ECG Paper Record to Digital Signal Conversion Challenge

Author:
Rafael López García

October 24, 2023

Abstract

This challenge involves developing an image processing algorithm using *OpenCV* or similar libraries to convert ECG paper records into digital signals. The process includes image preprocessing to enhance quality, extracting ECG traces from various layouts, and digitizing these traces. The provided dataset contains diverse ECG paper record images. The solution encompasses reading, preprocessing, segmentation, and digitization, enabling the conversion of paper records into digital ECG data.

Keywords: ECG (Electrocardiogram); Image Processing; Digital Signal Conversion; Medical Imaging

1 Introduction

This work is centered around a critical challenge in healthcare: the conversion of paper-based electrocardiogram (ECG) records into digital ECG signals. This task is essential for making the data accessible for more advanced digital processing and analysis. The work has been divided into three fundamental stages, each of which contributes to the success of the conversion process.

In the first phase, an "Image Preprocessing" step has been carried out. Here, the initial step involves reading the ECG record image and applying a series of techniques, filters, and transformations. The primary goal is to enhance the quality and clarity of the image. This enhancement not only aids in visual interpretation but also lays the foundation for the subsequent separation of the signal into various traces that will be digitized.

The second part of this project focuses on the extraction of ECG traces. This stage is critical for automation. Specific image processing techniques designed for these kind of problems are employed. The outcome is the automatic identification and separation of individual traces, for the different images presented in the studied dataset.

Finally, the third stage involves transforming the original signal into a digital representation. This is achieved through a digitization process that includes filtering the original signal. In the third stage, filtering and resizing not only simplifies the storage of the signal in digital format but also ensures that the signals are in the same format, making it easier to compare them and reproduce the information contained in the original paper record.

These three phases come together to create an automated system that effectively converts paper-based ECG records into digital ECG signals, enabling their access and advanced analysis in healthcare settings.

In addition to these three phases, I have undertaken the development of the Bonus section of the Challenge. This final segment has been divided into two distinct parts. Firstly, I've created a simple web application equipped with a friendly user interface. Within this application, users can effortlessly upload images of ECG paper records. The application leverages the code developed during the preceding three phases to perform pre-processing, trace extraction, and signal digitization, and allows users to obtain the digitizated singal and download it.

Moving on to the second part, I researched various deep learning techniques aimed at enhancing the tool's performance. Specifically, my focus was on image segmentation algorithms, especially those pretrained with databases of scanned documents, as I wanted to try a different approach on the trace extraction. However, despite these efforts, the results obtained failed to meet the desired level of improvement and did not surpass the performance achieved with the earlier version of the application. The primary challenge stemmed from the lack of specialized models and data tailored to this specific problem, ultimately limiting the efficacy of these more advanced techniques in enhancing the tool's accuracy.

2 Stage 1: Image Preprocessing

The primary objective of the Image Preprocessing stage is to prepare paper-based electrocardiogram (ECG) record images for further digital processing. This phase involves multiple steps aimed at improving image quality, aligning orientation, and ensuring consistency across all images:

Orientation Correction: To standardize and align all ECG images, orientation correction is applied, which unfolds in two distinct phases:

1. Orientation Determination: The purpose of this step is to determine the initial common orientation for the ECG image. It uses the image's histogram to establish

a reference point. This common orientation ensures that all ECG images are aligned consistently, making further processing more accurate. This step was made because using only corner and line detection was not sufficient in some cases, as when, for instance, the image was rotated 180º, the lines and corners were oriented in such a way that the corrected image ended up reversed.

2. Corner and Line Detection: After the initial orientation is established, corner and line detection algorithms (Canny and Hughes) are employed. These algorithms detect important features like corners and lines within the image. Their primary role is to fine-tune the image's orientation correction. By detecting these features, the algorithms calculate the required rotation angle and use a rotation matrix to rectify the image's orientation. This ensures that the image adheres to the desired alignment.

Image Enhancementand Preparation: Once the image is correctly oriented, the following steps are performed to prepare the image for the future digitization.

1. Grayscale Conversion: Grayscale conversion is performed to simplify the image and reduce noise. The conversion transforms the image from color to grayscale, which is a single-channel representation of the image. Grayscale simplification aids in subsequent processing steps by eliminating color variations that are not relevant for the analysis. It also enhances the image's suitability for further enhancements.

2. Gaussian Filtering: Gaussian filtering is used to enhance image details. This process applies a Gaussian filter to the grayscale image. A Gaussian blur filter is applied to the grayscale image to mitigate noise and irregularities. This smoothing operation enhances image quality by removing unwanted details, preparing it for thresholding.

3. Otsu Thresholding: Otsu thresholding is employed to convert the grayscale image into a binary image. In this step, the algorithm determines an optimal threshold value to distinguish between the foreground (the ECG signal) and the background. The process effectively binarizes the image, making it easier to isolate and analyze the ECG signal, which stands out from the background as a binary representation.

