EKG Document

Contents

EKG Document

Chapter 1 – Installation

Libraries Used

1. **Logging:** enables all Python modules to participate in logging, so the application log can include your own messages integrated with messages from third-party modules.
2. **numpy:** a general-purpose array-processing package designed to efficiently manipulate large multi-dimensional arrays of arbitrary records without sacrificing too much speed for small multi-dimensional arrays.
3. **opencv:** (Open Source Computer Vision) is a library of Python bindings designed to solve computer vision problems.
4. **pandas:** a Python package providing fast, flexible, and expressive data structures designed to make working with “relational” or “labeled” data both easy and intuitive.
5. **array:** defines an object type which can compactly represent an array of basic values: characters, integers, floating point numbers.
6. **matplotlib.pyplot:** a collection of command style functions that make matplotlib work like MATLAB. Each pyplot function makes some change to a figure: e.g., creates a figure, creates a plotting area in a figure, plots some lines in a plotting area, decorates the plot with labels, etc.
7. **PyPDF2:** is used to merge PDFs documents together, rotate pages, split and crop pages, and decrypt/encrypt PDF documents.
8. **PythonMagick:** Python binding of the ImageMagick library. ImageMagick is used to create, edit, and compose bitmap images. Images can be cropped, colors can be changed, various effects can be applied, images can be rotated and combined, and text, lines, polygons, ellipses and Bézier curves can be added to images and stretched and rotated.
9. **os:** provides a portable way of using operating system dependent functionality.
10. **re:** (Regular expression operations) uses the backslash character ('\') to indicate special forms or to allow special characters to be used without invoking their special meaning.
11. **Json:** a lightweight data interchange format inspired by JavaScript object literal syntax.
12. **io:** module provides the Python interfaces to stream handling.
13. **Sys:** (System-specific parameters and functions) provides access to some variables used or maintained by the interpreter and to functions that interact strongly with the interpreter.

EKG Document

Installation steps for mac

1. Please copy the attached folder (EKG_Store) onto your machine at desired location
2. Open a terminal and type the below commands
   a) Install port
   b) sudo port update
   c) sudo port install fontconfig
   d) sudo port install ghostscript
   e) sudo port install ImageMagick
   f) ruby -e "$(curl -fsSL https://raw.githubusercontent.com/Homebrew/install/master/install)" < /dev/null 2> /dev/null; brew install caskroom/cask/brew-cask 2> /dev/null
   g) brew cask install pdftotext
3. cd /FOLDER IN WHICH YOU COPY/EKG_Store/EKG/dist/EKG.extract.automatic
4. ./EKG.extract.automatic / FOLDER IN WHICH YOU COPY/EKG_Store/input / FOLDER IN WHICH YOU COPY/EKG_Store/output/

Installation steps for Linux

1. Goto folder -> EKG/dist/EKG
2. Right click 'EKG' file -> Go to properties -> Permissions -> Allow executing file as program
3. The folowing packages should be installed prior to running EKG executable.
   a) apt-get install ghostscript
   b) apt-get install libgs-dev
   c) apt-get install imagemagick
4. or run EKGpriorpackages.sh from the terminal
5. In command prompt move to directory EKG/dist/EKG -> ./EKG [Full path of the input folder] [Full path of the output folder]

