

# Pronalaženje ili prepoznavanje QRS kompleksa u EKG-u

---

**Mandić, Bojan**

**Master's thesis / Diplomski rad**

**2024**

*Degree Grantor / Ustanova koja je dodijelila akademski / stručni stupanj:* **Josip Juraj Strossmayer University of Osijek, Faculty of Electrical Engineering, Computer Science and Information Technology Osijek / Sveučilište Josipa Jurja Strossmayera u Osijeku, Fakultet elektrotehnike, računarstva i informacijskih tehnologija Osijek**

*Permanent link / Trajna poveznica:* <https://um.nsk.hr/um:nbn:hr:200:342240>

*Rights / Prava:* [In copyright](#)/[Zaštićeno autorskim pravom.](#)

*Download date / Datum preuzimanja:* **2025-02-23**

*Repository / Repozitorij:*

[Faculty of Electrical Engineering, Computer Science and Information Technology Osijek](#)



**SVEUČILIŠTE JOSIPA JURJA STROSSMAYERA U OSIJEKU  
FAKULTET ELEKTROTEHNIKE, RAČUNARSTVA I  
INFORMACIJSKIH TEHNOLOGIJA OSIJEK**

**Sveučilišni diplomski studij  
Računarstvo**

**Pronalaženje ili prepoznavanje QRS kompleksa u EKG-u  
Diplomski rad**

**Bojan Mandić**

**Osijek, 2024. godina.**

**Obrazac D1: Obrazac za ocjenu diplomskog rada na sveučilišnom diplomskom studiju**

**Ocjena diplomskog rada na sveučilišnom diplomskom studiju**

<b>Ime i prezime pristupnika:</b>	Bojan Mandić
<b>Studij, smjer:</b>	Sveučilišni diplomski studij Računarstvo
<b>Mat. br. pristupnika, god.</b>	D1303R, 07.10.2022.
<b>JMBAG:</b>	0036501741
<b>Mentor:</b>	prof. dr. sc. Krešimir Nenadić
<b>Sumentor:</b>	
<b>Sumentor iz tvrtke:</b>	izv. prof. dr. sc. Dario Galić
<b>Predsjednik Povjerenstva:</b>	doc. dr. sc. Hrvoje Leventić
<b>Član Povjerenstva 1:</b>	prof. dr. sc. Krešimir Nenadić
<b>Član Povjerenstva 2:</b>	Robert Šojo, univ. mag. ing. comp.
<b>Naslov diplomskog rada:</b>	Pronalaženje ili prepoznavanje QRS kompleksa u EKG-u
<b>Znanstvena grana diplomskog rada:</b>	<b>Obradba informacija (zn. polje računarstvo)</b>
<b>Zadatak diplomskog rada:</b>	Opisati svojstva EKG dijagrama. Dati pregled postojećih metoda pronalaženja i prepoznavanja QRS kompleksa u EKG dijagramu. Zadatak rada je napraviti aplikaciju za bilo koju platformu koja pronalazi ili prepoznaje QRS kompleks u EKG-u. Metode pronalaska i/ili prepoznavanja mogu biti: strojno učenje/neuronske mreže, waveleti, fraktali, prag, uvjet šiljatosti ili neka druga poznata metoda. Tema rezervirana: Bojan Mandić
<b>Datum ocjene pismenog dijela diplomskog rada od strane mentora:</b>	21.09.2024.
<b>Ocjena pismenog dijela diplomskog rada od strane mentora:</b>	Izvrstan (5)
<b>Datum obrane diplomskog rada:</b>	10.10.2024.
<b>Ocjena usmenog dijela diplomskog rada (obrane):</b>	Izvrstan (5)
<b>Ukupna ocjena diplomskog rada:</b>	Izvrstan (5)
<b>Datum potvrde mentora o predaji konačne verzije diplomskog rada čime je pristupnik završio sveučilišni diplomski studij:</b>	15.10.2024.



**FERIT**

FAKULTET ELEKTROTEHNIKE, RAČUNARSTVA  
I INFORMACIJSKIH TEHNOLOGIJA **OSIJEK**

## IZJAVA O IZVORNOSTI RADA

Osijek, 15.10.2024.

**Ime i prezime Pristupnika:**

Bojan Mandić

**Studij:**

Sveučilišni diplomski studij Računarstvo

**Mat. br. Pristupnika, godina upisa:**

D1303R, 07.10.2022.

**Turnitin podudaranje [%]:**

7

Ovom izjavom izjavljujem da je rad pod nazivom: **Pronalaženje ili prepoznavanje QRS kompleksa u EKG-u**

izrađen pod vodstvom mentora prof. dr. sc. Krešimir Nenadić

i sumentora

moj vlastiti rad i prema mom najboljem znanju ne sadrži prethodno objavljene ili neobjavljene pisane materijale drugih osoba, osim onih koji su izričito priznati navođenjem literature i drugih izvora informacija.

Izjavljujem da je intelektualni sadržaj navedenog rada proizvod mog vlastitog rada, osim u onom dijelu za koji mi je bila potrebna pomoć mentora, sumentora i drugih osoba, a što je izričito navedeno u radu.

Potpis pristupnika:

# SADRŽAJ

<b>1. UVOD .....</b>	<b>1</b>
1.1. Zadatak diplomskog rada .....	2
<b>2. GRAĐA SRCA I EKG .....</b>	<b>3</b>
2.1. Srce .....	3
2.2. Otkucaj.....	5
2.3. Elektrokardiografija.....	8
2.4. Karakteristike EKG-a .....	10
2.5. Elektrokardiograf .....	11
<b>3. Algoritmi za detekciju QRS kompleksa .....</b>	<b>12</b>
3.1. Pan-Tompkinsov Algoritam.....	12
3.2. Transformacija po valićima .....	18
3.3. Hamilton-Tompkinsov Algoritam .....	19
3.4. Hilbert-Huang transformacija.....	20
3.5. Podudarno filtriranje.....	22
<b>4. Implementacija .....</b>	<b>22</b>
4.1. Korištene tehnologije .....	23
4.2. Učitavanje podataka .....	23
4.3. Implementacija sučelja .....	26
4.4. Detekcija QRS kompleksa.....	27
4.5. Sučelje za prikaz rezultata .....	32
<b>5. Prikaz rada aplikacije.....</b>	<b>34</b>
5.1. Izgled sučelja aplikacije.....	34
5.2. Testiranje aplikacije .....	37
5.3. Analiza rezultata .....	44
<b>6. Zaključak .....</b>	<b>48</b>
<b>Literatura .....</b>	<b>49</b>

<b>Sažetak.....</b>	<b>51</b>
<b>Abstract .....</b>	<b>52</b>
<b>Životopis.....</b>	<b>53</b>

## 1. UVOD

Elektrokardiogram ili EKG je snimka koja je dobivena korištenjem elektrokardiografa, uređaja koji bilježi električnu aktivnost srca. Razvoj elektrokardiografa započeo je u drugoj polovici 19. stoljeća, nakon što je potvrđena ideja o električnoj aktivnosti unutar srčanog mišića. Luigi Galvan [1], talijanski znanstvenik, 1787. godine otkriva postojanje električne aktivnosti unutar mišića. Iako njegovo otkriće nije direktno povezano sa srcem i električnom aktivnosti unutar njega, ono postaje temelj za daljnja istraživanja. Krajem 19. stoljeća, 1883. godine, francuski znanstvenik August Waller [2], dokazuje mogućnost mjerenja varijacija električnog potencijala srca na površini kože. Međutim, njegov uređaj nije bio primjenjiv u medicini zbog svoje primitivnosti. Tek početkom 20. stoljeća, nizozemski liječnik Willem Einthoven [3] i [4] konstruira prvi upotrebljivi EKG uređaj, koji je na principu tankih, osjetljivih žica mogao izmjeriti male električne promjene generirane u srcu. Također, standardizira EKG snimanje, uvodi pojmove P, Q, R, S i T valova koji se koriste i danas, te za to dobiva i Nobelovu nagradu za fiziologiju ili medicinu, koja potvrđuje njegovu titulu oca elektrokardiografije. Nakon Einthovena, razvoj EKG-a se ubrzano nastavlja, uređaji postaju sve manji i jeftiniji, šire dostupni i postaju neizostavan alat u medicini i dijagnostici. Dodatnim napretkom tehnologije, 60-ih godina 20. stoljeća, pojavljuju se prvi računalno potpomognuti uređaji koji automatskom analizom signala smanjuju vrijeme očitavanja i povećavaju preciznost dijagnoze. Digitalizacijom EKG, koja se dogodila 1980-ih i 1990-ih, napravljen je dodatni korak u poboljšanju istoga, zbog kojeg postaje jedan od najvažnijih uređaja u modernoj medicini. Ručno prepoznavanje QRS kompleksa u EKG dijagramu može biti složeno i podložno pogreškama, posebno u situacijama s velikim brojem podataka ili u slučaju nejasnih signala. Razvijanjem automatizirane aplikacije koja koristi napredne metode povećava se točnost i brzina detekcije karakterističnih točaka EKG-a, što je vrlo bitno s obzirom na to da srčana oboljenja predstavljaju jedan od vodećih uzroka smrti u svijetu.

U drugom poglavlju ovog rada opisano je srce i građa, te funkcionalnosti istoga. Također, opisan je proces nastajanja električnog signala u srcu i njegov put, elektrokardiogram, te karakteristične točke na njemu. Treće poglavlje obuhvaća opis nekih od najpopularnijih algoritama za detekciju QRS kompleksa. U četvrtom poglavlju opisan je način implementacije aplikacije, koji obuhvaća korištene alate i biblioteke, te podatke. U petom poglavlju objašnjen je izgled sučelja aplikacije, kao i testiranje iste, te analiza dobivenih rezultata.

## **1.1. Zadatak diplomskog rada**

Zadatak rada je opisati svojstva EKG dijagrama, te dati pregled postojećih metoda pronalaženja i prepoznavanja QRS kompleksa u EKG dijagramu. Također, potrebno je napraviti aplikaciju za bilo koju platformu koja pronalazi ili prepoznaje QRS kompleks u EKG-u. Metode pronalaska i/ili prepoznavanja mogu biti: strojno učenje/neuronske mreže, waveleti, fraktali, prag, uvjet šiljatosti ili neka druga poznata metoda. Za izradu aplikacije korišten je programski jezik Python i popratne biblioteke, dok je validacija obavljena s pomoću MIT-Arrhythmia [5] baze podataka. Ona korisniku omogućava učitavanje EKG signala u .csv formatu, te na temelju njega detektira i označava sve karakteristične točke QRS kompleksa na grafu. Također, omogućava korisniku učitavanje anotacija sa svrhom validacije. Osim grafa, aplikacija prikazuje korisniku broj lažno pozitivnih i lažno negativnih vrhova te ih označava na istom, ako je učitana anotacija

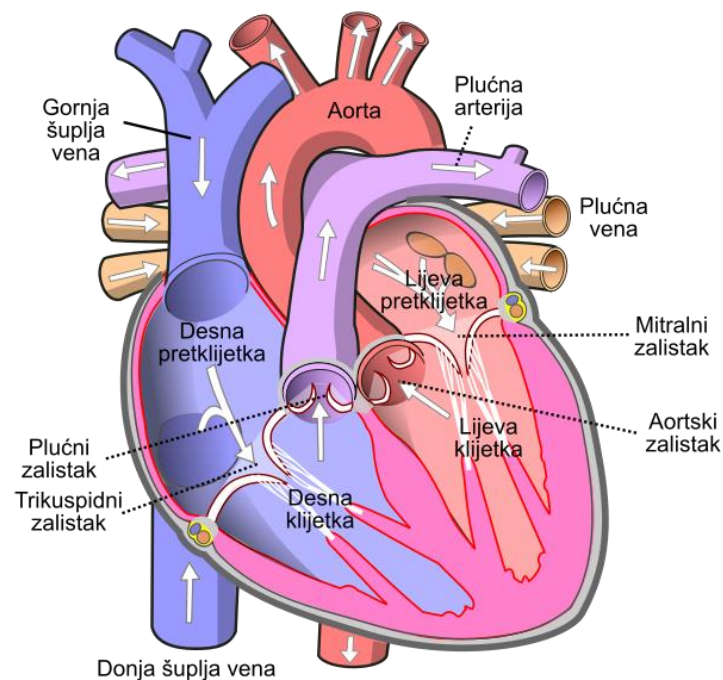


## 2. GRAĐA SRCA I EKG

U ovom poglavlju obrađuje se osnovna građa srca i kako nastaju otkucaji, s posebnim naglaskom na električnu aktivnost. Detaljno su opisani glavni dijelovi srca, njihov doprinos u procesu pumpanja krvi, te su prikazani osnovni principi elektrokardiografije i karakteristike elektrokardiograma.

### 2.1. Srce

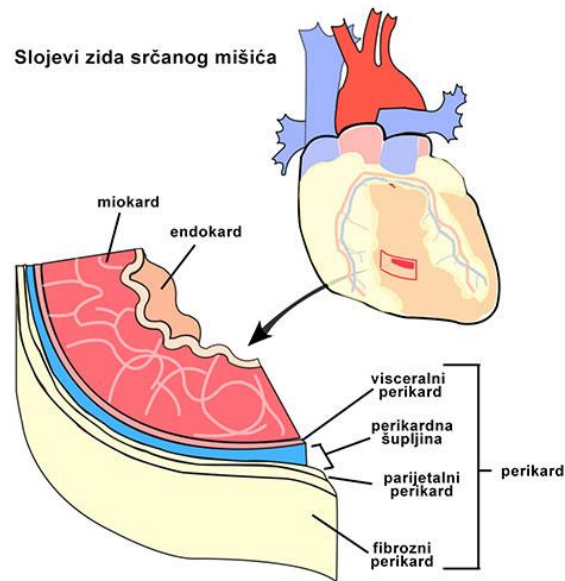
Prema [6], srce je mišićna pumpa veličine stisnute šake, mase između 200 i 425 grama. Nalazi se iza pluća u prsnom košu. Glavna funkcija je pumpanje krvi kroz krvne žile u različite dijelove tijela, a samim time i opskrba tijela hranjivim tvarima i kisikom koji se nalaze u krvi. Sadrži četiri komore: desnu i lijevu pretkljetku, te desnu i lijevu kljetku. U pretkljetkama se skuplja krv koja se vraća u srce, dok kljetke djeluju kao pumpe koje izbacuju krv u tijelo. Građa srca prikazana je na slici 2.1.



Slika 2.1 Građa srca [7]

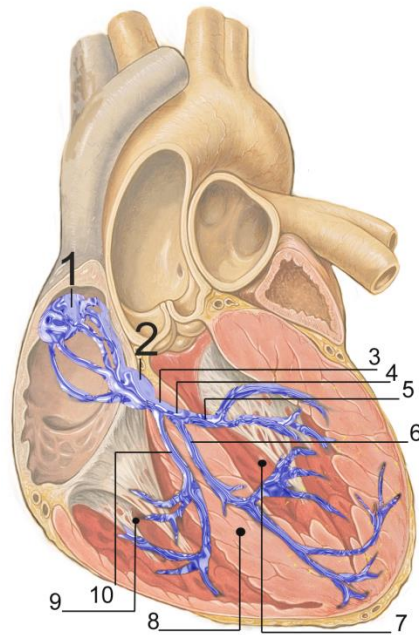
Srce je građeno od endokarda, miokarda i epikarda. Miokard je najdeblji sloj, a sadrži poprečnoprugaste mišićne stanice i provedbene srčane mišićne stanice. Ovaj sloj čini većinu

srčanog tkiva, a ima ulogu stvaranja impulsa koji uzrokuju otkucaje. Endokard je unutrašnji sloj srca koji oblaže srčane komore, ventrikule, i pretkomore, atrijske, kao i srčane zaliske. On je tanka, glatka membrana, a uloga mu je štititi srčano tkivo od patogena koji se nalaze u krvi. Epikard je vanjski sloj srca i predstavlja dio perikarda, odnosno srčane maramice. Prekriva srce, a njegova uloga je smanjenje trenja između srca i okolnih struktura pri otkucajima. Slojevi srca vidljivi su na slici 2.2.



**Slika 2.2** Slojevi srčanog mišića [8]

Provedbeni srčani mišići dijele se na sinusatrijski i atrioventrikularni sustav. Sinusatrijski sustav sastoji se od sinusatrijskog čvora, na slici 2.3 predstavljen brojem 1. On stvara električni impuls koji se širi kroz srce i uzrokuje kontrakcije. Atrioventrikularni sustav se sastoji od atrioventrikularnog čvora, na slici 2.3 vidljiv pod brojem 2, koji električno povezuje pretklijetke i klijetke, te Hissovog snopa, na slici 2.3 prikazan pod brojem 3, koji prenosi impulse do vrha srca i omogućuju sinkronizirane kontrakcije klijetki.



**Slika 2.3** Električni provodni sustav srca [9]

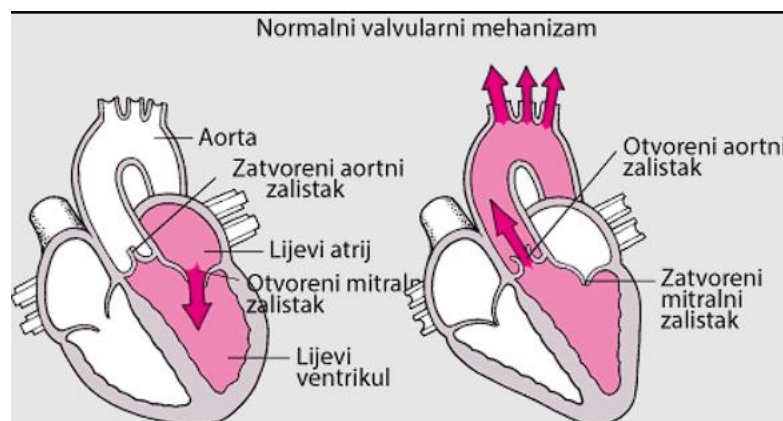
Srce je prekriveno perikardom, dvoslojnom membranom. Njegov vanjski sloj okružuje korijene glavnih krvnih žila i pričvršćen je ligamentima za kralježnicu, dijafragmu i druge dijelove tijela. Unutarnji sloj perikarda pričvršćen je na srčani mišić. Između dva sloja membrane nalazi se tekućina, koja omogućava srčanu mobilnost tijekom kucanja, a da ono ipak ostane pričvršćeno za tijelo.