These steps collectively prepare the paper-based ECG record image for subsequent digital processing. They correct its orientation, enhance its quality, and simplify its representation for more accurate and efficient analysis. The processed image serves as the foundation for the conversion of paper-based ECG records into digital signals.

The Figures 1, 2, 3, 4 show the original image and its transformations. In this case, the original image had the correct orientation, so the rotated image is not shown.

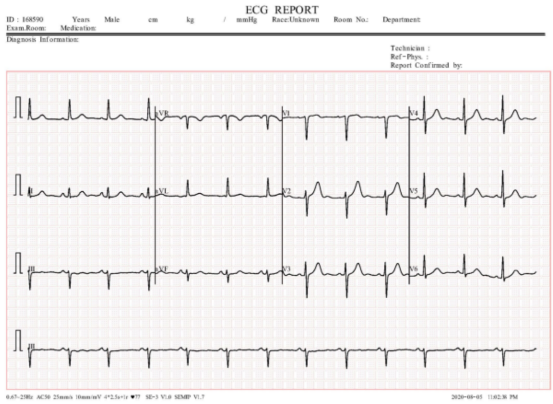

Figure 1: Original Image.

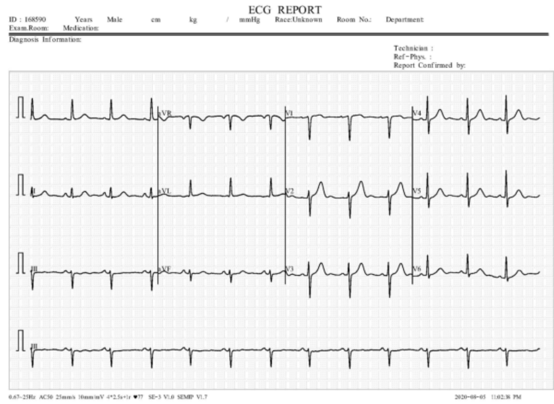

Figure 2: Greyscale Image.

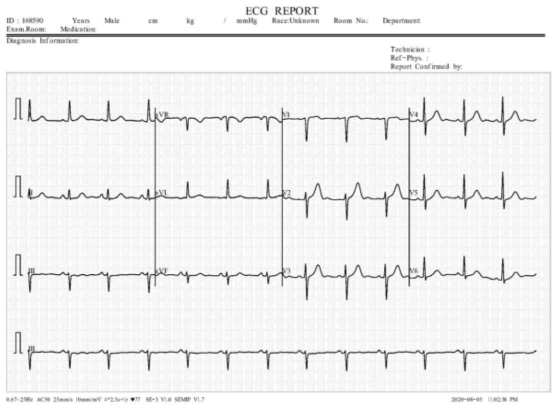

Figure 3: Smoothed Image.

Figure 4: Binary Image.

In the Appendix (section 6), various examples showcasing the transformations discussed earlier are provided. These examples include demonstrations of orientation corrections and enhancements and preparations for the image.

3 Stage 2: ECG Trace Extraction

In the second stage, the focus was on the extraction of electrocardiogram (ECG) traces from ECG paper records represented as images. This stage involved several image processing steps aimed at isolating and segmenting the individual ECG traces within the image.

Contour Detection and Sorting: The first step involved detecting contours in the binary image of the ECG paper record. Contours are regions of interest that can potentially contain the ECG traces. Contours were sorted in descending order of their areas to ensure that the largest contours corresponded to the ECG traces, which are typically the most prominent features in the image. The two largest contours correspond to the area where the 12 traces are together and to the area where the 13th trace is placed.

Figure 5: Crop with the 12 traces.

Figure 6: Crop without vertical lines.

Figure 7: Crop with the 13th trace.

Filtering and Cropping: To extract the ECG traces, the detected contours were filtered based on a threshold area. For each selected contour, a series of operations were applied to adjust and crop the region of interest around the contour. This process involved increasing the size of the bounding rectangle around the contour by a certain percentage and ensuring that the resulting coordinates were non-negative. This was made beacause without this enlargement, in the next step when the traces are differentiated, some of them would have truncated areas, and therefore, the optimal result would not be achieved. Figures 5 and 7 show the crops obtained, that belong to the 12 traces together and the 13th trace.

Vertical Edge Detection: In order to be able to diferenciate between the 12 connected traces, the Sobel operator was applied to detect vertical edges, as the traces are separated by a vertical line. The Sobel operator enhances areas of rapid intensity change in the image, which often correspond to edges. This step was only performed on the biggest area, where the 12 traces reamin together as the 13th contour is already separated from the rest.

