Database Article

SynthECG: Python Framework and ECG Image Datasets for Digitization, Lead Detection, and Waveform Segmentation

Masoud Rahimi*, Reza Karbasi*, Abdol-Hossein Vahabie

Department of Machine Intelligence and Robotics, School of Electrical and Computer Engineering, University of Tehran, Tehran, Iran

**These authors contributed equally.*

Abstract

Background: Digitizing electrocardiogram (ECG) images into structured time-series data is critical for clinical analysis, but it remains challenging due to the lack of standardized datasets, especially under realistic scenarios like overlapping waveforms. **Methods:** We introduce SynthECG, an open-source Python framework to generate four synthetic ECG datasets tailored for deep learning tasks, including ECG digitization, YOLO-based lead and lead name detection, and U-Net-based waveform segmentation. The framework supports customizable parameters (e.g., dataset size, lead layout, and visual style) and allows generating up to 21,799 images for multi-lead datasets and 261,588 for single-lead segmentation. Notably, it introduces a novel mechanism to simulate overlapping waveforms from adjacent leads while preserving clean segmentation masks. **Results:** Using our framework, we generated four open-access datasets: (1) 2000 ECG images in various lead configurations paired with time-series signals for ECG digitization, (2) 2000 ECG images in various lead configurations with YOLO-format annotations for detecting lead regions and lead names, (3) 20,000 cropped single-lead images with pixel-level segmentation masks (normal variant), and (4) 102 cropped single-lead images with overlapping waveforms from adjacent leads (overlapping variant). We validated these datasets through two case studies: digitization using a non–ML algorithm (mean squared error: 0.002, ρ: 0.93, signal-to-noise ratio [SNR]: 7.36 dB, SNR_{med}: 37.86 dB) and lead/name detection using YOLOv8. **Conclusions:** Our open-source framework enables the generation of large-scale, customizable ECG image datasets to support key deep learning-based tasks, including digitization under normal and overlapping conditions, as well as lead region and lead name detection. The full datasets and code are publicly available at: https://doi.org/10.5281/zenodo.15484519 and https://github.com/rezakarbasi/ecg-image-and-signal-dataset.

Keywords: *Deep learning, ECG digitization, electrocardiogram (ECG), lead detection, synthetic data, waveform segmentation*

Submitted: 08-Jun-2025 **Revised:** 27-Aug-2025 **Accepted:** 08-Sep-2025 **Published:** 30-Mar-2026

Introduction

Electrocardiogram (ECG) is a foundational tool in the diagnosis and monitoring of cardiovascular diseases, which remain a leading cause of death worldwide.[1] Access to ECG time-series data significantly improves the performance of deep learning–based clinical analysis.[2] For decades, healthcare institutions have stored ECG records in paper or scanned image formats. These legacy records contain valuable clinical information, including patient history and rare cardiac events.[3] In many hospitals, ECGs are often stored as images or PDFs rather than raw signals, as this method lowers costs and eliminates the need for specialized hardware or trained personnel.[4] As a result, digitizing ECG images into structured time-series data has become an essential task.

Lence *et al*.[4] highlighted the lack of open-access ECG digitization datasets for benchmarking. Since then, three large-scale, open-access datasets have been introduced that provide both ECG images and their corresponding ground truth signals: ECG-Image-Kit,[5] ECG-Image-Database,[3] and PMcardio ECG Image Database.[6] All three datasets consist of synthetic ECG images generated from the original signals, and they incorporate different augmentations to simulate real-world artifacts. Section 2 provides a detailed overview of these datasets.

One of the most critical components of fully automated ECG digitization is the

How to cite this article: Rahimi M, Karbasi R, Vahabie AH. SynthECG: Python framework and ecg image datasets for digitization, lead detection, and waveform segmentation. J Med Signals Sens 2026;16:8.

Address for correspondence:
Dr. Abdol-Hossein Vahabie, School of Electrical and Computer Engineering, University of Tehran, Tehran, Iran.
E-mail: h.vahabie@ut.ac.ir

Access this article online

Website: www.jmssjournal.net

DOI: 10.4103/jmss.jmss_58_25

Quick Response Code:

detection of regions of interest, such as lead region and lead names.[7-9] Several datasets available on Roboflow include bounding boxes around ECG leads, with a maximum sample size of 1227. However, no large-scale dataset currently provides bounding boxes for both ECG leads and their corresponding lead names. Furthermore, the U-Net[10] segmentation model has proven effective for extracting waveform traces from individual ECG lead images,[11,12] yet there is currently no public dataset that provides ECG images alongside both segmentation masks and ground truth signals. In addition, overlapping waveforms – a common occurrence in paper ECGs – pose a significant challenge for digitization. A recent nature article reported that digitization accuracy – measured by correlation with ground truth – dropped to 60% in the presence of overlapping leads.[13] Despite the importance of this problem, no open-access dataset currently includes ECG images with overlapping lead signals paired with clean (nonoverlapping) masks. Such a resource could play a key role in training deep learning models to handle this challenge.

To address the limitations in ECG image data availability, we introduce SynthECG, an open-source Python framework for generating large-scale, customizable datasets that support deep learning tasks such as ECG digitization, waveform segmentation, lead detection, and lead name recognition (e.g., Lead II, V1). Using this framework, we generate and release four ECG image datasets derived from the PTB-XL[14] time-series dataset. Our datasets include standard lead configurations such as 3 × 1, 3 × 4, 6 × 2, and 12 × 1, where each format defines the layout of leads across rows and columns (e.g., 3 rows × 4 columns in the 3 × 4 configuration). The dataset for waveform segmentation is provided in two versions: normal and overlapping. In the overlapping version, signals from adjacent leads (above or below) are superimposed onto a target lead. At the same time, the corresponding masks remain clean, containing only the true waveform of the target lead. This design enables the training and evaluation of digitization models under more realistic and challenging conditions. The full release includes:

- An open-source Python framework for generating multiple customizable and large-scale datasets
- ECG images paired with time-series signals
- ECG images annotated with object detection labels, including bounding boxes in YOLO format around each lead region and its name (e.g., Lead II, V1, etc.)
- Cropped single-lead images with pixel-level segmentation masks for U-Net-style segmentation (normal and overlapping versions) and paired time-series signals.

