A Data Pipeline for Extraction and Processing of Electrocardiogram Recordings

Joshua Prim[1], Tim Uhlemann[1], Nils Gumpfer[1], Dimitri Grün[2], Sebastian Wegener[2], Sabrina Krug[1], Jennifer Hannig[1], Till Keller[2], Michael Guckert[1,3]

[1] Cognitive Information Systems, Kompetenzzentrum für Informationstechnologie, Technische Hochschule Mittelhessen, 61169 Friedberg, Germany
[2] Department of Internal Medicine I, Cardiology, Justus-Liebig-University Gießen, 35390 Gießen, Germany
[3] Department of MND - Mathematik, Naturwissenschaften und Datenverarbeitung, Technische Hochschule Mittelhessen, 61169 Friedberg, Germany

Abstract

Algorithmic exploitation of medical data for diagnostic purposes has become state of the art in the modern medical world. Applying artificial intelligence algorithms is gaining importance and electrocardiogram recordings have successfully been used as input for deep learning models and produce viable diagnoses. Algorithms are non-invasive, relatively low-cost and promise to have high diagnostic leverage. However, for supervised learning algorithms such as deep learning models the amount of high quality data labelled with correct diagnoses required for training is considerable. In this paper, we present a pipeline that processes raw electrocardiogram recordings preparing them for use in training and validation of neural network models. Although, the electrocardiogram is widely used, appropriately labelled training data is rare and provided in different formats and from technically different sources. Therefore, our end-to-end pipeline not only processes data from modern digital ECG devices, e.g. in XML file format, but can also extract all necessary information from PDF files (both scanned hard copies and digitally generated PDFs). We present a use case in which data from XML and PDF sources is read, cleaned and combined into a unified dataset to be used by a model predicting myocardial scar. Our pipeline will become a cornerstone of our environment for building AI based diagnostic instruments.

1. Introduction

Artificial intelligence (AI) has attracted global attention as an algorithmic tool for innovative solutions in several disciplines, with medicine being a prominent example [1]. AI algorithms are increasingly being used to assist physicians in making diagnoses, e.g. AI is achieving competitive results in the analysis of electrocardiogram (ECG) recordings. The ECG is a human interpretable visualisation of the electrical activity of the heart, measured over time via skin electrodes. We recently showed the potential of AI-based methods in cardiology using a meta-analysis [2]. Additionally, we proposed an AI model for myocardial scar detection based on ECG data [3].

An important success factor for AI algorithms is the quality of training data. A significant amount of reliable data with associated correct labels and representative samples is required during training and validation when using supervised learning algorithms such as deep learning models. Although the ECG is a widely used diagnostic technique, the amount of labelled data available for training of AI models is often limited. Modern ECG devices record measured signals digitally and provide data in accessible formats, such as Extensible Markup Language (XML) files. However, existing structures often do not archive the digital version, further, some manufacturers only export Portable Document Format (PDF) files instead of raw time series information or even print the ECG on paper directly, requiring complicated subsequent processing of PDF files or images.

In this paper, we present an end-to-end data pipeline for extracting ECG recordings from XML and PDF, allowing researchers to use currently hard-to-access ECGs in PDF format, as well as data from native digital ECG devices delivered in XML, for training of AI models. In addition, the pipeline is also able to extract all the necessary information not only from digitally generated PDFs but also form scanned printouts. The proposed data pipeline unifies ECGs in heterogeneous formats to be processed flexibly in AI models.

The paper is organised as follows: First, we briefly discuss related work on PDF-based information extraction. Then, we describe the procedure of lead extraction from PDFs and processing as a fundamental step of our pipeline. Finally, we validate the proposed method with a use case where ECG recordings (XML and PDF) are used to train a deep learning model for myocardial scar detection.

