Fusion of Deep Learning and Rule-Based Techniques for Enhanced Paper-Based ECG Digitization

Amaan Kazi[1], Kelvin Nguyen[1], Varun Sendilraj[1], Shadi Manafi Avari[1], Sasan Esfahani[1], Zaniar Ardalan[1], Saman Parvaneh[1]

[1] Edwards Lifesciences, Irvine, United States of America

Abstract

The electrocardiogram (ECG) is crucial for diagnosing cardiovascular diseases. As a part of the Digitization and Classification of ECG Images: The George B. Moody PhysioNet Challenge 2024, our team, PapyrusECG, developed a pipeline combining deep learning and rule-based techniques to digitize paper-based ECGs. The pipeline includes image synthesis, skew correction, lead detection, lead binarization, and signal extraction. We generated images with augmentations using the ECG Image Kit, reserving 6000 for training and 1500 to serve as a held-out subset of the training set. Skew correction was implemented using multiple rule-based computer vision methods. Lead detection employed a fine-tuned YOLOv7 model while Lead binarization used a U-Net model. Signal extraction involved a rule-based algorithm that extracted pixel locations and converted them to millivolts, with challenge signal-to-noise ratio (SNR) results of 2.979±7.212 and 3.099±7.107 for our training and held-out subset datasets, respectively. Our team achieved a score of -5.192 on the hidden test set (ranked 15th out of 16 teams). Digitizing paper ECGs can enhance AI-based diagnostic systems and advance cardiac disease diagnosis. Future work will optimize the pipeline, address limitations, and expand its application to diverse ECG datasets to improve cardiovascular healthcare.

1. Introduction

The electrocardiogram (ECG) is a crucial tool for diagnosing cardiovascular disease [1]. Although many modern-day ECGs collected are in a digital format (allowing deep learning-based algorithms for ECG analysis [2]), an abundance of paper-based ECGs still exist from before the digital era. Many groups have hypothesized that converting these paper-based ECGs to a digital format can unlock a significant amount of information regarding cardiac electrophysiology, arrhythmia mechanisms, and cardiac disease [3]. Additionally, with the rise of AI-based diagnostic systems, having a large amount of digital ECG data to train these models is essential [4]. Thus, our team, PapyrusECG, developed a pipeline combining deep learning and rule-based techniques to digitize paper ECGs, as a part of the digitization task for the PhysioNet Challenge 2024 [5].

2. Methods

The pipeline consists of five parts: synthesizing the images, applying skew correction, lead detection, lead binarization, and signal extraction. We utilized the fusion of rule-based and deep learning techniques; skew correction and signal extraction employed rule-based methods, while lead detection and lead binarization employed deep learning techniques.

2.1. Image Synthesis

For our deep learning methods, we generated images for the training dataset and held-out subset using the ECG Image Kit [6, 7]. Each image we generated assumed the standard 3x4 12-lead format with one rhythm lead. To ensure robustness and generalizability, we generated 7500 augmented images from the PTB-XL database [8, 9], with 6000 for training and 1500 to serve as a held-out subset of the training set. Augmentations included noise, paper wrinkles, temperature changes, cropping, random rotations between -20 and 20 degrees, random resolutions between 50 and 200 dpi, and random grid colors. In addition to this, we generated the bounding boxes for each lead, which were stored in JSON files, and updated the ECG Image Kit [6, 7] to generate a binary mask for each generated image by applying an Otsu Threshold to the image before the grid lines were applied within the image synthesis pipeline. Furthermore, we updated the ECG Image Kit [6, 7] to parallelize image synthesis to speed up our pipeline.

 Page 1 ISSN: 2325-887X DOI: 10.22489/CinC.2024.495

2.2. Skew Correction

At this point, we have generated all the images and their corresponding masks, and split them into a training dataset and held-out subset. Given that some of the images are rotated, we want to apply skew correction to return the images back to their normal orientation. To find the skew angle, we convert the image to grayscale, perform Canny edge detection to find the edges, perform a Hough Transform to find the straight lines within the Canny edge detected image, obtain the most frequent line angles via peak detection in the Hough space, and determine the skew angle by choosing the most frequent line angle. After finding the skew angle, we use it to calculate a 2D rotation matrix, then apply the matrix to the image to return it to its normal orientation. There were occasionally cases where the image would have too much noise which would result in an incorrect angle estimation. Thus, in cases like this, we applied morphological operations to the image to remove the noise and then sent this image through the skew correction algorithm to obtain a better estimate of the skew angle. To perform these operations, we used an open-source third-party library called deskew [10] to calculate the skew angle and OpenCV to rotate the image.