Dilation and Line Detection: A dilation operation was performed on the Sobel-transformed image to improve the detection of vertical lines. Subsequently, the Hough Line Transform was used to detect lines in the dilated image. These lines represent the divisions between individual ECG traces. Once the lines that separate the traces are detected, these lines are removed so that the traces are not connected to each other. Once the traces are no longer connected, you can apply a contour-finding algorithm, which will return the separated traces as the contours found. Figure 6 shows the crop with 12 traces but without the connecting vertical lines.

Segmentation of 12 ECG Traces: To segment the 12 traces, the first step is to find the contours of the previously vertical-lines erased image. Then, bounding boxes for

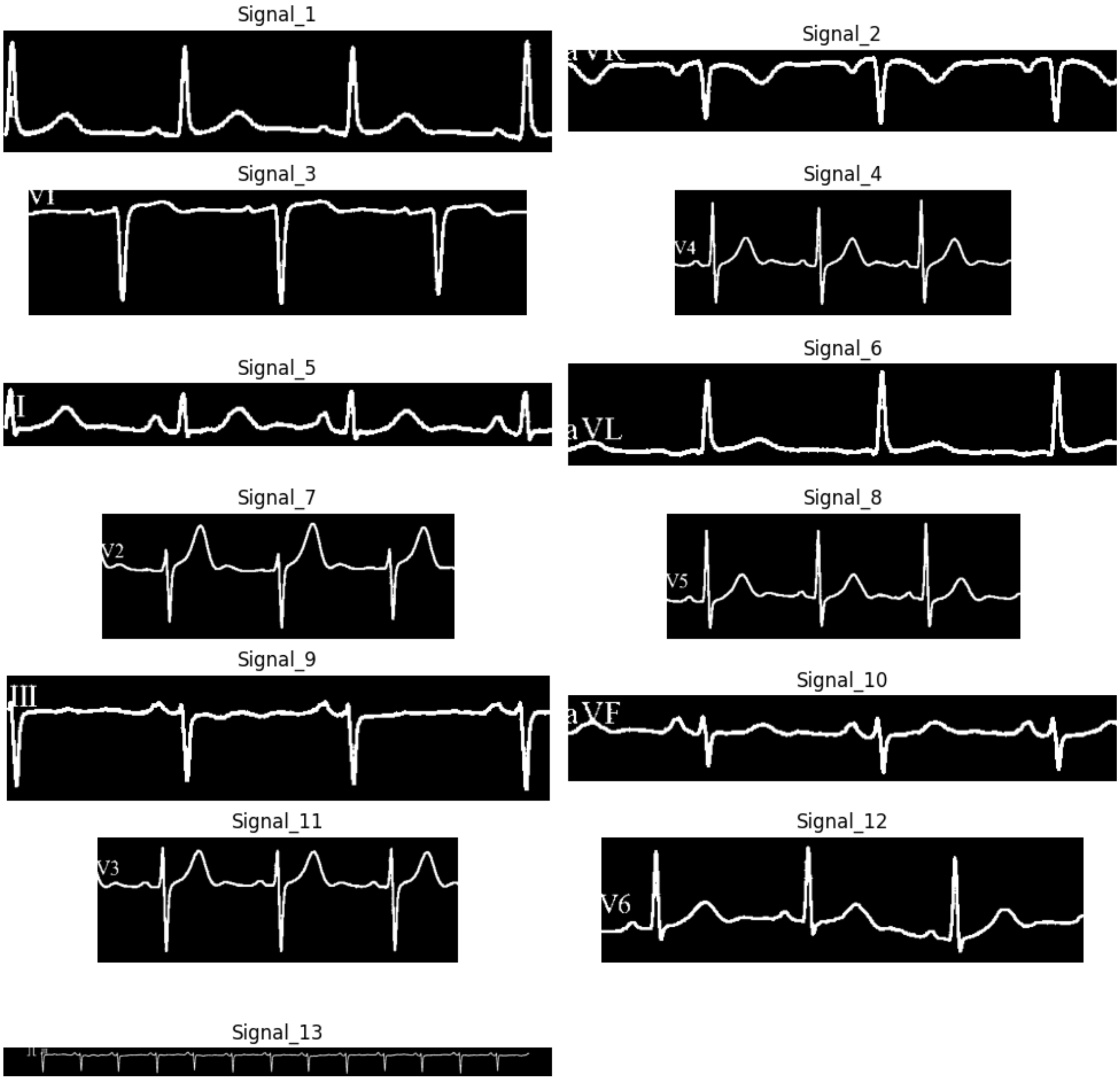

Figure 8: Traces segmented.

these contours are calculated , including their (x, y) coordinates, width, and height. These bounding boxes are filtered based on their area, retaining only those with an area greater than a threshold (to just obtain the traces signals). For each set of sorted coordinates, a crop is obtained with the corresponding region of each of the trace signals (the height of the cropped regions are adjusted by increasing it by 20% to ensure that the entire signal is captured). Figure 8 shows the 13 traces obtained after this step.