2. Related Work

ECG digitisation and conversion of ECG paper records to digital signals has variously been addressed. The automated data extraction system for converting ECG paper records to digital time databases was studied by Mitra et al. [4]. They performed the discrete Fourier transform of the generated database to observe the frequency response properties of every ECG signal. An image processing engine that first detects the underlying grid and then extrapolates the ECG wave-forms using a technique based on active contour modelling was described by Badilini et al. [5]. Swamy et al. [6] proposed an improved algorithm for the existing paper ECG trace to digital time series with adaptive and iterative image processing techniques. Further, their technique is enhanced to calculate the heart rate from the obtained time series and shows an accuracy of 95%. Jayaraman et al. [7] proposed a technique for the conversion of the scanned ECG record to a digital time series signal with an improved method of binarisation accurately. Kumar et al. [8] present a conversion of ECG signal from scanned ECG papers. The researchers use MATLAB for obtaining the data for the ECG taken from Indian patients and determine various parameters and abnormalities with an accuracy of 98%. Shi et al. [9] proposed a method based on K-means to extract ECG data from paper recordings. Based on paper recordings of 105 patients the researchers got a precision rate of 99% with their approach. An algorithm to extract ECG signals automatically from scanned 12 lead ECG paper recordings by operations including edge detection, image binarisation, and skew correction was designed by Sun et al. [10]. Mishra et al. [11] used a deep learning model to get a threshold value that separates ECG signal from its background and after applying various image processing techniques threshold ECG image gets converted into digital ECG.

3. Data Pipeline for ECG extraction

In this section, we describe the main steps of ECG lead extraction and preparation for XML and PDF files. The principal structure of the pipeline is provided in figure 1.

3.1. XML Extraction

Extraction from XML files is based on the health level seven version 3 (HL7v3) schema, which is the clinical standard for digital ECGs. To parse the leads from the XML structure, the python library *minidom* is used to generate an instantiation of a *document object model* (DOM). This model mainly consists of a hierarchical tree-structure of objects and attributes in which data contained in the XML document is inserted. The leads of the ECG are represented as sequences of numeric values and can be accessed by using their associated identifying labels. Extracted time series, lead-labels, and metadata are then exported in comma-separated-value (CSV) format or directly forwarded to the neural network for training.

3.2. Generated PDF Extraction

Extraction of structured numerical information such as time series from generated PDF documents is complex, as vector based plots have to be reverse-engineered to the original temporal structure without loss of information. In the PDF, the ECG time series curves are linearly approximated with small line segments (i.e. linear PEs) defined by the coordinates of their discrete endpoints. These lines have to be transformed from PDF coordinates to a sequence of correctly scaled amplitude values with the time structure of the original ECG.

Each end point of the line segments represents a deflection in the ECG. These deflections are collected, scaled, and translated to be zero centred. Depending on the orientation of the PDF document (portrait or landscape), the points may have to be rotated around the origin of the coordinate system.

For lead scaling, the *reference pulse* or *calibration jag* is extracted from the PDF. Its height h provides a reference length representing a deflection of $1mV$. This distance is used to calculate the scaling factor

$$= \frac{1\,mV}{h}$$

which is used to scale the extracted Y-coordinates by multiplication.

The values of each lead are shifted to be zero centred. For this, a window of size ws synchronously slides over the values in each lead locating the window w' for which the sum over the standard deviations (std) of the values in the window (notated as l_w in Algorithm 3.2) taken over all leads is minimal. Now, for each lead l the mean of the values of l'_w is subtracted from each value of l (see Algorithm 3.2).

The extracted line coordinates are transformed to equidistant points by linear interpolation so that the number of required discrete values is obtained, e.g. 5000 val-

