

UNIVERZITET U BEOGRADU  
MATEMATIČKI FAKULTET



Stefan Kapunac

# **DIGITALIZACIJA EKG DIJAGRAMA**

master rad

Beograd, 2020.

**Mentor:**

dr Nenad MITIĆ, redovni profesor  
Univerzitet u Beogradu, Matematički fakultet

**Članovi komisije:**

dr Gordana PAVLOVIĆ-LAŽETIĆ, redovni profesor  
Univerzitet u Beogradu, Matematički fakultet

dr Saša MALKOV, vanredni profesor  
Univerzitet u Beogradu, Matematički fakultet

**Datum odbrane:** \_\_\_\_\_

*Zahvaljujem se mentoru prof. dr Nenadu Mitiću, članovima komisije prof. dr Gordani Pavlović-Lažetić i prof. dr Saši Malkovu na izuzetno korisnim komentarima i savetima, kao i kolegi Aleksandru Veljkoviću na ustupanju EKG dijagrama i pomoći na početku izrade ovog rada.*

**Naslov master rada:** Digitalizacija EKG dijagrama

**Rezime:** Elektrokardiografija je metoda za snimanje električne aktivnosti srca. Elektrokardiogram (EKG) je grafički prikaz procesa elektrokardiografije, koji je izuzetno važno sredstvo za otkrivanje različitih oboljenja i procesa u organizmu. Da bi se omogućilo razvijanje modela ponašanja i predviđanja rada srčanog mišića, neophodno je da EKG dijagrami budu u digitalnom formatu. Cilj ovog rada je implementacija programa koji omogućava pretvaranje analognog EKG dijagrama (odštampanog na papiru) u digitalni zapis.

**Ključne reči:** EKG dijagram, digitalizacija

# Sadržaj

<b>1</b>	<b>Uvod</b>	<b>1</b>
<b>2</b>	<b>Osnove EKG-a</b>	<b>2</b>
<b>3</b>	<b>Pretprocesiranje</b>	<b>8</b>
3.1	Skeniran papir . . . . .	8
3.2	Paleta sivih boja . . . . .	8
3.3	Rotacija . . . . .	9
3.4	Nalaženje kvadrata . . . . .	11
3.5	Izbacivanje mreže sa slike . . . . .	12
3.6	Prepoznavanje karaktera . . . . .	14
3.7	Binarizacija . . . . .	16
<b>4</b>	<b>Ekstrakcija signala</b>	<b>19</b>
4.1	Lokacija kanala . . . . .	19
4.2	Ekstrakcija sirovih signala . . . . .	21
4.3	Podela na kanale . . . . .	23
4.4	Pretvaranje piksela u sekunde i volte . . . . .	24
<b>5</b>	<b>Grafički korisnički interfejs</b>	<b>27</b>
<b>6</b>	<b>Zaključak</b>	<b>31</b>
	<b>Bibliografija</b>	<b>32</b>

# Glava 1

## Uvod

S obzirom na važnost informacija koje se nalaze na EKG dijagramu, bilo bi dobro kada bi na osnovu njih mogli da se naprave modeli ponašanja i predviđanja rada srčanog mišića. Da bi se tako nešto postiglo, neophodna je velika količina ulaznih podataka u digitalnom formatu. Međutim, još uvek najveći broj EKG aparata daje izlaz samo u štampanom obliku na papiru. Upravo zbog toga potrebno je imati program koji će digitalizovati skenirani odštampani EKG dijagram.

S obzirom na to da se koristi termalni štampač, postojanost dobijene štampe nije velika. Tačnije, dolazi do relativno brzog gubljenja intenziteta boja. To je, samo po sebi, još jedan od razloga zašto je digitalizacija EKG dijagrama neophodna.

Postoje noviji EKG aparati koji prikazuju rezultat u PDF-u. Pretvaranje ovakve vrste dokumenta u vremensku seriju, to jest digitalni format koji je pogodan za dalju obradu, bilo bi olakšano, jer nema gubitaka u kvalitetu do kojih dolazi, što vremenom, što u procesu skeniranja.

Za implementaciju osmišljenog algoritma za digitalizaciju EKG dijagrama korišćen je programski jezik Python [18]. U toku rada su korišćene i sledeće biblioteke:

- za obradu slika - OpenCV [3],
- za rad sa višedimenzionim nizovima i numerička izračunavanja - NumPy [11],
- za iscrtavanje slika i grafika - Matplotlib [10].

## Glava 2

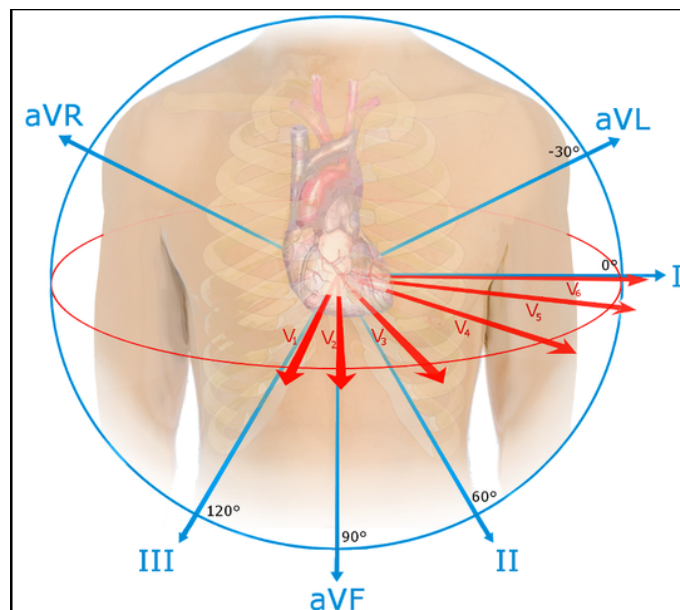
# Osnove EKG-a

Elektrokardiografija je proces snimanja električne aktivnosti srca koji se realizuje postavljanjem elektroda na određena mesta na koži. U pitanju je jeftina, a vrlo korisna metoda za dobijanje informacija o strukturi i funkciji srca, te se masovno koristi u medicini. Postoji već više od jednog veka i za to vreme postala je nezamenljivo sredstvo u dijagnostici različitih oboljenja među kojima su i infarkt miokarda (srčani udar), plućna embolija, sinusna aritmija (poremećaj srčanog ritma), sinusna bradikardija (usporen rad srca), sinusna tahikardija (ubrzan rad srca), kao i akutni perikarditis (upala srčane maramice). Takođe, koristi se i kao pomoćno sredstvo pri analizi rada pejsmejкера, provera pred hirurške zahvate, i praćenje stanja srčanog mišića pacijenata koji koriste lekove za srce.

Elektrokardiogram (skraćeno EKG) je grafički prikaz elektrokardiografije. Elektrode otkrivaju male promene elektriciteta na koži koje su posledica depolarizacije i repolarizacije srčanog mišića prilikom svakog srčanog ciklusa, odnosno kontrakcije i opuštanja prilikom svakog otkucaja srca. Kod uobičajenog dvanaestokanalnog EKG-a raspoređuje se 10 elektroda na grudni koš i ekstremitete pacijenta. Svaki od 12 kanala predstavlja 12 različitih uglova iz kojih se meri električni potencijal srca. Na taj način svaki kanal odgovara različitim anatomskim delovima srca. Vrednosti napona se beleže tokom određenog vremenskog perioda, koji najčešće iznosi 10 sekundi.

Kanal formiraju parovi elektroda koji mere razliku električnog potencijala između dva mesta na kojima su postavljene odgovarajuće elektrode. Od šest elektroda koje su postavljene na grudni koš pacijenta dobija se šest kanala označenih sa V1, V2, V3, V4, V5 i V6. Oni predstavljaju „pogled” na srce u horizontalnoj ravni. Četiri elektrode koje se nalaze na ekstremitetima pacijenta se koriste za dobijanje

šest preostalih kanala. Kanali I, II i III se direktno dobijaju kao razlika električnog potencijala između različitih elektroda na ekstremitetima pacijenta. Dodatni kanali aVR, aVL, aVF se izvode kao kombinacija više elektroda. Ovih šest kanala predstavljaju „pogled” na srce u vertikalnoj ravni. Ilustracija različitih uglova iz kojih kanali mere električnu aktivnost srca data je na slici 2.1.



Slika 2.1: 12 kanala

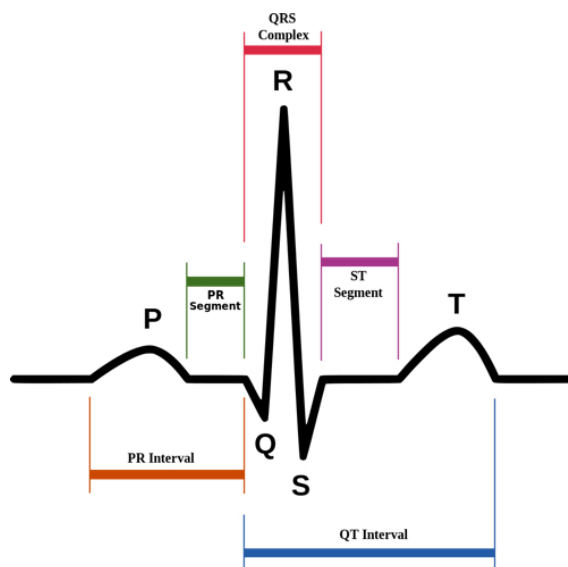
EKG zdrave osobe se sastoji iz sledećih komponenti - P talas, QRS kompleks i T talas. Označene komponente su prikazane na slici 2.2.

Nekada se dodaje i U talas kao komponenta koja se nalazi posle T talasa, ali da bi se on video na dijagramu, obično su neophodne dodatne elektrode. P talas predstavlja depolarizaciju pretkomora i kod zdravih osoba prethodi QRS kompleksu. QRS kompleks označava depolarizaciju komora. Sastoji se od tri međusobno bliska talasa - Q, R i S. T talas predstavlja repolarizaciju komora i nalazi se posle QRS kompleksa.

Pri analizi se dodatno koriste i podaci kao što su: PR interval, ST segment, QT interval i RR interval.

- PR interval predstavlja vreme koje prođe kretanjem električne aktivnosti između pretkomora i komora. Počinje na početku P talasa, a završava se na početku Q talasa.





Slika 2.2: Delovi EKG-a

- ST segment predstavlja vreme između depolarizacije i repolarizacije komora. Počinje na kraju S talasa, a završava se na početku T talasa.
- QT interval predstavlja trajanje depolarizacije i repolarizacije komora. Počinje na početku QRS kompleksa, a završava se na kraju T talasa.
- RR interval predstavlja vreme između dva QRS kompleksa. Počinje na piku jednog R talasa, a završava se na piku drugog R talasa [16].

