-Hemsey, J. L. Garvey, and M. G. Carey, "Cardiac Monitoring in the Emergency Department," *Critical Care Nursing Clinics of North America*, vol. 28, no. 3, pp. 331–345, 2016.

[15] A. Petrėnas, V. Marozas, G. Jaruševičius, and L. Sörnmo, "A modified Lewis ECG lead system for ambulatory monitoring of atrial arrhythmias," *Journal of Electrocardiology*, vol. 48, no. 2, pp. 157–163, 2015.

[16] K. Shameer, K. W. Johnson, B. S. Glicksberg, J. T. Dudley, and P. P. Sengupta, "Machine Learning in Cardiovascular Medicine: Are We There Yet?" *Heart*, vol. 104, no. 14, pp. 1156–1164, 2018.

[17] C. Krittanawong, H. Zhang, Z. Wang, M. Aydar, and T. Kitai, "Artificial Intelligence in Precision Cardiovascular Medicine," *Journal of the American College of Cardiology*, vol. 69, no. 21, pp. 2657 – 2664, 2017.

[18] E. Vocaturo and E. Zumpano, "Ecg analysis via machine learning techniques: News and perspectives," in *2021 IEEE International Conference on Bioinformatics and Biomedicine (BIBM)*. IEEE, 2021, pp. 3106–3112.

[19] G. Altan, N. Allahverdi, and Y. Kutlu, "A multistage deep learning algorithm for detecting arrhythmia," in *2018 1st international conference on computer applications & information security (ICCAIS)*. IEEE, 2018, pp. 1–5.

[20] G. Altan, A. Yayık, and Y. Kutlu, "Deep learning with convnet predicts imagery tasks through eeg," *Neural Processing Letters*, vol. 53, no. 4, pp. 2917–2932, 2021.

[21] G. Altan, N. Allahverdi, and Y. Kutlu, "A multistage deep learning algorithm for detecting arrhythmia," in *2018 1st International Conference on Computer Applications & Information Security (ICCAIS)*, 2018, pp. 1–5.

[22] A. E. Zadeh, A. Khazaee, and V. Ranaee, "Classification of the electrocardiogram signals using supervised classifiers and efficient features," *Computer Methods and Programs in Biomedicine*, vol. 99, no. 2, pp. 179–194, 2010. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S0169260710001100

[23] S. M. P. Dinakarrao, A. Jantsch, and M. Shafique, "Computer-aided arrhythmia diagnosis with bio-signal processing: A survey of trends and techniques," *ACM Comput. Surv.*, vol. 52, no. 2, Mar. 2019. [Online]. Available: https://doi.org/10.1145/3297711

[24] H. Li, X. Feng, L. Cao, E. Li, H. Liang, and X. Chen, "A new ecg signal classification based on wpd and apen feature extraction," *Circuits, Systems, and Signal Processing*, vol. 35, no. 1, p. 339–352, 2015.

[25] I. Güler and E. D. Übeyli, "Ecg beat classifier designed by combined neural network model," *Pattern Recognit.*, vol. 38, pp. 199–208, 2005.

[26] P. Rajpurkar, A. Y. Hannun, M. Haghpanahi, C. Bourn, and A. Y. Ng, "Cardiologist-level arrhythmia detection with convolutional neural networks," *CoRR*, vol. abs/1707.01836, 2017. [Online]. Available: http://arxiv.org/abs/1707.01836

[27] U. R. Acharya, H. Fujita, S. L. Oh, U. Raghavendra, J. H. Tan, M. Adam, A. Gertych, and Y. Hagiwara, "Automated identification of shockable and non-shockable life-threatening ventricular arrhythmias using convolutional neural network," *Future Generation Computer Systems*, vol. 79, pp. 952–959, 2018. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S0167739X17315248

[28] Y. Wu, F. Yang, Y. Liu, X. Zha, and S. Yuan, "A comparison of 1-d and 2-d deep convolutional neural networks in ECG classification," *CoRR*, vol. abs/1810.07088, 2018. [Online]. Available: http://arxiv.org/abs/1810.07088

[29] T. J. Jun, H. M. Nguyen, D. Kang, D. Kim, D. Kim, and Y. Kim, "ECG arrhythmia classification using a 2-d convolutional neural network," *CoRR*, vol. abs/1804.06812, 2018. [Online]. Available: http://arxiv.org/abs/1804.06812

[30] K. Simonyan and A. Zisserman, "Very deep convolutional networks for large-scale image recognition," 2015.