2.3. Lead Detection

After applying skew correction to all the images in the training dataset and held-out subset, we fine-tuned a YOLOv7 model [11] using the code available within the ECG Image Kit [6, 7] to detect all 12 short leads and the rhythm lead individually. We trained the model for 50 epochs, and saved the best checkpoint by evaluating the model based on a weighted sum of mAP@0.5, which measures the average precision calculated at a single IoU threshold of 0.5, and mAP@0.5:0.95, which measures the average precision at multiple IoU thresholds ranging from 0.5 to 0.95 in increments of 0.05, where the weights were 0.1 and 0.9, respectively. These measures were calculated from the held-out subset of the training set.

2.4. Lead Binarization

After training the lead detection model, we created a custom dataset to feed individual lead sections into a U-Net [12] and generate a binary mask. To elaborate, for each image and its corresponding mask, we used the bounding boxes stored in the JSON files to extract each lead section, resize each section to 256x256 pixels, and batch them together for input into the U-Net. For the rhythm lead, we separated its lead section into four separate parts, resized each part to 256x256 pixels, and batched them with the previous short lead sections. This set of lead sections translated to one batch. We trained the U-Net for 15 epochs, and saved the best checkpoint based on Dice Score, which measures the similarity between the segmented image and ground truth, calculated from the held-out subset.

2.5. Signal Extraction

After binarizing each lead section, we used a rule-based algorithm to extract the signal and convert it to millivolts. The first step of this algorithm involves resizing the 256x256 mask back to its original size and placing it in its original location on the image. We create a matrix filled with zeros and then fill in the area where the lead was originally located with its mask, creating a new mask for each detected lead. This step is necessary because we use the pixel locations of the signal to convert it to millivolts. The second step involves calculating the pixel location-to-millivolt ratio. For leads I, II, and III, we perform peak detection to identify the calibration pulse at the beginning of each signal, measure the height of the calibration pulse, and take the inverse of this height to obtain the pixel location-to-millivolt ratio. These three ratios are stored and used for converting the leads in the same row as one of these three leads. Next, we extract the pixel locations for each lead using the signal extraction algorithm created by ECG-Miner [13] and convert them to millivolts using the respective ratio. The same process is applied to the rhythm lead, where we calculate its ratio from the associated calibration pulse.

2.6. Inference Pipeline and Evaluation

After training the deep learning models, we can digitize an image as follows: the image undergoes skew correction to return it to a proper orientation, lead detection using YOLOv7, lead binarization using the U-Net, and finally signal extraction. After each signal has been extracted, it gets stored within a DAT file for evaluation on the hidden test set [14]. The evaluation metric used is signal-to-noise ratio (SNR) [15].

3. Results

For our best YOLOv7 model, we obtained a weighted sum of 0.897, where mAP@0.5 was 0.972 and mAP@0.5:0.95 was 0.888, from the held-out subset. From a visual inspection of the detected leads, the YOLOv7 model performed well for many of the standard images like the one shown in Figure 1, but did not perform so well for images that were noisy, had a darker grid color, and a smaller resolution like the image shown in Figure 2. Additionally, we observed that the model had the most difficulty in detecting the rhythm lead out of all the leads.

For our best U-Net model, we obtained a Dice Score of 0.991 on the held-out subset. When performing a visual