EKG-ovi se obično štampaju na mreži sačinjenoj od kvadrata čija je dužina stranice 5 milimetara. Pritom je svaki od tih kvadrata tanjim linijama podeljen na 5 manjih kvadrata čija je dužina stranice 1 milimetar. Na grafiku x osa predstavlja vreme, a y osa napon. Na standardnom EKG dijagramu, koji je štampan brzinom od 25 milimetara po sekundi, sa kalibracijom voltaže od 10 milimetara po milivoltu, može se izračunati da svaki manji kvadrat veličine  $1mm \times 1mm$  predstavlja  $0.1mV \times 0.04s$ . Slično, veliki kvadrat veličine  $5mm \times 5mm$  predstavlja  $0.5mV \times 0.2s$ . Međutim, nekad se koriste i druge brzine štampanja i drugačija kalibracija voltaže, što može dovesti do problema pri digitalizaciji, a detaljnije je opisano u sekciji o prepoznavanju karaktera 3.6.

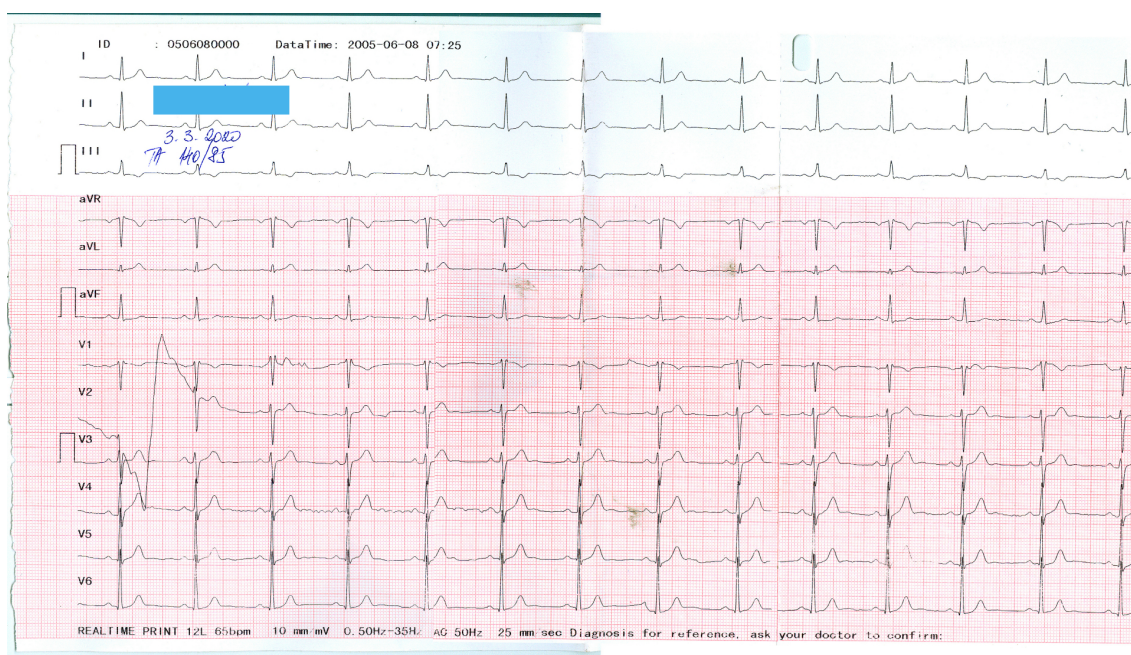
Postoje i različiti načini raspoređivanja 12 kanala na EKG dijagramu. Ako se EKG dijagram posmatra kao tabela sačinjena od redova i kolona, koje odgovaraju granicama između kanala, dimenzije takve tabele mogu biti:

## GLAVA 2. OSNOVE EKG-A

---

- dvanaest redova i jedna kolona, kao na slici 2.3 <sup>1</sup>,
- šest redova i dve kolone, kao na slici 2.4,
- tri reda i četiri kolone. kao na slici 2.5.

U svakom od ovih rasporeda, idući od gore na dole i s leva na desno, dolazi se do istog redosleda kanala - I, II, III, aVR, aVL, aVF, V1, V2, V3, V4, V5, V6.



Slika 2.3: EKG dijagram sa 12 redova i 1 kolonom

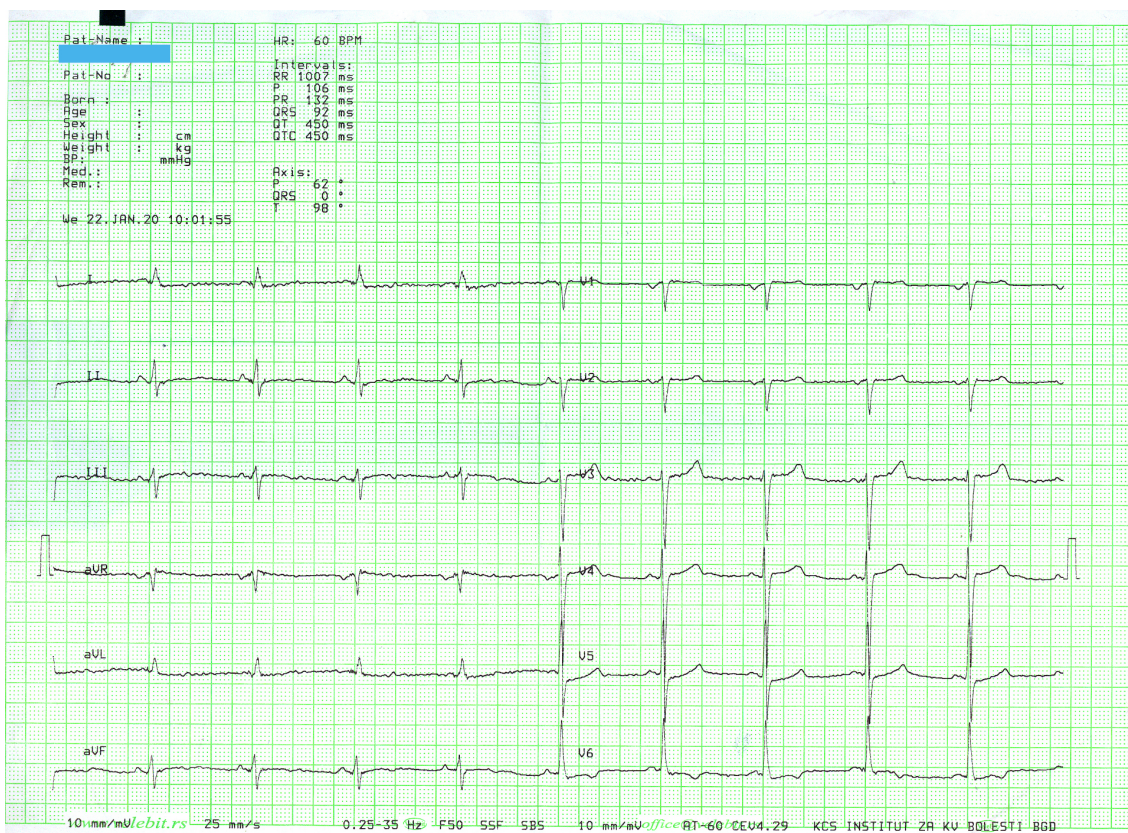
Već sa prethodna tri dijagrama može se zaključiti da postoji (bar) još jedna stvar koja nije standardizovana među EKG dijagramima, a to je boja mreže. Boja, u zavisnosti od aparata koji se koristi, može biti:

- crvena, kao na slici 2.3,
- zelena, kao na slici 2.4,
- siva, kao na slici 2.6.

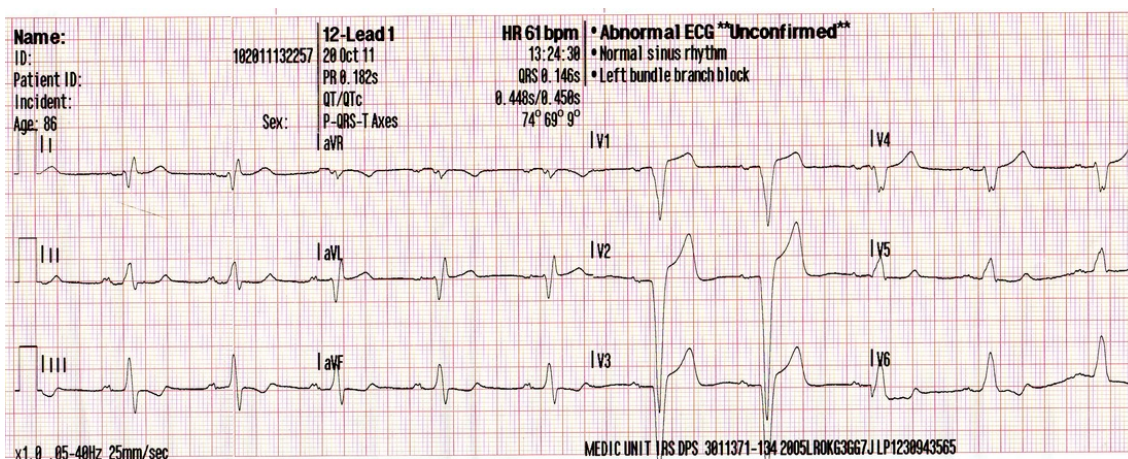
---

<sup>1</sup>Na ovoj, i na svim narednim slikama, plavim pravougaonikom su prekriveni lični podaci pacijenta.

GLAVA 2. OSNOVE EKG-A



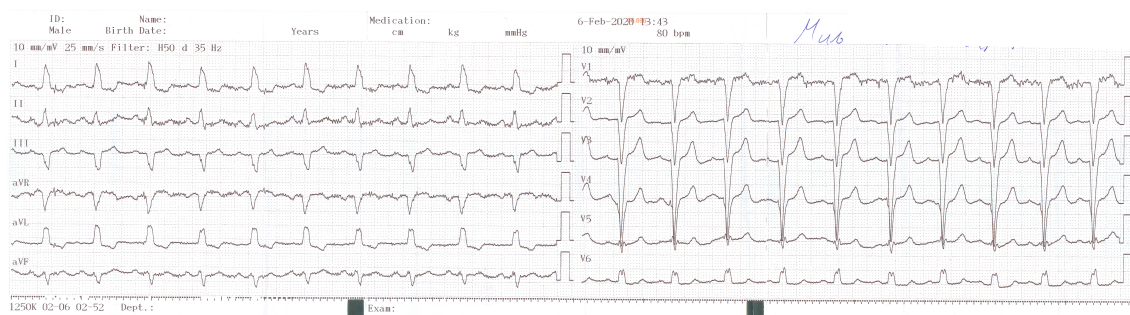
Slika 2.4: EKG dijagram sa 6 redova i 2 kolone



Slika 2.5: EKG dijagram sa 3 reda i 4 kolone

## GLAVA 2. OSNOVE EKG-A

---



Slika 2.6: EKG dijagram sa sivom bojom mreže

# Glava 3

## Pretprocesiranje

U ovom pogavlju biće dat opis svakog od pojedinačnih koraka algoritma digitalizacije koji se odnosi na pretprocesiranje. Koraci se primenjuju u redosledu u kom su navedeni, ali je, na mestima gde je to bilo moguće, prikazivan rezultat na EKG dijagramu u boji, da bi se lepše video rezultat primene.