In summary, the second stage of the work employed a combination of image processing techniques, including contour detection, edge detection, dilation, and line detection, to accurately extract and segment the 12 ECG traces from the input ECG paper record image. These segmented traces were subsequently available for advanced analysis and interpretation, contributing to the overall goal of digitizing and analyzing paper-based ECG records for healthcare applications.

4 Stage 3: Digitization

The third stage of this project focuses on transforming the paper-based electrocardiogram (ECG) signals into a digital format, making them accessible for advanced analysis and integration into healthcare systems. This stage involves the following key steps:

Signal extraction: After successfully extracting and separating individual ECG traces from the paper records in the previous stage, the next step is to prepare these traces for digitization. These traces are represented then as contours, and the goal is to convert them into a digital format. To identify contours, the approach used was thresholding. Thresholding is a technique that separates objects or features of interest from the background. It sets a certain pixel value as a threshold and classifies all pixels above that threshold as one type and all pixels below it as another. This helps isolate the lines or shapes of interest from the rest of the image. As the thresholding was already performed to obtain the binary image in the firste step, a contour detection algorithm was applied to directly this image. Contours are essentially the outlines of objects or shapes within the image. The algorithm identifies areas of the image where pixel intensity changes, signifying the presence of an object boundary or shape. Among the detected contours, the algorithm selects the largest one. This is often done because the largest contour is likely to represent the primary object or feature of interest in the image. The size of a contour is determined by the number of points that define it. This is the case, since the trace images biggest contour is the signal itself.

The selected largest contour may undergo further processing, such as resizing or smoothing, depending on the final application to ensure that it meets specific requirements or standards. This processing can enhance the quality and usability of the contour data. Figure 9 shows the 5 first traces with their corresponding contours. Figure 18 shows all the traces crops and thir corresponding contours. It is displayed in the appendix, as it is a large image.

Signal Smoothing: To ensure the accuracy and consistency of the digitized signals, a smoothing process is applied. This process involves reducing noise and enhancing the clarity of the ECG traces. It improves the visual interpretation of the signals and prepares them for further analysis. Depending on the final application, the smoothing applied will differ or will not be used.

Signal Normalization: After smoothing, the ECG signals are normalized to ensure that they fall within a consistent numerical range. Normalization is essential for standardizing the signals, making them comparable, and facilitating meaningful analysis. It ensures that all signals have a similar scale and range of values.

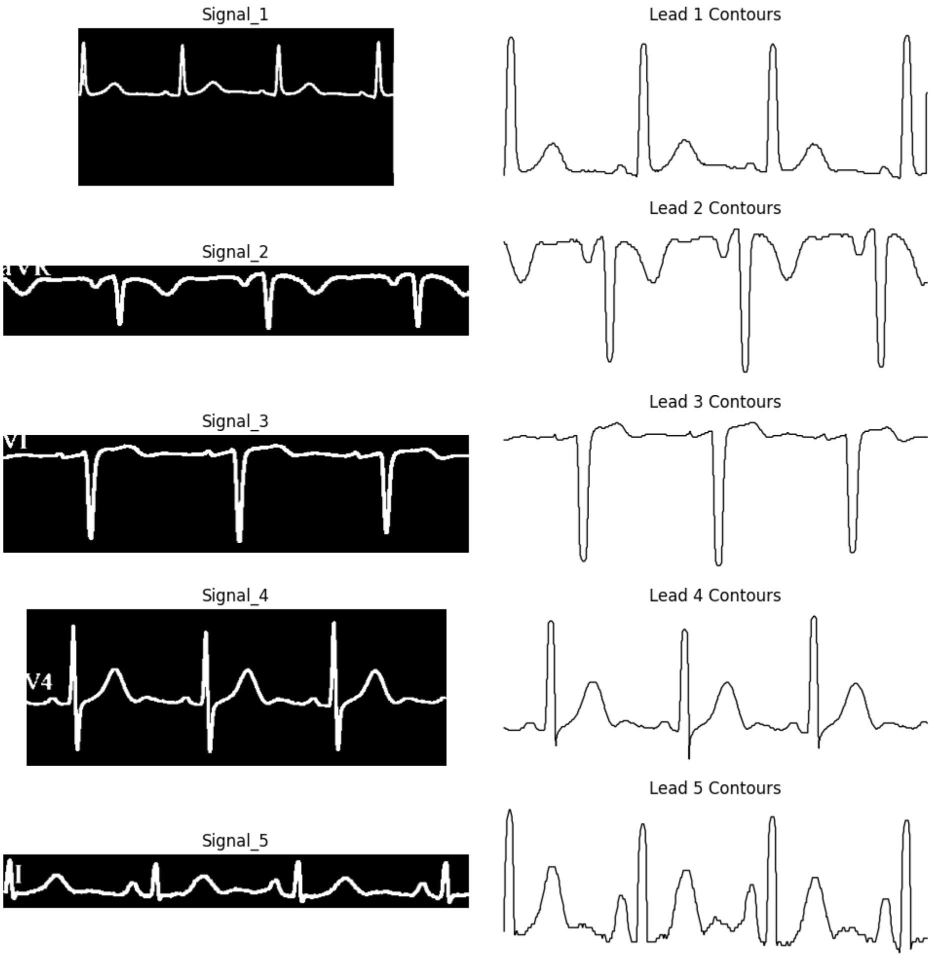

Figure 9: Traces and corresponding contours for the first 5 signals.