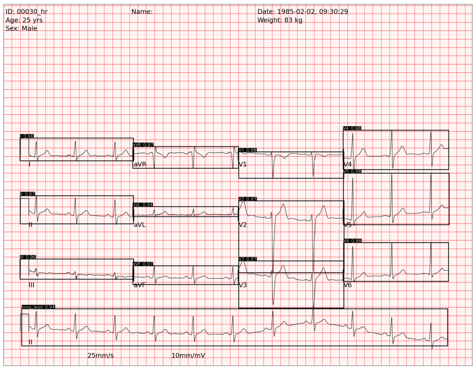

Figure 1. Example of good lead detection

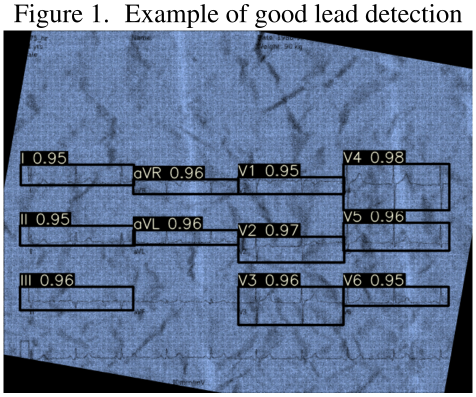

Figure 2. Example of sub-optimal lead detection

inspection, we noticed that the binary masks did well for many of the images with augmentations like wrinkles as seen in Figure 3. However, the quality of the binary masks decreased when the images were noisy and of a lower resolution. To illustrate, Figure 4 visualizes the outputted mask of lead V6 from the image within Figure 2.

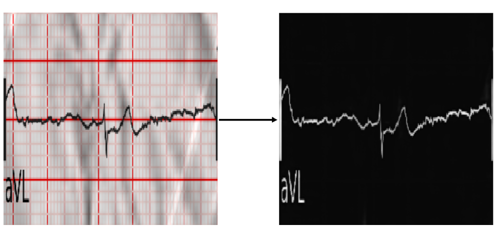

Figure 3. Example of good lead binarization

The challenge SNRs calculated for our training set and held-out subset were 2.979 ± 7.212 and 3.099 ± 7.107, respectively. However, these are not directly comparable to the official Challenge score on the hidden test set, which is summarized in Table 1 along with our ranking in the Challenge.

4. Discussion and Conclusions

We explored several other strategies for different steps.

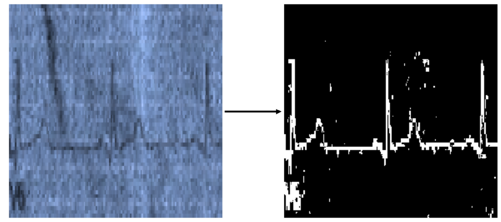

Figure 4. Example of sub-optimal lead binarization

Task	Score	Rank
Digitization	SNR: -5.192	15/16

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

Initially, we tried binarizing the entire image instead of each lead section. Our first attempt was converting the image from RGB to HSV and using Canny edge detection, which produced good masks for varying grid colors but failed with noisy images. Our second attempt involved training the U-Net to binarize the entire image, which worked for the augmented images when they had a uniform resolution, but performance dropped significantly for generated images of varying resolutions. Consequently, we switched to training YOLOv7 for lead detection followed by lead binarization, which improved performance and reduced training time and memory usage.

We attempted to fine-tune foundation models such as Segment Anything [16] to perform lead binarization. However, we had difficulties obtaining a fine-tuned model that performed lead binarization properly.

For lead detection, we initially fine-tuned YOLOv7 to detect short leads and rhythm lead on each image rather than the individual short leads and rhythm lead, binarize each lead section, then use optical character recognition (OCR) to detect the lead name. However, OCR led to sub-optimal results as it had trouble detecting leads I, II, and III due to them looking the least character-like.

For signal extraction, instead of using the signal extraction algorithm developed by ECG-Miner [13], we initially used an algorithm that would search each column for the first white pixel within the binarized lead section and store its location. Although this worked fine for many of the lead sections we tested, it worked poorly for lead sections that were not binarized well or when there was intereference between lead sections. Employing ECG-Miner's signal extraction algorithm resulted in cleaner signals and higher SNRs on our training and held-out subset datasets.

Our approach has a couple limitations. The skew detection algorithm fails for rotations of 45 degrees or more, making it unable to handle vertical scans. Implementing a vision transformer or another advanced computer vision technique could address this issue. For lead detection, YOLOv7 occasionally missed the rhythm lead, possibly due to rotation or the rhythm lead's length. Adjusting the class weights for the rhythm lead might improve detection. The U-Net struggled with noisy, low-resolution sections, which could be mitigated by applying morphological operations like dilation and erosion to remove noisy pixels. The pixel location-to-millivolt conversion in our signal extraction algorithm heavily relies on the calibration pulse, suggesting the need for a more advanced algorithm, such as a deep learning model, to predict these values. Lastly, our image synthesis could benefit from more augmentations, like random padding to simulate backgrounds, or more importantly, generating black and white images, lack of which resulted in poor SNR within the hidden test set. However, these were limited by difficulties in obtaining masks from padded images, and the training set was heavily imbalanced towards colored images as opposed to black and white images, suggesting that future work includes generating more black and white images.