## 2.2. Otkucaj

Rad srca se sastoji od skupljanja i širenja, koji zajedno čine bilo ili puls. Svaki otkucaj srca uzrokovan je sinkroniziranim slanjem električnih impulsa, kontrakcijom mišića i preciznim otvaranjem i zatvaranjem srčanih zalistaka. Otkucaj započinje depolarizacijom, koja je promjena električnog stanja srčanih mišićnih stanica. Događa se kada unutrašnjost stanica postaje manje negativna u odnosu na vanjsku stranu. Zapčinje u sinusatrijskom čvoru koji se nalazi u desnom atriju. Sinusatrijski čvor generira električni impuls koji se širi preko atrija, uzrokujući njihovu kontrakciju, atrijsku sistolu, koja zatim gura krv iz atrija u ventrikule kroz otvorene atrioventrikularne zaliske. Nakon atrijske sistole, električni impuls putuje do atrioventrikularnog čvora koji se nalazi na spoju između atrija i ventrikula. Čvor djeluje kao vrata, nakratko odgađa impuls kako bi ventrikule imale vremena napuniti se krvlju. Nakon što impuls prođe kroz atrioventrikularni čvor, dolazi u Hissov snop, gdje se zatim dijeli na desnu i lijevu granu. Nakon

toga se širi Purkinjeovim vlaknima, uzrokujući kontrakciju ventrikula koja izbacuje krv u pluća i ostatak tijela. Vlakna osiguravaju brzo i koordinirano prenošenje električnog impulsa koje dovodi do sinkronizirane kontrakcije ventrikula, odnosno ventrikularne sistole.

Tijekom ventrikularne sistole, ventrikuli se snažno skupljaju. Time potiskuju krv u sistemsku i plućnu cirkulaciju. Desni ventrikul pumpa deoksigeniranu krv, odnosno, krv siromašnu kisikom, kroz plućni zalistak u plućne arterije. U plućima se ugljični dioksid zamjenjuje kisikom, a sada kisikom bogata krv vraća se u lijevi atrij kroz plućne vene. Istovremeno, lijevi ventrikul pumpa krv punu kisika kroz aortalni zalistak u aortu. Iz aorte, najveće arterije u tijelu, krv se distribuira do svih organa i tkiva, te im dostavlja kisik i hranjive tvari. Zalisci osiguravaju da krv teče iz atrija u ventrikule tijekom atrijske sistole, a ostaju zatvoreni tijekom ventrikularne sistole kako bi spriječili njeno vraćanje, što je vidljivo na slici 2.4.



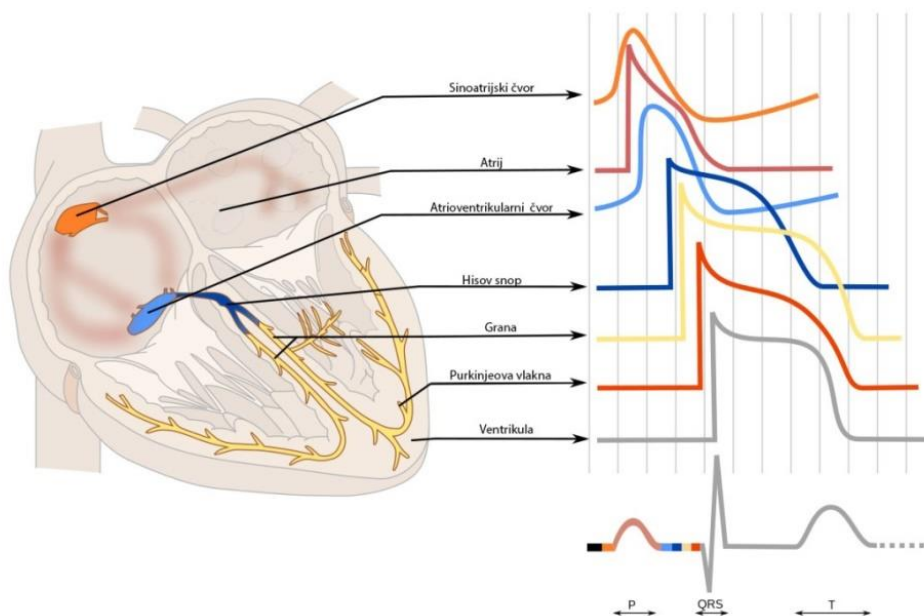
**Slika 2.4** Srčani zalisci pri otkucaju [10]

Nakon depolarizacije dolazi do repolarizacije, pri čemu se srčane mišićne stanice vraćaju u svoje mirno stanje, što znači da unutrašnjost postaje opet negativnija. Ovo je ključno za pripremu srčanog mišića za sljedeći ciklus depolarizacije i kontrakcije. Dijastola je faza srčanog ciklusa kada se srčani mišić opušta. Sastoji se od rane dijastole i kasne dijastole. Tijekom rane dijastole, ventrikuli se opuštaju, a tlak unutar njih pada. Ovo uzrokuje zatvaranje plućnih i aortalnih zalistaka. Kako se ventrikuli nastavljaju opuštati, tlak unutar njih pada ispod tlaka u atrijima, otvarajući trikuspidalne i mitralne zalistke. Krv pasivno teče iz atrija u ventrikule, ispunjavajući ih u pripremi za sljedeći otkucaj. Kasna dijastola obuhvaća atrijsku kontrakciju koja dovršava punjenje ventrikula. Ovim se osigurava da su ventrikuli adekvatno ispunjeni krvlju koja će biti izbačena tijekom sljedeće sistole.

Dakle, otkucaj srca se dijeli na pet faza:

- 1) **Atrijalna depolarizacija (P val):** električni impuls kreće se kroz atrije, uzrokuje njihovu kontrakciju i potiskivanje krvi u ventrikule
- 2) **Kašnjenje u atrioventrikularnom čvoru (PR segment):** impuls stiže do atrioventrikularnog čvora u kojemu dolazi do kratkog kašnjenja radi punjenja ventrikula krvlju, zatim kroz Hisov snop do Purkinjeovih vlakna, a na kraju do ventrikula
- 3) **Ventrikularna depolarizacija (QRS kompleks):** impuls putuje kroz ventrikule, te uzrokuje njihovu depolarizaciju i kontrakciju
- 4) **Ventrikularna repolarizacija (T val):** ventrikuli se repolariziraju i pripremaju za sljedeći ciklus otkucaja srca.
- 5) **Ventrikularna relaksacija i punjenje:** srce se opušta, pad tlaka u komorama uzrokuje otvaranje atrioventrikularnih zaliska, te se ono puni krvlju

Na slici 2.5 vidljivi su različiti oblici srčanog akcijskog potencijala koji zajedno tvore karakteristične točke EKG-a, detaljnije opisane u nastavku ovog rada.



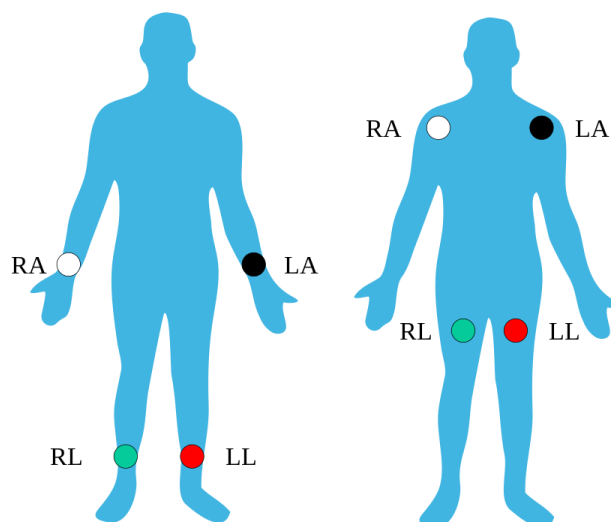
**Slika 2.5** Oblici srčanog akcijskog potencijala u srcu [11]

Autonomni živčani sustav regulira brzinu otkucaja srca i snagu kontrakcija. Simpatički živčani sustav oslobađanjem noradrenalina, neurotransmitera i hormona koji se proizvodi u

nadbubrežnim žlijezdama i u živčanom sustavu, povećava brzinu otkucaja srca i snagu ventrikularnih kontrakcija, pripremajući tijelo za stres. Parasimpatički živčani sustav putem vagusnog živca i oslobađanja neurotransmitera acetilkolina, koji je ključan za prijenos signala između živaca i mišića, smanjuje brzinu otkucaja srca i dovodi tijelo u stanje odmora. Adrenalin oslobođen iz nadbubrežnih žlijezda tijekom stresa povećava brzinu otkucaja srca i kontraktilnost, odnosno, svojstvo mišićnog tkiva da se steže pod utjecajem podražaja, dok hormoni štitnjače igraju ulogu u moduliranju brzine otkucaja srca.

### 2.3. Elektrokardiografija

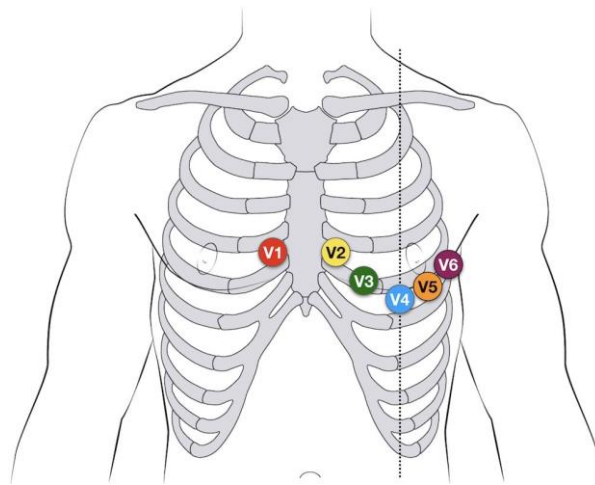
Elektrokardiografija (EKG) je neinvazivna, dijagnostička metoda koja se u medicini koristi za snimanje električne aktivnosti srca. Njome se bilježe električni signali koje proizvode srčani mišići tijekom srčanog ciklusa, a prikazuje ih se kao valne oblike na grafu u ovisnosti o vremenu. Primarna funkcija mu je otkrivanje i dijagnosticiranje različitih srčanih stanja. Električna aktivnost se mjeri pomoću elektroda koje se postavljaju na kožu pacijenta na određenim točkama oko prsnog koša i udova. Standardni 12-kanalni EKG uključuje 10 elektroda, po jedna na svakom udu i šest na prsima. Lokacije elektroda na udovima vidljive su na slici 2.6, dok su lokacije na prsima vidljive na slici 2.7.



**Slika 2.6** Lokacije elektroda na ekstremitetima [12]



Elektrode na prsima detektiraju električne impulse generirane depolarizacijom i repolarizacijom atrija i ventrikula, a svaka od njih bilježi električnu aktivnost iz specifičnog dijela srca: V1 elektroda se postavlja u četvrti međurebreni prostor desno od prsne kosti, V2 u četvrti međurebreni prostor lijevo od prsne kosti, V3 na sredinu između V2 i V4, V4 u peti međurebreni prostor na lijevoj srednjoj klavikularnoj liniji, V5 u peti međurebreni prostor na prednjoj aksilarnoj liniji, a V6 u peti međurebreni prostor na srednjoj aksilarnoj liniji. Aksilarna linija je anatomski referentna linija koja prolazi duž lateralne strane tijela, počevši od aksile, odnosno, pazuha. Za postavljanje elektroda koristi se prednja aksilarna linija koja prolazi uzduž prednjeg ruba aksile, duž vanjske strane prsnog mišića.



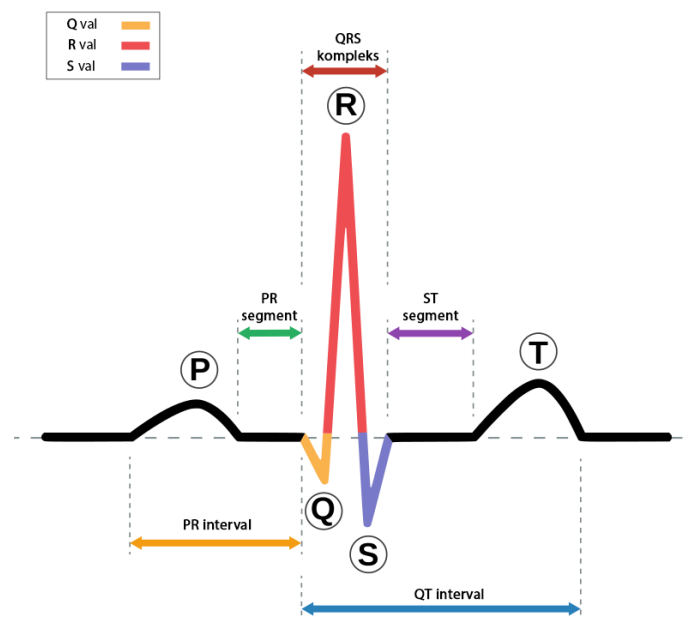
**Slika 2.7** Lokacije elektroda na prsima [13]

Aritmije, kao i strukturne abnormalnosti, moguće je identificirati karakterističnim promjenama u EKG valnom obliku. EKG-ovi se, osim za dijagnozu, koriste i za praćenje pacijenata s kroničnim srčanim bolestima, vođenje odluka o liječenju i procjenu učinkovitosti intervencija.

Postoje tri vrste EKG-a: odmarajući EKG, stresni EKG i ambulantni EKG. Odmarajući EKG se izvodi dok pacijent leži u opuštenom stanju i pruža osnovni zapis električne aktivnosti srca. Stresni EKG bilježi aktivnost srca tijekom fizičkog napora kako bi se procijenilo kako srce reagira na povećano opterećenje i identificiralo stanja kao što su aritmije inducirane vježbanjem ili koronarna bolest srca. Ambulantni EKG bilježi električnu aktivnost srca tijekom 24 sata ili dulje dok pacijent obavlja svoje svakodnevne aktivnosti. Koristan je za dijagnosticiranje povremenih aritmija ili drugih stanja koja se možda neće otkriti tijekom kratkog odmarajućeg EKG-a, a izvodi se s pomoću Holterovog monitora.

## 2.4. Karakteristike EKG-a

EKG se sastoji od karakterističnih točaka i intervala koji opisuju različite faze nastajanja otkucaja u srcu, spomenutih u poglavlju 2.2. P val predstavlja depolarizaciju atrija koja inicira kontrakciju atrija, trajanje mu je između 0.08 i 0.10 sekundi, a prosječna amplituda obično se kreće između 0.05 mV i 0.25 mV. PR segment se proteže od kraja P vala do početka QRS kompleksa, a predstavlja vrijeme potrebno za prolazak električnog impulsa kroz AV čvor, Hisov snop i Purkinjeova vlakna do mišića ventrikula. Obično traje između 0.05 i 0.12 sekundi. PR interval obuhvaća vrijeme od početka P vala do početka QRS kompleksa, a predstavlja vrijeme potrebno za provođenje električnog impulsa od atrija do ventrikula preko atrioventrikularnog čvora. Trajanje mu je obično između 0.12 i 0.20 sekundi. QRS kompleks predstavlja brzu depolarizaciju ventrikula, koja prethodi kontrakciji istih. Normalno trajanje je između 0.06 i 0.10 sekundi, dok je amplituda R vala obično je između 0.5 mV i 2.5 mV, Q vala ispod 0.1 mV, a S vala ispod 2 mV. T val predstavlja repolarizaciju ventrikula, a trajanje mu je generalno između 0.10 i 0.25 sekundi. QT interval se proteže od početka QRS kompleksa do kraja T vala, a predstavlja ukupno vrijeme potrebno za depolarizaciju i repolarizaciju ventrikula. Trajanje mu je obično manje od 0.44 sekunde. ST segment se proteže od kraja S vala do početka T vala, a predstavlja period kada su ventrikuli depolarizirani, to jest period kad nisu ni pozitivni ni negativni. Traje između 0.08 i 0.12 sekundi. Navedene karakteristične točke vidljive su na slici 2.8.

