ECG-DUaL: Pose-Invariant Digitization of Printed Electrocardiograms via U-Net and Local Edge Detection

Elias Stenhede[1], Bjørn-Jostein Singstad[1,2], Arian Ranjbar[1]

[1] Akershus University Hospital, Medical Technology and E-health, Lørenskog, Norway
[2] University of Oslo, Institute of Clinical Medicine, Oslo, Norway

Abstract

ECG recordings are often stored as paper scans, due to clinical accessibility and tradition. This paper presents a digitization pipeline for ECG printouts based on line detection to correct for rotation, and a U-net for segmentation. The digitization step is differentiable, so the U-net can be fine-tuned for any evaluation metric of interest. The tool achieves a maximum translation-adjusted SNR of $3.047\,\mathrm{dB}$ *on the George B. Moody PhysioNet Challenge 2024 hidden dataset, placing Ahus AI Lab fourth among the official entries.*

1. Introduction

Cardiovascular disease (CVD) remains the leading cause of death in the world [1]. Electrocardiogram (ECG) stands out as one of the most widely used diagnostic tools, accessible even in severely resource-constrained environments. In recent decades, computer-assisted analysis has improved clinicians' ability to interpret ECG data. This progress has accelerated dramatically with the advent of deep learning, unlocking algorithmic ECG classification accuracy on par with, or even above, human capabilities [2]. To achieve optimal performance in deep learning, high quality data is essential for both model training and inference. For ECGs, this means ensuring the accuracy of the recordings and maintaining consistency in the data format [3]. In this paper, we present an ECG conversion tool for general use, capable of digitizing printouts from various manifacturers and ecan layouts, see Figure 1 for an overview of the methodology.

2. Methods

The proposed digitization tool consists of five modules: pre-processing, semantic segmentation, signal identification, signal conversion (mapping from 2D to 1D), and lead identification. The pipeline, i.e., the underlying parameterized models, were trained using simulated ECG printouts.

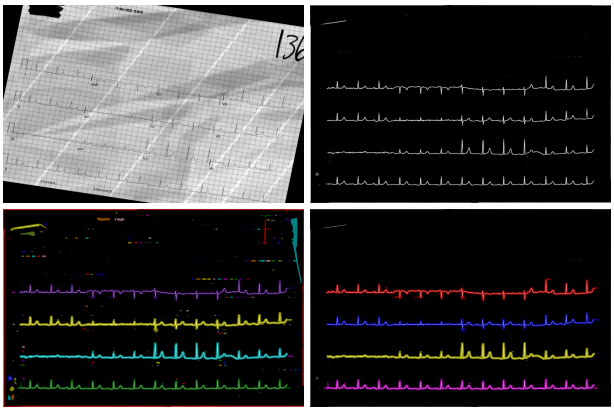

Figure 1. Overview of the effects by each module in the pipeline. Upper left corner: simulated ECG printout with artifacts such as rotation, handwritten digits, and redacted information. Upper right corner: output from the segmentation network. Lower left corner: The segmentation is thresholded and contiguous regions are separated, shown with different colors. Lower right corner: All small contiguous regions are removed, leaving the signal of interest intact.

Specifically, training data sets were generated using open source ECG Image Kit software [4], together with the public PTB-XL dataset containing 12-lead ECG records from 18869 patients [5]. The image generation source code was modified to output both images and segmentation masks with a shape of 2200 by 1700 pixels. The latter was generated to accommodate the training procedure of the semantic segmentation network.

2.1. Preprocessing

The initial preprocessing step aims to produce high-quality inputs for the segmentation network. Input images are converted from 8-bit RGB format to floating point grayscale, inverted, and finally Min-Max normal-

ized. These transformations result in images with mean pixel intensity close to zero, suitable for neural networks.

Real-world images and scans may have rotation in relation to the camera. To correct this, the grid lines on each ECG's gridded paper can be used for alignment. To detect grid lines, Canny edge detection is applied, followed by the Hough Line Transform. The result is a set of n line segments in various directions. As the grid is the predominant source of straight lines, most but not all lines can be assumed to belong to the grid. To reduce the effect of false positives, 20 % of the detected lines, specifically those with angles furthest from the mean, are discarded. Denoting the number of lines kept n and the angle of each line as θ_i, the average rotation Θ of the image is calculated as

$$\Theta = \frac{1}{n} \sum_{i=1}^{n} \left(\theta_i - \frac{\pi}{2} \left\lfloor \frac{\theta_i + \frac{\pi}{4}}{\frac{\pi}{2}} \right\rfloor \right).$$

After estimating Θ for each image, they are aligned by rotating with $-\Theta$ radians.