Related Works

Four publicly available datasets, including three large-scale ones, have recently contributed valuable resources for evaluating ECG digitization methods. ECG-Image-Kit[5] is an open-source toolkit for generating synthetic multi-lead ECG images from time-series data, incorporating realistic distortions including text artifacts, wrinkles, and creases on standard ECG paper backgrounds. Using the PhysioNet QT database,[15] the authors created a dataset of 21,801 ECG images with corresponding signals. The ECG-Image-Database[3] extended ECG-Image-Kit by combining its programmatic distortions with real-world physical effects, such as soaking, staining, and mold growth applied to printed ECGs. The physically altered ECGs were scanned or photographed under diverse lighting conditions to capture realistic imaging artifacts. The authors used 977 ECG records from the PTB-XL database and 1000 from Emory Healthcare to generate a dataset of 35,595 samples.

PTB-Image[16] used ECG-Image-Kit to generate 549 ECG images with corresponding signals, based on the PTB[17] signal dataset, providing paired ECG signals and their corresponding image formats. The PMcardio ECG Image Database[6] provides 6000 ECG images derived from 100 waveforms in the PTB-XL dataset. The dataset contains images with realistic distortions, including paper bends, crumbles, and variations introduced by scanning or capturing screens with mobile devices. It also features a wide range of augmentation techniques, including contrast and brightness changes, perspective transformations, rotations, blurring, JPEG compression, and resolution changes.

As mentioned earlier, lead detection is an important step for reaching fully automated ECG digitization. Several ECG lead detection datasets provide images annotated with bounding boxes around individual leads.[7-9] These datasets provide bounding box annotations for object detection tasks in multiple widely used formats, including COCO (JSON), Pascal VOC (XML), and YOLO (TXT). However, these datasets do not contain annotations for both lead name and lead, and the number of samples is limited. Moreover, U-Net has demonstrated strong performance in extracting waveform traces from single-lead ECG images.[11,12] However, no publicly available dataset currently offers ECG images paired with both segmentation masks and corresponding ground truth signals. Table 1 summarizes the key characteristics of our datasets alongside existing public ECG image datasets.

Methods

The open-source framework, SynthECG, is available at Github, and the open-access datasets are available at Zendo. The dataset structure is illustrated in Figure 1. Table 2 summarizes our released datasets, including their sample sizes, available file formats, and the tasks each dataset supports. Figure 2 illustrates the SynthECG pipeline that generates four datasets from a time-series signal dataset.

Table 1: Our electrocardiogram datasets versus public datasets

Dataset	Size	Artifacts	Supporting tasks			
			Digitization	LD	LND	Segmentation
Synth ECG (ours)	24.1k**	Overlapping signals (only in the segmentation dataset)	✓	✓	✓	✓
ECG-Image-Kit[5]	21.8k	Text artifacts, wrinkles, and creases	✓	×	×	×
ECG-Image-DB[3]	35.5k	Text artifacts, noise, wrinkles, stains, perspective shifts, soaking, staining, and mold growth, lighting variation	✓	×	×	×
PMcardio[6]	6k	Paper bends, crumples, screen capture distortions, contrast changes, brightness changes, perspective shifts, rotations, blurring, JPEG compression, resolution change	✓	×	×	×
PTB-Image[16]	0.6k	×	✓	×	×	×
3 Datasets on Roboflow[7-9]	1.8k	×	×	✓	×	×

**Total samples 21.4k=2k digitization + 2k lead/name detection + 20k normal segmentation + 102 overlapping segmentation. ECG – Electrocardiogram; LD – Lead detection; LND – Lead name detection, ✓– indicates the dataset supports the task; ×– indicates it does not.

Table 2: Overview of our released datasets: Size, format, and supported tasks

Dataset name	Size	Available files	Primary task
Digitization	2k	Multi-lead ECG image (.jpg) + ground truth signal (.json)	Digitization
Lead/name detection	2k	Multi-lead ECG image (.jpg) + Bounding Boxes (.txt)	YOLO-based lead region and name detection
Segmentation (normal)	20k	Single-lead ECG image (.jpg) + mask (.png/.bmp) + ground truth signal (.json)	U-net-based waveform segmentation
Segmentation (overlap)	102	Single-lead ECG image (.jpg) + clean mask (.png/.bmp) + ground truth signal (.json)	U-net-based waveform segmentation

ECG – Electrocardiogram; YOLO – You only look once

SynthECG: An open-source python framework

Our framework, SynthECG, allows researchers to generate and customize large-scale ECG image datasets directly from raw time-series signals. It supports a wide range of adjustable parameters, including dataset size (i.e., up to 21,799 images for multi-lead datasets and 261,588 for single-lead segmentation), train/validation/test split sizes, visual layout options (i.e., row height, horizontal and vertical scaling), input/output file paths, and stylistic elements such as grid visibility, lead names, font size, waveform and grid colors, and margin padding. The framework operates on time-series data converted to NumPy format (*.npy*) from the PTB-XL dataset, which is generated once and reused for efficient processing.