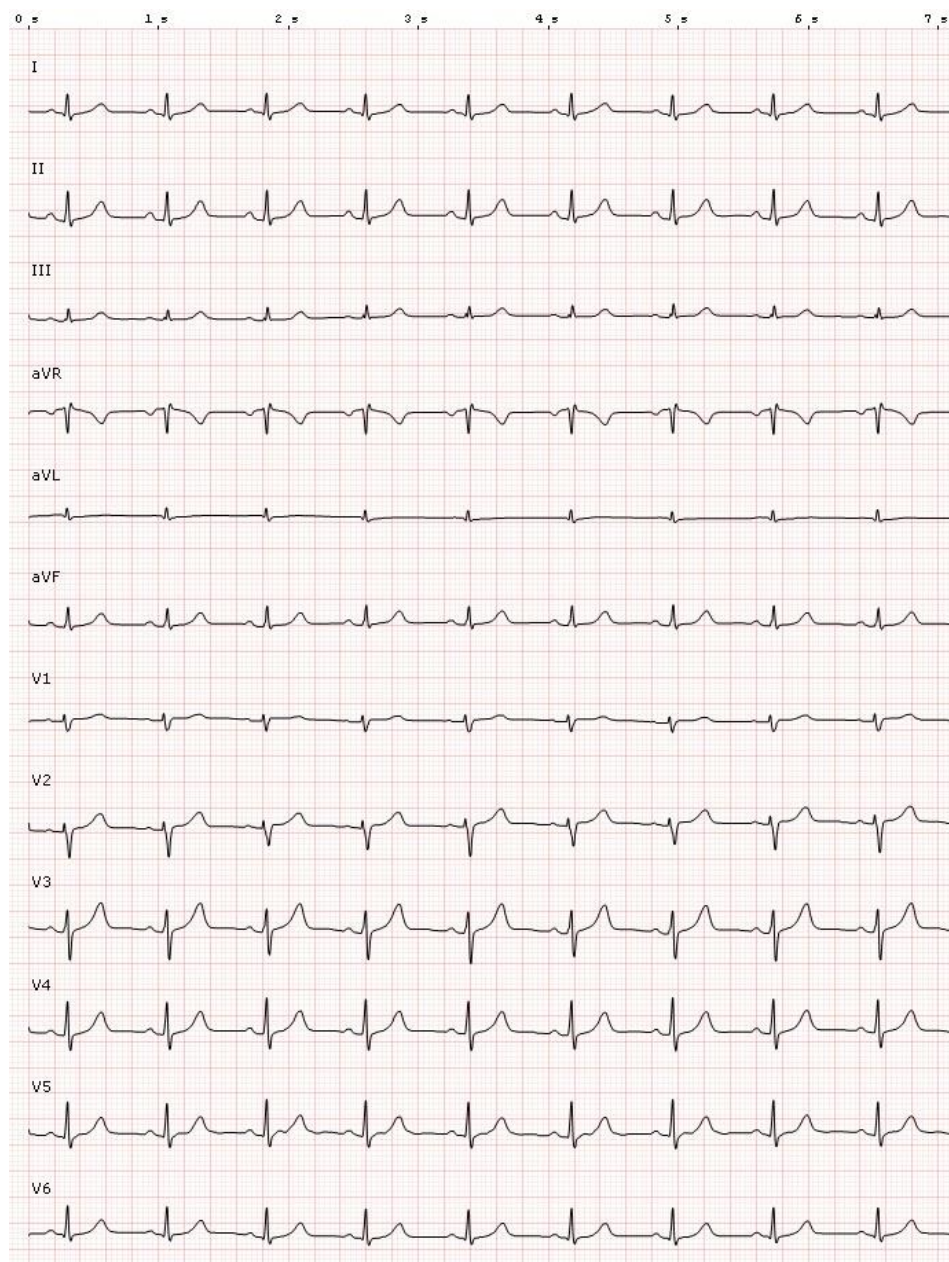


Slika 2.8 Idealni otkucaj na EKG-u [14]



## 2.5. Elektrokardiograf

Standardni EKG grafički papir, prikazan na slici 2.9, kreće se brzinom od 25 milimetara u sekundi, pri čemu svaki mali kvadrat predstavlja 0.04 sekunde, a svaki veliki kvadrat predstavlja 0.2 sekunde. Napon se mjeri na vertikalnoj osi, pri čemu svaki mali kvadrat predstavlja 0.1 milivolt. Analizom grafa i njegovih karakteristika moguće je otkriti abnormalnosti u funkciji srca.



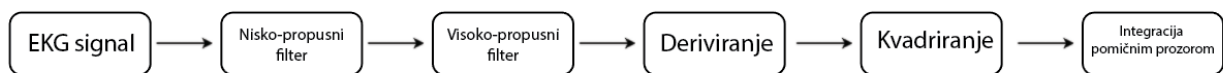
**Slika 2.9** Prikaz normalnog EKG-a [15]

### 3. Algoritmi za detekciju QRS kompleksa

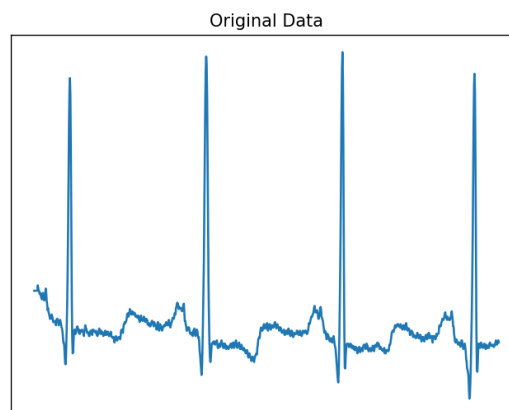
U ovom poglavlju opisane su različite metode i tehnike za prepoznavanje QRS kompleksa u EKG signalima, te njihove prednosti i nedostatci.

#### 3.1. Pan-Tompkinsov Algoritam

Pan-Tompkinsov algoritam je jedan od najpoznatijih i najčešće korištenih algoritama za detekciju QRS kompleksa. Kombinira digitalno filtriranje, deriviranje, kvadriranje i integraciju pomičnim prozorom sa svrhom pojačanja signala i smanjenja šuma. Algoritam su predložili Jiapu Pan i Willis J. Tompkins 1985. godine u časopisu *IEEE* [16]. Koraci algoritma vidljivi su na slici 3.1.



Slika 3.1 Koraci Pan-Tompkins algoritma za predobradu podataka



Slika 3.2 Izvorni signal

Prvi korak algoritma je primjena filtra propusnog pojasa na originalni signal koji je vidljiv na slici 3.2. Sirovi EKG signal je često kontaminiran šumom od mišićnih kontrakcija, pomicanja elektroda i drugih artefakata. Obično se primjenjuje filter u rasponu od 5 do 15  $Hz$ , a služi za uklanjanje visokofrekventnog šuma i baznog lutanja, pri tom čuvajući frekvencijske komponente relevantne za QRS kompleks. Prijenosna funkcija niskopropusnog filtera je

$$H(z) = \frac{(1-z^{-6})^2}{(1-z^{-1})^2}, \quad (3-1)$$

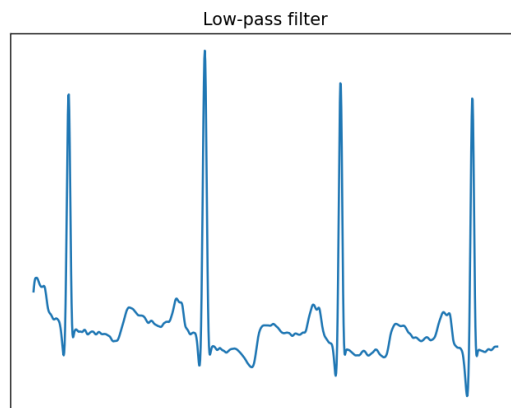
amplitudni odziv je

$$|H(\omega T)| = \frac{\sin^2(3\omega T)}{\sin^2\left(\frac{\omega T}{2}\right)}, \quad (3-2)$$

gdje je  $z$  kompleksna varijabla koja se koristi u  $z$ -transformaciji,  $\omega$  predstavlja frekvenciju u radianima u odnosu na uzorkovnu frekvenciju, a  $T$  period uzorkovanja. Diferencijalna jednadžba filtera je

$$y(nT) = 2y(nT - T) - y(nT - 2T) + x(nT) - 2x(nT - 6T) + x(nT - 12T), \quad (3-3)$$

gdje je granična frekvencija oko 11 Hz, a pojačanje 36. Filter uzrokuje kašnjenje od šest uzoraka. Izgled signala nakon primjene filtra vidljiv je na slici 3.3.



**Slika 3.3** Signal nakon nisko-pojasnog filtriranja

Prijenosna funkcija visoko-propusnog filtra je predstavljena sljedećom jednadžbom

$$H(z) = \frac{(-1+32z^{-16}+z^{-32})}{(1+z^{-1})}, \quad (3-4)$$

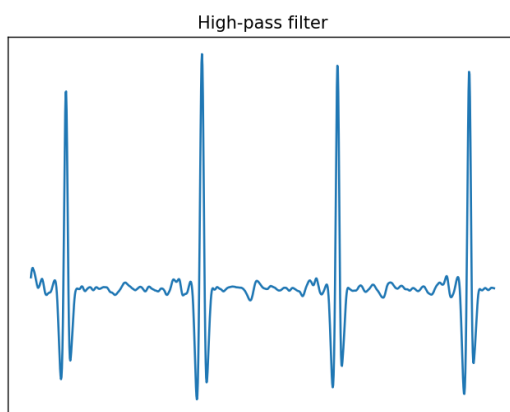
amplitudni odziv je

$$|H(\omega T)| = \frac{[256+\sin^2(16\omega T)]^{\frac{1}{2}}}{\cos\left(\frac{\omega T}{2}\right)}, \quad (3-5)$$

a diferencijalna jednadžba je

$$y(nT) = 32x(nT - 16T) - [y(nT - T) + x(nT) - x(nT - 32T)]. \quad (3-6)$$

Granična frekvencija je oko 5 Hz, a pojačanje 32. Filter uzrokuje kašnjenje od šesnaest uzoraka. Rezultantni signal vidljiv je na slici 3.4.



**Slika 3.4** Signal nakon visoko-pojasnog filtera

Nakon filtriranja propusnim pojasom, signal se podvrgava procesu derivacije. Derivacija naglašava visoke nagibe povezane s QRS kompleksom, ističući područja brze promjene signala. Koristi se derivacija u pet točaka s prijenosnom funkcijom

$$H(z) = \left(\frac{1}{8} T\right) (-z^{-2} - 2z^{-1} + 2z^1 + z^2), \quad (3-7)$$

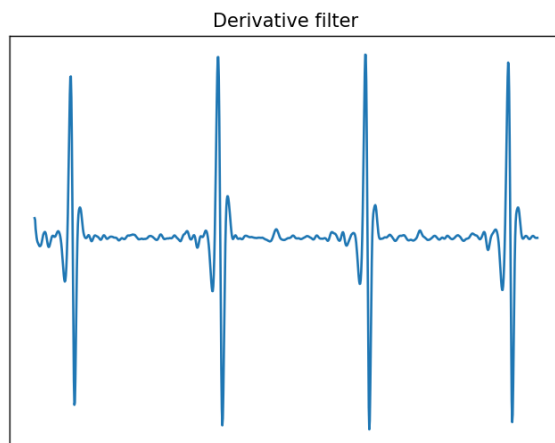
amplitudni odziv je

$$|H(\omega T)| = \left(\frac{1}{4} T\right) [\sin(2\omega T) + 2 \sin(\omega T)], \quad (3-8)$$

a diferencijalna jednačina

$$y(nT) = \left(\frac{1}{8} T\right) [-x(nT - 2T) - 2x(nT - T) + 2x(nT + T) + x(nT + 2T)]. \quad (3-9)$$

Filter uzrokuje kašnjenje od dva uzorka. Signal je prikazan na slici 3.5.

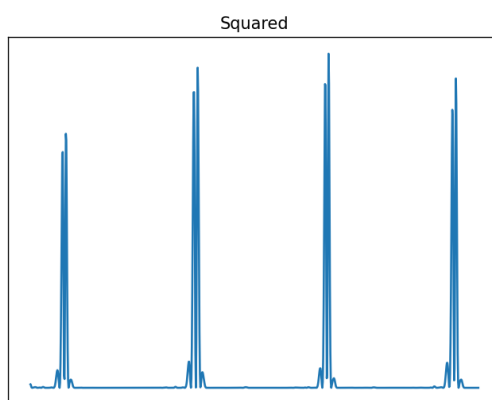


**Slika 3.5** Signal nakon derivacije

Sljedeći korak je kvadriranje kojim sve negativne vrijednosti postaju pozitivne i naglašavaju se visoke frekvencije koje su većinom važne za detekciju. Jednadžba operacije je

$$y(nT) = [x(nT)]^2 \quad (3-10)$$

Izgled kvadriranog signala prikazan je na slici 3.6.



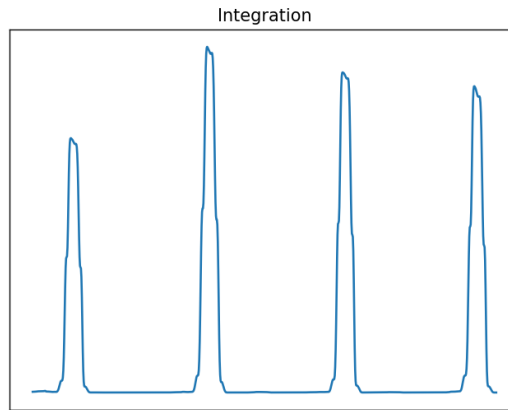
**Slika 3.6** Signal nakon kvadriranja

Nakon kvadriranja, primjenjuje se integracija pomičnim prozorom. Svrha ovoga je dobivanje informacija o značajkama valnog oblika uz nagib R vala. Računa se pomoću

$$y(nT) = \left(\frac{1}{N}\right) [x(nT - (N - 1) T) + x(nT - (N - 2) T) + \dots + x(nT)] \quad (3-11)$$

gdje je  $N$  širina integracijskog prozora. Optimalna širina prozora je jednaka najširem QRS kompleksu. Ako je prozor preširok, QRS kompleks i T val će biti spojeni. Ako je preuzak,

uzrokovat će lažne vrhove. Optimalna širina je 0.15 sekundi. Rezultantni signal vidljiv je na slici 3.7.



**Slika 3.7** Signal nakon primjene integracije pomičnim prozorom

Na dobivenom signalu se zatim traže vrhovi uz pomoć uvođenja praga (*engl. thresholding*). Za traženje vrhova koriste se dva praga: veći koji služi za prvotnu analizu signala i potragu kompleksa, a niži služi za pretragu promašenih kompleksa. Pragovi se primjenjuju na integrirani signal.

$$SPKI = 0.125 PEAKI + 0.875 SPKI \text{ (ako je PEAKI dio signala)} \quad (3-12)$$

$$NPKI = 0.125 PEAKI + 0.875 NPKI \text{ (ako je PEAKI dio šuma)} \quad (3-13)$$

$$THRESHOLD II = NPKI + 0.25 (SPKI - NPKI) \quad (3-14)$$

$$THRESHOLD I2 = 0.5 THRESHOLD II \quad (3-15)$$

gdje je SPKI procijenjena vrijednost signala u integriranom signalu, NPKI procijenjena vrijednost šuma u integriranom signalu, a PEAKI trenutni vrh u integriranom signalu. THRESHOLD II predstavlja procijenjenu vrijednost maksimuma u integriranom signalu, te se koristi za potvrđivanje vrha, a THRESHOLD I2 procijenjenu vrijednost maksimuma u integriranom signalu za potvrđivanje vrha u pretrazi promašenih vrhova.

Kada je QRS kompleks detektiran drugim pragom THRESHOLD I2, primjenjuje se

$$SPKI = 0.25 PEAKI + 0.75 SPKI. \quad (3-16)$$

Kada je vrh detektiran, potrebno je odrediti je li on dio signala ili je šum. Da bi bio klasificiran kao značajan, vrijednost mu mora biti veća od THRESHOLD I1 u prvotnom prijelazu ili veća od THRESHOLD I2 ako je u pitanju pretraga promašenih vrhova. Zasebni pragovi primjenjuju se na signal filtriran propusnim pojasom.

$$\text{SPKF} = 0.125 \text{ PEAKF} + 0.875 \text{ SPKF} \text{ (ako je PEAKF dio signala)} \quad (3-17)$$

$$\text{NPKF} = 0.125 \text{ PEAKF} + 0.875 \text{ NPKF} \text{ (ako je PEAKF dio šuma)} \quad (3-18)$$

$$\text{THRESHOLD F1} = \text{NPKF} + 0.25 (\text{SPKF} - \text{NPKF}) \quad (3-19)$$

$$\text{THRESHOLD F2} = 0.5 \text{ THRESHOLD F1} \quad (3-20)$$

gdje je SPKF procijenjena vrijednost signala nakon primjene propusnog filtera, NPKF procijenjena vrijednost šuma nakon primjene propusnog filtera, a PEAKF trenutni vrh u filtriranom signalu. THRESHOLD F1 predstavlja procijenjenu vrijednost maksimuma u filtriranom signalu, te se koristi za potvrđivanje vrha, a THRESHOLD F2 procijenjenu vrijednost maksimuma u filtriranom signalu za potvrđivanje vrha u pretrazi promašenih vrhova.

Kada je QRS kompleks detektiran drugim pragom THRESHOLD F2, primjenjuje se

$$\text{SPKF} = 0.25 \text{ PEAKF} + 0.75 \text{ SPKF}. \quad (3-21)$$

Za neregularni ritam otkucaja, vrijednosti pragova se prepolovljuju, čime se povećava osjetljivost i smanjuje vjerojatnost za lažno negativne vrhove. Vrh je dio QRS kompleksa ako je potvrđeni vrh u integriranom signalu i filtriranom signalu, odnosno, zadovoljava oba skupa navedenih pragova.

Kako bi se poboljšala točnost algoritma, u obzir se uzimaju i nedavni otkucaji i razmak između R vrhova (RR interval). Za to se uzimaju dvije vrijednosti: prosjek osam zadnjih RR intervala (*RR AVERAGE1*) i osam zadnjih RR intervala koji zadovoljavaju određeni uvjet (*RR AVERAGE2*):

$$\text{RR LOW LIMIT} = 92\% \text{ RR AVERAGE2} \quad (3-22)$$

$$\text{RR HIGH LIMIT} = 116\% \text{ RR AVERAGE2} \quad (3-23)$$

$$\text{RR MISSED LIMIT} = 166\% \text{ RR AVERAGE2} \quad (3-24)$$

Dakle, RR se nalazi između *RR LOW LIMIT* i *RR HIGH LIMIT*. Ako QRS kompleks nije pronađen u zadanom intervalu *RR MISSED LIMIT*, maksimum koji se nalazi u intervalu

$[RR\ LOW\ LIMIT, RR\ HIGH\ LIMIT]$  proglašava se vrhom QRS kompleksa. Ako se svaki od osam prethodnih RR intervala nalazi u intervalu  $[RR\ LOW\ LIMIT, RR\ HIGH\ LIMIT]$ , može se zaključiti da se radi o normalnom ritmu otkucaja srca. Kada je RR interval kraći od 0.36 sekunde i veći od 0.2 sekunde, provjerava se je li nagib trenutnog vala upola manji od vala prethodnog QRS kompleksa. Ako je navedeni uvjet ispunjen, val se proglašava T valom. U suprotnom se smatra QRS kompleksom.