The extracted line segments can also be utilized to identify the vanishing points in the image, which can then be used to estimate and correct the camera pose. Camera pose estimation was not performed in the official submission to the 2024 PhysioNet Challenge but is showcased in the qualitative evaluation.[2]

2.2. Semantic segmentation

After preprocessing, semantic segmentation is applied to identify components of the ECG print; specifically (1) ECG signal, (2) 1 mV control signal (3) other printed elements such as lead separators, text and redacted information, and (4) background. While the ECG signal is the only element of interest, introducing classes for other printed elements is done to improve segmentation results for the ECG class.

Given the necessity of using local context to distinguish between, e.g. control signal and ECG signal, a U-Net model is employed. The model uses cross-entropy loss without assigning different weights to the classes. The network output logits, which after applying the softmax function in the channel dimension can be interpreted as probabilities of pixels belonging to the four classes.

The output of the segmentation network runs through a postprocessing procedure, particularly to increase robustness against novel inputs. A Gaussian smoothing filter with kernel size $\sigma = 3$ pixels and binarized filter using a threshold value of $p = 0.001$ is applied to the ECG output channel. The filtering prevents smaller isolated islands of pixels with high probabilities of being ECG signals to build up, as these can be considered false positives.

2.3. Identification of signal start- and endpoints

To map the segmented ECG signal into a 1D time series, it is necessary to identify the start- and endpoints of the plotted ECG signals, carried out in two steps. First, start- and end-points are identified by projection onto the horizontal axis of the pixel map, as rotations are removed in the pre-processing, and solving the following optimization problem:

$$\min_{x_1, x_2} \| f(x) - h \cdot (\mathrm{H}(x - x_1) - \mathrm{H}(x - x_2)) \|_1 \qquad (1)$$

where H denotes the Heaviside step function, f(x) the target signal as a function of column index and h the median of the target signal. The problem can be illustrated most intuitively by an example, as depicted in Figure 2.

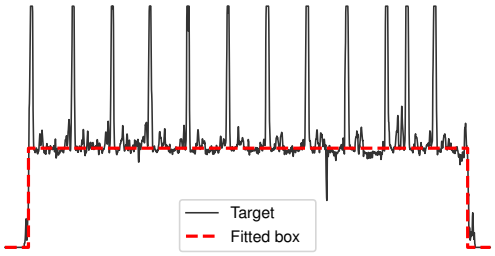

Figure 2. Box signal fitted to the projected pixel map in order to identify start and endpoints of the ECG channels.

Solving the optimization problem yields a fitted box to the respective ECG-channel. However, since the segmentation network is not specifically trained to find the true edges of a signal, it might fail in the presence of artefacts or obstructions in the image, e.g. when the start- or endpoint of a signal overlaps with the control signal. These scenarios could result in slight translations of the digitized signal, which could severely affect similarity metrics sensitive to lag, such as the signal-to-noise ratio (SNR). The box fitting estimates are therefore refined using a fully convolutional edge finding network, trained to detect the start and endpoints within each box, inspired by [6].

The networks aim is to output a tensor ρ the same shape as the input image, with local maxima at coordinates corresponding to signal start and endpoints. To illustrate the mechanics of the procedure, let S denote the sigmoid function, (x, y) the position of a start or end point in image coordinates and $\epsilon = 0.0001$ be a constant. During training, the network is trained with image patches with varying shape (W, H). The minimization objective is the sum of two functions:

$$L_1(\rho) = \sum_{i,j=1}^{H,W} S(\rho_{ij}) \sqrt{(j - x)^2 + (i - y)^2} \qquad (2)$$

and

$$L_2(\rho) = \left(x - \frac{\sum_{i,j=1}^{H,W} j w_{ij}}{\sum_{i,j=1}^{H,W} w_{ij}} \right)^2 + \left(y - \frac{\sum_{i,j=1}^{H,W} i w_{ij}}{\sum_{i,j=1}^{H,W} w_{ij}} \right)^2. \tag{3}$$

where w is a deterministic function of ρ,

$$w_{ij} = \frac{\epsilon + S(\rho_{ij})}{HW\epsilon + \sum_{i,j=1}^{H,W} S(\rho_{ij})}. \tag{4}$$

Note that ϵ in Equation (4) effectively defines a baseline weight for all pixels, compelling the network to produce higher values in the vicinity of (x, y) to minimize L_2. Meanwhile, L_1 encourages lower values for pixels farther from (x, y).