Data Storage: The digitized ECG signals, both in their original and normalized forms, are stored in separate data files. These files serve as the digital representation of the ECG data. Storing the data in this format enables easy access, sharing, and integration into healthcare information systems.

Data Visualization: To provide a visual comparison between the original and digitized ECG signals, visualizations are generated. These visualizations help demonstrate the effectiveness of the digitization process and provide a clear view of how the signals have been transformed from their paper-based form. Figure 10 represents the original and smoothed normalized signlas for the first 4 signlas. The 13 signlas are shown in Figure 19 in the appendix.

By completing this third stage, the work successfully achieves the conversion of paper-based ECG records into digital ECG signals. Figure 11 shows the final reconstruction of the image, where all the signals extracted from the traces and digitizated are shwon together to form an image like the one provided. These digital representations are now ready for advanced analysis, interpretation, and integration into healthcare applications,

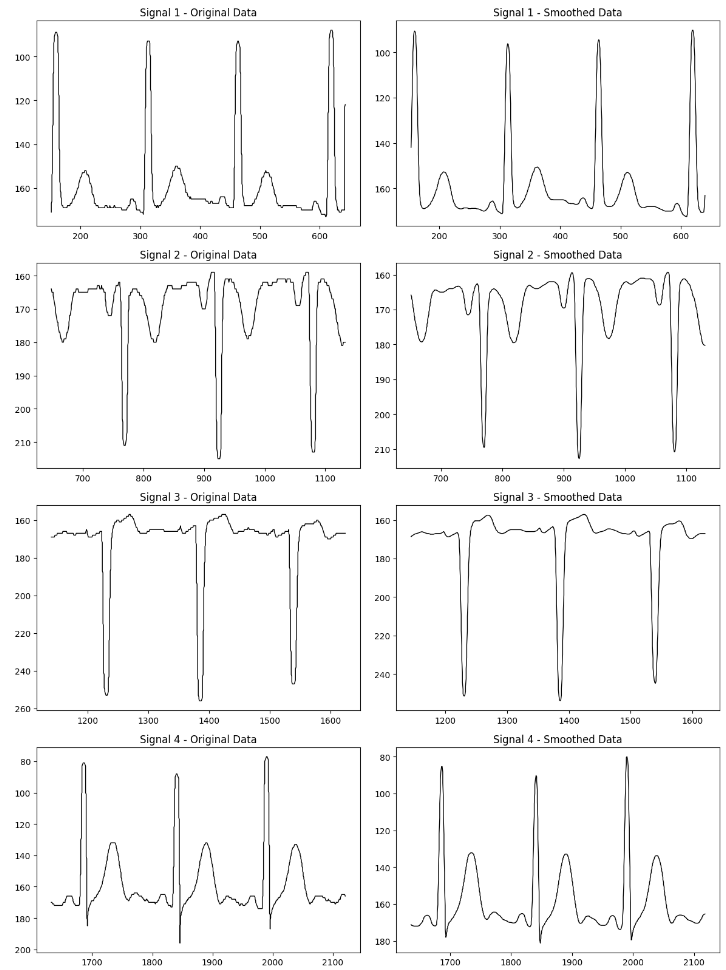

Figure 10: Original and smoothed first 4 signals representation read from csv.

ultimately contributing to improved patient care and diagnosis. This digitization process enhances the accessibility and usability of vital medical data in healthcare systems.

5 Bonus Part 1: App Web Development

Within the Challenge, an additional Bonus section was undertaken, divided into two specific parts. The primary focus of this section is the development of a web-based application made for user interaction and functionality.

The purpose of this web application is to allow users to upload ECG paper record images. Upon uploading, the application automatically processes the image, implementing stages such as pre-processing, trace extraction, and the subsequent digitization of the ECG signal. The end result provides users with a digitized version of their ECG record, available for download. In the app, users can view different image transformations (greysacle, smoothed, binary, etc) and can also view the different traces extracted, as well as the digitizated signals from them.

