

codECG: script Python per codifica da file pdf
proveniente da Kardia

Mauro Nicolì

Febbraio 2019

Indice

1	Introduzione	3
2	Implementazione	3
2.1	Scelte implementative	3
2.2	Casi d'uso	3
2.2.1	Codifica da file pdf	3
2.2.2	Decodifica da file ecg	4
3	Guida utente	4
3.1	Preparazione ambiente	4
3.2	Installazione librerie	5
3.3	Esecuzione	5
4	Sviluppi futuri	5

1 Introduzione

codECG è l'applicazione creata per ottenere una codifica della traccia elettrocardiografica proveniente dal sistema Kardia.

Sono previsti due casi d'uso:

- la codifica da file pdf (necessaria per poter eseguire la successiva)
- la decodifica da file codificato (.ecg)

All'esecuzione dello script, all'utente viene chiesto tramite una finestra di dialogo, di scegliere un file. Per semplicità avviene un unico controllo sull'estensione del file, e non sul nome del file. È proprio il controllo sull'estensione del file, che l'utente sceglie, il fattore discriminante dell'attivarsi di uno dei casi d'uso.

È stato scelto di realizzare la fase di decodifica per comprovare il risultato di codifica.

2 Implementazione

2.1 Scelte implementative

Per la realizzazione del software **codECG** si è scelto di lavorare con le librerie:

- **Tkinter, tkFileDialog**: standard GUI (Graphical User Interface) package
- **numpy**: per definire array multidimensionali e veloci funzioni matematiche associate che operano su di essi
- **Pillow PythonImagingLibrary**: per l'elaborazione di immagini
- **pdf2image**: per la conversione del file pdf in immagine
- **os**: che offre l'interfaccia con il sistema operativo
- **ast** Abstract Syntax Trees: modulo per processare la sintassi in Python

2.2 Casi d'uso

Nella sezione seguente viene illustrato il funzionamento delle due componenti dell'applicazione.

2.2.1 Codifica da file pdf

Scegliendo un file pdf, output di Kardia, il sistema produce il file "**codified.ecg.ecg**" contenente una stringa composta da array rappresentanti punti. I punti in questione sono frutto delle fasi di trasformazione del file pdf in immagine.

L'immagine viene prima ritagliata, ottenendo la sola superficie d'interesse ovvero l'elettrocardiogramma; in seguito viene eseguito il **load** per ottenere una

mappa di punti, sulla quale viene effettuato, per ogni pixel, il controllo sul suo valore.

Vengono salvate le coordinate dei soli punti che costituiscono il tracciato dell'ecg, all'interno di un array.

2.2.2 Decodifica da file ecg

Scegliendo un file ecg, precedentemente codificato tramite [2.2.1] *Codifica da file pdf*, il sistema produce il file "ecg_from_kardia.png".

L'immagine risultante è ricavata estraendo l'array codificato in "codified_ecg.ecg" e creando una nuova immagine bianca con dimensioni pari ai massimi valori presenti nell'array; a questa vengono aggiunti i pixel neri tramite le coordinate dei punti presenti nello stesso array.

Nel posizionamento dei pixel neri va considerato il sistema di riferimento delle coordinate (x,y) invertito, perché in input utilizza un array creato tramite la libreria numpy che utilizza un sistema di riferimento (y,x) .

3 Guida utente

È stata utilizzata la versione *python3* per la realizzazione di **codECG**. Se si dispone già di *python3* e *pip* procedere direttamente con la sezione [3.2].

3.1 Preparazione ambiente

Per accertarsi che sulla propria macchina sia presente la versione *python3* aprire un nuovo terminale e lanciare il seguente comando:

```
python3 --version
```

verrà così mostrata la versione corrente di *python3*.

Se non si dispone di alcuna versione potete scaricarla qui:

<https://www.python.org/downloads/>

NB: Se installate Python cliccate la spunta per creare automaticamente una variabile d'ambiente affinché, in ogni punto del vostro pc, possiate eseguire i comandi Python.

Occorre inoltre il tool pip: Python Installing Packages per poter installare le librerie utilizzate.

Per accertarsi che sulla propria macchina sia presente *pip* lanciare il seguente comando:

```
pip --version
```

verrà così mostrata la versione corrente di *pip*.

Solitamente pip viene scaricato automaticamente quando installate Python.

Se non si dispone di pip è reperibile al link di seguito:

<https://pip.pypa.io/en/stable/installing/>

3.2 Installazione librerie

A questo punto procedere con l'installazione delle librerie, eseguendo nel terminale i seguenti comandi:

- **numpy:** `pip3 install numpy`
- **Pillow:** `pip3 install pillow`
- **pdf2image:** `pip3 install pdf2image`

NOTA: solo per utenti che utilizzano il sistema operativo Windows

Occorre installare un modulo aggiuntivo, **Poppler**, che è una libreria per la visualizzazione di documenti PDF, nativa sui sistemi Unix.

<http://blog.alivate.com.au/poppler-windows/>

Scaricare la versione più recente, una volta decompresso lo zip, e posizionata la cartella nella sua destinazione finale, aprirla e copiare il path della cartella bin. (esempio : `C:\User\...\poppler\bin`)

Procedete con i seguenti passi per creare una variabile d'ambiente:

- In Cerca cercate e selezionate: Sistema (Pannello di controllo)
- Fate clic sul collegamento Impostazioni di sistema avanzate.
- Fate clic su Variabili di ambiente. Nella sezione Variabili di sistema, trovate la variabile di ambiente PATH e selezionatela. Fate clic su Modifica. Se la variabile di ambiente PATH non esiste, fate clic su Nuovo.
- Nella finestra Modifica variabile di sistema o Nuova variabile di sistema specificare il valore della variabile di ambiente PATH. (esempio : `C:\User\...\poppler\bin`) Fate clic su OK. Chiudere tutte le altre finestre facendo clic su OK.

3.3 Esecuzione

Lanciare l'esecuzione dello script:

```
python3 codECG.py
```

Verrà mostrata una finestra di dialogo: scegliere un file pdf output di Kardia per ottenere, nella stessa posizione del file pdf, un file con estensione ecg.

Eseguendo nuovamente:

```
python3 codECG.py
```

si può scegliere quindi un file di tipo ecg: il sistema produrrà nella stessa posizione un nuovo file png contenente l'elettrocardiogramma originale.

4 Sviluppi futuri

- Inserire **codEcg** all'interno di RESP: Registro Elettronico Sanitario Personale.

- Rendere dinamica l'area di cropping dell'immagine ricavata dal pdf, nel caso in cui il file pdf, output di Kardia, cambi dimensionamento, tramite riconoscimento dei pixel della superficie di interesse.