

Digitizing Paper ECGs at Scale: An Open-Source Algorithm for Clinical Research

Elias Stenhede^{1,2}, Agnar Martin Bjørnstad^{1,2}, Arian Ranjbar¹

¹Medical Technology & E-health, Akershus University Hospital, 1478 Lørenskog, Norway

²Faculty of Medicine, University of Oslo, 0317 Oslo, Norway

Millions of clinical ECGs exist only as paper scans, making them unusable for modern automated diagnostics. We introduce a fully automated, modular framework that converts scanned or photographed ECGs into digital signals, suitable for both clinical and research applications. The framework is validated on 37,191 ECG images with 1,596 collected at Akershus University Hospital, where the algorithm obtains a mean signal-to-noise ratio of 19.65 dB on scanned papers with common artifacts. It is further evaluated on the Emory Paper Digitization ECG Dataset, comprising 35,595 images, including images with perspective distortion, wrinkles, and stains. The model improves on the state-of-the-art in all subcategories. The full software is released as open-source, promoting reproducibility and further development. We hope the software will contribute to unlocking retrospective ECG archives and democratize access to AI-driven diagnostics.

Cardiovascular disease remains the leading cause of death globally, underscoring the importance of effective diagnostic tools [1]. Among these, the ECG is the most widely used, primarily due to its ease of use, non-invasiveness, low operator variability, and cost-effectiveness. Traditionally, ECG recordings have been acquired using analog instruments that produce printed signals on paper strips, subsequently analyzed by clinicians and archived in patient medical records. Despite technological advances, this practice persists today due to the familiarity for clinicians, convenience, and ease of interpretation. With the widespread adoption of electronic health records, these paper ECGs are now often scanned to be stored as images within the patient database.

Advancements in AI, particularly with the advent of deep learning during the last decade, have significantly improved computer-assisted analysis of ECGs [2], [3], even surpassing human diagnostic performance [4], [5]. State-of-the-art algorithms primarily utilize ECG data in standardized time series formats to perform both diagnostic classification [6], [7], [8], and risk stratification [9]. Furthermore, beyond typical cardiovascular assessment, similar ECG-based AI models have shown potential in e.g., identifying demographic characteristics [10].

While the healthcare sector is gradually adopting digital storage of original ECG signals, many hospitals and healthcare providers continue to rely on scanned paper records. Although AI models can analyze scanned ECG images, their performance and reliability significantly im-

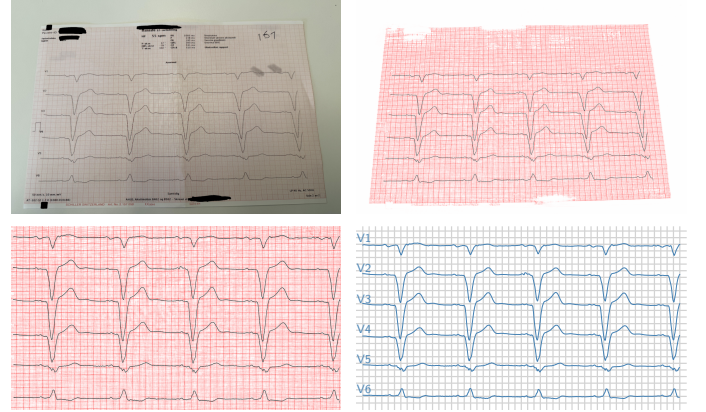


Figure 1: Shows the digitization pipeline for a paper ECG captured using a mobile phone. **Top left:** Mobile phone photo of a printed ECG. **Top right:** Segmented image showing detected ECG waveform and grid structure. **Bottom left:** Perspective-corrected and cropped ECG aligned. **Bottom right:** Final extracted waveforms in standardized format for archiving and downstream processing.

prove when standardized digital time-series data are available, mitigating complications arising from variations in channel configurations and scanning artifacts. Developing a robust and efficient tool to digitize paper ECG records accurately offers two critical advantages. First, it democratizes access to AI-driven ECG interpretation tools, making advanced diagnostic capabilities available regardless of digital infrastructure. Second, digitizing historical ECG archives substantially expands the volume of available data for training AI models, potentially by several orders of magnitude. This enhancement is particularly valuable in studying rare cardiovascular conditions, where extensive historical datasets can significantly increase sample sizes.

Related work

Over the past two decades, several semi-automatic methods have been proposed to address the challenge of ECG digitization, including Badilini et al., Ravichandran et al., and Baydoun et al., who in the period between 2005 and 2019 introduced MATLAB-based solutions [11], [12], [13], all requiring some degree of user input. In 2021, Mishra et al. proposed an algorithm [14] for crops of single leads. By 2021, Fortune et al. provided an openly available Python implementation of their semi-automatic method [15].

In the last few years, efforts have been directed to-

wards fully automated methods. In 2022, Wu et al. proposed a method including an accessible API for digitization tasks [16]. We were, however, not successful in digitizing any images from our database using the API. To accelerate the development of a general automated ECG digitization algorithm, the annual George B. Moody PhysioNet Challenge announced that the 2024 competition would be dedicated to the topic. This resulted in 17 distinct methods from the competing research groups, which were evaluated on a hidden test dataset consisting of scans and photographs of ECGs in varying conditions, including clean, strained, and significantly deteriorated samples. Performance was assessed quantitatively with the ground truth reference [17]. Among the methods, three demonstrated superior performance compared to a naive baseline of assigning zero values to all signal points [18], [19], [20]. Nevertheless, despite surpassing the baseline, none showed robust results on photos, and the source code is not available at the time of writing. In 2025, Demolder et al. introduced an ECG digitization method and released an evaluation dataset comprising 6,000 ECG images [21] derived from 100 ECGs. A commendable aspect of their work is the physical realism introduced by printing and photographing ECGs, mimicking real-world digitization scenarios. It does not, however, capture common degradations of handwritten notes by clinicians or variations introduced by printing on thermal paper, and the source code is not shared.

In summary, previous methods have been constrained by one or more of: (1) lack of automation or accuracy, (2) insufficient validation across realistic datasets, and (3) the absence of open-source code. To address these issues, we propose a new modular open-source framework for a general-purpose ECG digitization algorithm, capable of digitizing both scans and photos, validated on both publicly available datasets and real-world data. In short, our contributions include:

- A fully automated modular framework for general-purpose ECG digitization consisting of five modules: Perspective correction and cropping, Segmentation, Layout identification, Grid size extraction/scaling, and 2D to 1D conversion; publicly released as open source.
- Methodological and algorithmic advances, including dewarping, alignment, and signal extraction methods. We also release a training dataset of 8,658 simulated high-resolution ECG images with labels for segmentation and ground-truth signals, enabling reproducibility and further development.
- A real-world clinical dataset collected at Akershus University Hospital, containing 1,596 samples of photographed and scanned paper ECGs, provided together with the raw time series data.
- Extensive validation of the proposed framework on real-world datasets.

Results

The full ECG digitization framework is released with all code, the synthetic training dataset, and the clinical dataset from the prospective validation, as open source. In

this section*, we present the clinical data and results from the validation.