Algoritam je popularan zbog svoje točnosti, za koju je zaslužno njegovo dinamičko prilagođavanje signalu, te zbog jednostavne implementacije.

### 3.2. Transformacija po valićima

Transformacija po valićima (*engl.* wavelet transform [17] i [18]) omogućava analizu signala na različitim skalama ili frekvencijama. Koristi kratke, oscilirajuće funkcije, valiče (*engl.* wavelets), za analizu lokaliziranih promjena u signalu, a korisna je za analizu nestacionarnih signala. Waveleti su funkcije koje su lokalizirane u vremenu i frekvenciji. Postoje dva osnovna tipa transformacije: kontinuirana transformacija (*engl.* Continuous Wavelet Transform) i diskretna transformacija (*engl.* Discrete Wavelet Transform).

CWT se koristi za analizu signala na različitim skalama i vremenskim pozicijama. Definiira se kao:

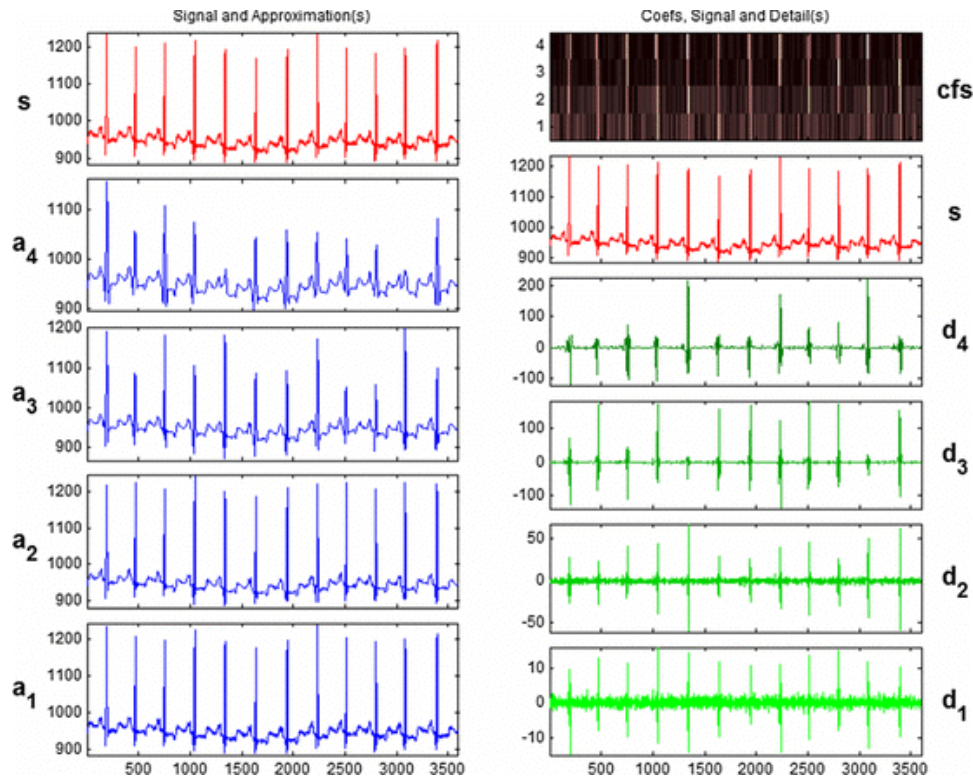
$$W(a, b) = \frac{1}{\sqrt{|a|}} \int_{-\infty}^{+\infty} x(t) \psi^*\left(\frac{t-b}{a}\right) dt \quad (3-25)$$

gdje je  $x(t)$  originalni signal,  $\psi$  funkcija valića,  $a$  skala,  $b$  vremenska pozicija, i  $\psi^*$  kompleksno konjugirana funkcija valića. Promjenom skale  $a$  i pomicanjem kroz vrijeme  $b$ , omogućava detaljnu vremensko-frekvencijsku analizu signala. Također, koristi diskretne skalne i vremenske korake za analizu signala razlažući signal na niz wavelet koeficijenata, koji predstavljaju signal na različitim skalama i rezolucijama. Proces se često provodi pomoću niskopropusnih i visokopropusnih filtera.

Transformacija po waveletima je korisna za analizu EKG signala zbog svoje sposobnosti za izolaciju i analizu različite frekvencijske komponente koje se pojavljuju u različitim vremenskim intervalima, za što koristi R val, koji ima najveću amplitudu u QRS kompleksu. On se detektira korištenjem četveroslojne diskretne transformacije po valićima putem binarnog „spline“ filtra za dobivanje skala  $S1$ ,  $S2$ ,  $S3$  i  $S4$ , na kojoj se R val naviše razlikuje ili ističe u odnosu na šum.



Skale su prikazane na slici 3.8 označeni s  $a_4$ - $a_1$ . Zatim se postavlja prag na  $S_4$  za pronalaženje ekstrema koji sadrže R vrh i njegovu točku presjeka s nulom. Ekstremi Q i S valova su manji i detektiraju se na  $S_1$  skali, korištenjem tri ekstrema prije i poslije R vrha.



Slika 3.8 Wavelet dekompozicija EKG signala u četiri razine [19]

### 3.3. Hamilton-Tompkinsov Algoritam

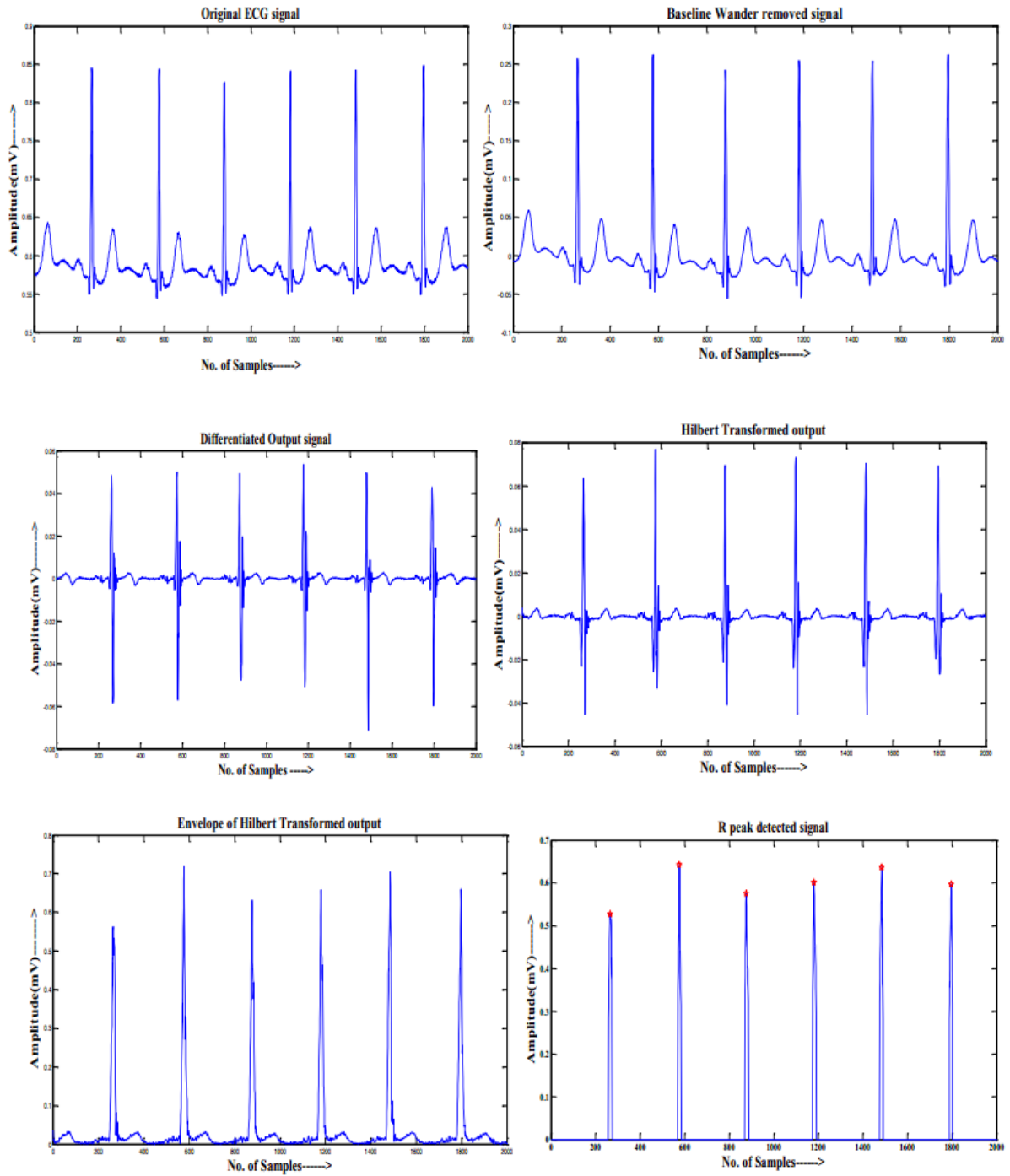
Hamilton-Tompkinsov [20] algoritam predstavlja poboljšanu verziju prethodno opisanog Pan-Tompkinsovog algoritma. Fokusira se na povećanje točnosti detekcije uz pomoć prilagodljivog filtriranja i dinamičkog podešavanje praga. S Pan-Tompkins algoritmom dijeli fazu predobrade signala, razlikujući se jedino u fazi donošenja odluka za koju koristi dinamički prag, aritmetičku sredinu, medijan i iterativnu vršnu razinu. Medijan se koristi kao najbolji procjenitelj točnosti detekcije i smanjenja lažno pozitivnih rezultata. Iako algoritam pruža visoku točnost i robusnost, najveći nedostatak mu je složenost i potreba za računalnim resursima.

### 3.4. Hilbert-Huang transformacija

Prema [21], Hilbert-Huang transformacija omogućuje detaljno razlaganje kompleksnih signala na jednostavnije komponente. Razvio ju je Norden E. Huang i njegovi kolege, a sastoji se od empirijske dekompozicijske metode (*EMD*) i Hilbertove spektralne analize (*HSA*).

Prva faza koristi iterativni proces za dekompoziciju originalnog signala u inherentne modove koji se nazivaju *IMF*-ovi (*engl.* intrinsic mode functions). Oni su oscilatorne komponente koje zadovoljavaju specifične kriterije simetričnosti i jedinstvenosti. Proces započinje identifikacijom svih lokalnih ekstrema u signalu, a zatim se provodi interpolacija kako bi se dobile gornje i donje „omotnice“ signala, to jest spoj svih maksimuma i minimuma signala. Srednja vrijednost ovih „omotnica“ se oduzima od originalnog signala kako bi se dobila prva aproksimacija *IMF-a*. Ovaj proces se ponavlja dok se ne zadovolje određeni kriteriji, a preostali signal, nakon što su svi *IMF-ovi* izdvojeni, predstavlja rezidualnu komponentu koja obično pripada originalnom signalu.

U sljedećem koraku se na signal primjenjuje Hilbertova transformacija. Ona svakom *IMF-u* pridružuje analitičku reprezentaciju u kompleksnom obliku, čime se omogućava računanje trenutne amplitude i frekvencije svakog *IMF-a* kroz vrijeme. Rezultirajući Hilbertov spektar pruža trodimenzionalni prikaz energije signala raspoređene po frekvenciji i vremenu, čime se omogućuje detaljna analiza frekvencijskih promjena signala tijekom vremena. Zatim je *IMF-ove* moguće dalje filtrirati, te se mogu koristiti različite metode analize kako bi se izdvojili dijelovi signala koji sadrže tražene karakteristične točke. Ovim fazama se postiže poboljšanje točnosti detekcije tako što se omogućava preciznija lokalizacija i uklanjanje šuma. Faze ove transformacije prikazane su na slici 3.9. Iz originalnog signala uklonjen je šum, te je isti diferenciran, a na njemu primjenjena Hilbert-Huang transformacija. Zatim su mu određene „omotnice“ na temelju kojih su detektirani R vrhovi.



**Slika 3.9** Korištenje Hilbert-Huangove transformacije za detekciju R vrhova [21]

### 3.5. Podudarno filtriranje

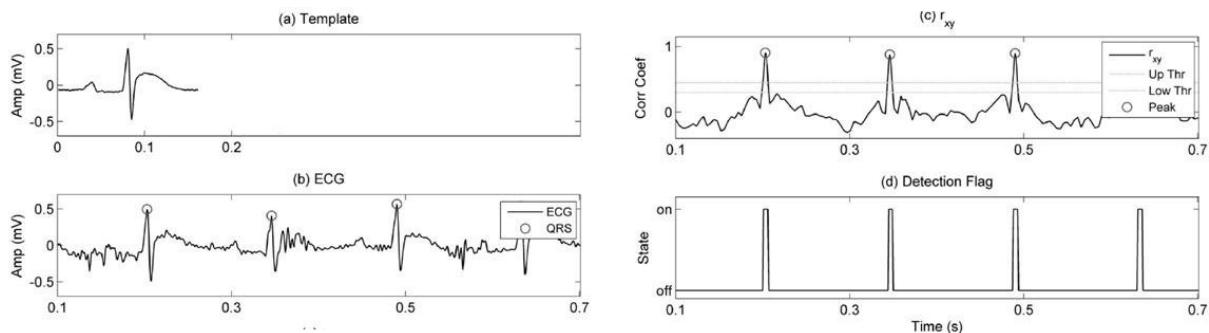
Podudarno filtriranje (*engl.* match filtering) [22] se koristi za identificiranje prisutnosti željenog oblika signala uz prisutnost šuma. Za detekciju QRS kompleksa u EKG-u koristi se njegov idealni oblik kao predložak za pretraživanje signala. Prvi korak je definiranje iste. Ona može biti empirijski izvedena ili teorijski definirana na temelju karakteristika kompleksa. Filter se konstruira tako da se oblik filtriranja podudara s inverzijom predložka QRS kompleksa, odnosno, podudarni filter  $h(t)$  je kompleksni konjugat signala  $x(t)$  pomaknutog unazad za  $t$

$$h(t) = x^*(-t) \quad (3-26)$$

Na signal se zatim primjenjuje konvolucija korištenjem dobivenog filtra  $h(t)$

$$y(t) = x(t) * h(t) \quad (3-27)$$

Dobiveni izlaz predstavlja mjeru podudaranja ulaznog signala i predložka, te se na njemu radi detekcija vrhova. Algoritam nudi visoku preciznost, otpornost na šum i sposobnost rada u nestacionarnim uvjetima, ali implementacija zahtijeva preciznu definiciju predložka i računalne resurse za izvođenje konvolucije. Na slici 3.10 vidljivi su predložak (a), originalni signal (b) i mjesta podudaranja s predložkom (d).



Slika 3.10 Koraci podudarnog filtriranja [23]

## 4. Implementacija

Za potrebe ovog rada, odabrana je predobrada signala opisana u Pan-Tompkins algoritmu, sa svrhom smanjenja utjecaja šuma. Za detekciju QRS kompleksa odabrana je kombinacija traženja maksimuma pomičnog prozora i metoda praga, metode koje će biti objašnjene u sljedećem poglavlju.

## 4.1. Korištene tehnologije

Za izradu aplikacije korišten je programski jezik Python [24] i njegove popratne biblioteke [25], [26]. Python je viši programski jezik kojeg je razvio Guido van Rossum 1991. godine. Podržava proceduralno, objektno orijentirano i funkcionalno programiranje. Ima bogatu standardnu biblioteku i veliku zajednicu korisnika, što ga čini pogodnim za primjenu u web razvoju, obradi podataka, umjetnoj inteligenciji, automatizaciji i drugim poljima. Jednostavna sintaksa omogućuje pisanje čistog i konciznog koda, odnosno, kratkog i jasnog koda, što olakšava razvoj i održavanje aplikacija. Za obradu signala, izradu sučelja i crtanje grafova korištene su biblioteke navedene u tablici 4.1.

Tablica 4.1 Korištene biblioteke

Zadatak	Biblioteka
Učitavanje i rad s CSV datotekama	csv
Primjena digitalnih filtera na signal	scipy.signal.lfilter
Rad s n-dimenzionalnim poljima i funkcije za efikasnu obradu velikih skupova podataka	numpy
Brojanje lažno negativnih i lažno pozitivnih kompleksa te analiza njihovih tipova	collections.Counter
Izrada korisničkog sučelja	tkinter
Dohvaćanje putanje do datoteke	os
Crtanje i rad s grafovima	matplotlib.pyplot

## 4.2. Učitavanje podataka

Za učitavanje podataka korištena je funkcija *load\_data* čiji je kod prikazan na slici 4.1. Funkcija na temelju predane putanje učitava zapis EKG signala u CSV formatu.

```

def load_data(f_name):
    with open(f_name, 'r') as file:
        next(file)
        reader = csv.reader(file)
        data = np.array([float(row[1]) for row in reader])

    return data

```

Slika 4.1 Funkcija `load_data`

Funkcija `load_annotation`, kod vidljiv na slici 4.2, koristi se za učitavanje anotacija, u slučaju kad su one dostupne. Anotacije se koriste za provjeru točnosti aplikacije, ali nisu potrebne za normalni rad aplikacije. Funkcija učitava anotacije u tekstualnom formatu, primjer dva retka anotacija vidljiv je na slici 4.3, i iz njih izbacuje one koje nisu vezane uz detekciju QRS kompleksa. Anotacije se sastoje od tri stupca: vremena događaja, broja uzorka i tipa događaja. Izbačene anotacije prikazane su u tablici 4.2.

```

def load_data_annotation(f_name):
    lines = []
    types = []
    first_line_read = False

    with open(f_name, 'r') as f:
        for line in f:
            if line:
                l = ' '.join((line.strip("\r\n\t")).split())
                t = l.split()[2]
                l = l.split()[1]
                if (l.isnumeric() and (t not in ["+", "~", "x", "|", "[", "]", "\"] or not first_line_read)):
                    lines.append(int(l))
                    types.append(t)
                    first_line_read = True
    return lines[1:], types[1:]

```

Slika 4.2 Funkcija `load_annotation`

Time	Sample #	Type
0:00.050	18	+
0:00.214	77	N

Slika 4.3 Izgled anotacija

**Tablica 4.2** Izbačene anotacije

Anotacija	Objašnjenje
+	Artefakt - Označava artefakte u EKG signalu koji nisu povezani s fiziološkim procesima.
~	Promjena ritma - Indikacija promjene u srčanom ritmu.
x	Neprovedeni P-val
[	Početak epizode - Označava početak specifične epizode ili događaja u EKG signalu.
]	Kraj epizode - Označava kraj specifične epizode ili događaja u EKG signalu.
"	Nenormalni ritam - Označava prisutnost nenormalnih ritmova ili srčanih aktivnosti.
	Izolirani artefakt sličan QRS-u

Anotacije koje su vezane uz detekciju QRS kompleksa prikazane su u tablici 4.3.