### 3.1 Skeniran papir

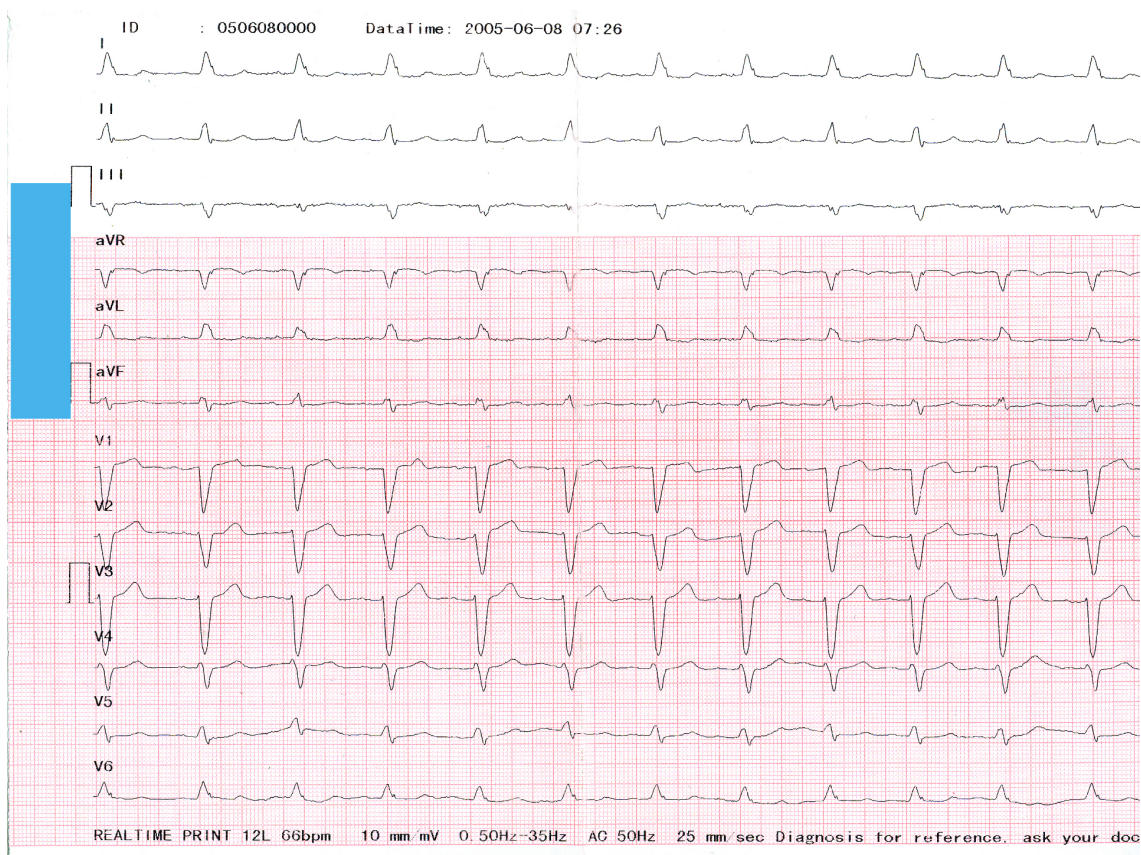
Ulaz algoritma predstavlja skeniran EKG dijagram. Podržani su uobičajeni formati datoteka kao što su JPG, PNG, TIFF. S obzirom na to da kvalitet dobijenih rezultata, naravno, zavisi od rezolucije slike, poželjno je da rezolucija bude što veća. Ipak, ne treba ni preterivati kako ne bi došlo do predugog trajanja izvršavanja.

Skenirani EKG dijagram koji će biti korišćen u nastavku za ilustraciju svakog od koraka dat je na slici 3.1.

### 3.2 Paleta sivih boja

Na papiru je najčešće signal crne, a mreža crvene boje. Međutim, mreža može biti i zelena ili siva. Takođe, postoje stariji skeneri koji nemaju opciju skeniranja u boji. Stoga se učitana slika pretvara u format palete sivih boja (engl. grayscale). Ovom transformacijom se smanjuje prostor koji slika zauzima u memoriji računara - prelazi se sa tri kanala (R, G i B) na jedan.

Cilj ove transformacije je što opštiji algoritam koji bi trebalo da radi sa različitim ulazima. Ukoliko bi postojala jednoobraznost među EKG dijagramima barem što se tiče boje mreže, tako da je ona uvek crvena (ili bilo koje druge fiksirane boje



Slika 3.1: Skenirani EKG dijagram

različite od crne), ova transformacija ne bi imala smisla. Mogao bi se konstruisati specijalizovan program koji bi na jednostavniji način mogao da izdvoji pozadinu sa slike, jer je unapred poznata boja mreže.

Primer EKG dijagrama pretvorenog u format palete sivih boja dat je na slici 3.2.

### 3.3 Rotacija

Prilikom skeniranja gotovo neizbežno dolazi do grešaka - dobijena slika je pod određenim uglom, što se može videti i na slici 3.1. Uglavnom taj ugao nije veliki, ali da bi se dobili ispravni rezultati neophodno je izvršiti rotaciju slike.

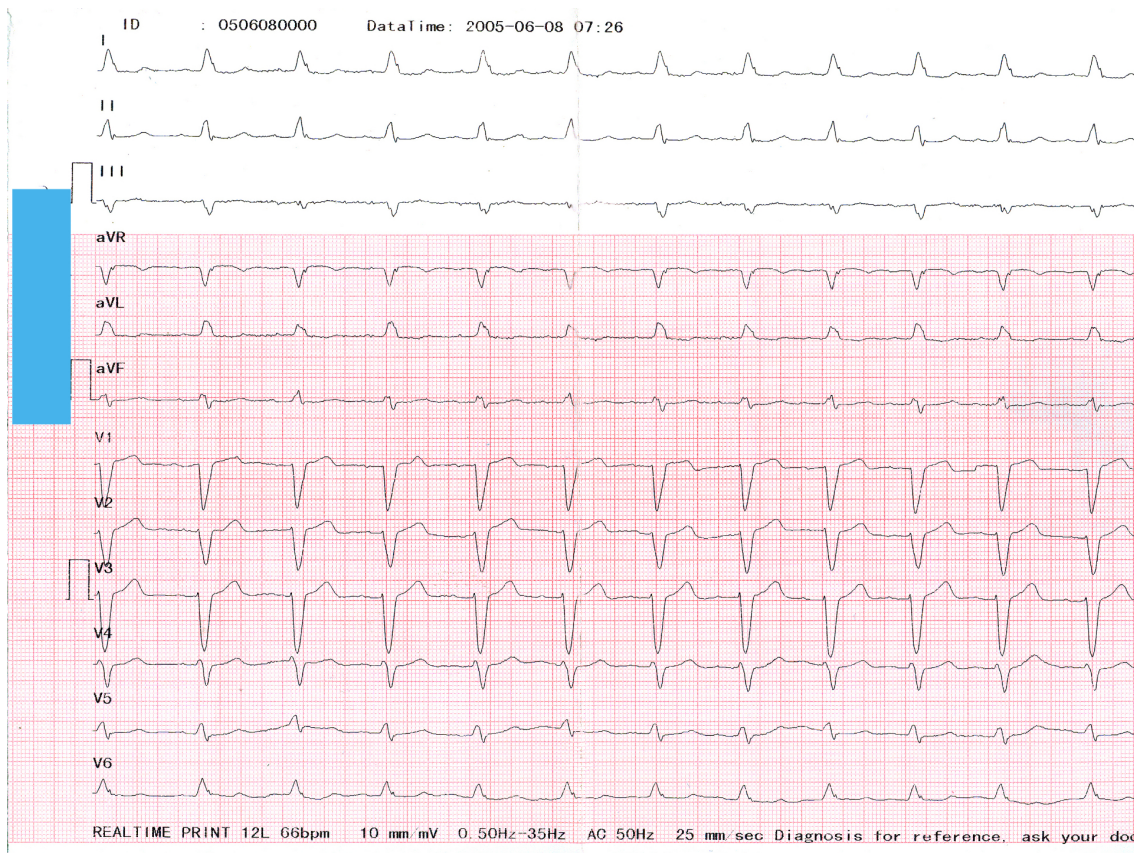
Za razliku od nekih predašnjih radova, kao što je [15], gde se od korisnika traži da obeleži dve tačke na jednoj liniji mreže da bi se odredio ugao, u ovom radu ugao se određuje automatski. Korišćenjem Hafove transformacije, nalaze se linije koje su deo



Slika 3.2: Paleta sivih boja

mreže i na osnovu njihovog nagiba se rotira celokupna slika. Hafova transformacija je metod za prepoznavanje krivih koji koristi dualnost između tačaka na krivoj i parametara te krive. Originalna Hafova transformacija je identifikovala prave na slici [9], a kasnije je proširena na identifikaciju proizvoljnih krivih [6].

S obzirom na to da OpenCV funkcija HoughLines prihvata samo binarne (crno-bele) slike, neophodno je prvo primeniti Kenijev algoritam za nalaženje ivica (engl. Canny edge detection) [4]. Funkciji HoughLines prosleđivan je u petlji argument za minimalnu dužinu pronađene linije. Na početku je njegova vrednost 90 posto širine slike, a ako ne postoje toliko dugačke linije, prelazi se u narednu iteraciju sa smanjenom vrednošću tog argumenta. Ovakav pristup ima za cilj pronalaženje linija koje su deo mreže, te linije su najduže i njih bi trebalo učiniti horizontalnim. Za svaku pronađenu liniju računa se ugao koji ona zaklapa sa x-osom i smešta u niz. Konačan ugao koji se koristi za rotaciju slike je modal (najčešća vrednost) tog niza, odnosno medijana ako modalna vrednost nije jedinstvena. Primer ispravljene slike ovakvim postupkom prikazan je na slici 3.3.