Validation data

The proposed algorithm is evaluated using two main datasets, the Ahus Paper Digitization ECG Database and the Emory Paper Digitization ECG Database. The first contains paper ECGs from Akershus University Hospital in Norway, reflecting data quality in a real-world clinical setting. The second dataset was used as the hidden test set in the 2024 George Moody PhysioNet Challenge. At the time of writing, the full Emory dataset is hidden, representing a zero-shot evaluation. Following the publication of the current article, both the Ahus and Emory Paper ECG Databases will be jointly released as a unified dataset. Both datasets are described in further detail in [22].

Validation metrics

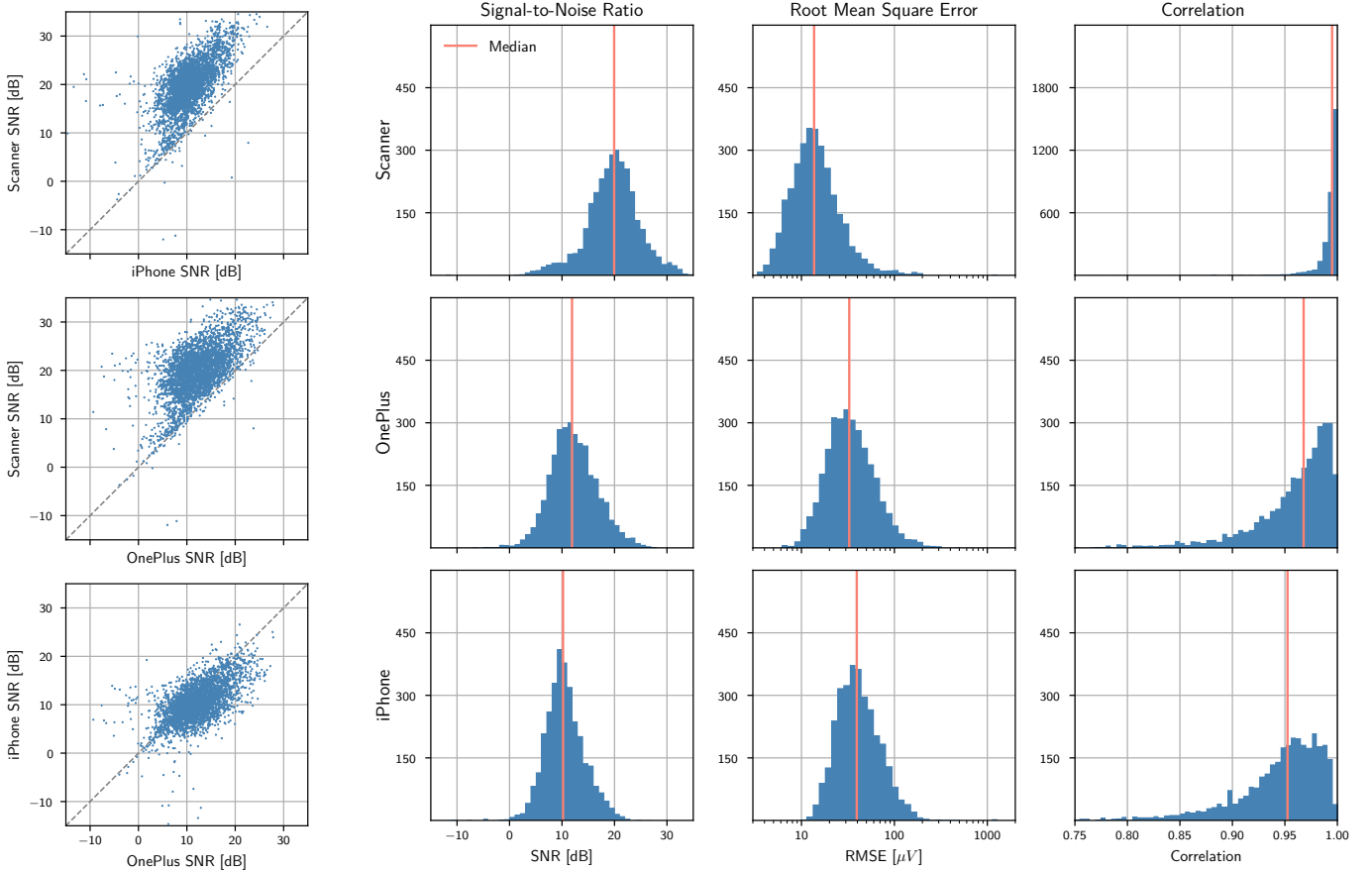
We use signal-to-noise ratio (SNR) as the main metric, and additionally report correlation and root mean squared error (RMSE). Metrics are calculated per lead and analyzed separately for the scanner and mobile phone cameras. The ground truth signals are sampled at 1000 Hz, and all reconstructed signals are resampled to match this rate. Denoting y as the ground truth signal and $\hat{y}$ as the reconstruction, SNR is calculated as

$$\text{SNR} = 10 \log_{10} \left(\frac{\sum_t y[t]^2}{\sum_t (y[t] - \hat{y}[t])^2} \right),$$

where the numerator is the power of the true signal and the denominator is the power of the noise signal. In evaluation, the ground truth and digitized signals are aligned using horizontal and vertical shifts. The horizontal alignment allows for a maximum shift of 100 ms, while the vertical alignment is performed by zero-centering both signals. Shifting is motivated by (a) there being no fixed baseline in the paper ECG and (b) translation having no clinical significance, and (c) previous work uses shifted metrics [18], [19], [20], [21].

Ahus Paper Digitization ECG Database

Paper ECGs were collected from the Department of Cardiology at Akershus University Hospital over 3 months in 2025. ECGs were printed on thermal paper with 1 mm grid lines, 50 mm/s, 10 mm/mV, then photographed with two mobile phone camera setups (iPhone: 5712×4284 pixels, OnePlus: $f/1.78$ and 4624×3468 pixels, $f/1.7$) and scanned into flatbed scans (7016×4964 pixels, 600 dpi). Each scan was then paired with the raw time series data extracted from the hospital’s databases [23]. As each 12-lead is printed on two papers, every ECG results in 6 images. The resulting dataset comprises 1,596 images derived from 266 unique ECGs and 266 unique patients, with 99 identified as female, 153 as male, and 14 with unknown gender due to no personal information being entered. Each image was de-identified and stored together with the raw time series in a DICOM file. The images and 266 corresponding de-identified DICOM files were saved for downstream processing and analysis [24]. Using the hardware described in Table 3, processing one image took on average 5.88 s, with a minimum processing time of 5.43 s and a maximum



(a) Pairwise scatter plot of SNR (b) Distribution of signal quality metrics for ECG recordings captured using three devices (Scanner, iPhone, OnePlus). Using the scanner results in better average results across all three metrics with the three different devices. The two phones perform similarly, with OnePlus resulting in slightly better results.