Time-series (PTB-XL) dataset

We selected publicly available PTB-XL[14] as the source dataset for generating synthetic ECG images due to its large and well-characterized patient cohort, availability of rich metadata and diagnostic labels, and expert validation of both signals and annotations. It contains 21,799 10-second, 12-lead ECG recordings from 18,869 patients, including all standard leads (I, II, III, aVR, aVL, aVF, V1–V6), and covers a broad patient population comprising 52% male and 48% female subjects, ranging from infants to elderly adults (0–95 years, median age 62). The ECG signals are provided in the WFDB format at a sampling frequency of 500 Hz, with an additional 100 Hz downsampled version. In addition, PTB-XL provides detailed metadata, including age, sex, height, and weight. The dataset features a wide range of cardiac diagnoses, such as normal ECG, myocardial infarction, ST/T changes, conduction disturbances, and hypertrophy, which are encoded using standardized SCP-ECG statements. These structured diagnostic labels enable pairing of our synthetic ECG images with clinical annotations, supporting future research in classification and condition-specific digitization. Furthermore, all signals were validated by technical experts for quality, and a large subset underwent additional review by a second cardiologist, ensuring reliability of both the signal and the label content.[3,14]

Digitization dataset

This dataset contains 2000 synthetic ECG images paired with ground truth signals. It supports multiple lead layouts, including 3 × 1, 3 × 4, 6 × 2, and 12 × 1. Figure 3 shows ECG images in several lead layout configurations. The ECG image contains separator lines, printed lead names (I, II, III, aVR, aVL, aVF, V1–V6), and the calibrated grid lines. Some layouts, such as 3 × 4, 6 × 2, may include lead II repeated in full 10-second length at the bottom of the image. The presence of a full lead, lead name, separating line between two leads, and gridlines can be customized using parameters in the SynthECG. Our pipeline converts raw multi-lead ECG signals directly into. jpg images that resemble standard clinical printouts. Each signal is plotted using Matplotlib on a calibrated grid, with a horizontal scale of 0.2 s and a vertical scale of 0.5 mV.

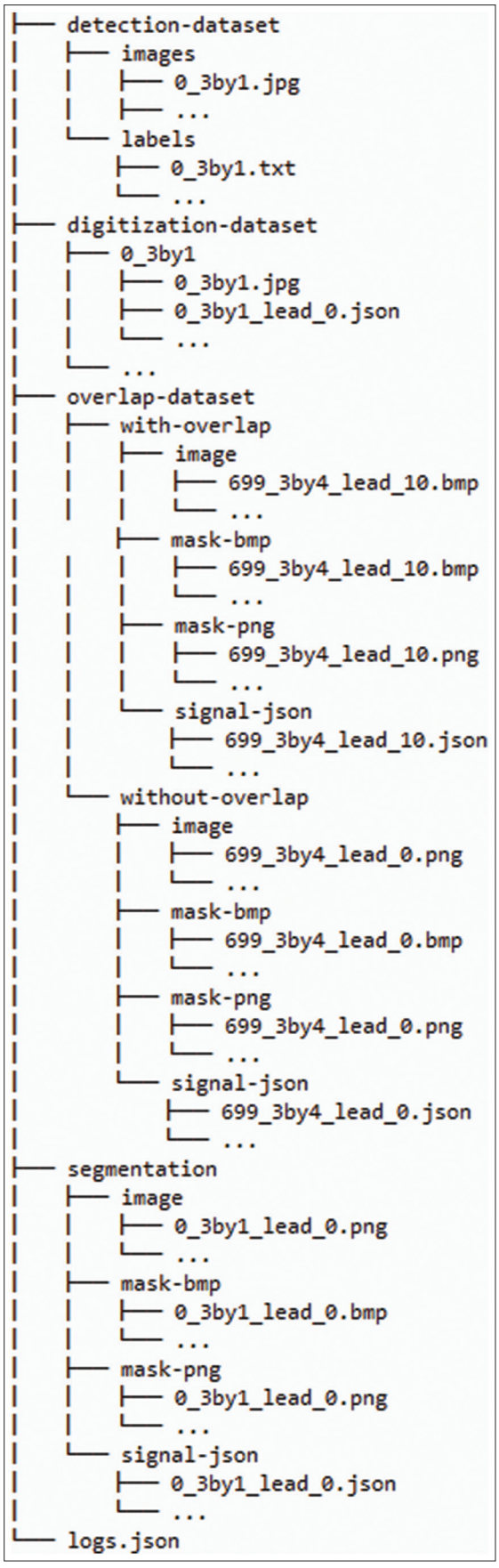

Figure 1: File structure of the ECG dataset release

Lead detection dataset

The detection dataset includes 2000 samples annotated in YOLO format for lead region and lead name detection in ECG images. Figure 4 demonstrates ECG images with different lead layouts and bounding boxes for the lead region and lead name. Each annotation is stored in a.txt file, with one line per object in the image. Each line contains five values: the class index as an integer c, the normalized coordinates of the bounding box center (x, y), and the normalized width and height (w, h), all expressed as ratios relative to the image dimensions. The top-left corner of the image is treated as the origin, with the positive x-axis extending to the right and the positive y-axis extending downward [Figure 5]. The class index c indicates the type of annotated object: $c = 0$ denotes lead waveform regions, while $c = 1$ to $c = 12$ correspond to the 12 lead names (I, II, III, aVR, aVL, aVF, V1-V6). For example, a bounding box with $c = 1$ identifies the "Lead I," and $c = 9$ corresponds to "V3."