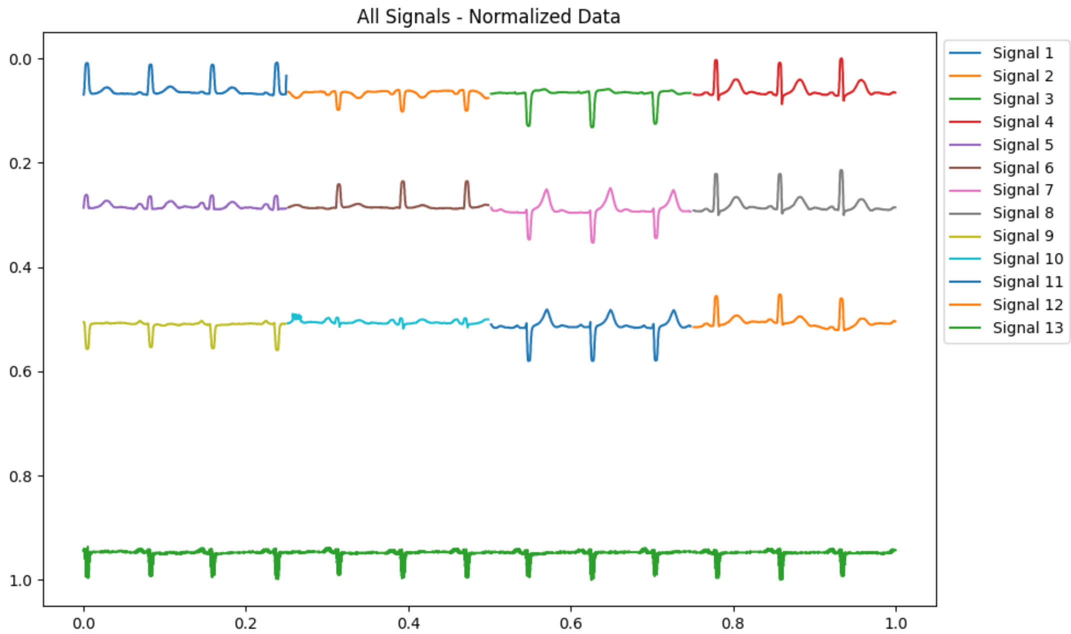

Figure 11: Normalized digitizated signals.

The base of this application is a Python script designed for the processing and analysis of ECG images. The development environment chosen for this project was Google Colab, which provides a platform for code development and execution, ensuring reproducibility and streamlined deployment. The application's functionality and accessibility are enhanced with the integration of tools such as Streamlit and Ngrok.

Streamlit is an open-source Python library used for the creation of web applications, especially in data science and machine learning topic.

Ngrok, offers a solution for exposing local servers to the internet. This tool securely connects a locally hosted web server to the web, producing a temporary public URL for the locally hosted application. This feature allows sharing the application to external users.

To operate the application, users are required to execute specific cells within the Google Colab notebook. The link to the notebook can be found in the following url (`https://github.com/Rafaloga/ECG-Paper-Record-to-Digital-Signal-Conversion-Challenge/blob/main/code/User_Friendly_Interface.ipynb`).

6 Bonus Part 2: Exploring Deep Learning Techniques

In the second segment of the Bonus section, deep learning techniques were explored with the intention of enhancing the application's performance. The primary focus was on image segmentation algorithms, particularly those pretrained on databases of scanned documents, to potentially offer a better approach to trace extraction.

Two main models were evaluated: Detectron2, pretrained on the Publaynet database, and U-net, pretrained on a generic database.

Detectron2 is recognized for its capabilities in image segmentation. The decision to use it was based on its broad applicability in segmentation tasks and its pretraining on Publaynet. Given that Publaynet is a database composed of scanned documents, it was anticipated that this combination might yield good segmentation results for scanned ECG records. However, the images in Publaynet are primarily of textual content with occasional tables and figures, which differ substantially from ECG trace images. This difference resulted in less than optimal segmentation outcomes.

U-net, on the other hand, is well-known for its effectiveness in medical image analysis. While its utility spans various domains, its pretraining on a generic database did not align well with the specific nature of ECG trace images, causing it to produce unsatisfactory results in this context.

In conclusion, while both models have demonstrated effectiveness in their respective domains, their application to this particular challenge was not successful. The primary limitation was the absence of specialized models and datasets made for ECG trace images.

A potential solution to enhance the performance of these models would be to obtain a database containing ECG images with their traces already segmented. Having such a database would allow for the training of models using data that mirrors the kind of images they would subsequently segment. This approach would likely yield more accurate and effective results.

7 Evaluation and Conclussions

The development of an image processing algorithm to convert ECG paper records into digital signals is an importnant process in healthcare technology. This section will provide a detailed evaluation of the algorithm's performance, discuss its limitations, and suggest potential improvements.

Performance Evaluation: The algorithm has shown promising results in converting paper-based ECG records into digital signals. Each stage of the process has been designed to ensure accuracy and efficiency. Here are some key points regarding the algorithm's performance:

- Image Preprocessing: The orientation correction, grayscale conversion, Gaussian filtering, and Otsu thresholding collectively enhance image quality and simplify the representation for further processing. It allows the application to use different types of records, varying in light, angle, quality, etc. The examples in the appendix demonstrate the effectiveness of these preprocessing steps.