Slika 3.3: Rotirana slika

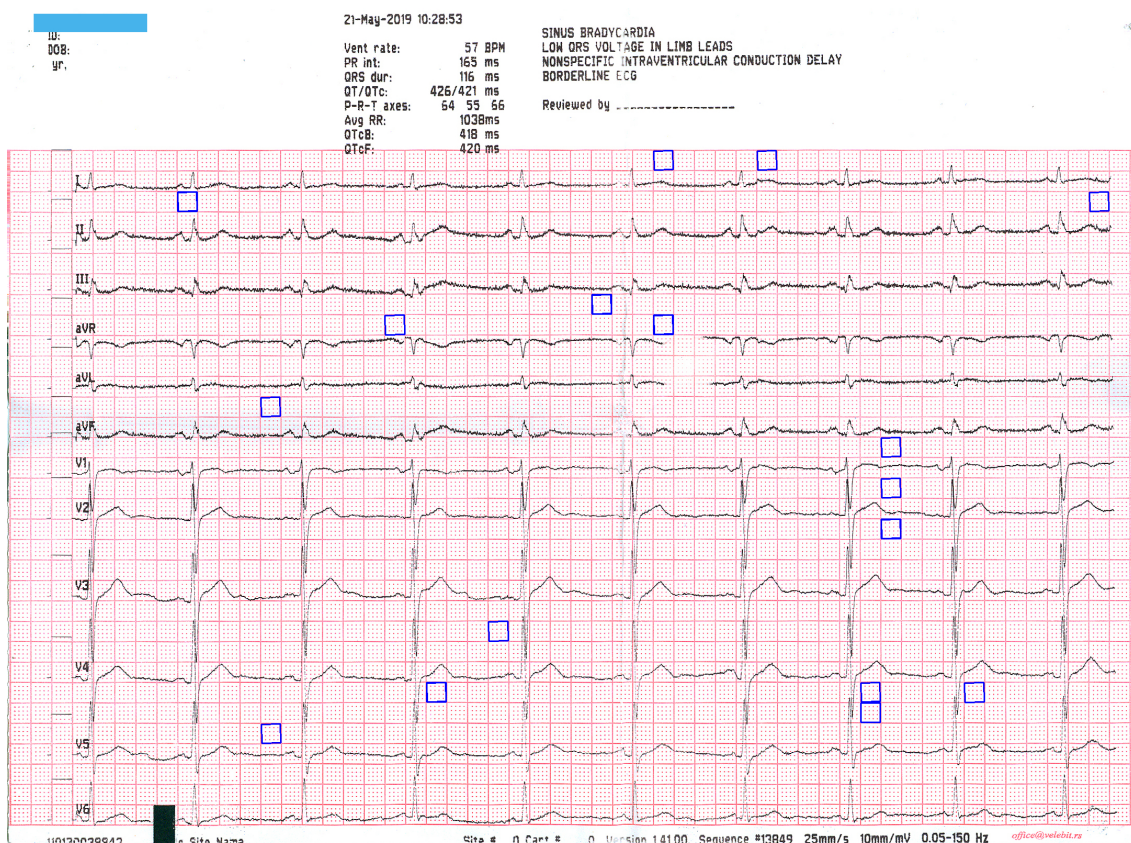
### 3.4 Nalaženje kvadrata

Kao što je već napomenuto, mreža je sačinjena od kvadrata čija je stranica veličine 5 milimetara. Međutim, na slici dimenzije kvadrata zavise od rezolucije slike. Krajnji cilj je dobijanje signala u obliku vremenske serije - parova vrednosti (vreme, voltaža). Svaki piksel koji je izabran kao deo signala treba pretvoriti u par (vreme, voltaža). Da bi to bilo moguće neophodno je izračunati dužinu stranice kvadrata u pikselima. Na osnovu te vrednosti lako može da se dobije veličina jednog piksela u milimetrima, koja će se dalje koristiti, uz podatke o brzini štampe (npr. 25 milimetara po sekundi) i naponu (npr. 10 milimetara po milivoltu), za određivanje konačnih rezultata.

Za samo pronalaženje kvadrata na slici korišćena je OpenCV funkcija `approxPolyDP` koja je zasnovana na Ramer-Daglas-Peker algoritmu. Ovaj algoritam za datu poliliniju (krivu sačinjenu od duži) pronalazi njoj sličnu krivu sa manje tačaka [14, 5]. Na konkretnom EKG dijagramu koji je korišćen kao primer u ovom radu,



ovakav metod nije dao dobre rezultate. Problem je u tome što mreža zapravo i nije sačinjena od povezanih crvenih linija, već se sastoji od izolovanih tačaka između kojih postoje beli pikseli. Na drugačijoj vrsti mreže, gde su petomilimetarski kvadrati uobičajeni kvadrati (bez belih piksela na stranicama kvadrata), ovaj metod se pokazao kao uspešan. Rezultat primene ovog algoritma na jednom takvom dijagramu prikazan je na slici 3.4.



Slika 3.4: Pronađeni kvadrati

## 3.5 Izbacivanje mreže sa slike

U prethodnim koracima mreža je iskorišćena za dobijanje korisnih informacija za digitalizaciju. Za sve naredne korake algoritma ona predstavlja šum, te se ovde izbacuje.

Izbacivanje je realizovano sledećim postupkom:

1. Originalna slika se prebacuje iz RGB (Red Green Blue) formata u HSV (Hue Saturation Value) format.
2. Konstruiše se maska invertovanjem V kanala.
3. Pozadina se izračunava tako što se maska oduzima od invertovane slike u formatu palete sivih boja.
4. Slika bez pozadine se dobija sabiranjem pozadine i slike u formatu palete sivih boja.

Primer izvršavanja je dat na slici 3.5.



Slika 3.5: Izbacivanje pozadine

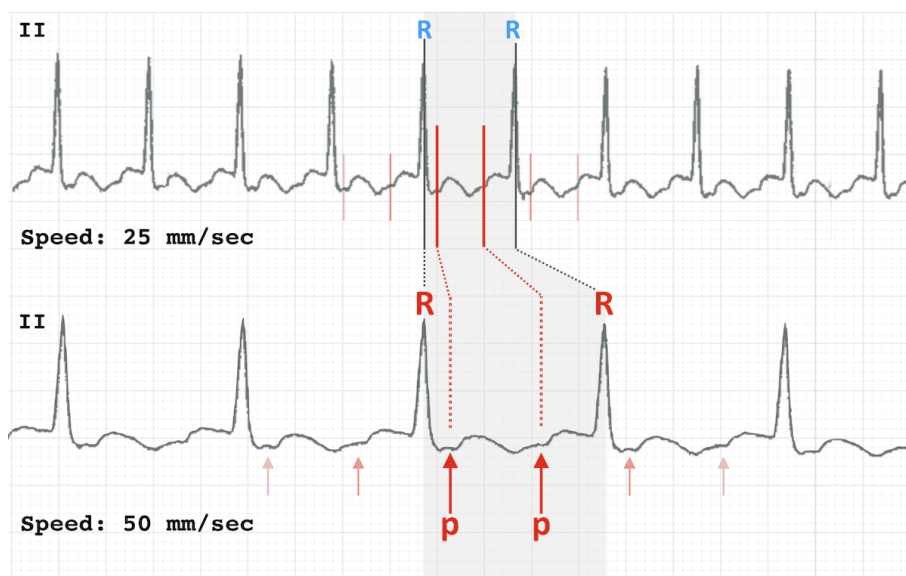
Razmatrana je i mogućnost izbacivanja pozadine korišćenjem binarizacije na osnovu praga (engl. thresholding). Pokušano je sa više različitih algoritama među kojima se najbolje pokazao Otsuov [13]. To je metod koji je zasnovan na minimizaciji međuklasne varijanse (gde je jedna klasa pozadina, a druga prednji deo slike). Za

različite EKG dijagrame ova metoda je davala rezultate različitog kvaliteta. Stoga je izabran prethodno opisan metod za otklanjanje pozadine, koji se bolje pokazao u opštem slučaju.

Ovaj deo algoritma se može proširiti, tako što se prvo pokuša sa izbacivanjem crvenih, zelenih i plavih piksela sa slike, jer su to boje koje uobičajeno predstavljaju mrežu ili beleške hemijskom olovkom. Za svaku od ovih boja se jednostavno može odrediti njen opseg u HSV formatu i svi pikseli sa slike koji su u tom opsegu se mogu postaviti na belo.

### 3.6 Prepoznavanje karaktera

EKG dijagrami se mogu štampati različitom brzinom. Standardno je brzina 25 milimetara po sekundi, ali može da bude i 50 ili 10 mm/s. Štampanje većom brzinom može da pomogne pri otkrivanju suptilnih promena koje pri manjoj brzini mogu da ostanu neprimećene [1, 7]. Odnos između EKG dijagrama štampanih brzinom 25 i 50 mm/s prikazan je na slici 3.6.



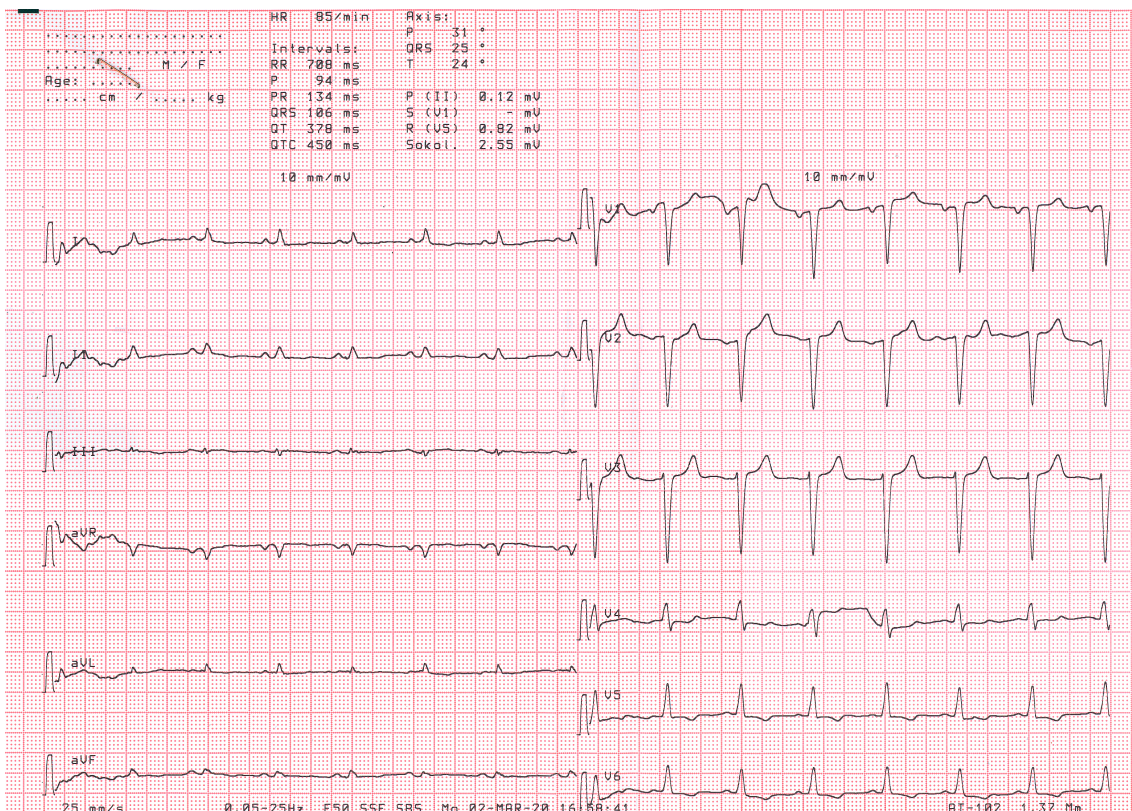
Slika 3.6: Različite brzine štampanja

Slično je i sa naponom. Standardna kalibracija je 10 milimetara po milivoltu, ali može da bude i 20 ili 5 mm/mV.