Normal segmentation dataset

The segmentation dataset includes 20,000 cropped single-lead ECG images, corresponding .png and .bmp masks, and *.json* time-series files. The top panel of Figure 6 shows an example of a cropped ECG image alongside its corresponding *.png* mask. For each cropped lead, two types of masks are generated: a grayscale. *png* mask and a binary .bmp mask, where foreground pixels are set to 1 and background pixels to 0. The BMP format is particularly useful for training U-Net-style segmentation models.

To create the segmentation dataset, we first cropped each lead region using bounding boxes from the lead detection dataset. For mask generation, ECG images were regenerated in grayscale without grid lines or lead names. The resulting mask images were saved as *.png* files and then converted to .bmp format to produce binary masks that serve as labels for training U-Net models.

Overlapping segmentation dataset

ECG digitization accuracy drops sharply when printed leads overlap – a challenge highlighted in a recent study.[13] To address this, we created an overlapping segmentation dataset containing 102 overlapping samples. This dataset contains single-lead ECG images where signals from adjacent leads (i.e., above or below) are superimposed onto a target lead. Meanwhile, the segmentation masks remain clean, retaining only the target lead waveform. Examples of overlapping images and their clean masks are shown in the middle and bottom panels of Figure 6.

To generate this dataset, we reduced the vertical spacing between leads by adjusting the *row_height* parameter in the SynthECG framework. Reducing this parameter decreases the space between the target lead and adjacent upper or lower leads, causing waveforms

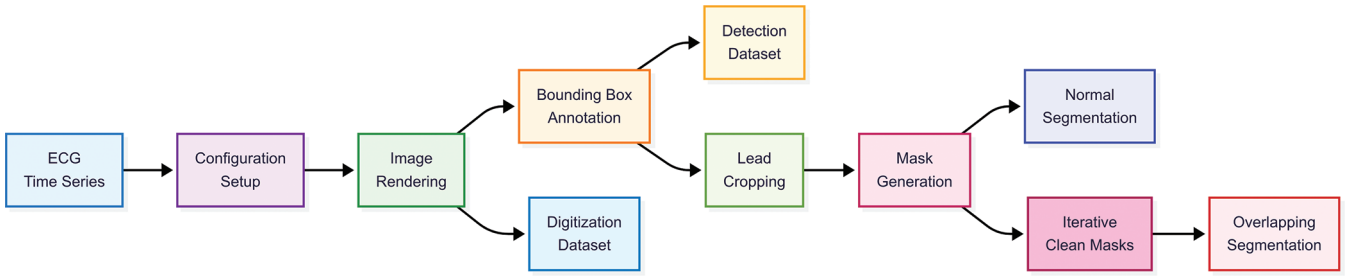

Figure 2: Proposed pipeline for generating synthetic ECG images

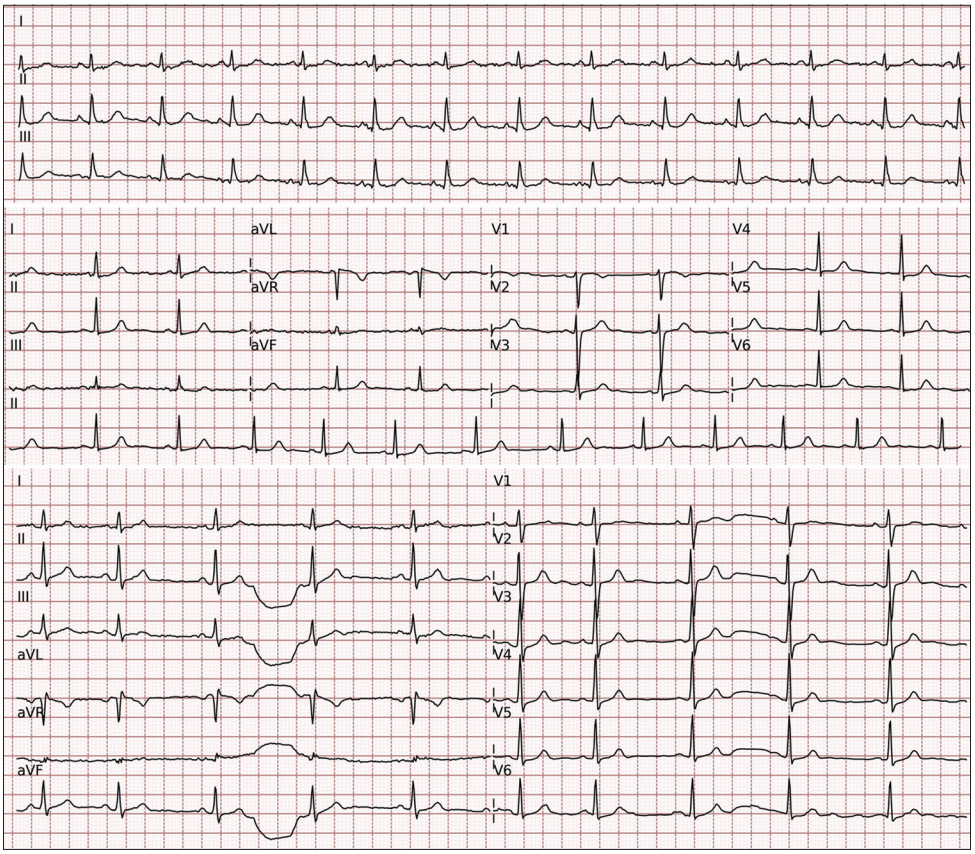

Figure 3: Sample ECG recordings displayed in different lead layout configurations: Top panel: 3×1 layout. Middle panel: 3×4 layout. Bottom panel: 6 × 2 layout

from those leads to appear partially superimposed onto the target lead's region, mimicking overlapping artifacts commonly seen in clinical printed ECGs. We then cropped individual lead regions and manually reviewed the resulting single-lead images to categorize them into two groups: With overlap (102 samples) and without overlap (186 samples).