- ECG Trace Extraction: The algorithm successfully extracts and segments individual ECG traces from diverse paper record layouts. The combination of contour detection, filtering, cropping, edge detection, and line detection has proven to be robust in handling variations in paper records. This automatization of the ECG extraction helps reducing the need for manual intervention and significantly accelerates the process of converting paper-based ECG records into digital signals.

- Digitization: The algorithm's digitization process, including signal extraction, smoothing, and normalization, prepares the ECG traces for digital storage and analysis. The visualizations provided give a clear indication of the quality of the digitized signals.

Limitations: Despite its performance, the algorithm has certain limitations that should be acknowledged:

- Variability in Paper Records: While the algorithm handles variations in paper record layouts among the data provided, other complex or damaged paper records may pose challenges. Further robustness testing and adaptations for new cases may be required.

- Parameter Tuning: Some stages of the algorithm, such as thresholding and contour detection, rely on predefined parameters. Fine-tuning these parameters for different datasets or paper record qualities may be necessary to achieve optimal results.

- Real-time Processing: The algorithm has been designed for batch processing of ECG paper records. Adapting it for real-time applications, such as continuous monitoring, would require optimization.

Potential Improvements: To further enhance the algorithm's capabilities and address its limitations, the following improvements can be considered:

- Deep Learning Integration: Incorporating machine learning models for image recognition and segmentation can improve the algorithm's ability to handle diverse paper record layouts and variations. These techniques could be used for segmenting the different traces and extracting their contours for future digitization. To do this, it would be advisable to obtain an annotated database for the specific problem or manually annotate the data used in this problem. This dataset should include both the original image and the already separated traces to train a model.

- User Interface: Developing a user-friendly interface that allows healthcare professionals to interact with the algorithm and manually intervene or validate results when needed can improve usability.
- Dynamic Parameter Adjustment: Implementing a mechanism to dynamically adjust parameters based on the characteristics of the input image can enhance adaptability and robustness. With the help of an expert, it could be possible to address the common and differentiating characteristics for each type of ECG, enabling the use of specific parameters for each type.

In conclusion, the development of an automated system for converting ECG paper records into digital signals using image processing techniques is a significant step toward improving healthcare data accessibility and analysis. With further development and integration of advanced technologies, this system can contribute significantly to healthcare diagnosis and research.

8 Appendix

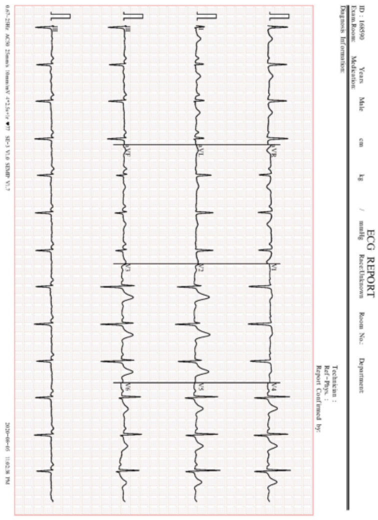

Figure 12: Original Image.

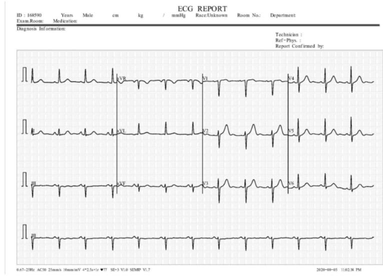

Figure 13: Rotated Greyscale Image.

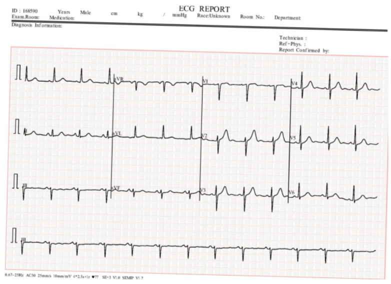

Figure 14: Original Image.

Figure 15: Rotated Greyscale Image.

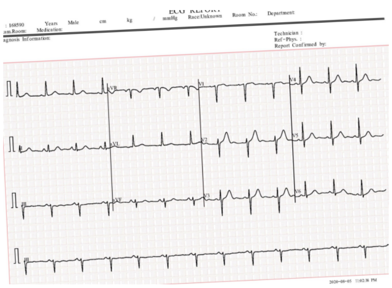

Figure 16: Original Image.

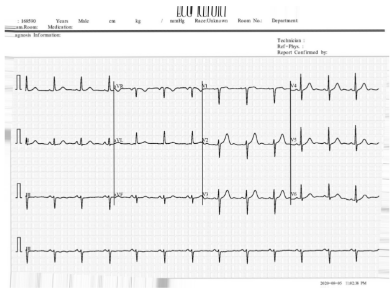

Figure 17: Rotated Greyscale Image.