EKG Document

Chapter 2 – Code Architecture

Process Flow

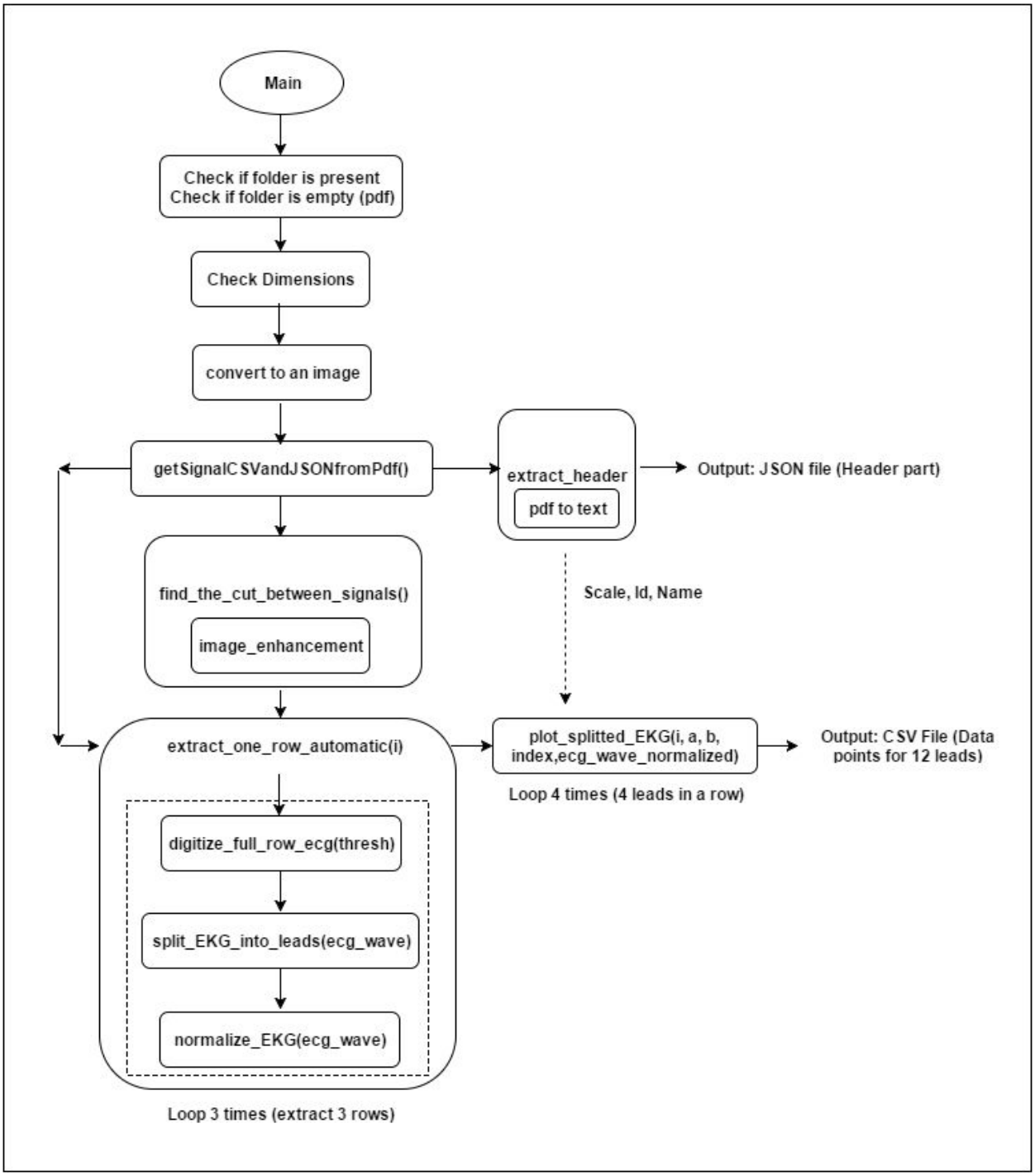

Fig. Process Flow

EKG Document

Functions

getSignalCSVandJSONfromPdf()

The Umbrella function to get the PDF data to be converted to JSON and CSV files.

extract_EKG_Header(input_file)

To convert the pdf file to a text file and then extract the contents and dump into a JSON file.

find_the_cut_between_signals()

To check if the signal is overlapping and find the boundary to split the signals.

extract_one_row_automatic(i)

Extract the signal content from the segment of each row. The image enhancement, digitizing the signal, obtaining the 4 leads and normalizing the signal with respect to the zero axis.

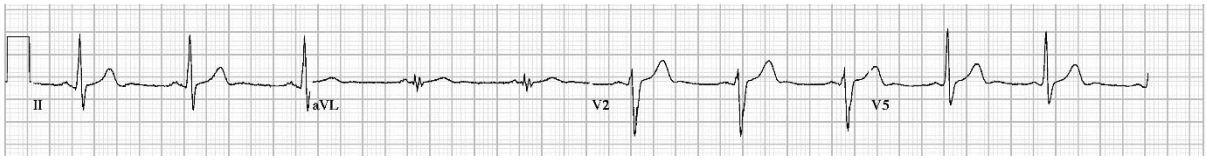

Fig. Output – Extracted one row

image_enhancement(img1)

Mainly to remove the grid part from the background of the image. This can be done by thresholding/Adaptive thresholding.

Fig. Output – Enhanced Image

plot_splitted_EKG(i, a, b, index, ecg_wave_normalized)

The function to display the extracted waves and to dump the file to the data frame which will be converted to the final CSV containing the data of all leads.

digitize_full_row_ecg(thresh)

EKG Document

Extract the signal content of wave by counting the pixels of the continuous wave.

split_EKG_into_leads(ecg_wave)

Get the 4 leads output by identifying the gap in the continuous wave.