To create the clean segmentation masks (i.e., U-Net labels), we re-rendered grayscale versions of the ECG images using the same configuration parameters as the original images but with several key modifications: Gridlines and lead names were removed, the background was set to black, and the ECG waveforms were plotted in white. This produces a binary-style image suitable for U-Net segmentation. To ensure clean segmentation masks, we employed an iterative approach for each lead in the ECG. For a 12-lead image, this process was repeated 12 times, once for each lead. In each iteration, we plotted only the waveform of the target lead, and we did not plot the waveforms of any other leads. However, the regions corresponding to the nontarget leads were still rendered as empty areas using the same layout configuration to preserve spatial consistency. We then cropped the target lead's region to generate its corresponding mask. This ensured that each mask captured solely the target lead's waveform, fully isolated from any overlapping or adjacent signals.

Dataset Validation

To assess the quality and practical utility of our datasets, we conducted two case studies: (1) we evaluated the

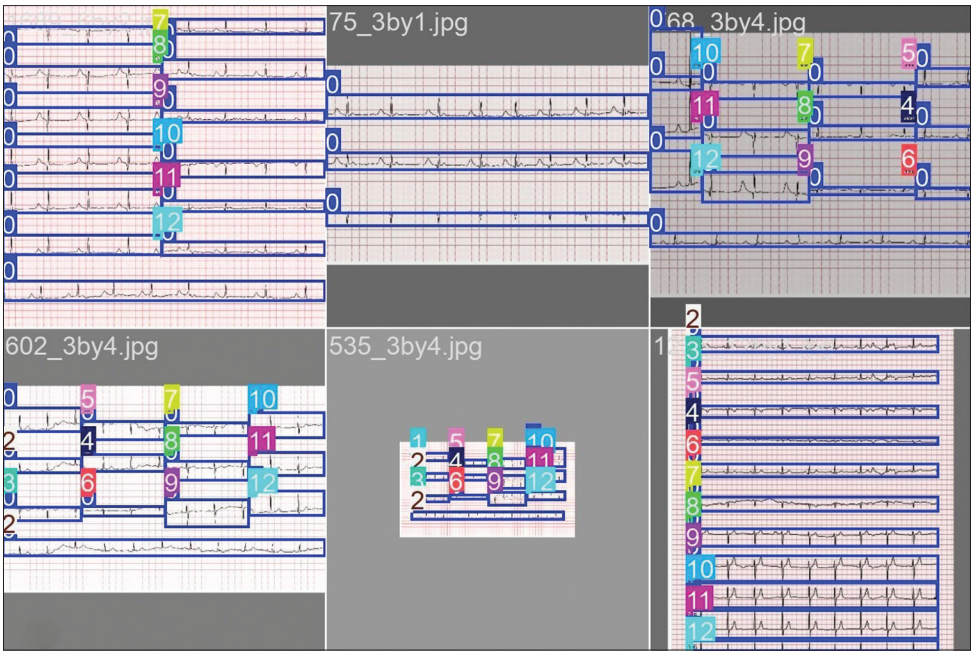

Figure 4: Detection dataset samples: ECG images annotated with bounding boxes for lead regions and lead names. The color of each box indicates the class ID. Class 0 corresponds to lead waveform regions, whereas classes 1 to 12 represent the lead names I, II, III, aVR, aVL, aVF, and V1–V6, respectively. The bounding boxes for lead names are notably smaller than those for the waveform regions

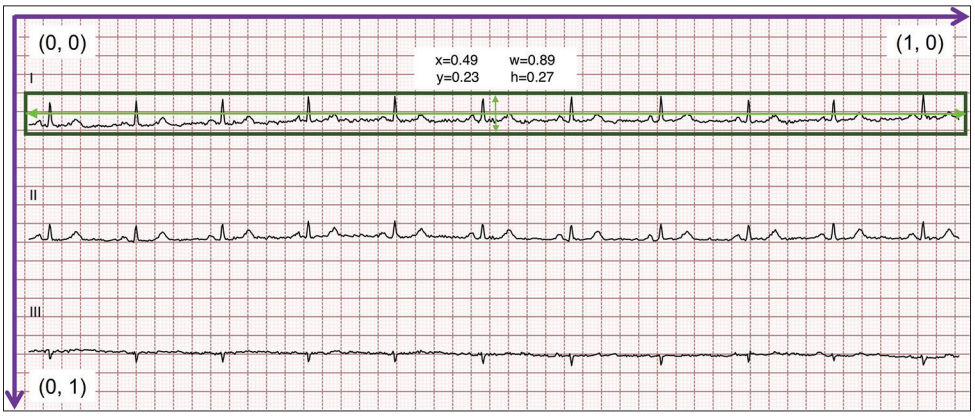

Figure 5: Illustration of the YOLO bounding box format, which includes the class ID, normalized center coordinates (x, y), and the bounding box width and height (w, h), all expressed relative to the image dimensions

Figure 6: Top – Single-lead ECG image with its corresponding mask shown on the right. Middle and bottom – Overlapping single-lead ECG images with clean masks displayed on the right

digitization dataset using an open-source digitization algorithm developed by Fortune *et al*.[18] and (2) we validated the lead region and lead name detection datasets by training a YOLOv8 object detection model.