**Tablica 4.3** Zadržane anotacije

Anotacija	Objašnjenje
N	Normalni otkucaj
L	Otkucaj bloka lijeve grane snopa
R	Otkucaj bloka desne grane snopa
V	Preuranjena kontrakcija ventrikula
/	Otkucaj uzrokovan „pacemakerom“
A	Preuranjeni otkucaj atrijski
f	Spoj otkucaja uzrokovanog „pacemakerom“ i normalnog otkucaja
F	Spoj ventrikularnog i normalnog otkucaja
!	Ventrikularna aritmija
j	Kasni atrioventrikularni otkucaj
a	Aberirani preuranjeni otkucaj atrijski
E	Ventrikularni izlazni otkucaj

J	Preuranjeni atrioventrikularni otkucaj
Q	Ne-klasificirani otkucaj
e	Kasni atrijalni otkucaj
S	Supraventrikularni prijevremeni otkucaj

### 4.3. Implementacija sučelja

Početno sučelje se sastoji od tri gumba: gumb za učitavanje EKG-a, gumb za učitavanje anotacija i gumb za pokretanje detekcije. Glavno sučelje, čiji kod je prikazan na 4.4, koje se prikazuje nakon pritiska na gumb bit će objašnjeno u sljedećem poglavlju.

```

root = tk.Tk()
root.title("QRS Complex Detection")
root.state('zoomed')

main_frame = tk.Frame(root)
main_frame.pack(side=tk.TOP, pady=20)

label_frame = tk.Frame(main_frame)
label_frame.pack(side=tk.TOP)

button_frame = tk.Frame(main_frame)
button_frame.pack(side=tk.TOP)

file_label = tk.Label(label_frame, text="No file loaded")
file_label.pack(pady=5)
load_file_button = tk.Button(button_frame, text="Load CSV", command=load_csv)
load_file_button.pack(side=tk.LEFT, padx=10)

annotation_label = tk.Label(label_frame, text="No annotation loaded")
annotation_label.pack(pady=5)
load_annotation_button = tk.Button(button_frame, text="Load annotation", command=load_annotation)
load_annotation_button.pack(side=tk.LEFT, padx=10)

detect_qrs_button = tk.Button(button_frame, text="Detect QRS Complex", command=detect_qrs_complex)
detect_qrs_button.pack(side=tk.LEFT, padx=10)

root.mainloop()

```

Slika 4.4 Sučelje aplikacije



## 4.4. Detekcija QRS kompleksa

Na učitani signal se primjenjuje niskopropusni i visokopropusni filter. Za potrebe ove aplikacije korišteni su filtri koji se koriste u Pan-Tompkins algoritmu, opisani u poglavlju 3.1. Kod funkcije *detect\_qrs\_complex* prikazan je na slici 4.5

```
def detect_qrs_complex():
    global data, annotation, types, high_pass_filter, possible_peaks_index, possible_peaks_value, root, false_negative, false_positive, qrs_intervals

    low_pass_filter = apply_low_pass_filter(data)
    high_pass_filter = apply_high_pass_filter(low_pass_filter)

    high_pass_filter = normalize(high_pass_filter)
    possible_peaks_index, possible_peaks_value = get_local_maxima(high_pass_filter, 200)
    starting_threshold = determin_starting_threshold(possible_peaks_index, possible_peaks_value)

    possible_peaks_index, possible_peaks_value = remove_false_positives(possible_peaks_index, possible_peaks_value, starting_threshold)

    qrs_intervals = find_qrs_interval(possible_peaks_index)

    if (annotation != None):
        false_positive, false_negative = evaluate_precision(possible_peaks_index, annotation.copy(), 15)

        print("False positives: ", len(false_positive))
        print("False negatives: ", len(false_negative))
        print("False negatives counter: ", Counter(types[annotation.index(x)] for x in false_negative))
        print("False negatives counter: ", Counter(types[annotation.index(x)] for x in false_positive))

    plot_graph()
```

Slika 4.5 Funkcija koja započinje detekciju QRS kompleksa

Filtri se primjenjuju korištenjem *lfilter* funkcije koja prima brojnik i nazivnik prijenosne funkcije filtra, zapisanih u obliku polja. Ovim se smanjuje utjecaj šuma i samim time olakšava detekcija QRS kompleksa. Kod je vidljiv na slici 4.6.

```
def apply_low_pass_filter(data):
    data = np.pad(data, [0, 6], mode='constant', constant_values=0)
    gain = 36
    numerator = [1, 0, 0, 0, 0, 0, -2, 0, 0, 0, 0, 0, 1]
    denominator = [1 * gain, -2 * gain, 1 * gain]
    return lfilter(numerator, denominator, data)[6:]

def apply_high_pass_filter(data):
    data = np.pad(data, [0, 16], mode='constant', constant_values=0)
    gain = 32
    numerator = [-1] + [0] * 15 + [32, -32] + [0] * 14 + [1]
    denominator = [1 * gain, -1 * gain]
    return lfilter(numerator, denominator, data)[16:]
```

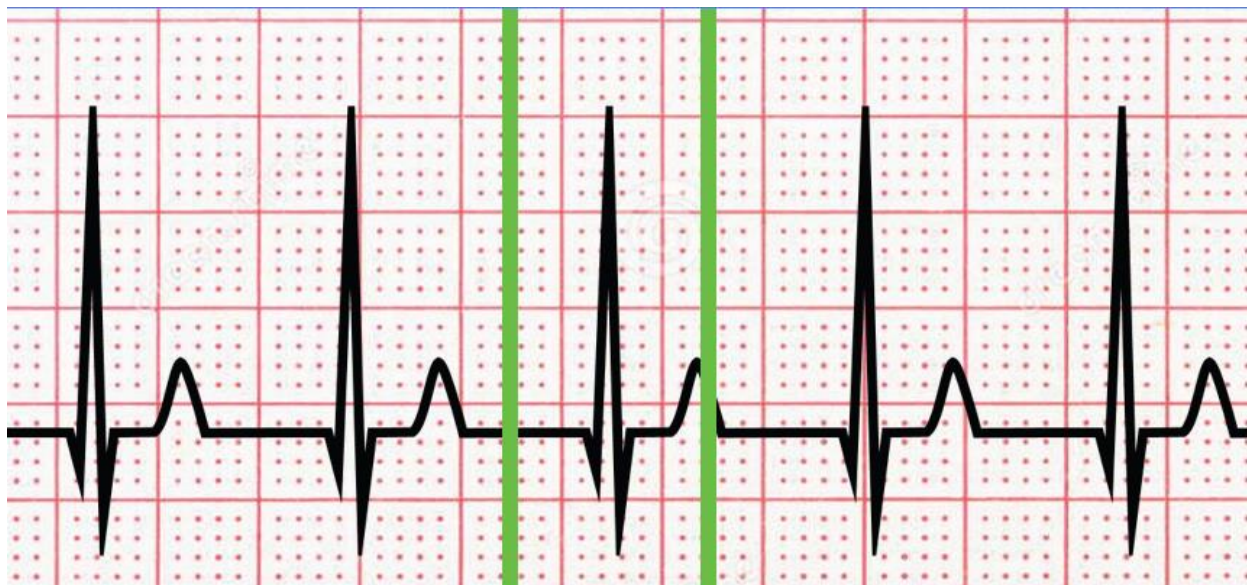
Slika 4.6 Funkcije za niskopropusni i visokopropusni filter

Dobiveni filtrirani signal se zatim normalizira, funkcijom na slici 4.7, a na njemu se zatim provodi detekcija vrhova korištenjem pomičnog prozora.

```
def normalize(data):  
    mean = np.mean(data)  
    data = data - mean  
    return data / max(abs(data))
```

**Slika 4.7** Funkcija za normalizaciju podataka

Funkcija traži maksimum koji se nalazi točno u sredini prozora širine uzorkovanih vrijednosti (oko 0.56 sekundi). Ova širina je malo dulja od trajanja prosječnog otkucaja srca, koje traje oko 0.46 sekundi, a odabrana je testiranjem. Ako se maksimum nalazi na sredini prozora, najvjerojatnije je riječ o R-vrhu, te se njegova vrijednost i lokacija pohranjuju. Primjer pomičnog prozora vidljiv je na slici 4.8, dok je kod funkcije prikazan na slici 4.9.



**Slika 4.8** Pomični prozor

```

def get_local_maxima(data, window_size):
    local_maxima_indexes = []
    local_maxima_values = []

    padded_data = np.concatenate((np.zeros(window_size - 1), data, np.zeros(window_size - 1)))
    for i in range(window_size - 1, len(padded_data)):
        window = padded_data[i - (window_size - 1):i + 1]
        possible_peak = max(window)
        possible_peak_index = np.where(window == possible_peak)[0][0]

        if (possible_peak_index == int(window_size / 2) - 1):
            local_maxima_indexes.append(i - (window_size + int(window_size / 2) - 1))
            local_maxima_values.append(possible_peak)

    return local_maxima_indexes, local_maxima_values

```

Slika 4.9 Funkcija za detekciju vrhova

Nakon detekcije R-vrhova, slijedi izbacivanje lažno pozitivnih vrhova. Prvo se određuje početna vrijednost praga, a ona se dobiva tako da se detektirani vrhovi prvo grupiraju. Vrhovi koji se pojavljuju unutar prvih 250 uzorkovanih vrijednosti se grupiraju zajedno, zatim sljedećih 250 i tako do kraja signala. Početni prag se postavlja na 28% vrijednosti medijana svih minimalnih i maksimalnih vrhova koji su pronađeni u svakoj grupi. Ovim se smanjuje utjecaj uzastopnih malih ili velikih amplituda na konačnu vrijednost praga, a samim time i smanjuje broj izbacivanja pravih vrhova (broj lažno negativnih) i broj zadržavanja lažnih (broj lažno pozitivnih). Kod ove funkcije vidljiv je na slici 4.10.

```

def determin_starting_threshold(maxima_indexes, maxima_values):
    grouped_indexes, grouped_values = split_into_groups(maxima_indexes.copy(), maxima_values.copy(), 250)
    local_extrema = []
    for g in grouped_values:
        if (len(g) != 0):
            minimal = min(g)
            if (minimal != 0):
                local_extrema.append(minimal)

            maximal = max(g)
            if (maximal != 0):
                local_extrema.append(maximal)

    median = 0.28 * np.median(local_extrema)
    return median

def split_into_groups(indexes, values, group_size):
    grouped_indexes = []
    grouped_values = []
    for i in range(group_size - 1, indexes[-1] + group_size, group_size):
        index_group = []
        value_group = []

        for p in indexes:
            if (p < i and p > i - group_size - 1):
                index_group.append(p)
            else:
                break

        num_of_items = len(index_group)
        for i in range(num_of_items):
            value_group.append(values[index_group[i]])
            del values[index_group[i]]
            del indexes[index_group[i]]

        grouped_indexes.append(index_group)
        grouped_values.append(value_group)
    return grouped_indexes, grouped_values

```

Slika 4.10 Funkcije za određivanje početne vrijednosti praga za izbacivanje lažno pozitivnih vrhova

Lažni vrhovi se izbacuju na temelju prethodno određenog početnog praga i na temelju udaljenosti između dva susjedna vrha. Vrh se smatra lažnim ako je razlika na vremenskoj osi od sljedećeg detektiranog vrha manja od 0.32 sekundi i ako je njegova amplituda manja od sljedećeg detektiranog vrha. Također, vrh se smatra lažnim ako je razlika na vremenskoj osi prethodno detektiranog vrha manja od 0.32 sekundi i ako je njegova amplituda manja od prethodnog. Ako je amplituda vrha manja od vrijednosti početnog praga, on se također smatra lažnim i izbacuje se. Vrijednost od 0.32 sekundi je odabrana jer za sve manje intervale postoji šansa da je umjesto R-vrha, detektiran T-vrh. Kod je prikazan na slici 4.11.

```

def remove_false_positives(indexes, values, fp_threshold):
    false = []
    for i in range(len(indexes)):
        if (i != len(indexes) - 1 and indexes[i] + 0.32 * sampling_rate > indexes[i + 1] and values[i] < values[i + 1]):
            false.append(indexes[i])
        if (i != 0 and indexes[i] - 0.32 * sampling_rate < indexes[i - 1] and values[i] < values[i - 1] and indexes[i] not in false):
            false.append(indexes[i])
        if (values[i] < fp_threshold and indexes[i] not in false):
            false.append(indexes[i])

    for f in false:
        index = indexes.index(f)
        indexes.remove(f)
        del values[index]
    return indexes, values

```

Slika 4.11 Funkcija za izbacivanje lažno pozitivnih vrhova

Nakon izbacivanja lažno pozitivnih vrhova, slijedi određivanje QRS kompleksa. QRS kompleks se određuje na temelju detektiranih R-vrhova. Kao što je spomenuto u poglavlju 2.4., QRS kompleks obično traje između 0.06 i 0.10 sekundi. To znači da se Q-vrh nalazi 0.05 sekundi prije R-vrha, a S-vrh 0.05 sekundi poslije. Za potrebe ove aplikacije, početak QRS kompleksa se traži unutar 0.125 sekundi, da bi se spriječilo krivo detektiranje Q i S vrhova. Funkcija *find\_qrs\_interval*, čiji je kod vidljiv na slici 4.12, traži prve minimume prije i nakon R-vrha i njih proziva Q i S vrhovima. Nakon toga, traži prvi prijašnji i sljedeći maksimum i označava ih kao početak i kraj QRS kompleksa.. Lokacije Q i S vrhova, te interval početka i kraja QRS kompleksa se zatim objedinjuje i pohranjuje u polje.

```

def find_qrs_interval(r_peak_locations):
    global high_pass_filter
    qrs_intervals = []
    for r_index in range(len(r_peak_locations)):
        interval = []

        window = high_pass_filter[max(r_peak_locations[r_index] - round(0.125 * sampling_rate), 0) : r_peak_locations[r_index]]
        minimum = min(window)
        q_peak = max(r_peak_locations[r_index] - round(0.125 * sampling_rate), 0) + np.where(window == minimum)[0][0]

        for q_index in range(q_peak, max(r_peak_locations[r_index] - round(0.125 * sampling_rate), 0), -1):
            if (high_pass_filter[q_index] >= high_pass_filter[min(q_index + 1, len(high_pass_filter) - 1)] and
                high_pass_filter[q_index] >= high_pass_filter[max(q_index - 1, 0)]):
                interval.append(q_index)
                break

        if (len(interval) != 1):
            interval.append(max(r_peak_locations[r_index] - round(0.125 * sampling_rate), 0))

        window = high_pass_filter[r_peak_locations[r_index]: min(r_peak_locations[r_index] + round(0.125 * sampling_rate), len(high_pass_filter) - 1)]
        if (len(window) == 0):
            break
        minimum = min(window)
        s_peak = r_peak_locations[r_index] + np.where(window == minimum)[0][0]

        for s_index in range(s_peak, min(r_peak_locations[r_index] + round(0.125 * sampling_rate), len(high_pass_filter) - 1)):
            if (high_pass_filter[s_index] >= high_pass_filter[s_index + 1] and high_pass_filter[s_index] >= high_pass_filter[s_index - 1]):
                interval.append(s_index)
                break

        if (len(interval) != 2):
            interval.append(min(r_peak_locations[r_index] + round(0.125 * sampling_rate), len(high_pass_filter)))

        qrs_intervals.append((interval, q_peak, r_peak_locations[r_index], s_peak))

    return qrs_intervals

```

Slika 4.12 Funkcija za pronalazak QRS intervala, Q i S vrhova.