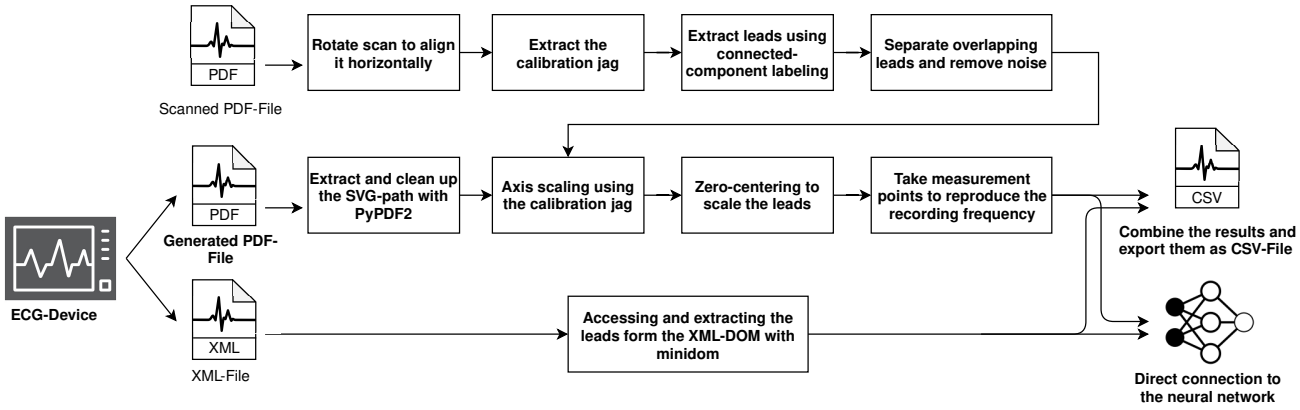

Figure 1. **Overview of the Processing Pipeline.** PDFs and XMLs from ECG devices are extracted, processed and saved as CSV, or directly used.

Algorithm 1: ECG lead zero centring

Data: set of ECG *leads*; window size $ws = 124$; *length* of leads
Result: *leads* shifted to be zero centred
$W = \{\{i, ..., i + ws\} : i = 0, ..., length - ws - 1\}$;
$w' = \underset{w \in W}{\operatorname{argmin}} \sum_{l \in leads} std(l_w)$;
for $l \in leads$ **do**
  $m = mean(l_{w'})$;
  for $value \in l$ **do**
    $value = value - m$;
  end
end

ues for 10 seconds at 500 Hz. The resulting time series finally are labelled and exported to CSV format or directly forwarded to the neural network.

3.3. Scanned PDF Extraction

For the extraction and preparation of the leads from scanned PDFs, meta information such as page number, writing speed, and the time length of the existing recording are first extracted. The open-source OCR engine *Tesseract* in combination with regular expressions are used for this purpose.

Since in practice ECGs are often obliquely scanned, a rotation is performed at this point if necessary. For this purpose, the image is converted to a grey-scale image, possible image noise is removed with the help of Gaussian blur and Canny Edge Detection [12] is performed. The edges found are then examined for straight lines which are used to calculate the angle with which the centre of the image is then rotated before the image is cropped. Finally, the scanned PDFs are converted into arrays of pixels using the Python library NumPy. The grid of the graph paper is filtered out by colouring all pixels with RGB colour values above 150 white. All remaining pixels are colour intensified. Now, all connected components [13] are captured on the recording, labelled, and individual coordinates within the components are stored. During this process, the minimum and maximum number of pixels in the components is restricted to ensure that only the relevant leads of the ECG are found. In the same way, the *calibration jag* is extracted and validated by its typical shape to use for scaling the leads. If two leads overlap and are combined to a connected component, they have to be separated. For this purpose, the number of y-value ranges is counted for each x-value, i.e. if the number of leads is less than the number of y-value ranges, two leads touch at this x-location. By assigning different value ranges to the respective lead, they can thus be separated from each other. The further steps are equivalent to the extraction of generated PDFs described above (see figure 1). The result of the extraction can be seen in figure 2.

4. Validation

To test the implemented PDF and scan extraction, we used a set of 111 ECGs for which data was available in XML and PDF format. All formats were separately used for training and validation of a deep learning architecture for myocardial scar detection as described in [3]. All models were then assessed by their prediction capability. They produced comparable results, reaching AUC (area under the curve) values of $0{:}78 \pm 0{:}10$ (XML), $0{:}82 \pm 0{:}09$ (generated PDF), and $0{:}75 \pm 0{:}13$ (scanned PDF). Note that in contrast to the XML time series, those extracted from generated PDF and scanned PDF underwent standard filtering in the ECG device, e.g. high-/low-pass filters. Possibly, positive influence of these filters may explain the high quality of PDF-based results. However, the results encourage the use of ECG extracted from generated PDF and scanned PDF for training of AI models.