Takve razlike među EKG dijagramima prave problem za automatsku obradu, ali na većini dijagrama su odštampani ti dodatni podaci o brzini štampanja i naponu.

Zbog toga je korišćeno optičko prepoznavanje karaktera za dobijanje tih izuzetno važnih informacija sa slike.

Optičko prepoznavanje karaktera (engl. Optical Character Recognition, OCR) predstavlja prepoznavanje odštampanog ili rukom napisanog teksta sa slike. Ova metoda primenjuje se u različitim domenima među kojima je i digitalizacija skeniranih dokumenata. U nekim ranijim radovima koji se takođe bave digitalizacijom EKG dijagrama, kao što je [15], korišćen je OCR za prepoznavanje demografskih informacija o pacijentima (npr. ime, prezime, godine). Korišćenje optičkog prepoznavanja karaktera u ovom radu je, pre svega, posvećeno pronalazanju informacija o brzini štampanja i naponu. OCR može da se iskoristi i za prepoznavanje imena 12 kanala (I, II, III, aVR, aVL, aVF, V1, V2, V3, V4, V5, V6). Ova imena označavaju početak novog, odnosno kraj prethodnog kanala, te bi njihovo pronalazanje olakšalo proces ekstrakcije signala. Ipak, često se dešava preklapanje imena sa samim signalom, kao na slici 3.7. U takvim situacijama OCR znatno teže može da ih locira.



Slika 3.7: Preklapanje imena kanala sa signalom

Nakon pronalaska korisnih reči, slova predstavljaju šum za nastavak algoritma,

pa se izbacuju. Sva prepoznata slova se brišu sa slike, odnosno postavlja se na belo svaki piksel u okviru pravougaonika unutar koga su se nalazila slova. Primer izvršavanja ovog dela algoritma dat je na slici 3.8.



Slika 3.8: Izbacivanje slova

Korišćena je biblioteka pytesseract [8], koja predstavlja Python omotač za sistem za optičko prepoznavanje karaktera Tesseract [17]. Tesseract je alat koji je originalno 1980-ih razvijao HP (Hjult-Pakard). 2005. godine je postao projekat otvorenog koda, a od 2006. Gugl sponzoriše njegov razvoj.

### 3.7 Binarizacija

Sledeći korak u algoritmu je pretvaranje slike u crno-belu, odnosno binarizacija. U ovom slučaju se prethodno pomenuti Otsuov algoritam [13] pokazao kao adekvatno rešenje. Primer binarizovane slike dat je na slici 3.9.



Slika 3.9: Binarizovana slika

## Invertovanje slike

Poslednja stvar koju treba uraditi pre glavnog dela algoritma je invertovanje prethodno binarizovane slike. Ovim procesom crni pikseli postaju beli, a beli pikseli postaju crni. Sam po sebi, ovaj postupak ne donosi nikakvo poboljšanje, ali određene funkcije korišćene u nastavku (npr. morfološka dilatacija) očekuju da pozadina bude crna, a objekti na slici beli. Primer invertovane slike dat je na slici 3.10.



Slika 3.10: Invertovana slika

# Glava 4

## Ekstrakcija signala

U ovom poglavlju biće dat opis svakog od pojedinačnih koraka centralnog dela algoritma digitalizacije, odnosno ekstrakcije signala.

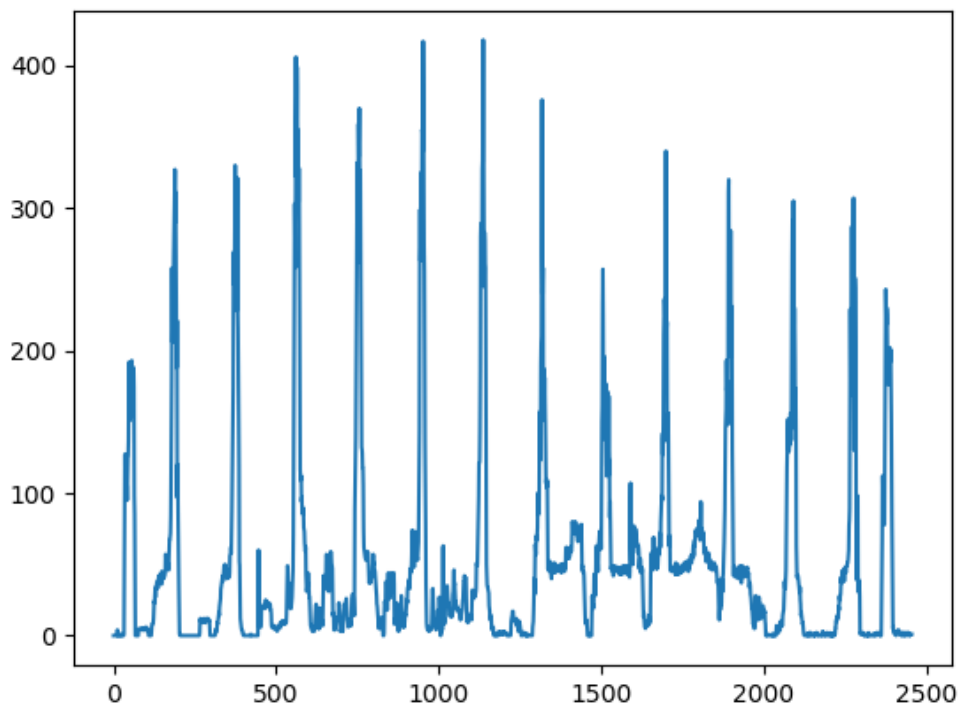
### 4.1 Lokacija kanala

Kao što je već prikazano na slikama u delu 2, postoje brojne varijante EKG dijagrama među kojima se i raspored 12 kanala razlikuje. Na nekim dijagramima je u 12 odvojenih redova prikazano 12 kanala - jedan kanal u svakom redu. Na drugima u 6 redova po 2 kanala. Na trećima u 3 reda po 4 kanala.

Da bi se na isti način mogle obrađivati sve navedene vrste EKG dijagrama, potrebno je na početku locirati kanale. Za ovaj problem osmišljeno je sledeće rešenje, a sličan pristup je korišćen i u [2]. Za svaki red piksela u slici izračunava se ukupan broj belih piksela (piksela koji su potencijalno deo signala). Trebalo bi da se jasno izdvajaju redovi u čijoj bliskoj okolini se nalazi signal, to jest neki kanal. Maksimum u okviru vrha bi trebalo da odgovara nuli signala, s obzirom na to da je u nuli EKG signal najravniji. Grafik broja belih piksela u odnosu na red prikazan je na slici 4.1.

Na grafiku se vidi da, zaista, postoje vrhovi na mestima koja odgovaraju redovima gde se nalazi signal. Ali, takođe se vidi da postoje i, manji, ali ipak problematično veliki vrhovi na mestima koja odgovaraju redovima gde se nalazi tekst koji nije prepoznat i izbačen sa slike prilikom pretprocesiranja. Da bi se ovaj problem prevazišao, za svaki red piksela računa se dužina najdužeg regiona crnih piksela. Taj broj predstavlja veličinu najveće praznine u tom redu piksela. Što je praznina veća, to je manja verovatnoća da se u tom redu nalazi nula signala. Stoga se ukupan broj





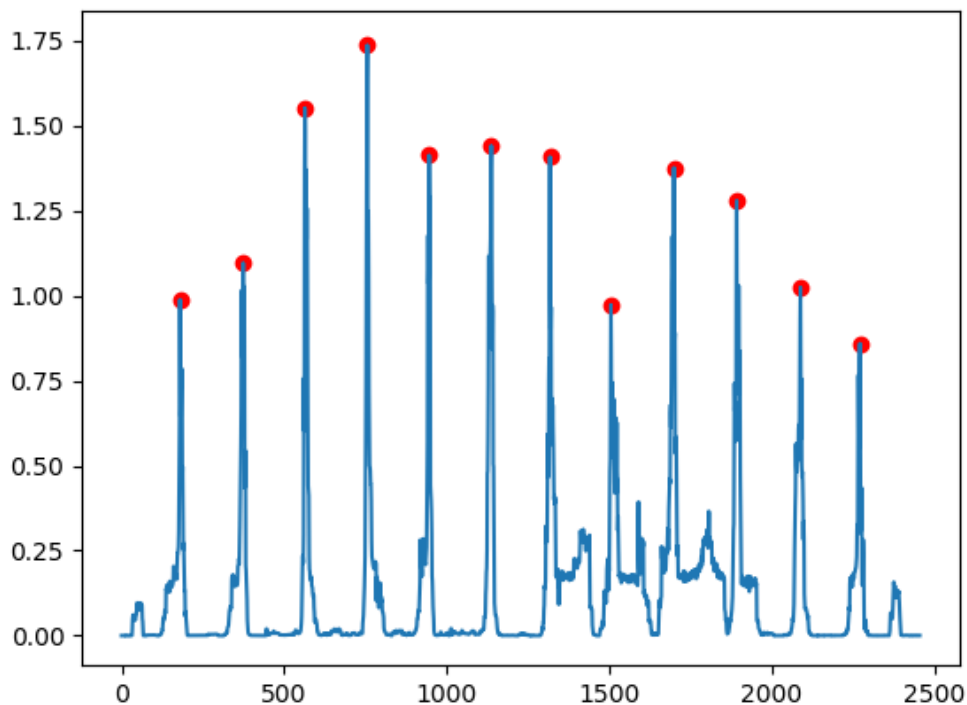
Slika 4.1: Broj belih piksela u redu

belih piksela u redu deli dužinom najduže praznine u redu. Grafik ovih vrednosti u odnosu na red prikazan je na slici 4.2.