[31] R. Nishihara, P. Moritz, S. Wang, A. Tumanov, W. Paul, J. Schleier-Smith, R. Liaw, M. Niknami, M. I. Jordan, and I. Stoica, "Real-Time Machine Learning: The Missing Pieces," in *Proceedings of the 16th Workshop on Hot Topics in Operating Systems*, ser. HotOS '17. ACM, 2017, pp. 106–110.

[32] D. Crankshaw, P. Bailis, J. E. Gonzalez, H. Li, Z. Zhang, M. J. Franklin, A. Ghodsi, and M. I. Jordan, "The Missing Piece in Complex Analytics: Low Latency, Scalable Model Management and Serving with Velox," in *CIDR 2015, Seventh Biennial Conference on Innovative Data Systems Research*, 2015.

[33] D. Silver, A. Huang, C. J. Maddison, A. Guez, L. Sifre, G. van den Driessche, J. Schrittwieser, I. Antonoglou, V. Panneershelvam, M. Lanctot, S. Dieleman, D. Grewe, J. Nham, N. Kalchbrenner, I. Sutskever, T. Lillicrap, M. Leach, K. Kavukcuoglu, T. Graepel, and D. Hassabis, "Mastering the game of Go with deep neural networks and tree search," *Nature*, vol. 529, pp. 484–503, 2016.

[34] A. Nair, P. Srinivasan, S. Blackwell, C. Alcicek, R. Fearon, A. D. Maria, V. Panneershelvam, M. Suleyman, C. Beattie, S. Petersen, S. Legg, V. Mnih, K. Kavukcuoglu, and D. Silver, "Massively Parallel Methods for Deep Reinforcement Learning," *CoRR*, vol. abs/1507.04296, 2015.

[35] J. J. Oresko, Z. Jin, J. Cheng, S. Huang, Y. Sun, H. Duschl, and A. C. Cheng, "A Wearable Smartphone-Based Platform for Real-Time Cardiovascular Disease Detection Via Electrocardiogram Processing," *IEEE Transactions on Information Technology in Biomedicine*, vol. 14, no. 3, pp. 734–740, 2010.

[36] Z. Jin, Y. Sun, and A. C. Cheng, "Predicting Cardiovascular Disease from Real-time Electrocardiographic Monitoring: An Adaptive Machine Learning Approach on a Cell Phone," in *2009 Annual International Conference of the IEEE Engineering in Medicine and Biology Society*, 2009, pp. 6889–6892.

[37] E. Ashley, E. Ashley, and J. Niebauer, *Cardiology Explained*, ser. Explained Series. Remedica, 2004.

[38] A. L. Goldberger, L. A. N. Amaral, L. Glass, J. M. Hausdorff, P. C. Ivanov, R. G. Mark, J. E. Mietus, G. B. Moody, C.-K. Peng, and H. E. Stanley, "PhysioBank, PhysioToolkit, and PhysioNet: Components of a new research resource for complex physiologic signals," *Circulation*, vol. 101, no. 23, pp. e215–e220, 2000 (June 13).

[39] Z. Ahmad, A. Tabassum, L. Guan, and N. M. Khan, "Ecg heartbeat classification using multimodal fusion," *IEEE Access*, vol. 9, pp. 100 615–100 626, 2021.

[40] F. Zhou and D. Fang, "Multimodal ecg heartbeat classification method based on a convolutional neural network embedded with fca," *Scientific Reports*, vol. 14, no. 1, p. 8804, 2024.

[41] I. Loshchilov and F. Hutter, "Sgdr: Stochastic gradient descent with warm restarts," *arXiv preprint arXiv:1608.03983*, 2016.

[42] Z. Wang, A. C. Bovik, H. R. Sheikh, and E. P. Simoncelli, "Image quality assessment: from error visibility to structural similarity," *IEEE transactions on image processing*, vol. 13, no. 4, pp. 600–612, 2004.