Figure 2: Digitization metrics across devices.

of 7.45s, across the 1,596 images in the dataset. Using the same setup but on CPU only, processing one image took on average 16.78s with the minimum and maximum values being 15.53s and 23.39s. On average, the signals were shifted by 0.86 ms (s.d. 1.66 ms) for scanner images, 1.89 ms (s.d. 4.31 ms) for iPhone images, and 1.96 ms (s.d. 5.23 ms) for OnePlus images. A histogram of time shifts is shown in Figure 3. The algorithm may assign an error code to individual elements (sets the value to NaN) in the reconstruction, for example, if a lead is not visible or if the algorithm partly fails. Across all three devices, an average of 0.21 % of samples were set to NaN. Specifically, 0.15 % of samples were set to zero for the scanned paper, with corresponding numbers for OnePlus and iPhone being 0.23 % and 0.25 %.

We observed a mean SNR across all 12 leads of 19.65 dB for scanned images, compared to 12.19 dB for OnePlus photos and 10.47 dB for iPhone photos. Corresponding RMSE and correlation values also followed this ranking, see Table 1. The same results are also displayed in Figure 2b, allowing for assessment of the spread, and pairwise SNR comparisons across all leads and devices are plotted in Figure 2a.

To illustrate representative reconstructions from our algorithm and highlight the limitations of SNR, we plot lead II of five signals from the scanner, iPhone, and OnePlus cameras. The signals are selected based on their respective SNR, and correspond to the 5th, 25th, 50th, 75th,

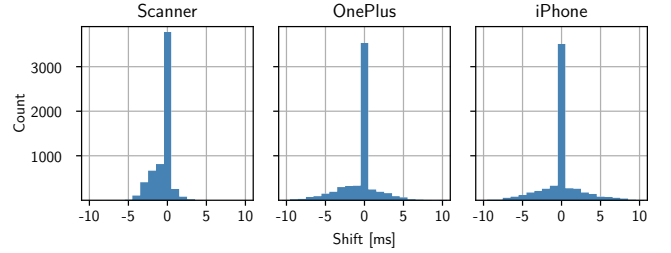


Figure 3: Shows the necessary shift to align each lead with the ground truth. In general, smaller shifts are needed for images captured with a flatbed scanner, and for all three devices, no shift is the most common.

and 95th percentiles. Signals reconstructed from the scanner are shown in Figure 4, iPhone images in Figure 5, and OnePlus images in Figure 6.

Emory Paper Digitization ECG Database

The dataset comprises 35,595 paper ECG images with print speed of, 25 mm/s, and 10 mm/mV. The images are captured under varying conditions, including color scans, black-and white scans, and mobile photos of papers that had been partially degraded, stained, or left to mold. This allows for systematic assessment of model performance across varying conditions [22].

Table 1: Mean and standard deviation (s.d.) of evaluation metrics across the three devices tested.

	SNR [dB]		RMSE [μ V]		Correlation [-]	
	Mean	s.d.	Mean	s.d.	Mean	s.d.
Scanner	19.65	5.20	19.3	35.4	0.987	0.038
OnePlus	12.19	4.65	42.8	44.8	0.947	0.083
iPhone	10.47	3.94	51.3	80.3	0.935	0.080

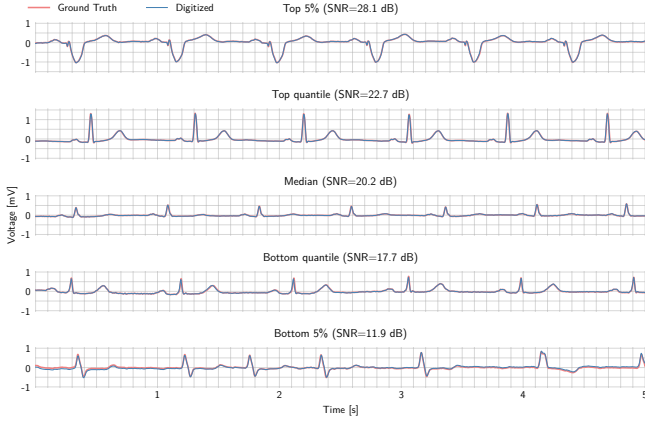


Figure 4: Lead II as found in the original recording and after digitization of high-resolution **scanner** images using our proposed algorithm. Five different ECGs are selected based on their SNR. In the lowest-SNR examples (bottom two), a slight baseline drift appears in the scanned-digitized traces, possibly due to the error margins in detecting the rotation of the paper.

As the Emory Paper Digitization ECG Database was used as the hidden test set for the 2024 George Moody PhysioNet Challenge, it is possible to directly compare our proposed algorithm with previously published methods, even as some of them have not publicly released their code. The main metric used in evaluation is shifted SNR, and results are presented in Table 2. Our proposed algorithm outperforms previously published work across all types of images. Just as on the Ahus Paper Digitization ECG Dataset, the algorithm performs best on scans, achieving an SNR of 7.34dB. There is only a minor performance drop for black-and-white scans, with a corresponding SNR of 7.24dB. For mobile photos and scans of deteriorated papers, the algorithm achieves SNRs ranging from 1.05 to 2.03dB. Somewhat surprisingly, the algorithm performs the best on stained papers in the mobile photo category.

Discussion

The algorithm shows strong performance across all three scenarios, both quantitatively and qualitatively. The average processing time of 5.88s is competitive compared to other methods, with the fastest competitor being [21] reporting an average of 4.86s with a minimum of 3.88s and a maximum of 6.33s. However, the authors do not specify the environment used in obtaining these measurements, nor share code allowing for replication.

Our quantitative results on the Ahus Paper Digitization ECG Dataset showed an average SNR of 19.65dB on

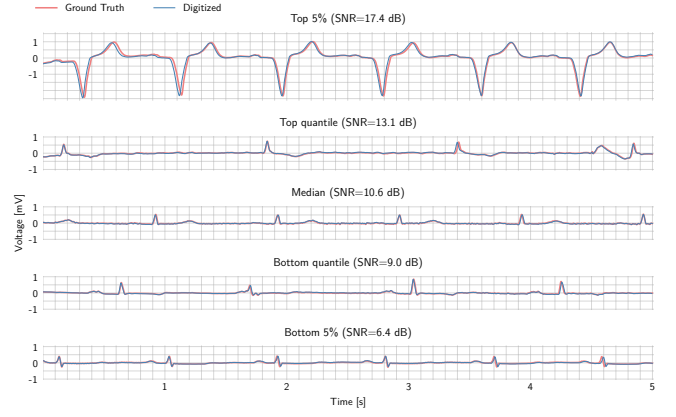


Figure 5: Lead II as found in the original recording and after digitization of images taken with an **iPhone** Pro Max 16 using our proposed algorithm. Five different ECGs are selected based on their SNR. The signals are not always perfectly aligned in time, resulting in lower SNR as compared to scanner images.