Ako su dostupne anotacije, sljedeći korak je evaluacija točnosti algoritma. Lažno pozitivni vrhovi se još jednom provjeravaju uz toleranciju jesu li zapravo dobro detektirani. Za toleranciju se uzima prozor od 50 uzorkovanih vrijednosti, kao što je to navedeno u članku [20]. Kod za evaluaciju preciznosti prikazan je na slici 4.13.

```
def evaluate_precision(p, a, tolerance):
    p = list(set(p))
    a = list(set(a))
    peaks_not_in_annotation = [x for x in p if x not in a]
    annotation_not_in_peaks = [x for x in a if x not in p]
    peaks_within_tolerance = []

    for i in range(len(peaks_not_in_annotation)):
        peak_locations_with_tolerance = list(range(peaks_not_in_annotation[i] - tolerance, peaks_not_in_annotation[i] + tolerance + 1, 1))
        for j in peak_locations_with_tolerance:
            if j in annotation_not_in_peaks:
                annotation_not_in_peaks.remove(j)
                peaks_within_tolerance.append(peaks_not_in_annotation[i])
                break

    for t in peaks_within_tolerance:
        peaks_not_in_annotation.remove(t)
    return peaks_not_in_annotation, annotation_not_in_peaks
```

Slika 4.13 Funkcija za evaluaciju preciznosti

## 4.5. Sučelje za prikaz rezultata

Za crtanje i upravljanje grafom na kojemu su prikazani filtrirani signal i detektirani R-valovi, zajedno s Q i S valovima, koristi se funkcija *plot\_graph*, čiji kod je prikazan na slici 4.14. Osim što označava detektirane vrhove na filtriranom signalu, označava i početak i kraj QRS kompleksa, lažno pozitivne i lažno negativne vrhove. Također, omogućava korisniku pristup kontrolama za pomicanje, približavanje ili spremanje grafa i crta legendu pored grafa.

```

def plot_graph():
    global high_pass_filter, canvas, toolbar, false_negative, false_positive, annotation, types, qrs_intervals, text_frame

    if high_pass_filter is not None:
        fig, ax = plt.subplots()

        x = np.arange(len(high_pass_filter))
        ax.plot(x, high_pass_filter)

        q_peaks, r_peaks, s_peaks = zip(*[(qrs[1], qrs[2], qrs[3]) for qrs in qrs_intervals])

        ax.scatter([q for q in q_peaks if q is not None], [high_pass_filter[peak] for peak in q_peaks if peak is not None], marker='x', color='purple', label='Q-Peaks')
        ax.scatter(r_peaks, [high_pass_filter[peak] for peak in r_peaks], marker='o', color='green', label='R-Peaks')
        ax.scatter([s for s in s_peaks if s is not None], [high_pass_filter[peak] for peak in s_peaks if peak is not None], marker='+', color='blue', label='S-Peaks')

        handles, labels = plt.gca().get_legend_handles_labels()

        for l in [*[q[0][0] for q in qrs_intervals], *[s[0][1] for s in qrs_intervals]]:
            plt.axvline(l, color='lightblue', linestyle='--')

        handles.append(Line2D([0], [0], color='lightblue', linestyle='--', label='QRS Intervals'))
        labels.append('QRS Intervals')

        if (annotation):
            for i in range(len(false_negative)):
                ax.axvline(false_negative[i], color="red", linestyle='--')

            for i in range(len(false_positive)):
                ax.axvline(false_positive[i], color="purple", linestyle='--')

            if (len(false_negative) != 0):
                handles.append(Line2D([0], [0], color='red', linestyle='--', label='False Negative'))
                labels.append('False Negative')
            if (len(false_positive) != 0):
                handles.append(Line2D([0], [0], color='purple', linestyle='--', label='False Positive'))
                labels.append('False Positive')

    if not canvas:
        canvas = FigureCanvasTkAgg(fig, master=root)
        canvas.get_tk_widget().pack(side=tk.TOP, fill=tk.BOTH, expand=True)
        canvas.draw()

        text_frame = tk.Frame(root)
        text_frame.pack(side=tk.LEFT, fill=tk.Y, padx=10, pady=10)

        fp_label = tk.Label(text_frame, text=f"False Positives: {len(false_positive)}", font=('Arial', 16))
        fp_label.pack(anchor='nw')

        fn_label = tk.Label(text_frame, text=f"False Negatives: {len(false_negative)}", font=('Arial', 16))
        fn_label.pack(anchor='nw')

        toolbar = NavigationToolbar2Tk(canvas, root)
        toolbar.update()

    ax.set_xlabel('Sample Number')
    ax.set_ylabel('Amplitude')
    ax.set_title('Filtered Signal')

    plt.subplots_adjust(left=0.07, bottom=0.25, right=0.97, top=0.97)
    plt.grid(True)
    plt.xticks(rotation=45)
    plt.tight_layout()
    plt.legend(loc='upper left', bbox_to_anchor=(-0.1, 1), handles=handles, labels=labels)

```

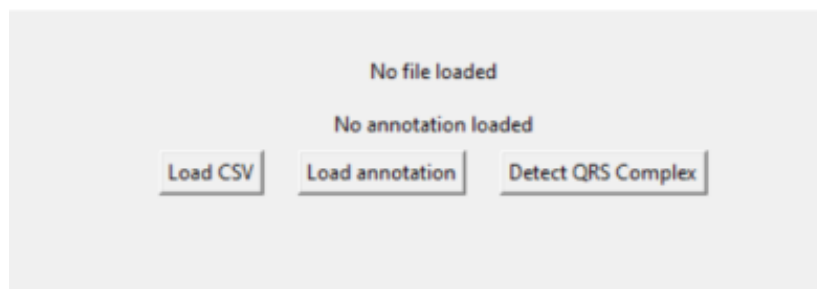
Slika 4.14 Funkcija *plot\_graph*

## 5. Prikaz rada aplikacije

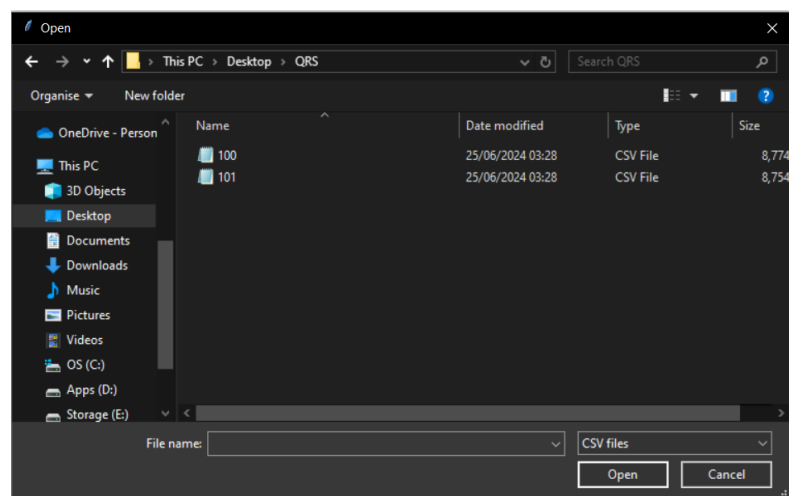
U ovom poglavlju objašnjene su funkcionalnosti aplikacije, te su iste testirane na različitim primjerima. Rezultati testova su zatim ukratko analizirani.

### 5.1. Izgled sučelja aplikacije

Kao što je spomenuto u poglavlju 4.3., početno sučelje se sastoji od 3 gumba: gumb za učitavanje EKG-a, gumb za učitavanje anotacija i gumb za pokretanje detekcije. Klikom na *Load CSV* gumb, otvara se prozor za odabir CSV datoteke koja predstavlja EKG signal. U prozoru za odabir datoteke EKG signala automatski se filtriraju .csv datoteke radi lakšeg pronalaženja odgovarajuće. Početno sučelje prikazano je na slici 5.1, dok je izgled prozora za odabir datoteke EKG signala vidljiv na slici 5.2.



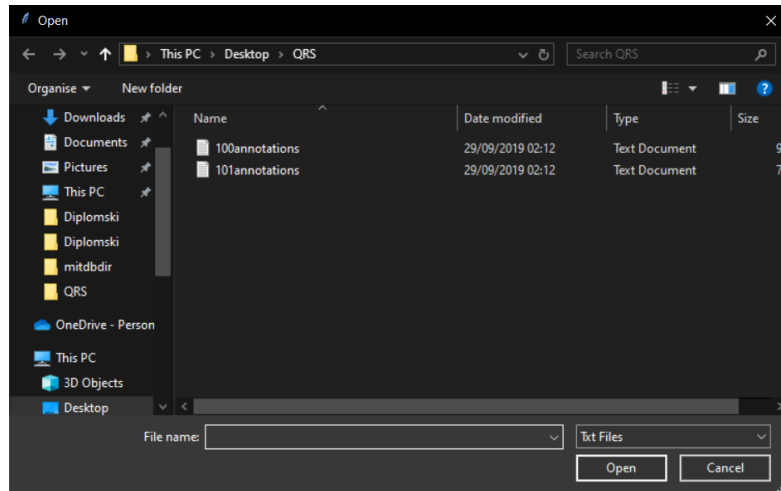
Slika 5.1 Izgled početnog sučelja



Slika 5.2 Prozor za učitavanje datoteke EKG signala



Pritiskom na gumb *Load annotation* otvara se prozor za odabir tekstualne datoteke koja predstavlja anotacije. Ovu datoteku nije potrebno učitati za normalni rad aplikacije, samo ako korisnik želi analizu točnosti algoritma. U prozoru za odabir anotacija također se automatski filtriraju *.txt* datoteke radi lakšeg pronalaženja odgovarajuće. Izgled prozora za odabir anotacija prikazan je na slici 5.3.



**Slika 5.3** Prozor za učitavanje anotacija

Nakon odabira odgovarajućih datoteka, korisnik može kliknuti na gumb *Detect QRS Complex* ako želi pronaći sve QRS komplekse u učitanoj signalu ili može ponovno odabrati datoteke, ako je učitao pogrešne. Klikom na gumb *Detect QRS Complex* u glavnom prozoru se iscrtava graf sa svim detektiranim karakterističnim točkama, vidljivo na slici 5.4.



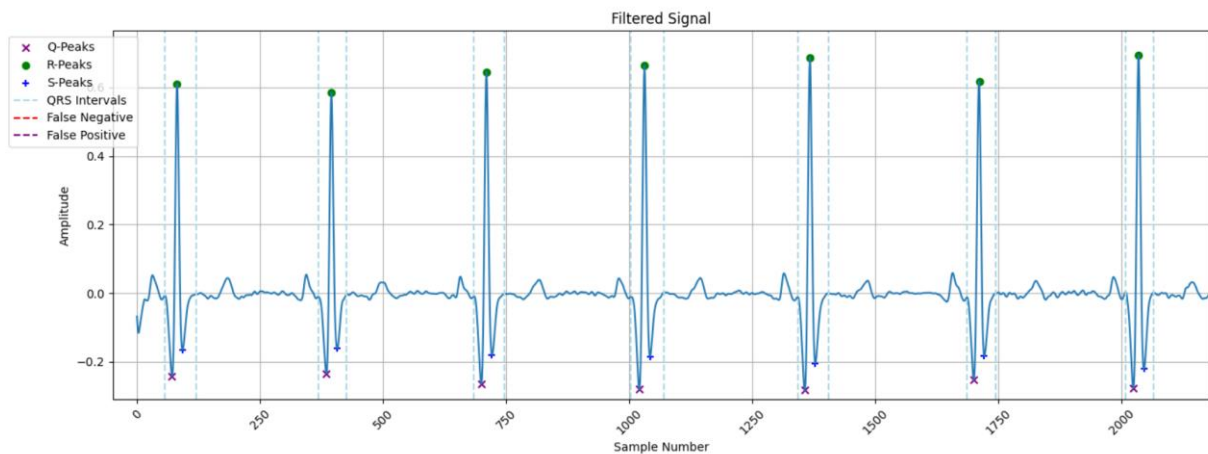
**Slika 5.4** Sučelje nakon detekcije QRS-a

Budući da signal sadrži puno uzoraka, početni graf izgleda nepregledno. Korisnik može klikom na jednu od ikona u donjem lijevom kutu odabrati alat za pomicanje po grafu, približavanje, namještati poziciju grafa i pohraniti sliku istoga. U okviru koji sadrži alate za rad s grafom također se pojavljuje broj lažno pozitivnih i negativnih vrhova, ako su učitane anotacije. Izgled alatne trake vidljiv je na slici 5.5.



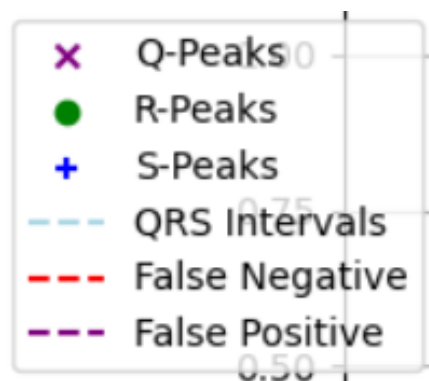
**Slika 5.5** Prozor s alatima za rad s grafom i prikaz broja lažno pozitivnih i negativnih vrhova

Klikom na ikonu povećala korisnik može približiti graf radi boljeg pregleda. Nakon povećavanja grafa, mogu se vidjeti označene karakteristike. Izgled prozora nakon povećavanja vidljiv je na slici 5.6.



Slika 5.6 Graf nakon približavanja

Na grafu su označeni Q, R, S vrhovi (na grafu označeni s X, O, i +), te početak i kraj QRS kompleksa (na grafu označeni plavom isprekidanom linijom), te lažno negativni i lažno pozitivni vrhovi (na grafu označeni crvenom i ljubičastom isprekidanom linijom). Značenje specifičnih oznaka također je moguće vidjeti na legendi, slika 5.7, koja se iscrtava u gornjoj lijevoj strani grafa.



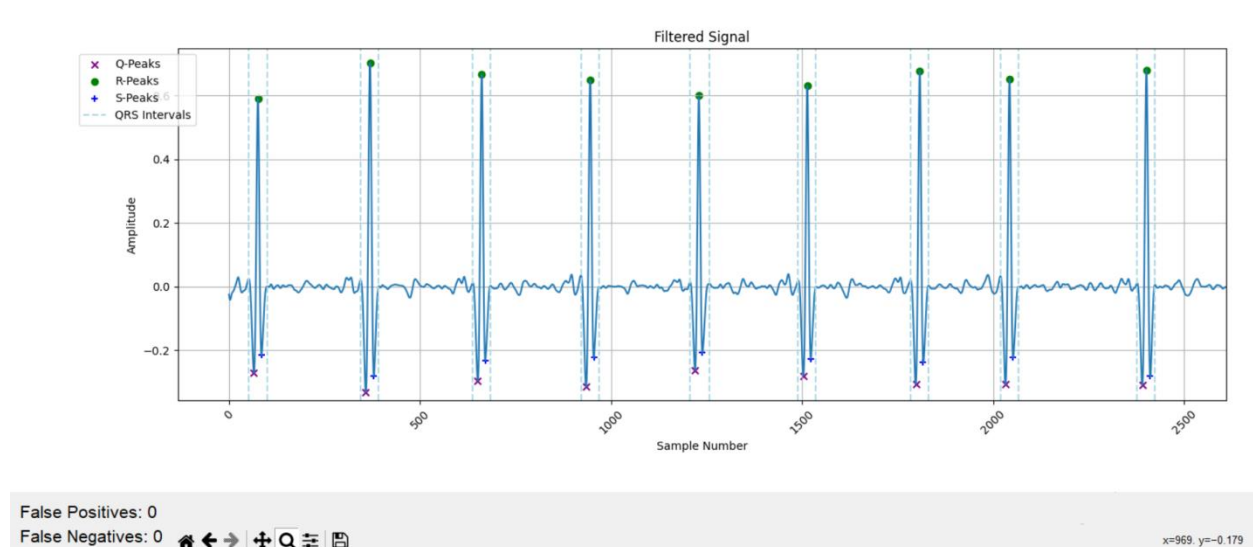
Slika 5.7 Legenda grafa

## 5.2. Testiranje aplikacije

Aplikacija je testirana na uzorcima EKG preuzetim iz *MIT-BIH Arrhythmia* baze podataka [5], na temelju koje će se i evaluirati preciznost iste. *MIT-BIH Arrhythmia* baze podataka je široko korištena kolekcija EKG zapisa koja se koristi za razvoj i evaluaciju algoritama za detekciju srčanih aritmija. Sastoji se od 48 dvokanalnih EKG zapisa, svaki u trajanju od 30 minuta,

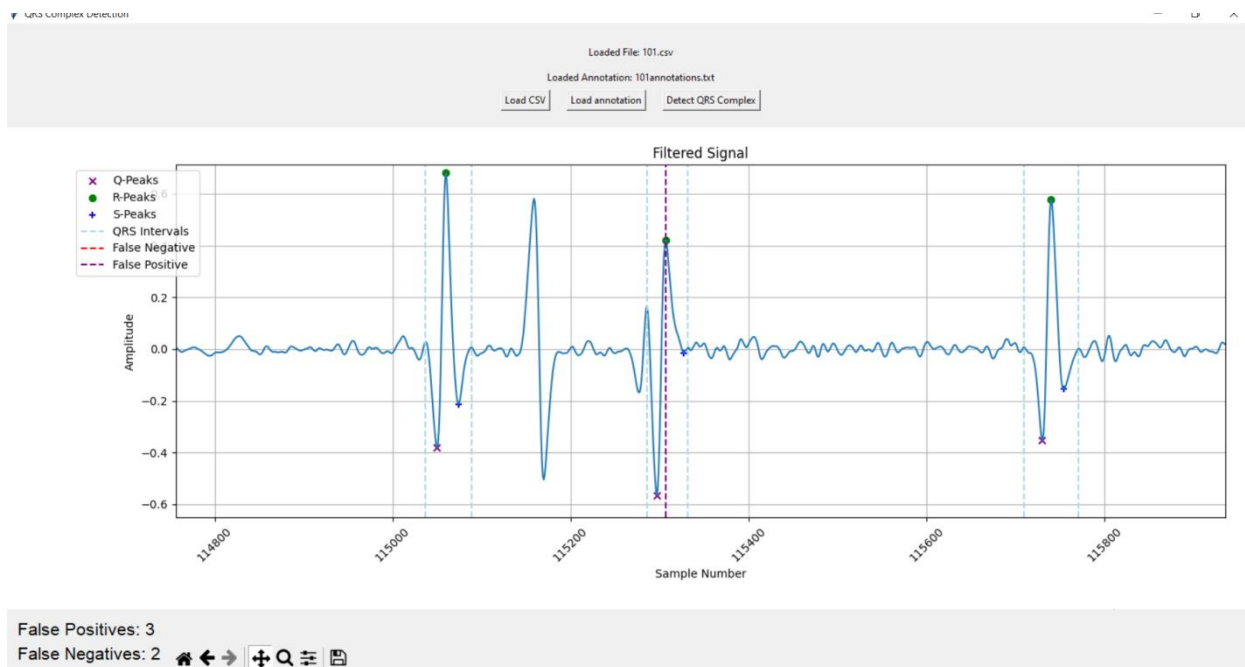
prikupljenih od 48 različitih pacijenata. Podaci su anotirani kako bi se označile različite vrste srčanih otkucaja i ritmova, te različite vrste aritmija. Baza je razvijena kasnih 1970-ih i ranih 1980-ih godina u Laboratoriju za biomedicinsko inženjerstvo na Massachusetts Institute of Technology (MIT) i Beth Israel Hospital (sada dio Beth Israel Deaconess Medical Center) u Bostonu. Cilj je bio pružiti standardizirani skup podataka za istraživanje i razvoj algoritama za automatsku detekciju aritmija. Prva verzija baze podataka objavljena je 1980. godine i od tada je postala temelj za razvoj mnogih algoritama za analizu EKG-a. Korištena je u velikom broju studija i istraživačkih radova, značajno doprinoseći napretku u računalnoj kardiologiji i medicinskoj dijagnostici. U nastavku su prikazani rezultati testiranja aplikacije na uzorcima 100-105 sa slikama grafova. Na ostalim uzorcima su prikazani samo rezultati.