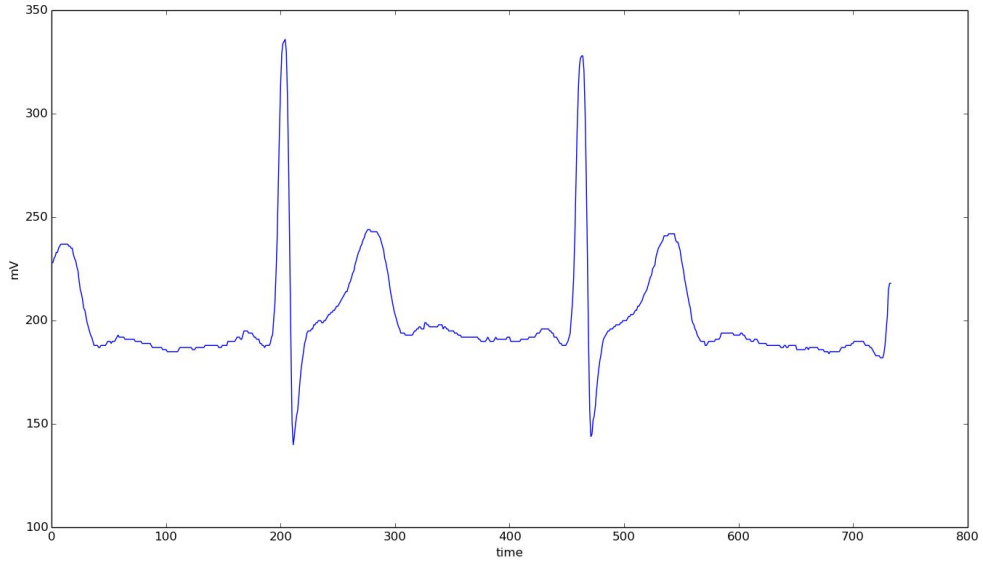

Fig. Output – Digitized and Split

normalize_EKG(ecg_wave)

Normalize the wave to the zero by comparing to the start of the calibration peak.

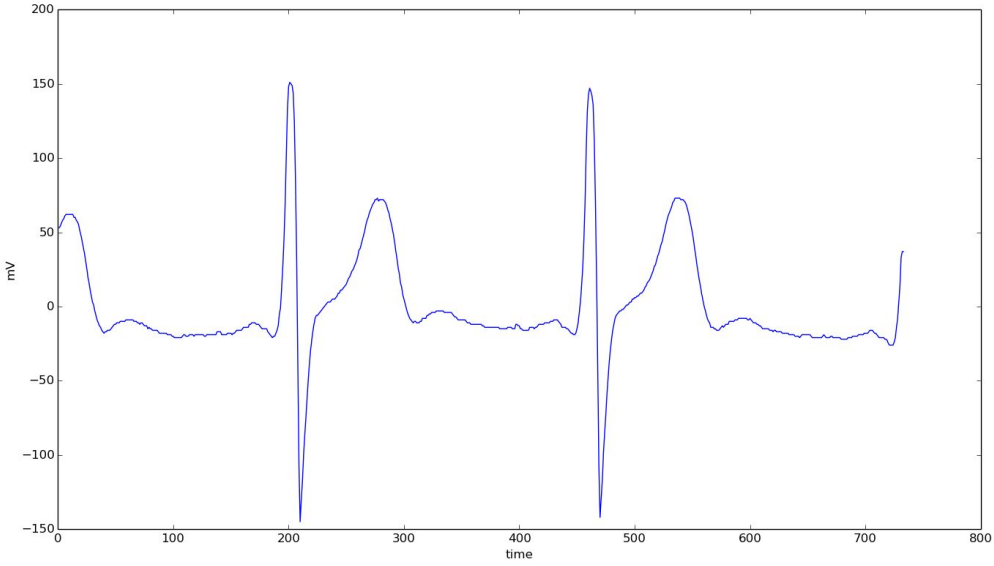

Fig. Output - Normalized

EKG Document

Chapter 3 – Test Cases and Validation

Output directory exists

a) **Validation -** if (os.path.isdir(output_folder))
b) **Checks** for the existence of input directory
c) **Error –** "NO such directory ! Please check your Input path/ Input folder"

Input directory exists

a) **Validation -** if (os.path.isdir(input_folder))
b) **Checks** for the existence of output directory
c) **Error –** "NO such directory ! Please check your Output path/ Output folder"

Folder is empty

a) **Validation** - if len(onlyfiles) > 0
b) **Checks** if the input folder is empty (proceed if not empty)
c) **Error** - "Input directory Empty ! Please check your input folder"

Dimensions of pdf

a) **Validation** - if PDF_DIM_X == pdf_x_length and PDF_DIM_Y == pdf_y_length and Input_as_pdf.getNumPages()==1
b) **Checks** expected dimensions of pdf
c) **Error** - "PDF dimension mismatch.The dimension of %s is [%d X %d] instead of [%d X %d]."

Dimensions of image

a) **Validation** - if img.shape[0] == EXPECTED_Y_LENGTH and img.shape[1] == EXPECTED_X_LENGTH global df_merged df_merged = pd.DataFrame(np.nan, index=list(range(0, 0)), columns=[])
b) **Checks** expected dimensions of image

Overlapping

a) **Checks** lead-text and lead-lead overlapping
b) **Print** - "Over lap ....."