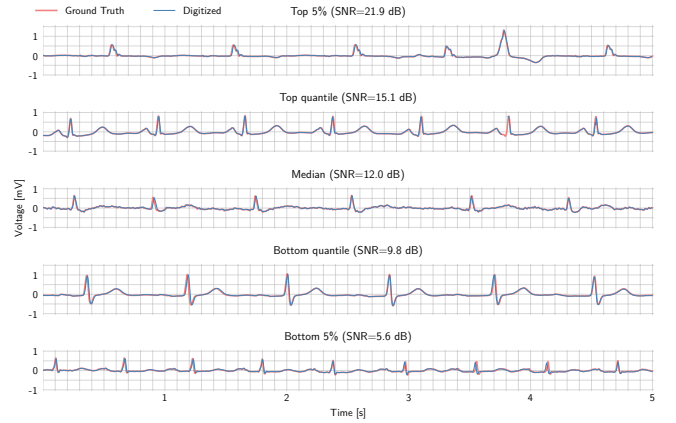


Figure 6: Lead II as found in the original recording and after digitization of images taken with a **OnePlus** Nord CE 2 using our proposed algorithm. Five different ECGs are selected based on their SNR. The signals are not always aligned in time, resulting in lower SNR as compared to scanner images.

scanned images is the highest reported SNR across any previous work, even considering closed-source and data claims. The SNR is lower for mobile photos, reflecting the increased difficulty in handling paper that is often not flat, uneven lighting, and motion blur. However, we believe that the reconstruction quality is still sufficient in this case.

Results on the Emory Paper Digitization ECG Dataset, while surpassing previously proposed methods, were comparatively lower. One likely explanation is the higher printing speed of 25mm/s, which results in greater information loss during the original signal printing compared to 50mm/s at Ahus. Notably, our algorithm is the first to achieve positive SNR across all categories, including mobile photos and degraded paper samples. The remaining performance gap might stem from curvature in deteriorated papers, misaligning digitized signals with their ground truth. With the public release of the Emory Paper Digitization ECG Dataset, identification of main bottlenecks will be possible, enabling further development of the methods.

Table 2: SNR [dB] by capture condition across the Emory Paper ECG Database. Negative scores are not shown, as an SNR of zero decibels can be achieved by setting the digitized signal to all zeros. Det. stands for deteriorated, the best values are bold, while the second best are underscored.

	Scans				Mobile photos			Screenshots
	Color		B&W		Color			Color
	Clean	Det.	Clean	Det.	Clean	Stained	Det.	Clean
Open ECG Digitizer	7.34	1.82	7.24	1.73	1.05	2.03	1.59	2.11
Yoon et al. [19]	<u>6.22</u>	<u>0.91</u>	4.74	0.36	–	–	<u>0.81</u>	<u>0.12</u>
Jammoul et al. [25]	5.18	–	<u>4.89</u>	–	–	–	–	–
Krones et al. [18]	4.93	0.51	3.48	<u>0.51</u>	–	–	–	–
Stenhede et al. [20]	3.32	0.67	2.78	–	<u>0.14</u>	–	0.79	–

The qualitative results show that in cases with lower SNR, misalignment is likely the main contributing cause of seemingly degraded performance, rather than obvious morphological inaccuracies. Unlike previously proposed ECG digitization solutions, our algorithm does not reject challenging images, but instead sets leads to NaN at points where it is not capable of digitizing. This comes with both advantages and disadvantages, which should be considered in clinical applications. One suggested approach could be to let the user manually determine what appropriate steps are for each challenging sample. Examples of cases with NaNs in the reconstructions are shown in Figure 7.

Limitations and future work

All work is best understood in light of its limitations. We present an algorithm for digitizing paper ECGs at scale, suitable for use on a computer. To fully unlock the potential of this algorithm in clinical practice, its efficacy needs to be validated in further studies. Similar studies have been conducted, but not for open-source software [26]. Another limitation is the use of standardized metrics such as SNR, correlation, and RMSE. These metrics might not always be completely aligned with the clinical utility of an algorithm. However, we believe that reporting quantitative data is suitable in the first stage of algorithm development, and that further validation should be performed to confirm the clinical utility. Lastly, it is always important to consider the effect of domain shift in evaluating deep learning models. We try to demonstrate the robustness of our algorithm by training it on data very unlike the testing datasets. The algorithm will likely need tweaking under certain circumstances. To aid in tweaking, we provide instructions for debugging the algorithm along with the source code.

Future directions also include improvements to the methodology. While we have aimed to keep the building blocks of our algorithm modular to enable further improvements and modifications, an end-to-end approach might prove superior, especially for the segmentation-to-trace conversion, as it is difficult to build rule-based solutions in cases of overlapping signals, see for example Figure 7c where noise in lead V5 interferes with the other leads. In addition, the limb leads are linearly dependent, a feature which we have not exploited in this work. By utilizing this, it is likely possible to get even more robust digitization results. For example, the missing part of lead III in Figure 7a could be filled in with good precision even though it is missing.

Methods

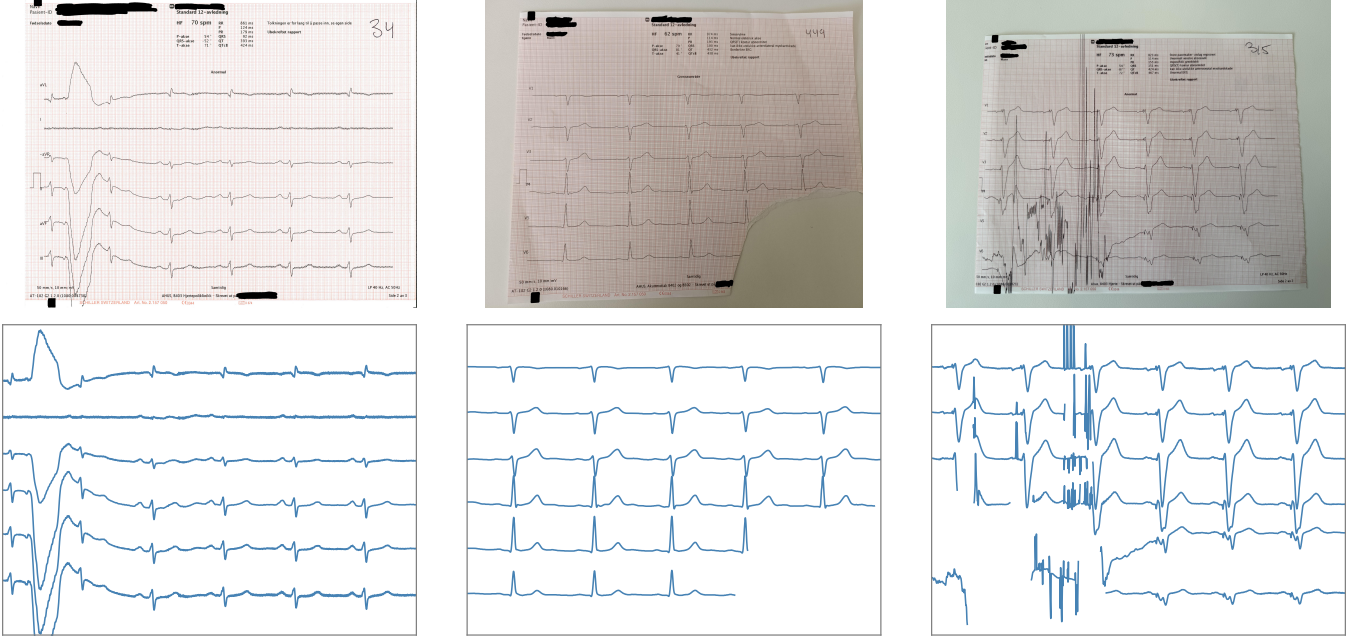
The following section contains the methodology for the digitization framework, training, and data collection for the clinical validation. All development was done on synthetic data, and clinical data were only used in the prospective testing. The study was approved by the Norwegian Directorate of Health, case no. 21/39600-3, dated 01 July 2022.