Na ovom grafiku je skala drugačija, ali to i nije bitno, bitan je međusobni odnos. Vidi se da su smanjeni vrhovi gde su bila slova, a 12 vrhova koji odgovaraju kanalima je postalo izraženije.

Sami vrhovi se određuju tako da ispunjavaju sledeće uslove:

- Visina je veća od 30 posto maksimalne visine.
- Prethodni pik je dovoljno daleko (5 posto duzine slike). Signal je deblji od jednog piksela, pa bi se bez ovog uslova biralo više pikova nego sto ih zapravo ima.
- Trenutna visina je najveća u nekoj manjoj okolini (1 posto duzine slike). Bez ovog uslova bi se birali početni delovi „brda” koji su dovoljno visoki, a ne sam vrh



Slika 4.2: Skalirani broj belih piksela u redu

Pošto su određene y koordinate početaka svakog od kanala, potrebno je odrediti i vrednosti za x koordinate. Ideja je da se za svaku nulu signala pronađe prvi beli piksel idući sa leva na desno. Pritom se pretraga vrši u maloj okolini (1 posto visine slike) reda piksela u kome se nalazi nula signala. Takođe, ne kreće se baš od prve kolone, zato što je moguće da postoji okvir koji bi predstavljao šum. Rezultat izvršavanja je prikazan na slici 4.3.

Za konačnu x koordinatu početka bira se medijana, kao robusniji statistički podatak u odnosu na prosek, dobijenog niza vrednosti x koordinata. Na isti način se može odrediti i kraj signala, pretragom s desna na levo.

## 4.2 Ekstrakcija sirovih signala

Ovaj korak u algoritmu ima za cilj pronalaženje piksela na slici koji su deo EKG signala. Vrednosti piksela predstavljaju sirov signal koji kasnije, korišćenjem pret-



Slika 4.3: Počeci signala

hodno izračunatih parametara vezanih za veličinu kvadrata u mreži, brzinu štampe i napon, treba pretvoriti u vremensku seriju, odnosno parove vrednosti (vreme, voltaža).

Za implementaciju je korišćena OpenCV funkcija `connectedComponents` za pronalaženje grupa povezanih belih piksela na slici. Piksela A se smatra povezanim sa pikselom B ukoliko je A iste boje kao B i A je jedan od osam susednih piksela piksela B. Korišćena funkcija svim pikselima iste grupe dodeljuje istovetnu oznaku. Primenom ove funkcije, u idealnom slučaju, za EKG dijagram koji ima 12 pojedinačnih redova za 12 kanala dobilo bi se tačno 12 povezanih komponenti koje u potpunosti odgovaraju signalu. Na žalost, u realnim okolnostima, usled različitih problema (izbledelost ili oštećenja na samom EKG dijagramu, spajanje dvaju pikova iz različitih kanala...) primena samo ove funkcije neće dati dovoljno dobre rezultate. Može da se desi da se pronađe više (ako u okviru signala postoje prekidi), a i manje (ako se vrhovi susednih kanala dodiruju) od 12 povezanih komponenti. Problem takođe mogu da predstavljaju i slova koja, iz različitih razloga, ostanu neprepoznata, a

samim tim i neuklonjena.

X i y koordinate početaka signala su određene u prethodnom koraku algoritma 4.1. Od njih se počinje, a zatim se u svakoj sledećoj koloni, za svaki kanal, bira po jedan piksel koji je deo signala. Odabir se vrši na osnovu oznake komponente koja je funkcijom `connectedComponents` pridružena svim pikselima. Dakle, ako u trenutnoj koloni postoji piksel koji pripada istoj komponenti kao i odabrani piksel iz prethodne kolone, on biva odabran. A ako takav piksel ne postoji (što označava kraj povezane komponente, odnosno prekid u signalu), traži se piksel koji je najbliži odabranom pikselu iz prethodne kolone. Pritom se pretraga u trenutnoj koloni ograničava na neposrednu blizinu odabranog piksela iz prethodne kolone. Dakle, može da se desi da u koloni ne bude odabran nijedan piksel, ali pretraga se nastavlja u sledećim kolonama i ako se u nekoj od njih pronađe adekvatan piksel, sve prethodno prazne kolone se popunjavaju linearnom interpolacijom.

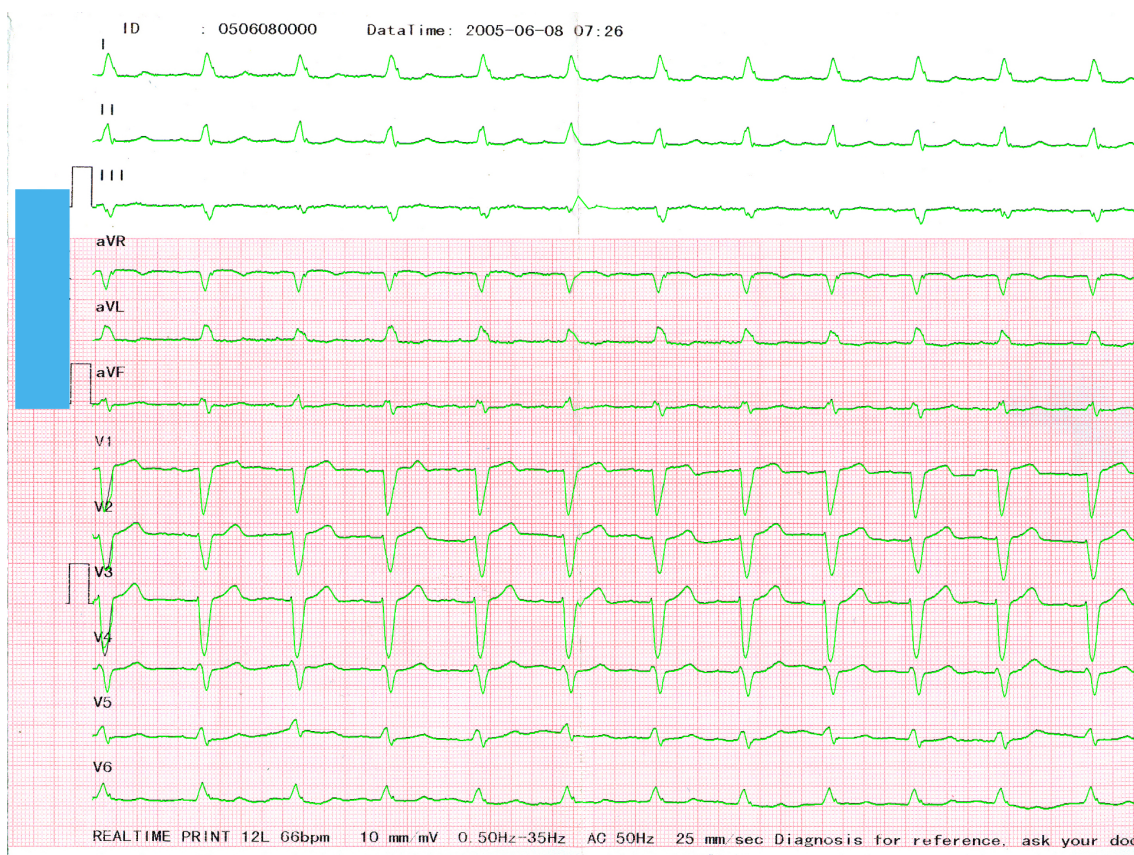
S obzirom na to da je debljina signala veća od jednog piksela, svako pojavljivanje reči „piksel” u prethodnom pasusu se može shvatiti kao skraćunica za „piksel koji je sredina vertikalnog povezanog regiona belih piksela”.

Rezultat izvršavanja ovog dela algoritma dat je na slici 4.4. Na ovom EKG dijagramu grešaka u pronađenom signalu gotovo i da nema. Problemi koji postoje su praktično neotklonjivi, jer su posledica nesavršenog kvaliteta skenirane slike. Jedan problem predstavlja to što je dijagram bio savijen po sredini, pa se duž te linije obrisao deo signala. Posledica je greška kod šestog pika u trećem kanalu. Preostale greške koje su uočene su nastale zbog mešanja slova koja predstavljaju imena kanala i samog signala. Konkretno imena kanala V2, V3 i V4 se preklapaju sa delom signala, pa tu dolazi do manjih grešaka.

### 4.3 Podela na kanale

U prethodnom koraku određene su vrednosti signala u pikselima. Kao što je već napomenuto u delu 2, postoje EKG dijagrami na kojima je 12 kanala raspoređeno u 3, 6 ili 12 redova. Ako na dijagramu postoji:

1. 12 redova, u prethodnom koraku je već prepoznato svih 12 kanala pojedinačno;
2. 6 redova, u prethodnom koraku su prepoznati parovi kanala;
3. 3 reda, u prethodnom koraku su prepoznate četvorke kanala.



Slika 4.4: Prepoznat signal

U drugom (odnosno trećem) slučaju potrebno je podeliti dobijene signale na dva (odnosno četiri) dela, da bi se dobile vrednosti za svih 12 kanala pojedinačno. Ovaj postupak je opravdan, jer svaki kanal jednako dugo traje.

#### 4.4 Pretvaranje piksela u sekunde i volte

U prethodnom delu bi trebalo da su određene vrednosti dužine stranice kvadrata mreže u pikselima  $L$  3.4, kao i brzine štampe  $V$  i napona  $U$  3.6. Ukoliko to nije slučaj, koriste se podrazumevane vrednosti za svaki od ovih parametara. Kako je dužina stranice kvadrata na EKG dijagramu 5 milimetara, dužina jednog piksela u milimetrima se računa deljenjem broja 5 sa  $L$ . Da bi se dobilo trajanje jednog piksela u sekundama, deli se njegova dužina u milimetrima sa  $V$ . Slično, da bi se dobio napon jednog piksela u milivoltima, deli se njegova dužina sa  $U$ .