Na uzorku 100 uspješno su detektirani svi kompleksi i sve karakteristične točke. Nema lažno-positivnih i negativnih, vidljivo na slici 5.8.



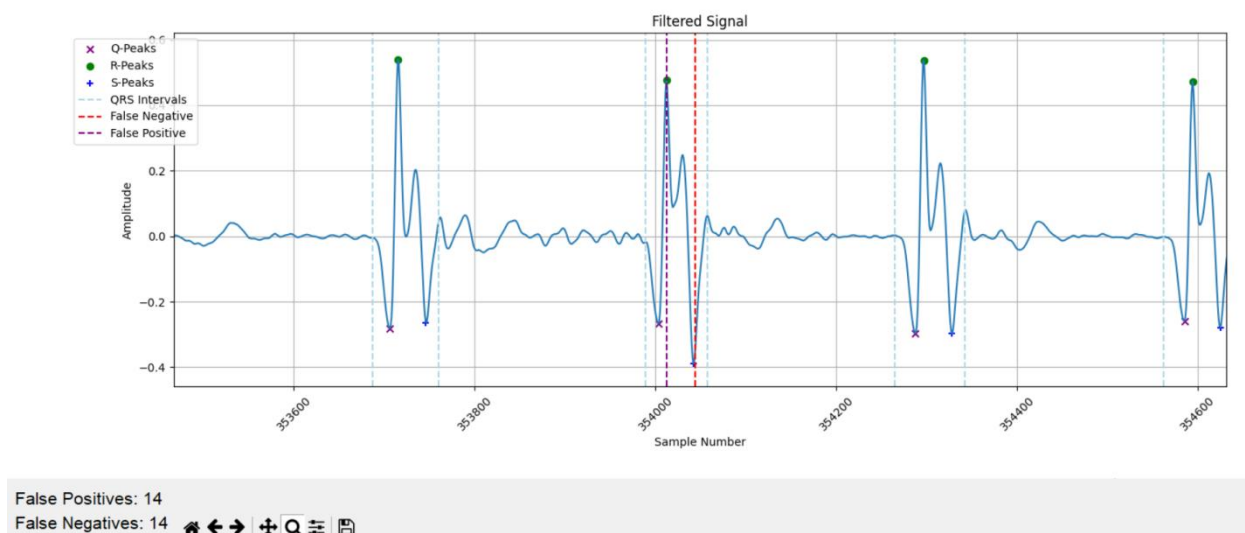
**Slika 5.8** Dio grafa dobiven testiranjem na uzorku 100

Na uzorku 101 neuspješno je detektirano 2 kompleksa, dok su 3 lažno pozitivna, vidljivi na slici 5.9. Algoritam na ovom uzorku ima problema s detekcijom Q-tipa koji predstavlja neklasificirani otkucaj i N-tip koji predstavlja normalni otkucaj.



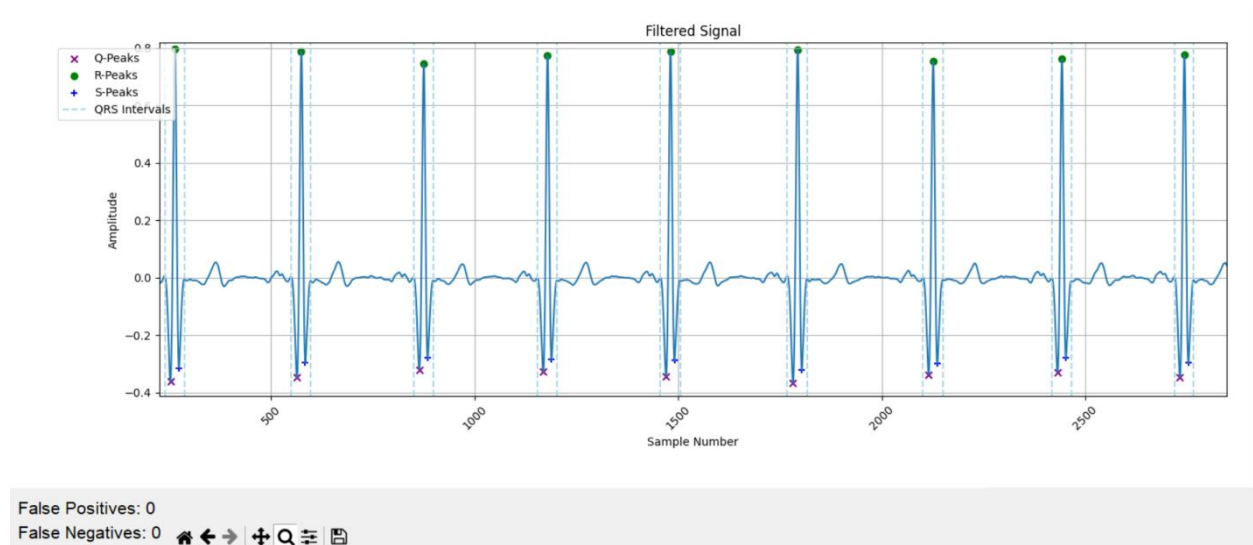
Slika 5.9 Primjer lažno detektiranog vrha na uzorku 101

Na uzorku 102 pogrešno je detektirano 14 kompleksa, dok je promašeno 14 kompleksa. Algoritam je neuspješno detektirao  $-$ tip, koji u anotacijama predstavlja otkucaj uzrokovan pacemakerom, medicinskim uređajem koji pomaže u regulaciji srčanog ritma. Primjer pogrešno detektiranog kompleksa vidljiv je na slici 5.10.



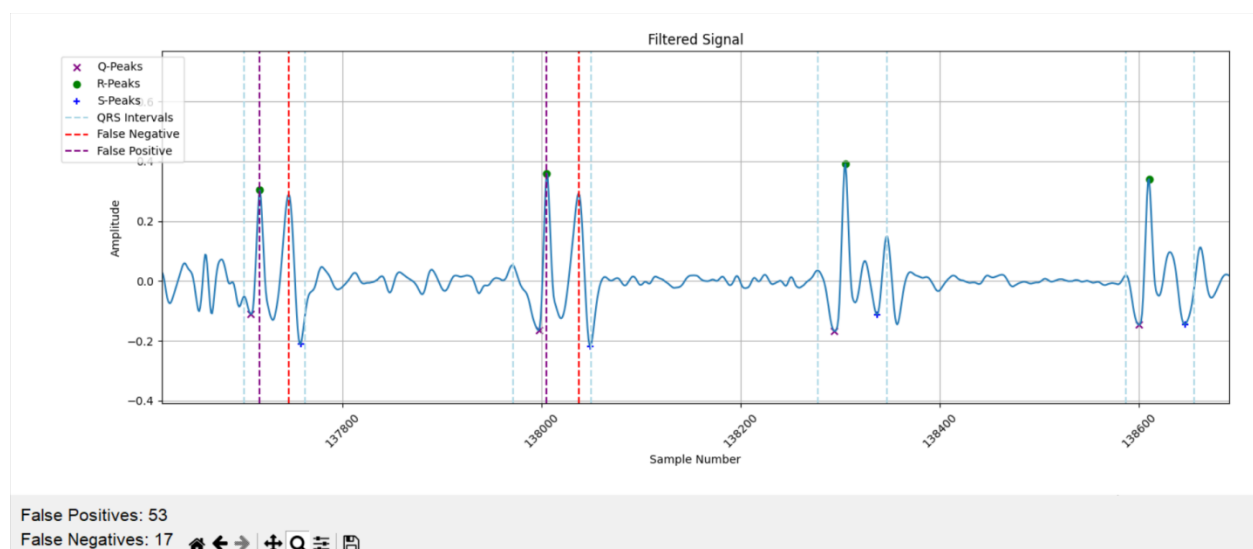
Slika 5.10 Primjer lažno detektiranog vrha i ne-detektiranog vrha na uzorku 102

Na uzorku 103 su uspješno detektirani svi kompleksi, nema ni lažno pozitivnih, ni lažno negativnih, što je vidljivo na slici 5.11.



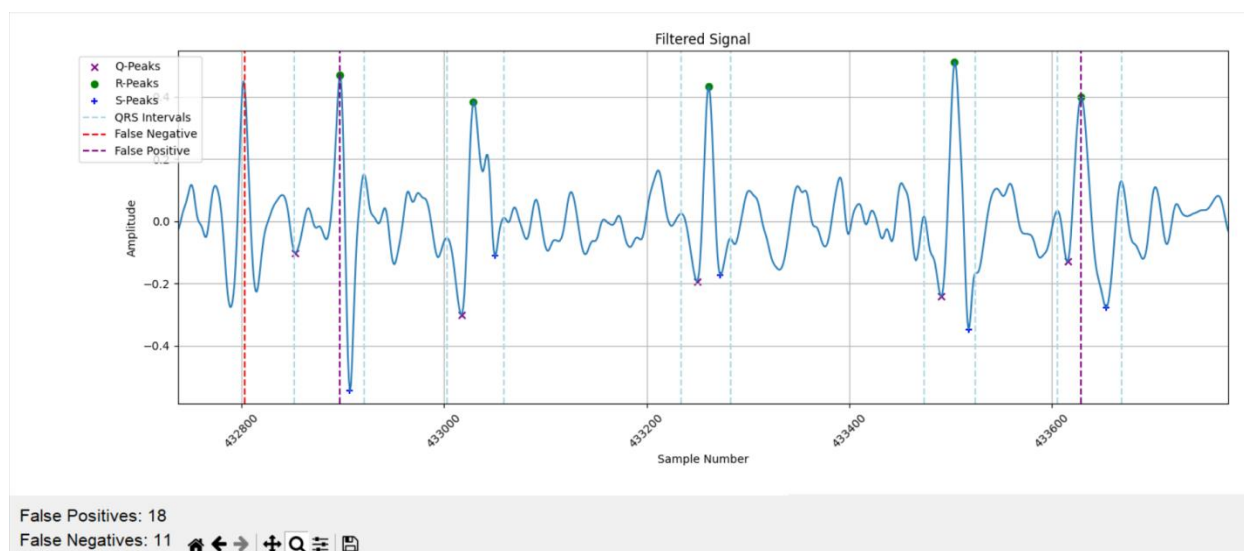
Slika 5.11 Dio grafa dobiven testiranjem na uzorku 103

Testiranje na uzorku 104 rezultiralo je s 53 pogrešno detektirana kompleksa i 17 lažno negativna. Algoritam na ovom uzorku ima problem s detektiranjem f i / tipa, koji u anotacijama predstavljaju spoj otkucaja uzrokovanog elektrostimulatorom srca (*engl. pacemaker*) i normalnog otkucaja, te otkucaj uzrokovan elektrostimulatorom srca. Primjer je vidljiv na slici 5.12.



Slika 5.12 Primjer lažno detektiranog vrha i ne-detektiranog vrha na uzorku 104

Na uzorku 105 lažno je detektirano 18 kompleksa, dok je promašeno 11. U ovom uzorku algoritam ima problema s pronalaskom N-tipa, koji u anotacijama predstavlja normalni otkucaj srca. Lažno su detektirani vrhovi koji su najvjerojatnije rezultat šuma. Primjer lažno detektiranog vrha i ne-detektiranog vrha vidljiv je na slici 5.13.



**Slika 5.13** Primjer lažno detektiranog vrha i ne-detektiranog vrha na uzorku 105

Testiranje je provedeno na ostalim uzorcima u *MIT-Arrhythmia* bazi [5], te su rezultati zapisani u tablicu. Za svaki uzorak izračunava se preciznost i osjetljivost, njihov prosjek, kao i prosjek lažno pozitivnih i lažno negativnih. Preciznost je metrika koja mjeri koliko su točne pozitivne predikcije algoritma. Osjetljivost, također poznata kao prisjetljivost (*engl.* recall), je metrika koja mjeri sposobnost ispravne identifikacije stvarno pozitivnih slučajeva algoritma. Ove vrijednosti izračunavaju se prema formulama 5-1, 5-2, 5-3 i 5-4.

$$\text{Preciznost} = \frac{\text{ispravno detektirani otkucaji (TP)}}{\text{ispravno detektirani otkucaji (TP)} + \text{lažno pozitivni (FP)}} \quad (5-1)$$

$$\text{Osjetljivost} = \frac{\text{ispravno detektirani otkucaji (TP)}}{\text{ispravno detektirani otkucaji (TP)} + \text{lažno negativni (FN)}} \quad (5-2)$$

$$\text{ispravno detektirani otkucaji (TP)} = \text{detektirani otkucaji} - \text{lažno pozitivni (FP)} \quad (5-3)$$

ili

ispravno detektirani otkucaji (TP) = stvarni broj otkucaja – lažno negativni (FN) (5-4)

Rezultati testiranja na ostalim uzorcima prikazani su u tablici 5.1.

**Tablica 5.1** Rezultati testiranja na MIT-Arrhythmia bazi podataka [5]

#	Očekivani broj otkucaja	Ukupni broj detektiranih otkucaja	Broj lažno-pozitivnih	Broj lažno-negativnih	Tip lažno-negativnih	Preciznost [%]	Osjetljivost [%]
100	2273	2273	0	0	nema	100	100
101	1865	1866	3	2	Q: 1, N: 1	99.84	99.89
102	2187	2187	14	14	/: 14	99.36	99.36
103	2084	2084	0	0	nema	100	100
104	2229	2265	53	17	f: 13, /: 4	97.66	99.24
105	2579	2572	18	11	N: 11	99.3	99.57
106	2027	2019	3	11	V: 11	99.85	99.46
107	2137	2134	15	18	V: 17, /: 1	99.3	99.16
108	1763	1766	130	127	N: 127	92.64	92.8
109	2532	2528	1	5	V: 5	99.96	99.8
111	2124	2124	2	2	L: 2	99.91	99.91
112	2539	2540	1	0	nema	99.96	100
113	1795	1797	2	0	nema	99.89	100
114	1879	1902	23	0	nema	98.79	100
115	1953	1953	0	0	nema	100	100
116	2412	2391	3	24	N: 23, V: 1	99.87	99
117	1535	1535	0	0	nema	100	100
118	2278	2279	1	0	nema	99.96	100
119	1987	1990	3	0	nema	99.85	100
121	1863	1862	0	1	N: 1	100	99.95
122	2476	2476	0	0	nema	100	100
123	1518	1518	0	0	nema	100	100
124	1619	1625	10	4	V: 4	99.38	99.75



200	2601	2606	8	3	V: 2, F: 1	99.69	99.88
201	1963	1938	0	25	a: 24, N: 1	100	98.73
202	2136	2131	0	5	a: 4, A: 1	100	99.77
203	2980	2884	32	128	V: 74, N: 51, Q: 3	98.89	95.7
#	Očekivani broj otkucaja	Ukupni broj detektiranih otkucaja	Broj lažno-pozitivnih	Broj lažno-negativnih	Tip lažno-negativnih	Preciznost [%]	Osjetljivost [%]
205	2656	2636	0	20	V: 17, N: 2, F: 1	100	99.25
207	2332	2064	88	356	!: 328, V: 25, L: 2, E: 1	95.74	84.73
208	2955	2937	4	22	N: 10, V: 9, F: 2, Q: 1	99.86	99.26
209	3005	3005	0	0	nema	100	100
210	2650	2573	2	79	V: 71, F: 5, a: 3	99.92	97.02
212	2748	2748	0	0	nema	100	100
213	3251	3245	23	29	V: 27, F: 2	99.29	99.11
214	2262	2258	2	6	V: 4, L: 2	99.91	99.73
215	3363	3354	0	9	V: 8, N: 1	100	99.73
217	2208	2206	1	3	V: 1, /: 1, f: 1	99.95	99.86
219	2154	2154	0	0	nema	100	100
220	2048	2048	0	0	nema	100	100
221	2427	2422	0	5	V: 5	100	99.79
222	2483	2483	1	1	A: 1	99.96	99.96
223	2605	2601	0	4	V: 2, N: 1, A: 1	100	99.85
228	2053	2076	27	4	N: 3, V: 1	98.7	99.81
230	2256	2256	0	0	nema	100	100
231	1571	1571	0	0	nema	100	100
232	1780	1786	6	0	nema	99.66	100
233	3079	3070	0	9	V: 8, N: 1	100	99.71
234	2753	2750	0	3	V: 3	100	99.89
Σ	109 979	109 482	476	947			
PROSJEK			9.92	19.73		99.52	99.16

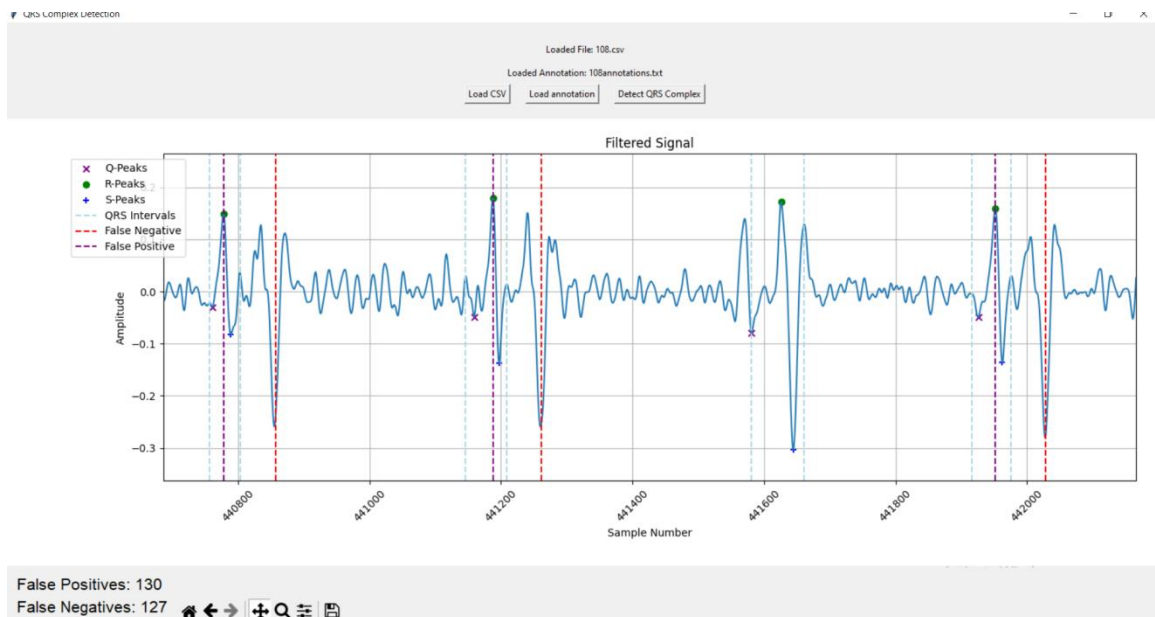
### 5.3. Analiza rezultata

Testiranje na prethodno opisanom skupu podataka rezultiralo je prosječnom preciznošću od 99.52%, dok je osjetljivost 99.16%. U nastavku poglavlja analizirana su tri najgora uzorka po preciznosti i osjetljivosti, te kategorije lažno negativnih uzoraka.