Our proposed digitization pipeline comprises five main modules, namely (1) semantic segmentation, (2) perspective correction and cropping, (3) layout identification, (4) grid size extraction, and (5) conversion to 1D time series. The first two steps are aimed at creating a high-quality 2D representation of the ECG signal: after semantic segmentation, details such as lighting and background should be removed. After perspective correction, the image coordinates should be aligned with the time-voltage axes. The last three steps involve taking the 2D representation of the segmented ECG signals and converting it to a 1D time series with physical units such as millivolt and seconds, as opposed to pixel coordinates. The pipeline is illustrated in Figure 8. The following section*s describe each module in detail.

Segmentation

The initial module of our framework focuses on segmentation, tasked with identifying ECG signals, gridlines, text annotations, and background elements within the ECG images. This segmentation process facilitates the subsequent isolation and extraction of the ECG signals. We adopt a neural network in a residual U-Net configuration for the task [27], due to its demonstrated performance across various semantic segmentation tasks [28]. Additional architectures were explored, but it was found that a diverse training set was more important than the choice of segmentation network, and thus, no formal ablations were deemed necessary.

The model architecture comprises eight encoder and -decoder blocks. The feature map widths in the encoder are: (32, 64, 128, 256, 320, 320, 320, 320). Each encoder block contains two Conv - InstanceNorm - LeakyReLU stages, followed by a 2×2 strided convolution for spatial downsampling. Bias terms are omitted in convolution layers since they are directly followed by normalization. In the decoder, each block upsamples via bilinear interpolation, concatenates skip connections, and applies



(a) Shows a scanner image together with its reconstruction, in which lead III has been partially set to NaN due to the signal being outside of the paper.

(b) Shows a OnePlus image where the paper presumably got stuck in the ECG machine. The reconstruction puts NaNs in the part where the paper is missing.

(c) Shows an iPhone image with severe noise in lead V5, interfering with other leads. The reconstruction has several leads that have some parts with NaNs.

Figure 7: Three examples of signals with significant numbers of NaNs in them, across the three devices. Instead of rejecting the images, the leads are left blank, and the user can decide what to do with them.

a Conv - InstanceNorm - LeakyReLU block. A final 3×3 convolution projects to the four output channels.

Perspective correction and cropping

The preprocessing required for an ECG image depends on whether it is scanned or photographed. For scanned paper ECGs, rotating the image is typically enough to align it with the image coordinates. However, for photographs taken with a camera (e.g., a phone), the image has some amount of perspective distortion, requiring additional transformations to properly align the ECG with the image coordinates. The same algorithm can handle both rotation and perspective adjustments, as perspective correction also corrects for rotation.

We propose a robust two-step method for correcting perspective in ECG images. The first step is to separate the grid lines from the rest of the image, a task that is handled by the segmentation network. The second step builds upon the insight that each set of parallel grid lines will correspond to a line in the angle-radius domain, even when the grid is viewed at an angle. By applying two subsequent transformations, the task of identifying the perspective reduces to finding local maxima in a grid.

The Hough transformation is a well-proven method for detecting straight line segments [29]. The transform builds on parameterizing lines using an angle θ and a radius from the image origin ρ as

$$y \sin \theta + x \cos \theta = \rho, \quad (1)$$

and obtaining a dual domain that we denote the angle-radius domain, parameterized by ρ and θ instead of x and y . The maxima in the angle-radius domain correspond to lines in the image domain, but it is often hard to identify

the correct maxima under moderately noisy circumstances. An illustration of the relationship between lines in the image domain and points in the angle-radius domain is shown in the two leftmost panes of Figure 9.

Now, we take advantage of the fact that we are not trying to identify a single line, but a set of nearly parallel line segments. By identifying sets of lines, the noise can be significantly suppressed. The full process, starting from segmentation, followed by angle-radius domain transformation, and angle-angle transformation, is shown in Figure 10. When calculating the discrete Hough transform, a set of angles θ has to be determined a priori, setting the resolution of the angle-radius domain. To ensure both good resolution and computational efficiency, we compute the transform twice, first with low resolution over the range $[-\pi/4, 3\pi/4]$, and then zoom with higher resolution. To go from the angle-radius domain to the angle-angle domain (depicted as the rightmost plot in Figure 10), we search for lines in the angle-radius domain, in essence, the value at (θ_i, θ_j) is the variance of the pixels along the line going through the points $(\theta_i, \rho_{\min})$ and $(\theta_j, \rho_{\max})$ in the discretized angle-radius domain.

To clarify the relationship between the angle-radius and angle-angle domains, we map the green line in Figure 9 to a green mark in the rightmost pane. Note that the values of θ where the green line exits the figure correspond to the coordinates of the green mark in the angle-angle domain. After locating the two most prominent maxima in the angle-angle domain, we essentially know the relationship between the camera and the gridded paper. All that remains is to resample and crop the image, while ensuring all of the signal is preserved. To accomplish this, the cropping is informed by the ECG signal channel of the segmentation network, as well as the two coordinates from the

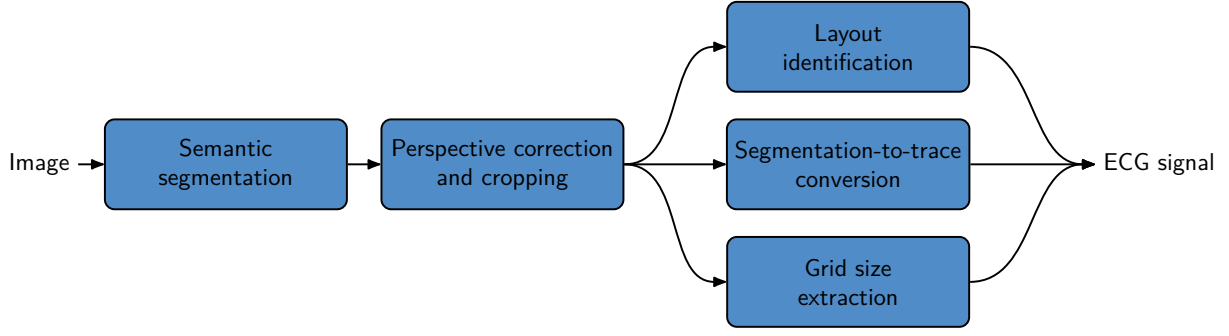


Figure 8: The digitization pipeline consists of semantic segmentation, that separates gridlines, ecg signal, text and background, followed by perspective correction and cropping. These first two modules produce an image where the gridlines on the paper are aligned with the image. The last part converts the aligned image to 1D signals, identifies the lead layout and outputs a structured time series in the desired sample rate.