Case study 1: ECG digitization

We evaluated our digitization dataset using the open-source algorithm from Fortune *et al*.[18] implemented in.[19] As the algorithm operates on single-lead ECG images, we cropped individual leads from our multi-lead configurations to match the required input format. The digitization pipeline converts single-lead images to binary masks, estimates horizontal grid spacing, and extracts amplitude arrays. We applied postprocessing, including resampling to 100 Hz and baseline correction via median subtraction. Testing was performed on 285 samples, comprising 100 with overlapping signals and 185 without overlap, using mean squared error (MSE) and Pearson correlation coefficient (ρ) against ground truth signals. On the normal samples, the algorithm achieved an MSE of 0.002 and a

correlation of 0.93, consistent with the original paper's results.[18] This outcome supports that our synthetically generated ECG images preserve appropriate ECG structure and signal fidelity for digitization tasks. As expected, performance degraded on overlapping samples (MSE: 0.018, ρ: 0.86), since the algorithm was not designed to address waveform overlaps. Figure 7 demonstrates the original signal and the digitized outputs for both normal and overlapping single-lead ECG images.

In addition to these metrics, we computed the signal-to-noise ratio (SNR) and its median-based variant (SNR_{med}) to assess digitization performance. SNR was computed as:

$$SNR = \frac{mean_k(y_k^2)}{mean_k[(y_k - \hat{y}_k)^2]}$$

Figure 7: Digitization performance using the method from Fortune *et al.*[18] on our single-lead ECG dataset. Top: input single-lead ECG images. Bottom: extracted signals (red) overlaid on original images. Left: example from the normal dataset; Right: example from the overlapping dataset

7 where y_k is the ground truth signal and $\hat{y}_k$ is the digitized signal. The SNR_{med} uses the median noise power in the denominator for greater robustness to outliers:

$$SNR_{med} = \frac{mean_k\left(y_k^2\right)}{median_k\left[\left(y_k - \hat{y}_x\right)^2\right]}$$

On the normal dataset, the digitized outputs achieved an average SNR of 7.36 ± 3.89 dB and an SNR_{med} of 37.86 ± 8.46 dB. On overlapping samples, performance declined slightly, with an average SNR of 5.78 ± 3.99 dB and an SNR_{med} of 37.66 ± 7.87 dB, reflecting the algorithm's sensitivity to waveform interference. Figure 8 presents the histograms of the SNR and SNR_{med} distributions across samples.

Case study 2: Lead and lead name detection

To illustrate the applicability of our lead detection dataset for object detection tasks, we trained a YOLOv8-nano model to detect both lead waveform regions and lead name labels. The dataset includes annotations for 13 object classes: One class for the lead region and 12 classes corresponding to individual lead names (I, II, III, aVR, aVL, aVF, V1–V6). We trained the model for 100 epochs using default YOLOv8 settings with a minimal augmentation strategy to preserve visual characteristics critical to detection. The applied augmentations included limited HSV color space variations (hue: 0.015, saturation: 0.7, value: 0.4), image scaling (±0.5), and minor BGR shifts (±0.1). Figure 9 presents detection results on the validation set using the

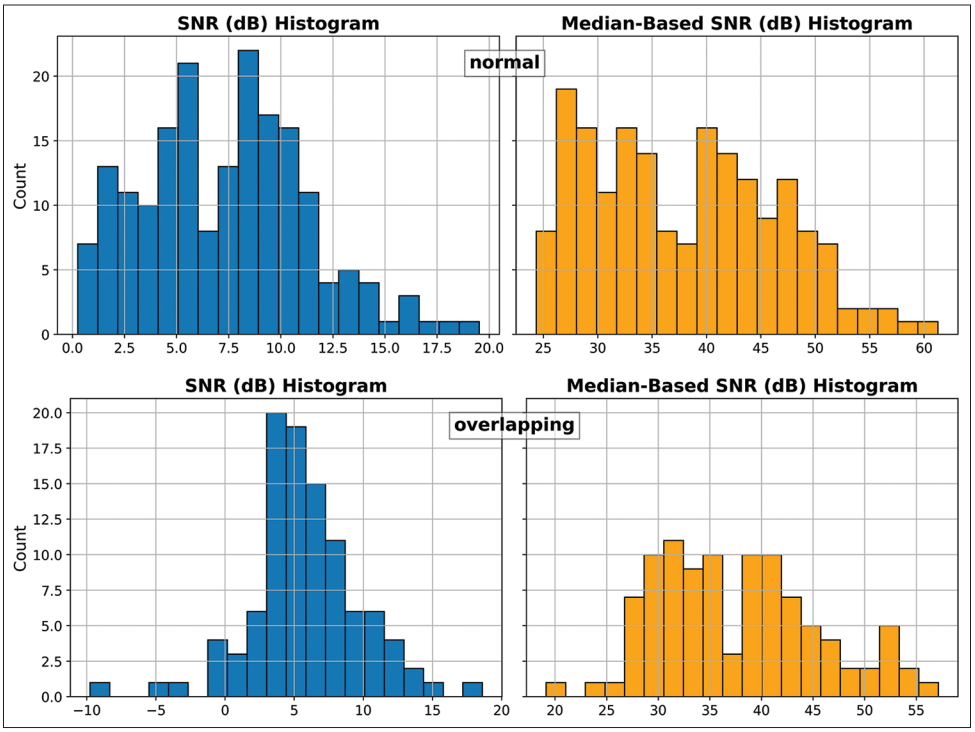

Figure 8: Histogram distributions of signal-to-noise ratio (SNR) (left) and median-based SNR (right) for the digitization results. Top row corresponds to the normal dataset; bottom row shows results for the overlapping dataset

YOLOv8-nano model, showing predicted bounding boxes around lead waveform regions and lead name labels, along with associated class confidence scores. Figure 10 displays the training and validation performance metrics over 100 epochs, including bounding box loss, classification loss, distribution focal loss (used for precise bounding box localization), precision, recall, and mean average precision. This case study demonstrates the usability of our annotated dataset for training lightweight object detection models tailored to ECG-specific tasks.