In conclusion, the PapyrusECG pipeline effectively combines deep learning and rule-based techniques to digitize paper-based ECGs, unlocking valuable cardiac information. The challenge SNR results of 2.979 ± 7.212 for the training dataset and 3.099 ± 7.107 for the held-out subset of the training set indicate the effectiveness of our pipeline, despite some limitations. On the hidden test set, we received a score of -5.192, highlighting areas for further improvement, including a more diverse set of images, better skew correction, more robust solutions for lead detection and binarization, and a less calibration pulse dependent signal extraction algorithm. Despite these challenges, digitizing ECGs can significantly enhance AI-based diagnostic systems, contributing to advancements in cardiac electrophysiology and disease diagnosis. Future work will focus on optimizing the pipeline, addressing limitations, and expanding its application to diverse ECG datasets, ultimately aiming to improve cardiovascular healthcare outcomes.

Acknowledgments

We have no conflicts of interest to declare.

References

[1] Barold SS. Willem Einthoven and the Birth of Clinical Electrocardiography a Hundred Years Ago. Cardiac Electrophysiology Review Jan. 2003;7:99–104.

[2] Parvaneh S, Rubin J, Babaeizadeh S, Xu-Wilson M. Cardiac arrhythmia detection using deep learning: A review. Journal of Electrocardiology 2019;57:S70–S74. ISSN 0022-0736.

[3] Waits GS, Soliman EZ. Digitizing paper electrocardiograms: Status and challenges. Journal of Electrocardiology 2017;50(1):123–130.

[4] Wu H, Patel K, Li X, et al. A fully-automated paper ECG digitisation algorithm using deep learning. Scientific Reports 2022;12:20963.

[5] Goldberger A, Amaral L, Glass L, Hausdorff J, Ivanov P, Mark R, Mietus J, Moody G, Peng C, Stanley H. Physiobank, Physiotoolkit, and PhysioNet: Components of a new research resource for complex physiologic signals. Circulation 2000;101(23):e215–e220.

[6] Shivashankara K, Deepanshi, Shervedani A, Reyna M, Clifford G, Sameni R. ECG-Image-Kit: a synthetic image generation toolbox to facilitate deep learning-based electrocardiogram digitization. Physiological Measurement 2024; 45:055019.

[7] Deepanshi, Shivashankara K, Clifford G, Reyna M, Sameni R. ECG-Image-Kit: A Toolkit for Synthesis, Analysis, and Digitization of Electrocardiogram Images, January 2024.

[8] Wagner P, Strodthoff N, Bousseljot R, Kreiseler D, Lunze F, Samek W, et al. PTB-XL, a large publicly available electrocardiography dataset. Scientific Data 2020;7:154.

[9] Strodthoff N, Mehari T, Nagel C, Aston P, Sundar A, Graff C, et al. PTB-XL+, a comprehensive electrocardiographic feature dataset. Scientific Data 2023;10:279.

[10] Brunner S. deskew: Library used to deskew a scanned document. `https://github.com/sbrunner/deskew`, 2024.

[11] Wang CY, Bochkovskiy A, Liao HYM. YOLOv7: Trainable bag-of-freebies sets new state-of-the-art for real-time object detectors, 2022. URL `https://arxiv.org/abs/2207.02696`.

[12] Ronneberger O, Fischer P, Brox T. U-Net: Convolutional Networks for Biomedical Image Segmentation, 2015. URL `https://arxiv.org/abs/1505.04597`.

[13] Santamónica AF, Carratalá-Sáez R, Larriba Y, Pérez-Castellanos A, Rueda C. ECGMiner: A flexible software for accurately digitizing ECG. Computer Methods and Programs in Biomedicine 2024;246:108053. ISSN 0169-2607.

[14] 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`.

[15] Reyna M, Deepanshi, Weigle J, Koscova Z, Elola A, Seyedi S, Campbell K, Clifford G, Sameni R. Digitization and Classification of ECG Images: The George B. Moody PhysioNet Challenge 2024. PhysioNet Challenge 2024;51:1–4.

[16] Kirillov A, Mintun E, Ravi N, Mao H, Rolland C, Gustafson L, Xiao T, Whitehead S, Berg AC, Lo WY, Dollár P, Girshick R. Segment Anything, 2023. URL `https://arxiv.org/abs/2304.02643`.

Address for correspondence:

Saman Parvaneh
1 Edwards Way, Irvine, CA, USA 92614
parvaneh@ieee.org