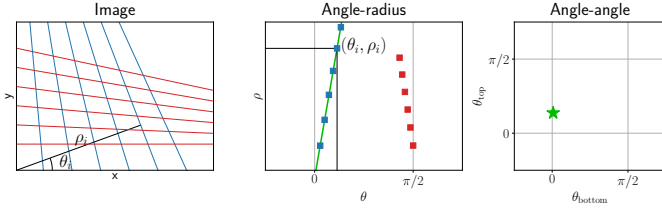


Figure 9: The left image represents a paper with gridlines. Parallel lines viewed at an angle in the image domain end up, to a good approximation, on a line in the angle-radius domain. Sets of lines that are parallel in the image domain are mapped to vertical lines in the angle-radius domain. The green line in the angle-radius domain maps to the green star in the angle-angle domain. Note that the θ -values where the line exits the plot correspond to the coordinates in the angle-angle domain.

angle-angle domain. All the aforementioned steps, including segmentation, iterative Hough transforms, and transformation to angle-angle domain, have been implemented to run on the GPU (if available), ensuring the time spent at these steps in the process is negligible.

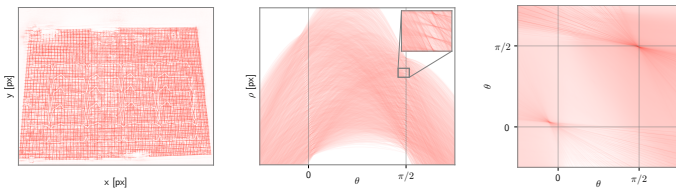


Figure 10: Steps to identify the perspective in a photograph of a paper ECG. The leftmost panel shows the gridlines as detected by the neural network. The middle panel shows the grid transformed into the angle-radius domain using the Hough transform. Finally, the rightmost panel illustrates the domain where the lines of maximum variance are identified. Note that the maxima are much more pronounced in this domain and correspond to the vertical and horizontal gridlines. We encourage the reader to zoom in on the middle panel at around 0 and $\pi/2$

Grid size extraction

The standard units for ECG recordings are millimetres per second (usually 25 or 50 mm/s) along the horizontal axis

and millimetres per millivolt (usually 10 mm/mV) along the vertical axis. In the dewarped ECG image, both axes are represented in pixels. Therefore, it is necessary to estimate conversion factors between mm/s, mm/mV, and pixels. To this end, we use an iterative grid search algorithm to determine the pixel spacing between grid lines in the dewarped image.

The algorithm operates independently on each axis. For clarity, its operation is described here for the horizontal axis. As input, the algorithm takes the dewarped feature map, where each pixel has a value in the range $(0, 1)$, representing the probability of it being a grid line. The first step of the algorithm is to collapse the feature map from 2D to 1D by taking the sum of the columns and computing the autocorrelation. The discrete autocorrelation function R_{xx} at index m of a discrete signal x is computed as

$$R_{xx}[m] = \sum_n x[n] \cdot x[n + m].$$

where m is the lag and the sum is taken over all valid indices. We now find the correct conversion factor by matching the autocorrelation with a known template containing both major and minor gridlines, with the assumption that the major gridlines are spaced with $5d$ and the minor lines by d , with d being the parameter of interest. The search is implemented as an adaptive grid search over d . Figure 11 shows the autocorrelation and a matched template, where a suitable value for d has been found.

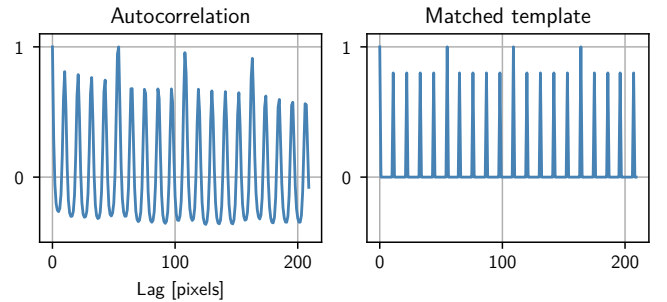


Figure 11: The left pane shows the autocorrelation for a column-sum signal for lags between 0 to 220 pixels. The left pane shows the template that is matched to the autocorrelation function.

Layout identification

ECG recordings appear in various layout formats, with 6×2 , 12×1 , and 3×4 being common in clinical practice. Accurately identifying the layout is a critical step for reliable ECG digitization. To achieve this, we perform layout identification in two stages. First, a residual U-Net is used to segment lead text annotations (e.g., III or aVR) from the dewarped text predictions generated by the initial segmentation model. Once the lead names are identified, their positions are estimated by computing the weighted centroids of the segmented text regions. This allows the system to infer the layout by matching detected leads to the best matching configuration, supporting both standard and arbitrary 12-lead ECG arrangements, including any subset of leads. The lead text segmentation network shares the same architectural principles as the first segmentation U-Net but is designed to be more lightweight, with feature map widths set to (32, 64, 128, 256, 256). It processes a single-channel probability map as input and outputs 13 channels, corresponding to the 12 standard leads and one background class. Notably, this algorithm allows the user to specify arbitrary layouts in a configuration file. The algorithm for finding the best match among the set of user-defined templates is described in Algorithm 1. The lead markers in the generated template layout are assigned 2D-positions in $[0, 1]^2$, and missing lead markers are handled by assigning them a fixed distance corresponding to half the image width (parameterized by λ in Algorithm 1). Thus, even if the U-Net fails to detect one or more lead markers, the lowest cost layout will still often be the correct one.

Algorithm 1 Get best matching layout from lead markers.

Require: Set of candidate layouts $\mathcal{L} = \{L_j\}$, List of detected lead markers and their coordinates P .

Ensure: Best matching layout L^*

```

1: Initialize  $L^* \leftarrow \emptyset$ 
2: Initialize  $\lambda \leftarrow 0.5$ 
3: Initialize  $\text{lowest\_cost} \leftarrow +\infty$ 
4: for all  $L_j \in \mathcal{L}$  do
5:    $G \leftarrow \text{generate\_layout}(L_j)$ 
6:    $(M, M_{\text{miss}}) \leftarrow \text{match}(P, G)$ 
7:    $\text{scale, translate} \leftarrow \text{estimate\_transform}(M)$ 
8:    $M' \leftarrow \text{apply\_transform}(M, \text{scale}, \text{translate})$ 
9:    $\text{cost} = \frac{1}{|G|} \left( \lambda |M_{\text{miss}}| + \sum_{(p'_i, g_k) \in M'} \|p'_i - g_k\| \right)$ 
10:  if  $\text{cost} < \text{lowest\_cost}$  then
11:     $L^* \leftarrow L_j$ 
12:     $\text{lowest\_cost} \leftarrow \text{cost}$ 
13:  end if
14: end for
15: return  $L^*$ 

```

Segmentation-to-trace conversion

The next step is to convert the dewarped segmentation output into distinct ECG lead traces. Segmented lines from the dewarped image are used as input to the lead detection process. The algorithm begins by estimating the initial lead locations. This is done by identifying connected components in the feature map. In cases where a connected component yields poor results, such as when overlapping signals cause multiple leads to merge into a single com-

ponent, a snipping algorithm is applied to subdivide the component into smaller segments.