Discussion

The results of our two case studies demonstrate the practical usability of the SynthECG datasets. The digitization dataset supported accurate waveform reconstruction aligned with existing benchmarks. Compared to ECG-Image-Kit, which reported an SNR of 12.7 ± 4.8 dB and a median-based SNR of 27.6 ± 5.4 dB, our results (7.36 ± 3.89 dB and 37.86 ± 8.46 dB, respectively) demonstrate comparable performance despite differences in dataset generation and digitization algorithm. This supports that our synthetic images preserve key ECG signal characteristics relevant for downstream digitization tasks. The detection dataset enabled effective lead and lead name localization using a YOLOv8 model. These results validate the utility of our synthetic datasets. By generating diverse, task-specific datasets from a shared source of time-series signals, SynthECG fills a critical gap in the availability of standardized ECG image datasets. Notably, the inclusion of overlapping waveform scenarios with clean segmentation masks introduces a novel and previously unavailable resource for evaluating model performance under challenging conditions. In future work, we plan to leverage this overlapping segmentation dataset to train and evaluate digitization models specifically designed to handle waveform interference caused by lead overlaps.

While the current version of SynthECG does not include visual artifacts such as wrinkles, stains, or scanner distortions, it is designed to be modular and easily extensible. In future work, we plan to simulate common real-world imperfections in printed or scanned ECGs to better reflect clinical scenarios. These may include text overlays, creases, paper bends, crumples, glare, and smudges, as well as digital distortions such as

Figure 9: Detection example from the validation set using YOLOv8. Each bounding box represents either a lead region or a lead name, and the accompanying value indicates the model's confidence score. Bounding boxes for the same class share the same color for clarity

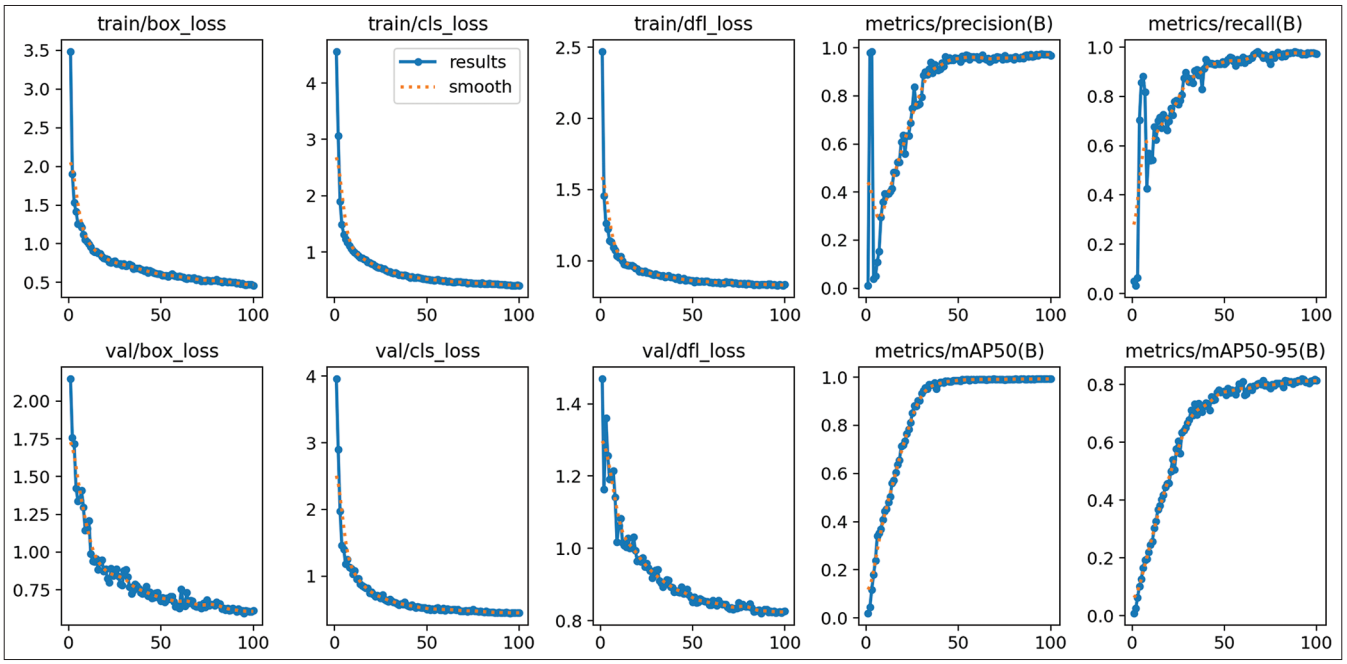

Figure 10: Training and validation performance of the YOLOv8-nano model over 100 epochs. Metrics include bounding box loss, classification loss, Distribution focal loss for localization, precision, recall, and mean average precision

brightness/contrast shifts, perspective warping, rotation, blurring, JPEG compression, and mobile capture artifacts. Our open-source framework allows researchers to implement these suggested artifacts or to develop additional ones, supporting broader use cases and more robust model training and evaluation.

In generating the segmentation datasets, a manual review of single-lead ECG images (n = 288) was conducted to separate overlapping from nonoverlapping samples. We acknowledge that this manual approach is not scalable for large-scale dataset development. To address this limitation, we suggest a future direction: The development of a lightweight Python-based graphical user interface to support rapid human verification of overlap presence. This tool would sequentially display single-lead ECG images and allow the reviewer to assign labels using simple keyboard input (e.g., pressing "1" for overlapping and "2" for nonoverlapping). Such an interface could reduce per-image review time to under 10 seconds, making it feasible to annotate 20k samples within a reasonable timeframe (e.g., ~42 h). This level of effort would enable the creation of a human-validated segmentation dataset.