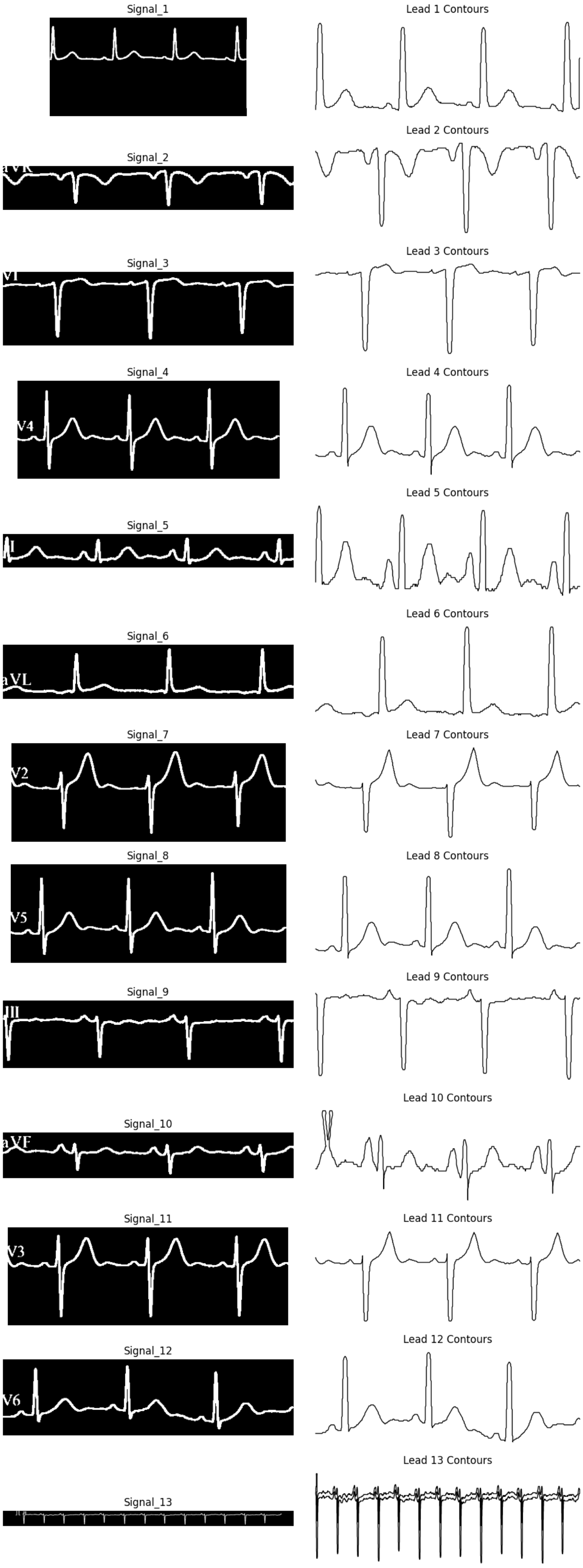

Figure 18: Traces and corresponding contours for all the signals.

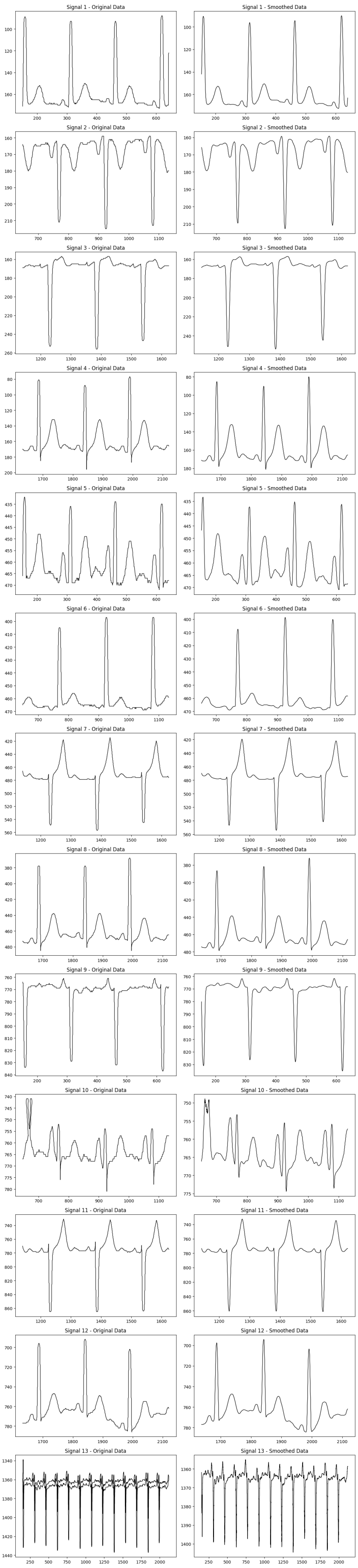

Figure 19: Original and smoothed signals representation read from csv.