For a problematic component, the snipping algorithm attempts to divide it horizontally by drawing a path from the left to the right edge of the component, separating it into at least two disjoint parts. The path is identified via a greedy, one-step lookahead search that selects the path of least resistance, i.e., the path minimizing overlap with the segmented signal at each step. After snipping, the connected components are recalculated. This process, connected component detection followed by component snipping, is repeated iteratively until convergence or until a predefined maximum number of iterations is reached. As a result of this stage, each lead is divided into multiple disjoint connected components from the original dewarped segmentation output. The disjoint components need to be re-merged to match the number of leads in the image. This is formulated as a minimum-weight matching problem in a bipartite graph, where the leftmost endpoint of each component should be connected with the rightmost part of some other lead segment, i.e., the linear sum assignment problem. It is solved with a modified Jonker-Volgenant algorithm [30]. To ensure a correct matching, wrap-around is needed for some components; see Figure 12 for an illustration of how the component merging process looks visually.

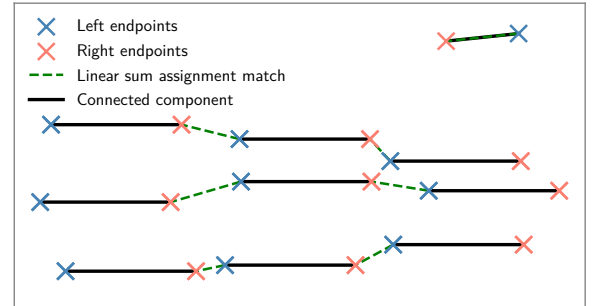


Figure 12: Example of a solution to a linear sum assignment problem. Note that three of the leftmost endpoints and three of the rightmost endpoints are unmatched, illustrating that wrapped matches are dropped. Further, note that the connected component in the top right corner is matched with itself. Self-connections are later rejected as noise.

The cost function for matching is based on a weighted Manhattan distance between component endpoints. Matches that imply a backward connection in time (i.e., most likely incorrect matches) are penalized by multiplying the cost by 2.

Training

The initial segmentation model is trained using synthetic ECG images generated using real ECGs from the CODE15% ECG dataset [31], and a modified version of ECG-Image-Kit [32]. Each training image is procedurally augmented with randomized parameters for the width and color of the ECG line, text overlays, and perspective transformations using real photo backgrounds to simulate varied imaging conditions. The generated images follow a 3×4 with lead II as a rhythm lead layout. By using a dataset

collected in Brazil (i.e., a big demographic shift with regards to the test datasets), random coloring, and a layout not present in the test set, we aim to verify that our algorithm can withstand significant domain shifts without degraded performance.

Training samples are obtained by randomly cropping 1024×1024 pixel regions from the generated images. The loss function is a sum of soft Dice loss and focal loss, which has shown increased robustness in multi-class segmentation tasks [33]. Both terms are suitable for unbalanced segmentation problems, including the present task, where the ECG signal is of high importance but takes up a small minority of the total number of pixels.

Model weights are updated using the Muon optimizer [34] for convolutional layers, while AdamW [35], [36] is applied to the affine parameters in the instance normalization layers. Using the Muon optimizer results in significantly faster convergence and improved final accuracy compared to using AdamW alone. Automatic mixed precision is employed to accelerate training. Full training details, including hardware and software, are provided in Table 3, and hyperparameters are provided in Table 4.

Table 3: Hardware and software specifications used for model development and training.

Component	Specification
System	Debian 12
CPU	Intel i9-14900KF
RAM	2×48 GB; 4800 MT/s
GPU	NVIDIA RTX 5090 (32 GB)
CUDA version	12.8
Programming language	Python 3.12
Deep learning framework	PyTorch 2.7.0

Table 4: Training protocol used for the main model. FLOPs are calculated by assuming a 1:2 ratio of computation between the forward and backward passes, and correspond to a full training run.

Parameter	Value
Batch size	12
Patch size	1024×1024
Loss function	Soft Dice + Focal
Optimizer	Muon & AdamW
Muon momentum	0.95
AdamW betas	0.9, 0.999
Weight decay	0.001
Initial learning rate	0.0037
Learning rate scheduler	Cosine to constant
Batches until final learning rate	20,000
Final learning rate	0.00037
Training time	5 hours
Total epochs	45
Model parameters	22.6 M
FLOPs	260 PFLOP

Code Availability

The full software, including instructions, can be found on GitHub: github.com/Ahus-AIM/Open-ECG-Digitizer.

Acknowledgements

We thank the staff at the Department of Cardiology at Akershus University Hospital for helping us in data collection, and James Weigle and Matthew Reyna for kindly evaluating our framework on the Emory Paper Digitization ECG Dataset.

Author Contributions

E.S. and A.B. designed the software as well as designed and performed experiments for evaluation, with supervision from A.R.. E.S. and A.B. wrote the article with input from A.R.

Competing interests

The authors declare no competing interests.

References

- [1] World Health Organization, *Cardiovascular diseases (cvds)*, WHO Fact Sheet, Estimated 17.9 million deaths in 2019 (32% of global deaths). Retrieved July 2025 from [https://www.who.int/news-room/fact-sheets/detail/cardiovascular-diseases-\(cvds\)](https://www.who.int/news-room/fact-sheets/detail/cardiovascular-diseases-(cvds)), 2022.
- [2] X. Liu, H. Wang, Z. Li, and L. Qin, “Deep learning in ECG diagnosis: A review,” *Knowledge-Based Systems*, vol. 227, Sep. 2021, DOI: 10.1016/j.knsys.2021.107187
- [3] Z. Wu and C. Guo, “Deep learning and electrocardiography: Systematic review of current techniques in cardiovascular disease diagnosis and management,” *BioMedical Engineering OnLine*, vol. 24, no. 1, 2025.
- [4] R. Jiang et al., “Deep Learning-Augmented ECG Analysis for Screening and Genotype Prediction of Congenital Long QT Syndrome,” *JAMA Cardiology*, vol. 9, no. 4, Apr. 2024, DOI: 10.1001/jamacardio.2024.0039
- [5] H. Zhu et al., “Automatic multilabel electrocardiogram diagnosis of heart rhythm or conduction abnormalities with deep learning: A cohort study,” English, *The Lancet Digital Health*, vol. 2, no. 7, Jul. 2020, Publisher: Elsevier, DOI: 10.1016/S2589-7500(20)30107-2
- [6] A. H. Ribeiro et al., “Automatic diagnosis of the 12-lead ecg using a deep neural network,” *Nature communications*, vol. 11, no. 1, 2020.
- [7] S. S. Al-Zaiti et al., “Machine learning for ECG diagnosis and risk stratification of occlusion myocardial infarction,” en, *Nature Medicine*, vol. 29, no. 7, Jul. 2023, Publisher: Nature Publishing Group, DOI: 10.1038/s41591-023-02396-3