Za svaki piksel koji je prethodno odabran kao deo signala može se izračunati

vremenski trenutak u kome se nalazi i njegov napon. Parovi vrednosti (vreme, napon) se konačno zapisuju u datoteku koja kasnije može da se obrađuje za razne potrebe. Ovde je, zbog veličine, prikazan samo deo JSON datoteke sa digitalnim zapisom, a celokupna datoteka je dostupna u elektronskoj verziji rada.

```
{
  "I" : [[0.0, 0.0], ..., [2.032, -0.4233333333333334], ..., [4.064, 0.13546666666666668], ...],
  "II" : [[0.0, 0.0], ..., [2.032, -0.4487333333333334], ..., [4.064, 0.04233333333333334], ...],
  "III" : [[0.0, 0.0], ..., [2.032, 0.008466666666666667], ..., [4.064, -0.01693333333333335], ...],
  "aVR" : [[0.0, 0.0], ..., [2.032, 0.3048], ..., [4.064, -0.01693333333333335], ...],
  "aVL" : [[0.0, 0.0], ..., [2.032, -0.1778], ..., [4.064, 0.06773333333333334], ...],
  "aVF" : [[0.0, 0.0], ..., [2.032, -0.04233333333333334], ..., [4.064, 0.008466666666666667], ...],
  "V1" : [[0.0, 0.0], ..., [2.032, 0.7112], ..., [4.064, -0.16086666666666669], ...],
  "V2" : [[0.0, 0.0], ..., [2.032, 0.8636], ..., [4.064, -0.22013333333333335], ...],
  "V3" : [[0.0, 0.0], ..., [2.032, 1.0329333333333335], ..., [4.064, -0.34713333333333335], ...],
  "V4" : [[0.0, 0.0], ..., [2.032, 0.508], ..., [4.064, -0.10160000000000001], ...],
  "V5" : [[0.0, 0.0], ..., [2.032, 0.025400000000000002], ..., [4.064, -0.03386666666666667], ...],
  "V6" : [[0.0, 0.0], ..., [2.032, -0.008466666666666667], ..., [4.064, 0.08466666666666667], ...]
}
```

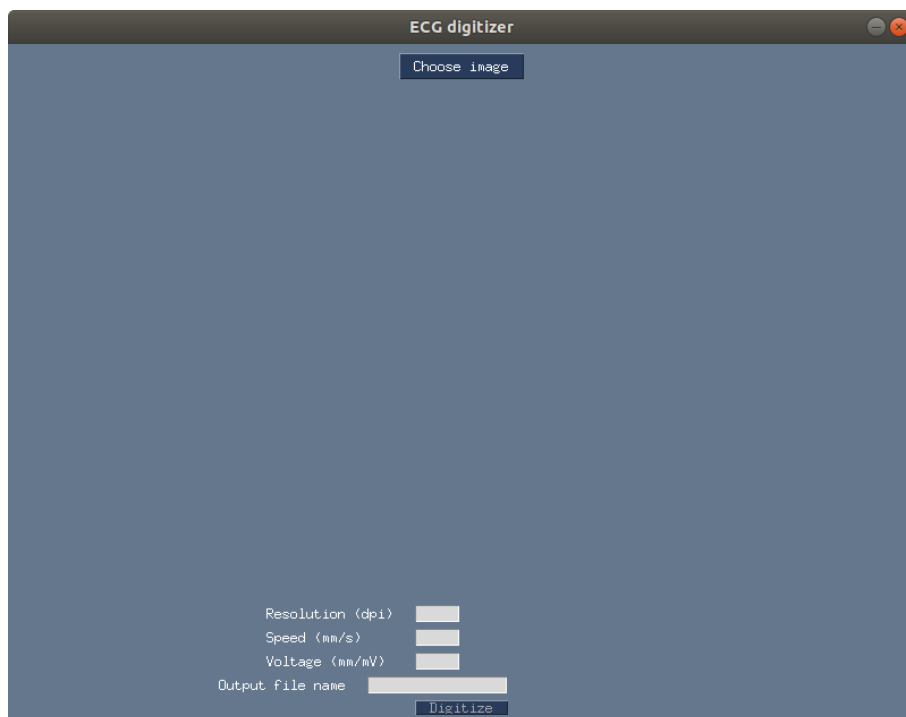
Uz male izmene u formatu JSON datoteke može se dobiti dosta manji i pregledniji izlaz. S obzirom na to da je veličina piksela zajednička za sve kanale, može se samo jednom navesti veličina piksela u sekundama i u milivoltima, a kao vrednosti signala se onda mogu koristiti celi brojevi (y koordinate piksela pronađenih u prethodnom delu 4.2). Dodatno, da bi se u kasnijoj analizi mogla iskoristiti ta informacija, za sve kolone u kojima nije pronađen piksel koji je deo signala korišćena je specijalna vrednost **null**. U nastavku je prikazan format ovakve datoteke, čiji „info” deo se može proširiti i drugim korisnim informacijama vezanim za konkretan EKG dijagram (npr. podaci o pacijentu).

```
{
  "info" : {
    "pixel_time": 0.0033866666666666667,
    "pixel_voltage": 0.008466666666666667
  },
  "data" : {
    "I" : [0, ..., null, ..., 12, ... , -10, ...],
    "II" : [...],
    "III" : [...],
    "aVR" : [...],
    "aVL" : [...],
    "aVF" : [...],
    "V1" : [...],
    "V2" : [...],
    "V3" : [...],
    "V4" : [...],
    "V5" : [...],
    "V6" : [...]
  }
}
```

## Glava 5

# Grafički korisnički interfejs

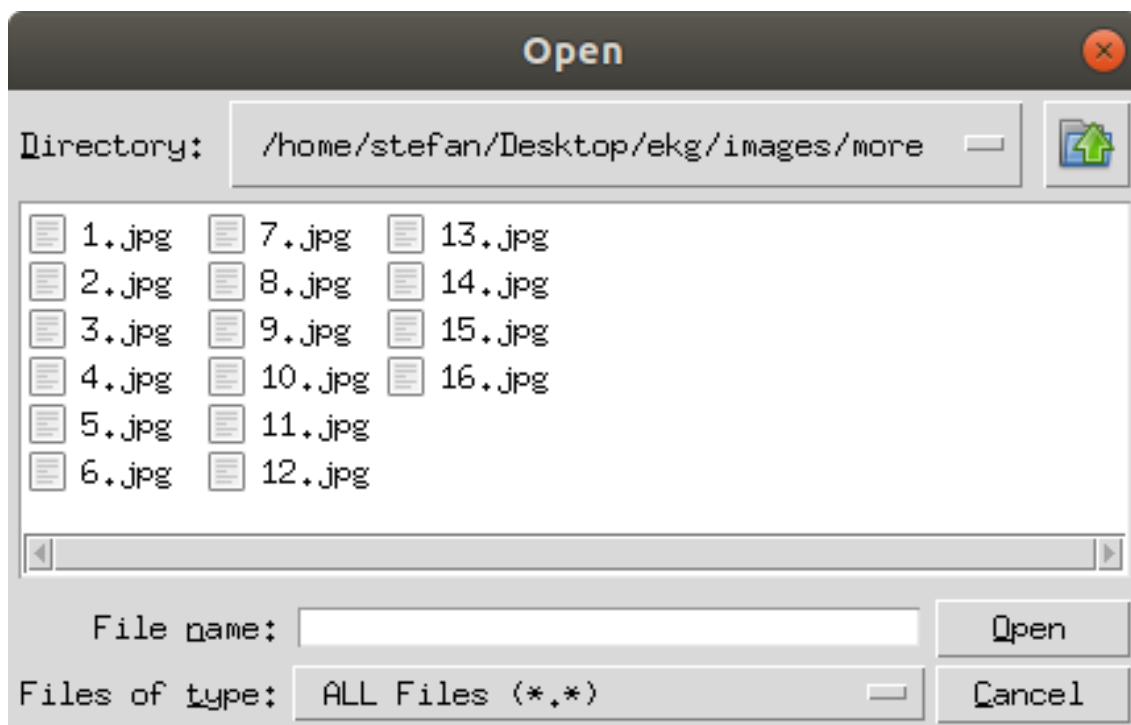
Osmišljeni algoritam za digitalizaciju je implementiran u programskom jeziku Python. S obzirom na to bi potencijalni korisnici ovakvog programa mogli da budu i medicinski radnici ili biolozi, pored programa koji se može pozivati iz terminala, napravljen je i jednostavan grafički korisnički interfejs. Korišćena je biblioteka PySimpleGUI [12], a celokupan kôd je dostupan na adresi <https://github.com/StefanKapunac/ekg>. Po pokretanju prikazuje se prozor koji je prikazan na slici 5.1.



Slika 5.1: Početni prozor



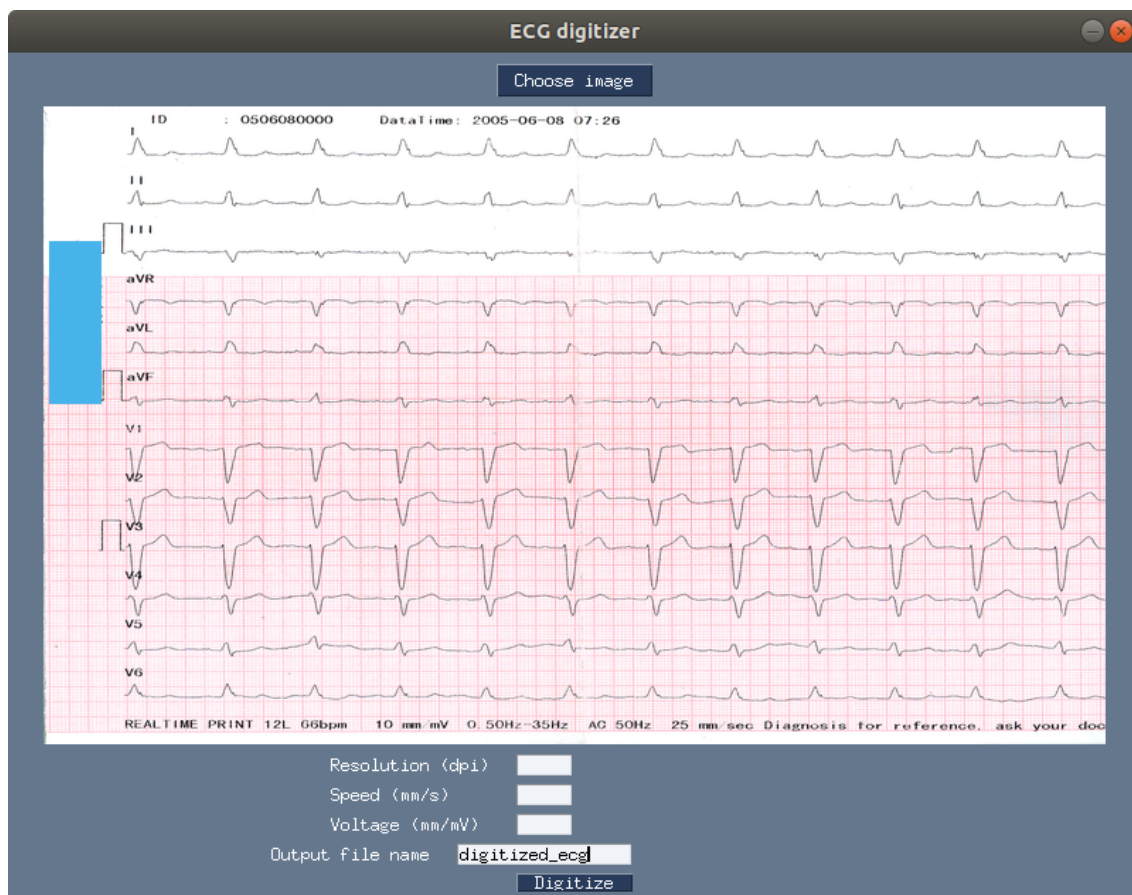
Jedino što korisnik može da uradi u ovom trenutku jeste da učita sliku koristeći dugme „Choose image”. Kada pritisne to dugme, otvara se prozor nalik onom na slici 5.2.