During inference, ρ is normalized to $w_{ij} = S(\rho_{ij})$ and detection is performed by choosing a threshold value for w. The location of start- and endpoints is found by calculating the center of mass of each island. Notably, the network can take input images of any size and find any number of start and endpoints. However, to save computational resources, crops around the estimated start and end points were used, as illustrated in Figure 3

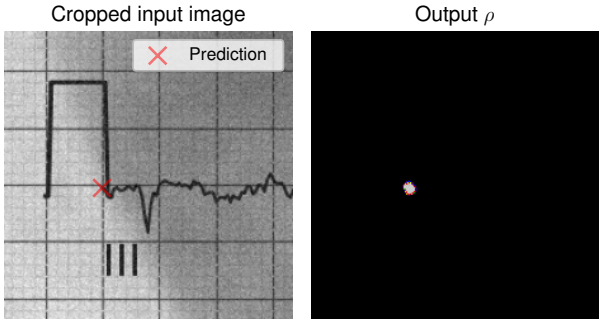

Figure 3. The edge finding network returns a weightmap; the local center of mass is used to predict the position of start- and endpoints of ECG signals with high accuracy.

2.4. Signal Conversion

After identifying the start and endpoints of the ECG signals, the full image is cropped and resampled along the horizontal axis. Resampling is performed to match the target sample rate of the digitized signal, e.g., a desired sample rate of 100 Hz results in an image with shape (1000, H) for a 10 second signal. Transformation of the 2D probability tensor to the 1D time series is carried out in parallel for each column and lead, identified via contiguous regions as depicted in Figure 1. For a given column $c \in \mathbb{R}^{\mathrm{H}}$, the mV value is calculated as $\mathrm{softmax}(\log(\frac{c}{1-c})) \cdot v$ where $v_i = \alpha i$ and α is the conversion factor determined via the printer settings, i.e. the ratio between $\mathrm{mm/mV}$ and $\mathrm{mm/s}$, and i is the vertical pixel coordinate. The conversion process is depicted in Figure 4.

Since the full process is now differentiable, the pipeline can be fine-tuned end-to-end for the desired metric, e.g. SNR. For the challenge experiment, fine-tuning is used to maximize both SNR and segmentation cross-entropy, resulting in a more quantized signal, see Figure 5.

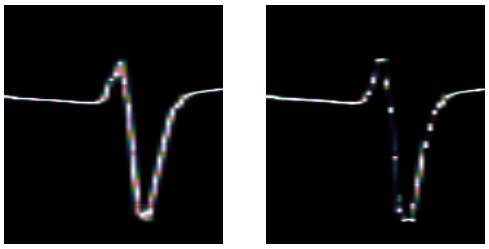

Figure 5. To the left, output of segmentation network trained only with cross-entropy. To the right, segmentation network trained to minimize both cross-entropy and negative SNR. Note that the fine-tuned network gives a more concentrated output.

2.5. Lead identification

Lead identification is carried out assuming a 3-by-4 channel layout, where each lead is plotted for 2.5 s, with an arbitrary number of rythm leads. Identification of the short leads is done via relative geometric location, and rythm leads are identified by calculating the corresponding correlation of each 2.5 s segment.

3. Experiments

Experiments were carried out using the simulated ECG printout dataset and PyTorch. The U-net model was implemented with a standard block structure of `2×[Conv(3x3, k), ReLU, BatchNorm2d]` where k is the number of output channels, chosen to be `1, 16, 32, 54, 128, 256` in the downsampling with `MaxPool2d(2,2)` between each block. The upsampling used the mirrored structure, replacing the max pooling operation with bilinear upsampling.

The edge finding network was implemented as a series of 11 blocks of `[Conv(3x3, 64), ReLU, BatchNorm2d]` followed by `Conv(1x1, 1)`.

The networks were trained using AdamW, learning rate 0.0001, batch size 1 and weight decay 0.01. A low batch size was used because of the high resolution of the input