Algoritam je imao najnižu preciznost na uzorku 108 i to 92.64%, dok je po osjetljivosti na istom uzorku drugi najgori s 92.8%. Lažno je detektirano 130 otkucaja, dok je promašeno 127 otkucaja, od kojih su svi klasificirani kao normalni, što je prikazano u tablici 5.2. Algoritam je najvjerojatnije imao problema s detekcijom zbog šuma, što je vidljivo na slici 5.14.

**Tablica 5.2** Rezultati testiranja uzorka 108

#	Očekivani broj otkucaja	Ukupni broj detektiranih otkucaja	Broj lažno-pozitivnih	Broj lažno-negativnih	Tip lažno-negativnih	Preciznost [%]	Osjetljivost [%]
108	1763	1766	130	127	N: 127	92.64	92.8



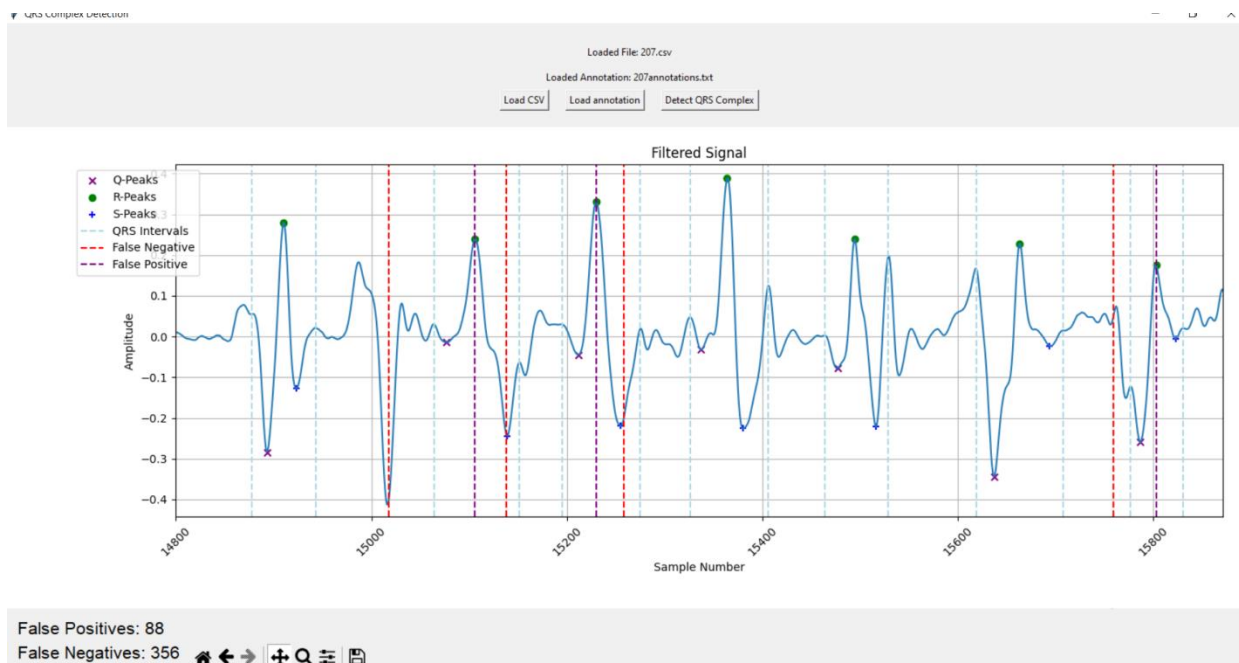
**Slika 5.14** Lažno pozitivni i lažno negativni vrhovi na uzorku 108

Drugu najlošiju preciznosti, a prvu najlošiju osjetljivost algoritam ima za uzorak 207, i to s vrijednostima 95.74% za preciznosti i 84.73% osjetljivost, vidljivo iz tablice 5.3. Lažno pozitivnih otkucaja detektirano je 88, dok je lažno negativnih 356. To su otkucaji koji predstavljaju ventrikularnu aritmiju (označeni s !), preuranjenu kontrakciju ventrikula (označeni

s V), otkucaji bloka lijeve grane snopa (označeni s L) i ventrikularni izlazni otkucaj (označen s E). Iz grafa na slici 5.15 nije jasno zašto su pogrešno detektirani vrhovi, najvjerojatnije se nalaze izvan tolerancije od 50 uzoraka, pa ih samim time algoritam ne klasificira kao ispravno detektirane, iako najvjerojatnije jesu.

**Tablica 5.3** Rezultati testiranja uzorka 207

#	Očekivani broj otkucaja	Ukupni broj detektiranih otkucaja	Broj lažno-pozitivnih	Broj lažno-negativnih	Tip lažno-negativnih	Preciznost [%]	Osjetljivost [%]
207	2332	2064	88	356	!: 328, V: 25, L: 2, E: 1	95.74	84.73



**Slika 5.15** Lažno pozitivni i lažno negativni vrhovi na uzorku 207

Treći najlošiji rezultat po preciznosti algoritam je ostvario na uzorku 104, uz vrijednost 97.66% prikazanu u tablici 5.4. Kao što je spomenuto u prethodnom potpoglavlju, algoritam je lažno detektirao 53 otkucaja, dok je promašio 17, i to većinom spojeve otkucaja uzrokovanih pacemakerom i normalnih otkucaja (označenih s f) i otkucaje uzrokovane pacemakerom (označene s /). Iz grafa na slici 5.12 se može vidjeti da su lažno detektirani šumovi.

**Tablica 5.4** Rezultati testiranja uzorka 104

#	Očekivani broj otkucaja	Ukupni broj detektiranih otkucaja	Broj lažno-pozitivnih	Broj lažno-negativnih	Tip lažno-negativnih	Preciznost [%]	Osjetljivost [%]
104	2229	2265	53	17	f: 13, /: 4	97.66	99.24

Treći najlošiji rezultat po osjetljivosti algoritam je ostvario na uzorku 203 s vrijednošću 95.7%, prikazano u tablici 5.5. Algoritam je detektirao 32 lažno pozitivna, a lažno negativnih otkucaja 128, većinom otkucaja s preuranjenom kontrakcijom ventrikula (označeni s V), normalnih (označenih s N) i 3 ne-klasificirana otkucaja (označeni s Q).

**Tablica 5.5** Rezultati testiranja uzorka 203

#	Očekivani broj otkucaja	Ukupni broj detektiranih otkucaja	Broj lažno-pozitivnih	Broj lažno-negativnih	Tip lažno-negativnih	Preciznost [%]	Osjetljivost [%]
203	2980	2884	32	128	V: 74, N: 51, Q: 3	98.89	95.7

Broj lažno negativnih otkucaja prema tipu prikazan je u tablici 5.6. Iz nje je vidljivo da najviše lažno negativnih otkucaja koji predstavljaju ventrikularnu aritmiju kojih je 328, zatim preuranjenih kontrakcija ventrikula kojih je 295, a poslije njih dolaze normalni otkucaji kojih je 233.

**Tablica 5.6** Broj lažno negativnih otkucaja prema tipu

Tip	Objašnjenje	Broj lažno-negativnih otkucaja
!	Ventrikularna aritmija	328
V	Preuranjena kontrakcija ventrikula	295
N	Normalni otkucaj	233
a	Aberirani preuranjeni otkucaj atrijski	31
/	Otkucaj uzrokovan „pacemakerom“	20
f	Spoj otkucaja uzrokovanog „pacemakerom“ i normalnog otkucaja	14
F	Spoj ventrikularnog i normalnog otkucaja	11

L	Otkucaj bloka lijeve grane snopa	6
Q	Ne-klasificirani otkucaj	5
A	Preuranjeni otkucaj atrijski	3
E	Ventrikularni izlazni otkucaj	1
j	Kasni atrioventrikularni otkucaj	0
R	Otkucaj bloka desne grane snopa	0
J	Preuranjeni atrioventrikularni otkucaj	0
e	Kasni atrijalni otkucaj	0
S	Supraventrikularni prijevremeni otkucaj	0

## 6. Zaključak

U sklopu diplomskog rada izrađena je aplikacija za detekciju QRS kompleksa i njegovih karakterističnih točaka na temelju zadane EKG snimke. Nakon učitavanja snimke, aplikacija prikazuje originalni EKG signal na kojemu su označene prethodno spomenute karakteristike. Također, korisnik ima mogućnost detaljnijeg pregleda i pohrane dijela grafa korištenjem različitih alata, te pregleda broja lažno pozitivnih i negativnih vrhova u slučaju učitavanja anotacija.

Radom je detaljno opisan postupak izrade aplikacije, koji uključuje opis podataka i anotacija koji su potrebni za rad aplikacije, jednostavno sučelje za učitavanje istih i prikaz rezultata, te samog algoritma za detekciju QRS kompleksa. Aplikacija je zatim testirana na 48 uzoraka koji pripadaju *MIT-Arrhythmia* bazi podataka [5], a rezultati testova prikazani i analizirani. Algoritam za detekciju ima prosječnu preciznost od 99.52%, dok je osjetljivost 99.16%. Najčešći lažno negativni tip detektiranih otkucaja je otkucaj ventrikularne aritmije, zatim otkucaj koji ima preuranjenu kontrakcija ventrikula i poslije nje normalni otkucaj.

Najveći izazov tijekom izrade aplikacije bio je podešavanje parametara za optimalnu detekciju QRS kompleksa. Iako je preciznost i osjetljivost aplikacije visoka, algoritam za detekciju i dalje ima problema s detekcijom uzoraka u kojima su prisutne velike količine šuma. Dodavanjem dinamičkog postavljanja praga najvjerojatnije bi se smanjio broj lažno pozitivnih i negativnih vrhovi. Također, moguće je poboljšanje izgleda sučelja za učitavanje EKG signala i anotacija, te mogućnost učitavanja različitih formata istih.

## Literatura

- [1] C. Cajavilca J. Varon, G. L. Sternbach, „Luigi Galvani and the foundations of electrophysiology“, 2009.
- [2] B. Lüderitz, „Augustus Desiré Waller (1856–1922)—The First to Record the Electrical Activity of the Human Heart“, 2003.
- [3] A. M. Salam, „The Invention of Electrocardiography Machine“, 2019.
- [4] G. E. Burch, N. P. DePasquale, „A History of Electrocardiography“, Norman Publishing 1990.
- [5] PhysioNet, Dostupno na <https://physionet.org/content/mitdb/1.0.0/>, [Pristupljeno 13.06.2024.]
- [6] J. Krmpotić-Nemanić, A. Marušić, „Anatomija čovjeka“, 2007.
- [7] „Presjek građe srca“, Wikipedia, 17.07.2021. Dostupno na <https://hr.wikipedia.org/wiki/Srce>
- [8] „Slojevi zida srčanog mišića“, Dostupno na <https://zonamedicine.com/anatomija-srca/>
- [9] „Electrical conduction system of the heart“, Wikipedia, 02.03.2007.
- [10] „Avantatges De L'ecocardiografia 3d En El Diagnòstic De La Vàlvula Mitral“, 03.12.2021., Dostupno na [https://premisreerca.uvic.cat/sites/default/files/webform/webform\\_1/3151/TDR%20avantatges%20eco%20premi%20definitiu.pdf](https://premisreerca.uvic.cat/sites/default/files/webform/webform_1/3151/TDR%20avantatges%20eco%20premi%20definitiu.pdf)
- [11] „The different shapes of the cardiac action potential in the heart“, Wikipedia, 28.12.2011. Dostupno na [https://en.wikipedia.org/wiki/Cardiac\\_conduction\\_system](https://en.wikipedia.org/wiki/Cardiac_conduction_system)
- [12] „Electrode positions for limb leads and augmented limb leads.“, A Cost-Effective Multichannel Wireless ECG Acquisition System, prosinac 2018.
- [13] „ECG Course“, AFEducation, Dostupno na <https://www.afeducation.org/courses/ecg>
- [14] „Fast QRS Complex Detection Algorithm Based on RMS Shifting Concept for Heart Rate Estimation Using an Electrocardiogram“, lipanj 2023.
- [15] „Prikaz normalnog EKG-a“, Wikipedia, 25.06.2023. Dostupno na <https://hr.wikipedia.org/wiki/Elektrokardiogram>
- [16] J. Pan, W. J. Tompkins, „A Real-Time QRS Detection Algorithm“, 1985.
- [17] C. Lin, H. Chang, Y. Huang, C. Yeh, „A Novel Wavelet-Based Algorithm for Detection of QRS Complex“, 2019.
- [18] N. Li, F. He, W. Ma, R. Wang, L. Jiang, X. Zhang, „The Identification of ECG Signals Using Wavelet Transform and WOA-PNN, 2022.
- [19] A. Kumar, D. Berwal, Y. Kumar, „Design of High-Performance ECG Detector for Implantable Cardiac Pacemaker Systems using Biorthogonal Wavelet Transform“, rujan 2018.

- [20] R. A. Álvarez, A. J. M. Penín, X. A. V. Sobrino, „Comparison of Three QRS Detection Algorithms Over a Public Database“, 2013.
- [21] S. Sahoo, P. Biswal, T. Das, S. Sabut, „De-noising of ECG Signal and QRS Detection Using Hilbert Transform and Adaptive Thresholding“, 2013.
- [22] Q. Xue, Y. H. Hu, W. J. Tompkins, „Neural-network-based adaptive matched filtering for QRS detection“, travanj 1992.
- [23] A. Shdefat, S.-J. Park, H.-C. Kim, „Real-time processing structure for handling HRV tuples“, kolovoz 2017
- [24] Python Documentation, 2024., Dostupno na <https://docs.python.org/3/> [Pristupljeno: 16.06.2024.]
- [25] SciPy Documentation, 2024., Dostupno na <https://docs.scipy.org/doc/scipy/> [Pristupljeno: 16.06.2024.]
- [26] NumPy Documentation, 2024., Dostupno na <https://numpy.org/doc/> [Pristupljeno: 16.06.2024.]



## Sažetak

Tema ovog rada je izrada aplikacije za detekciju QRS kompleksa u zadanom EKG signalu. EKG je neizostavan alat za dijagnostiku u medicini, za što je bitna detekcija i analiza njegovih karakterističnih točaka. U opsegu ovoga rada objašnjena je građa srca, gdje električni signal nastaje i kako putuje kroz njega, te EKG i njegove karakteristike. Također, opisana je implementacija i način rada aplikacije, a rad iste je testiran na uzorcima iz *MIT-Arrhythmia* baze podataka. Rezultati testiranja su zatim detaljno analizirani i prikazani, te je na temelju njih donesen zaključak o preciznosti aplikacije. Za izradu aplikacije korišten je programski jezik Python i njegove pripadajuće biblioteke.

**Ključne riječi:** Python, EKG, QRS kompleks, srce

## **Abstract**

### **Finding or recognizing the QRS complex in the EKG**

The subject of this paper is the implementation of an application for the detection of QRS complexes in a given EKG signal. The EKG is an indispensable diagnostic tool in medicine, for which the detection and analysis of its characteristic points is essential. The structure of the heart, where the electrical signal originates and how it travels through it, and the EKG and its characteristics are explained in the scope of this paper. Also, the application's implementation and mode of operation are described, and its operation was tested on samples from the *MIT-Arrhythmia* database. The test results were then analyzed and presented in detail, and based on them, a conclusion was made about the accuracy of the application. The Python programming language and its associated libraries were used to create the application.

**Key words:** Python, EKG, QRS complex, heart

## **Životopis**

Bojan Mandić rođen je 28.08.1997. u Našicama. Nakon završetka osnovne škole upisuje prirodoslovno-matematičku gimnaziju u Srednjoj školi Isidora Kršnjavoga u Našicama. Nakon srednje škole 2016. godine upisuje preddiplomski studij računarstva na Fakultetu elektrotehnike i računarstva u Zagrebu s kojeg se 2019. godine prebacuje na preddiplomski studij računarstva na Fakultetu elektrotehnike, računarstva i informacijskih tehnologija u Osijeku kojeg završava 2022. godine. Iste godine upisuje diplomski studij programskog inženjerstva na istom fakultetu.

---

Potpis autora