- [8] J.-m. Kwon et al., "A deep learning algorithm to detect anaemia with ECGs: A retrospective, multicentre study," English, *The Lancet Digital Health*, vol. 2, no. 7, Jul. 2020, Publisher: Elsevier, DOI: 10.1016/S2589-7500(20)30108-4
- [9] D. Ouyang et al., "Electrocardiographic deep learning for predicting post-procedural mortality: A model development and validation study," English, *The Lancet Digital Health*, vol. 6, no. 1, Jan. 2024, Publisher: Elsevier, DOI: 10.1016/S2589-7500(23)00220-0
- [10] Z. I. Attia et al., "Age and sex estimation using artificial intelligence from standard 12-lead ecgs," *Circulation: Arrhythmia and Electrophysiology*, vol. 12, no. 9, 2019.
- [11] F. Badilini, T. Erdem, W. Zareba, and A. J. Moss, "ECGScan: A method for conversion of paper electrocardiographic printouts to digital electrocardiographic files," eng, *Journal of Electrocardiology*, vol. 38, no. 4, Oct. 2005, DOI: 10.1016/j.jelectrocard.2005.04.003
- [12] L. Ravichandran, C. Harless, A. J. Shah, C. A. Wick, J. H. McClellan, and S. Tridandapani, "Novel Tool for Complete Digitization of Paper Electrocardiography Data," eng, *IEEE journal of translational engineering in health and medicine*, vol. 1, 2013, DOI: 10.1109/JTEHM.2013.2262024
- [13] "High Precision Digitization of Paper-Based ECG Records: A Step Toward Machine learning," *IEEE Journal of Translational Engineering in Health and Medicine*, vol. 7, Nov. 2019, DOI: 10.1109/JTEHM.2019.2949784
- [14] S. Mishra et al., "ECG Paper Record Digitization and Diagnosis Using Deep Learning," eng, *Journal of Medical and Biological Engineering*, vol. 41, no. 4, 2021, DOI: 10.1007/s40846-021-00632-0
- [15] J. D. Fortune, N. E. Coppa, K. T. Haq, H. Patel, and L. G. Tereshchenko, *Digitizing ECG image: New fully automated method and open-source software code*, en, Jul. 2021, DOI: 10.1101/2021.07.13.21260461
- [16] H. Wu et al., "A fully-automated paper ECG digitization algorithm using deep learning," *Scientific Reports*, vol. 12, Dec. 2022, DOI: 10.1038/s41598-022-25284-1
- [17] M. A. Reyna et al., "Digitization and Classification of ECG Images: The George B. Moody PhysioNet Challenge 2024," Dec. 2024, DOI: 10.22489/CinC.2024.011
- [18] F. H. Kroner, T. J. Lyons, A. Mahdi, and B. Walker, "Combining Hough Transform and Deep Learning Approaches to Reconstruct ECG Signals From Printouts," Dec. 2024, DOI: 10.22489/CinC.2024.097
- [19] H.-C. Yoon, D.-K. Kim, H.-S. Kim, W.-Y. Seo, C.-H. Heo, and S.-H. Kim, "Segmentation-based Extraction of Key Components from ECG Images: A Framework for Precise Classification and Digitization," Dec. 2024, DOI: 10.22489/CinC.2024.227
- [20] E. Stenhede, B.-J. Singstad, and A. Ranjbar, "ECG-DUaL: Pose-invariant Digitization of printed electrocardiograms via U-net and Local edge detection," Dec. 2024, DOI: 10.22489/CinC.2024.262
- [21] A. Demolder et al., "High precision ECG digitization using artificial intelligence," *Journal of Electrocardiology*, vol. 90, May 2025, DOI: 10.1016/j.jelectrocard.2025.153900
- [22] M. A. Reyna 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*, 2024, arXiv: 2409.16612 [q-bio.QM].
- [23] A. Ranjbar and J. Ravn, "Data quality in health-care for the purpose of artificial intelligence: A case study on ecg digitalization," in *Healthcare Transformation with Informatics and Artificial Intelligence*, IOS Press, 2023.
- [24] A. Ranjbar, J. Ravn, E. Ronningen, and O. Hanseth, "Enabling clinical trials of artificial intelligence: Infrastructure for heart failure predictions.," *Studies in Health Technology and Informatics*, vol. 302, 2023.
- [25] S. Jammoul, A. Hassan, E. Zhang, P. Warrick, and J. Afilalo, "Layout-Invariant U-Net Segmentation for Time-Series Reconstruction from Noisy ECG Images," Dec. 2024, DOI: 10.22489/CinC.2024.481
- [26] J. C. L. Himmelreich and R. E. Harskamp, "Diagnostic accuracy of the PMcardio smartphone application for artificial intelligence-based interpretation of electrocardiograms in primary care (AMSTELHEART-1)," *Cardiovascular Digital Health Journal*, vol. 4, no. 3, Jun. 2023, DOI: 10.1016/j.cvdhj.2023.03.002
- [27] O. Ronneberger, P. Fischer, and T. Brox, *U-Net: Convolutional Networks for Biomedical Image Segmentation*, arXiv:1505.04597, May 2015, DOI: 10.48550/arXiv.1505.04597
- [28] R. Azad et al., "Medical image segmentation review: The success of u-net," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2024.
- [29] R. O. Duda and P. E. Hart, "Use of the Hough transformation to detect lines and curves in pictures," *Commun. ACM*, vol. 15, no. 1, Jan. 1972, DOI: 10.1145/361237.361242
- [30] D. F. Crouse, "On implementing 2D rectangular assignment algorithms," *IEEE Transactions on Aerospace and Electronic Systems*, vol. 52, no. 4, Aug. 2016, DOI: 10.1109/TAES.2016.140952
- [31] A. H. Ribeiro et al., *CODE-15%: A large scale annotated dataset of 12-lead ECGs*, Jun. 2021, DOI: 10.5281/zenodo.4916206
- [32] K. K. Shivashankara, Deepanshi, A. Mehri Shervedani, G. D. Clifford, M. A. Reyna, and R. Sameni, "ECG-Image-Kit: A synthetic image generation toolbox to facilitate deep learning-based electrocardiogram digitization," *Physiological Measurement*, vol. 45, no. 5, May 2024, DOI: 10.1088/1361-6579/ad4954

- [33] J. Ma, *Segmentation Loss Odyssey*, arXiv:2005.13449, May 2020, DOI: 10.48550/arXiv.2005.13449
- [34] *Muon: An optimizer for hidden layers in neural networks* / Keller Jordan blog.
- [35] D. P. Kingma and J. Ba, *Adam: A Method for Stochastic Optimization*, arXiv:1412.6980, Jan. 2017, DOI: 10.48550/arXiv.1412.6980
- [36] I. Loshchilov and F. Hutter, *Decoupled Weight Decay Regularization*, arXiv:1711.05101, Jan. 2019, DOI: 10.48550/arXiv.1711.05101