Ethical approval

This study is entirely based on the PTB-XL[14] dataset, a large, publicly available, and fully anonymized resource published in Scientific Data. The creators of PTB-XL obtained approval from their Institutional Ethics Committee (Approval ID: PTB-2020-1) to release these data in an open-access format. We did not collect or access any new patient data. All data used were already anonymized and ethically approved for open access. This study solely involved processing PTB-XL signals to generate synthetic ECG image datasets.

Funding

This research received no funding.

Availability of data and materials

The full datasets and code are publicly available at: https://doi.org/10.5281/zenodo.15484519 and https://github.com/rezakarbasi/ecgimage-and-signal-dataset, respectively.

Financial support and sponsorship

Nil.

Conflicts of interest

There are no conflicts of interest.

References

1. Birnbaum Y, Nikus K, Kligfield P, Fiol M, Barrabés JA, Sionis A, *et al*. The role of the ECG in diagnosis, risk estimation, and catheterization laboratory activation in patients with acute coronary syndromes: A consensus document. Ann Noninvasive Electrocardiol 2014;19:412-25.
2. Adedinsewo DA, Siddiqui H, Johnson PW, Douglass EJ, Cohen-Shelly M, Attia ZI, *et al*. Digitizing paper based ECG files to foster deep learning based analysis of existing clinical datasets: An exploratory analysis. Intell Based Med 2022;6:100070.
3. Reyna MA, Weigle J, Koscova Z, Campbell K, Shivashankara KK, Saghafi S, et al. ECG-image-database: A dataset of ECG images with real-world imaging and scanning artifacts; a foundation for computerized ECG image digitization and analysis. arXiv preprint arXiv:2409.16612. 2024. doi.org/10.48550/arXiv.2409.16612.
4. Lence A, Extramiana F, Fall A, Salem JE, Zucker JD, Prifti E. Automatic digitization of paper electrocardiograms – A systematic review. J Electrocardiol 2023;80:125-32.
5. Shivashankara KK, Mehri Shervedani A, Clifford GD, Reyna MA, Sameni R. ECG-Image-Kit: A synthetic image generation toolbox to facilitate deep learning-based electrocardiogram digitization. Physiol Meas 2024;45:ad4954.
6. Iring A, Krešňáková V, Hojcka M, Boza V, Rafajdus A, Vavrik B. PMcardio ECG Image Database (PM-ECG-ID): A Diverse ECG Database for Evaluating Digitization Solutions; 2024. Available from: https://doi.org/10.5281/zenodo.13617673. [Last accessed on 2025 Nov 24].
7. A I. ECG Dataset; 2023. Available from: https://universe.roboflow.com/ia-q0vuc/ecg-ijg1u. [Last accessed on 2025 Aug 27].
8. Enetcom. Ecg Lead Detection Dataset; 2023. Available from: https://universe.roboflow.com/enetcom-sfidt/ecg-lead-detection. [Last accessed on 2025 Aug 27].
9. ECG Artivatic. ECG Final Dataset; 2021. Available from: https://universe.roboflow.com/ecgartivatic/ecg_final_2. [Last accessed on 2025 Aug 27].
10. Ronneberger O, Fischer P, Brox T. U-net: Convolutional Networks for Biomedical Image Segmentation. Cham, Switzerland: Springer; 2015. p. 234-41.
11. Li Y, Qu Q, Wang M, Yu L, Wang J, Shen L, *et al*. Deep learning for digitizing highly noisy paper-based ECG records. Comput Biol Med 2020;127:104077.
12. Demolder A, Kresnakova V, Hojcka M, Boza V, Iring A, Rafajdus A, *et al*. High precision ECG digitization using artificial intelligence. J Electrocardiol 2025;90:153900.
13. Wu H, Patel KH, Li X, Zhang B, Galazis C, Bajaj N, *et al*. A fully-automated paper ECG digitisation algorithm using deep learning. Sci Rep 2022;12:20963.
14. Wagner P, Strodthoff N, Bousseljot RD, Kreiseler D, Lunze FI, Samek W, *et al*. PTB-XL, a large publicly available electrocardiography dataset. Sci Data 2020;7:154.
15. Laguna P, Mark RG, Goldberg A, Moody GB. A Database for Evaluation of Algorithms for Measurement of QT and other Waveform Intervals in the ECG. IEEE; 1997. p. 673-6.
16. Nguyen CV, Nguyen HX, Minh DD, Do CD. PTB-Image: A scanned paper ECG dataset for digitization and image-based diagnosis. arXiv preprint arXiv:2502.14909. 2025. doi.org/10.48550/arXiv.2502.14909.
17. Bousseljot R, Kreiseler D, Schnabel A. Use of the PTB CARDIODAT ECG signal database via the Internet. Biomedizinische Technik. 1995;40:317-8.
18. Fortune JD, Coppa NE, Haq KT, Patel H, Tereshchenko LG. Digitizing ECG image: A new method and open-source software code. Comput Methods Programs Biomed 2022;221:106890. Available from: https://www.sciencedirect.com/science/article/abs/pii/S0169260722002723. [Last accessed on 2025 Jul 28].
19. GitHub – Tereshchenkolab/paper-ecg: OSU Capstone Project 2020-21 – Natalie and Julian. Available from: https://github.com/Tereshchenkolab/paper-ecg. [Last accessed on 2025 Jul 28].