Slika 5.2: Odabir slike

## GLAVA 5. GRAFIČKI KORISNIČKI INTERFEJS

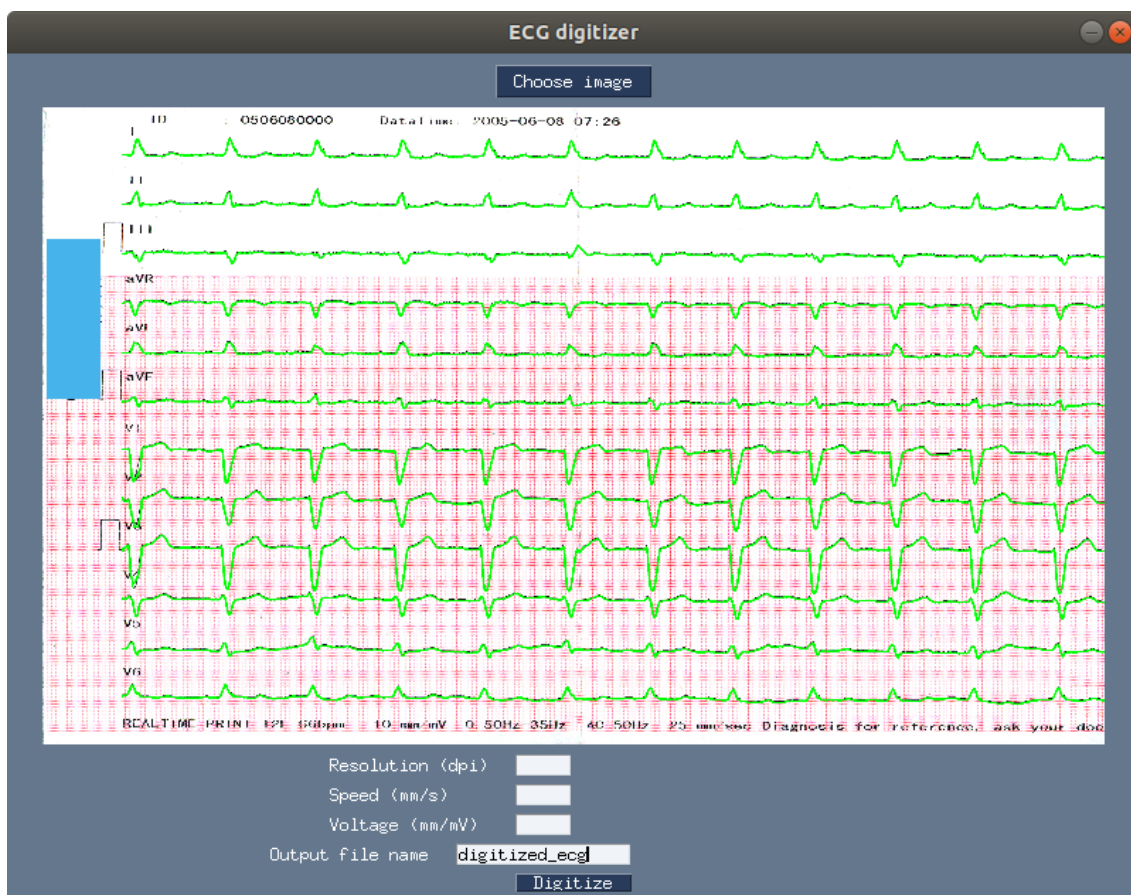
Korisnik može da pretražuje sistem datoteka i da izabere sliku skeniranog EKG dijagrama koji želi da digitalizuje. Kada to uradi, odabrana slika se prikazuje u centralnom delu prozora, kao na slici 5.3.



Slika 5.3: Učitana slika

Korisnik sada ima opciju da popuni polja vezana za rezoluciju, brzinu štampe i voltažu. Ova polja nisu obavezna, ali trajanje procesa digitalizacije će biti manje ako su ovi podaci poznati. Ukoliko je poznata rezolucija u kojoj je EKG dijagram skeniran, nema potrebe za nalaženjem kvadrata na slici opisanog u delu 3.4, jer se na osnovu rezolucije može izračunati veličina piksela u milimetrima. Slično, ukoliko su poznati brzina štampe i kalibracija napona, nema potrebe za optičkim prepoznavanjem karaktera, koje je opisano u delu 3.6.

Postoji još jedno polje koje korisnik treba da popuni, a to je ime izlazne datoteke u koju će biti smešten digitalizovan EKG, kao vremenska serija, odnosno parovi vrednosti (vreme, napon). Nakon toga, korisnik pokreće proces digitalizacije koristeći dugme „Digitize”. Kada se digitalizacija završi, napravljena je datoteka sa digitalnim EKG-om, a korisniku se, na mestu gde je prethodno bila originalna slika, prikazuje slika sa prepoznatim EKG signalom, kao što je prikazano na slici 5.4.



Slika 5.4: Prepoznat signal

## Glava 6

# Zaključak

U ovom radu rešavan je problem digitalizacije EKG dijagrama, to jest pretvaranja slika skeniranih dijagrama u digitalni format, pogodniji za dalju analizu i obradu. Kvalitet digitalizacije, naravno, zavisi od kvaliteta samog dijagrama (odnosno njegove izbledeposti, savijenosti ili čak pocepanosti) i kvaliteta skeniranja (odnosno rezolucije). U skladu sa tim, ni program koji je napisan ne može da radi savršeno za svaki mogući ulaz. Za jedan konkretan dijagram prikazani su rezultati koji deluju obećavajuće. Na EKG dijagramima sličnog kvaliteta, mogu se očekivati i slični rezultati.

Do sada se kvalitet dobijenih rezultata ocenjivao na osnovu vizuelnog utiska. Za ozbiljniju validaciju napisanog programa moguće je koristiti specijalizovane metrike, bazirane na korelaciji između digitalizovanog EKG signala i onog na papiru. Pritom se mogu koristiti različiti standardni EKG parametri (RR interval, PR interval...). Takođe, neophodna je veća količina EKG dijagrama koji bi se koristili za testiranje.

Naravno, ima prostora i za poboljšanja samog algoritma. Na primer, može se pokušati sa skaliranjem ulazne slike faktorom 2, 2.5, 3, 3.5 sa ciljem preciznijeg određivanja signala, kao i umanjivanja uticaja pretprocesiranja na signal. Takođe, potrebno je poboljšati rad programa u slučajevima kada se na ulaznom dijagramu preklapaju signali. Ovo predstavlja glavni pravac za dalji rad, uz dodatna finija podešavanja u samom algoritmu.

# Bibliografija

- [1] Andrew J. Accardi, Robert Miller, and James F Holmes Jr. Enhanced diagnosis of narrow complex tachycardias with increased electrocardiograph speed. *Journal of Emergency Medicine*, 22(2):123–126, 2002.
- [2] Mohammed Baydoun, Lise Safatly, Ossama Abou Hassan, Hassan Ghaziri, Ali Hajj, and Hussain Isma'eel. High precision digitization of paper-based ecg records: A step toward machine learning. *IEEE Journal of Translational Engineering in Health and Medicine*, PP:1–1, 11 2019.
- [3] G. Bradski. The OpenCV Library. *Dr. Dobb's Journal of Software Tools*, 2000.
- [4] John Canny. A computational approach to edge detection. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, PAMI-8:679 – 698, 12 1986.
- [5] DAVID H DOUGLAS and THOMAS K PEUCKER. Algorithms for the reduction of the number of points required to represent a digitized line or its caricature. *Cartographica: The International Journal for Geographic Information and Geovisualization*, 10(2):112–122, 1973.
- [6] Richard O. Duda and Peter E. Hart. Use of the hough transformation to detect lines and curves in pictures. *Commun. ACM*, 15(1):11–15, 1972.
- [7] Joao Luis Gaspar and Richard Body. Differential diagnosis of narrow complex tachycardias by increasing electrocardiograph speed. *Emergency Medicine Journal*, 22(10):730–732, 2005.
- [8] Samuel Hoffstaetter. Python-tesseract. <https://pypi.org/project/pytesseract/>.
- [9] Paul C. Hough V. Method and means for recognizing complex patterns. (3069654), December 1962.

- [10] J. D. Hunter. Matplotlib: A 2d graphics environment. *Computing in Science & Engineering*, 9(3):90–95, 2007.
- [11] Travis E Oliphant. *A guide to NumPy*, volume 1. Trelgol Publishing USA, 2006.
- [12] PySimpleGUI Organization. Pysimplegui. <https://pysimplegui.readthedocs.io/en/latest/>.
- [13] Nobuyuki Otsu. A Threshold Selection Method from Gray-level Histograms. *IEEE Transactions on Systems, Man and Cybernetics*, 9(1):62–66, 1979.
- [14] U. Ramer. An iterative procedure for the polygonal approximation of plane curves. *Comput. Graph. Image Process.*, 1:244–256, 1972.
- [15] L. Ravichandran, C. Harless, A. J. Shah, C. A. Wick, J. H. McClellan, and S. Tridandapani. Novel tool for complete digitization of paper electrocardiography data. *IEEE Journal of Translational Engineering in Health and Medicine*, 1:1800107–1800107, 2013.
- [16] Araz Rawshani. Clinical ecg interpretation. <https://ecgwaves.com/course/the-ecg-book/>.
- [17] R. Smith. An overview of the tesseract ocr engine. volume 2, pages 629 – 633, 10 2007.
- [18] Guido Van Rossum and Fred L. Drake. *Python 3 Reference Manual*. CreateSpace, Scotts Valley, CA, 2009.