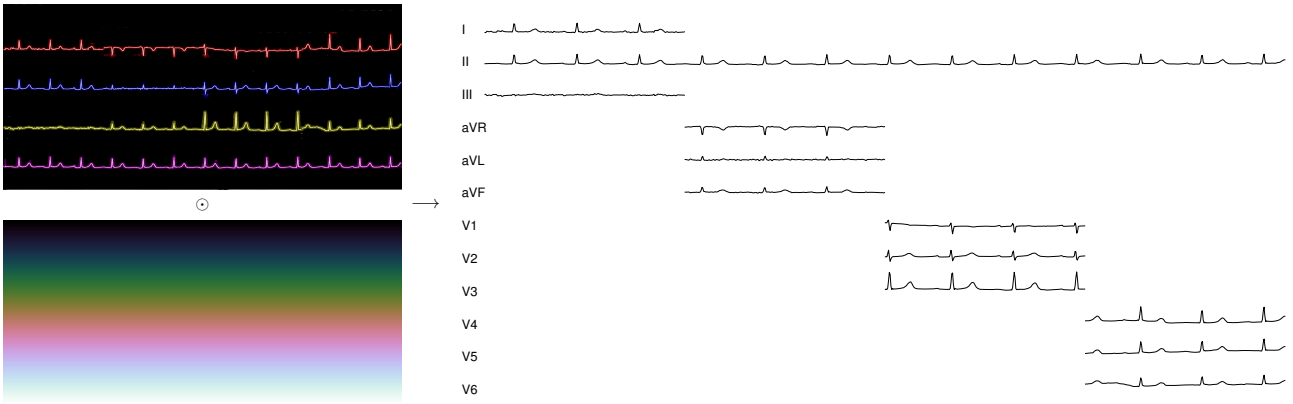

Figure 4. The weightmap is multiplied with the linearly increasing tensor to produce the multivariate 1D time series.

images. The edge finding network was initially trained on (128, 128) sized crops from the simulated images, before fine-tuning both networks jointly for 300 batches.

Using batch loss, convergence was achieved already before one epoch, or 4 hours of training on a Tesla T4 GPU. The edge finding pre-training converged to subpixel accuracy after being trained with 2,000 images, i.e. 16,000 (128, 128) crops. The official challenge [7] score obtained by evaluation on [8] is presented in Table 1.

Table 1. Signal-to-noise (SNR) ratio of our team's model on the hidden data for the digitization task.

Task	Score	Rank
Digitization	SNR: 3.047	4/16

4. Discussion and Conclusions

While testing the tool on held-out subset of the training set it performs better than on the hidden challenge dataset, indicating limitiations in generalization capabilities. To demonstrate this, inference is made on the held out subset[1] and a collection of real-world images[2]. In the former, a SNR of $10.509\,\mathrm{dB}$ was obtained, the latter gives indications on where the method fails. As suspected, the tool fails to assign the correct identity to a channel, when non-supported layouts are used. When the paper ECG is not flat, the dewarping step might fail.

Summa summarum, this paper presented a tool to digitize paper ECGs, showing proficient performance for the task. To further improve accuracy, better channel identification, dewarping and handling of overlapping signals might be necessary.

[1] https://github.com/eliasstenhede/ECG-DUaL-Quantitative-Test/
[2] https://github.com/eliasstenhede/ECG-DUaL-Qualitative-Test/

References

[1] World Health Organization. The Top 10 Causes of Death. URL www.who.int/news-room/fact-sheets/detail/the-top-10-causes-of-death.

[2] Siontis KC, et al. Artificial Intelligence-Enhanced Electrocardiography in Cardiovascular Disease Management. Nature Reviews Cardiology 2021;18(7):465–478.

[3] Ranjbar A, Ravn J. Data Quality in Healthcare for the Purpose of Artificial Intelligence: A Case Study on ECG Digitalization. In Studies in Health Technology and Informatics. IOS Press, June 2023; .

[4] Shivashankara KK, et al. ECG-Image-Kit: A Synthetic Image Generation Toolbox to Facilitate Deep Learning-Based Electrocardiogram Digitization, February 2024. ArXiv:2307.01946 [cs].

[5] Wagner P, et al. PTB-XL, a Large Publicly Available Electrocardiography Dataset v1.0.3. Physionet 2022;.

[6] Midtvedt B, et al. Single-Shot Self-Supervised Object Detection in Microscopy. Nature Communications December 2022;13:7492. ISSN 2041-1723.

[7] Reyna M, et al. Digitization and Classification of ECG Images: The George B. Moody PhysioNet Challenge 2024. Proceedings of Computing in Cardiology 2024 ;.

[8] Reyna MA, Deepanshi, Weigle J, Koscova Z, Campbell K, Shivashankara KK, Saghafi S, Nikookar S, Motie-Shirazi M, Kiarashi Y, Seyedi S, Clifford GD, Sameni R. 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. URL https://arxiv.org/abs/2409.16612.

Address for correspondence:

Elias Stenhede
Sykehusveien 25, 1478 Nordbyhagen, Norway
elias.stenhede@ahus.no