
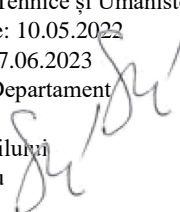

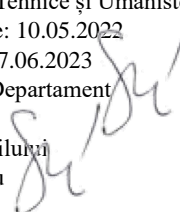


**UNIVERSITATEA „SAPIENTIA” DIN CLUJ-
NAPOCA
FACULTATEA DE ȘTIINȚE TEHNICE ȘI UMANISTE, TÎRGU
MUREȘ
SPECIALIZAREA CALCULATOARE**

**ANALIZA SEMNALELOR ECG PRIN
METODA SPECTOGRAMELOR
UTILIZATĂ ÎN DETECTAREA
ARITMIILOR CU AJUTORUL
REȚELELOR NEURONALE DE TIP
DEEP LEARNING
PROIECT DE DIPLOMĂ**

**Coordonator științific:
Prof. Dr. Ing. Dávid László**

**Absolvent:
Kántor Zsuzsa-Krisztina**

**Consultant medical:
Dr. Heidenhoffer Erhald**

UNIVERSITATEA „SAPIENTIA” din CLUJ-NAPOCA Facultatea de Științe Tehnice și Umaniste din Târgu Mureș Specializarea: Automatică și informatică aplicată		Viza facultății: 
LUCRARE DE DIPLOMĂ		
Coordonator științific: prof dr. ing. Dávid László	Candidat: Kántor Zsuzsa-Krisztina Anul absolvirii: 2023	
a) Tema lucrării de licență: <div style="text-align: center;"> ANALIZA SEMNALELOR ECG PRIN METODA SPECTOGRAMELOR UTILIZATĂ ÎN DETECTAREA ARITMIILOR CU AJUTORUL REȚELOR NEURONALE DE TIP DEEP LEARNING </div>		
b) Problemele principale tratate: Se va studia metode care asigură identificarea celor mai frecvente boli de cardiovasculare sentimente prin procesarea unor semnale ECG captate. Astfel metodele clasice de prelucrarea a semnalelor ECG în timp fetei pot fi îmbunătățite. Totodată prin utilizarea unor elemente de inteligență artificială, se va asigura o diagnoză automată. Datele brute colectate se vor procesa, după care va fi folosită pentru a construi o baza de date și a învăța o rețea neuronală artificială deep learning, care a fost creată cu ajutorul motorului Tensorflow și Keras. Totodată se va examina spectrul probelor și aplicând diferite transformări (spectrograma, scalograma), pentru a le compara. - Se va realiza o aplicație pentru prelucrarea semnalelor ECG		
c) Desene obligatorii: - Schema bloc al aplicației - rezultate practice obținute		
d) Softuri obligatorii: -Soft capture, filtrare, prelucrare, transformare a semnalelor ECG, soft învățare rețe neuronală deep learning. Soft aplicație practică		
e) Bibliografia recomandată: Antoun Khawaja, Automatic ECG analysis, using PCA and Wavelet Transformation, Karlsruhe 2006 Rajarsi Gupta, ECG acquisition and automated remote processing, Springer 2014 Adam Gacek, ECG signal processing, Springer, 2012 Gari D. Clifford, Advanced methods for ECG data analysis, Artech House Engineering in Medicine & Biology Series, 2006 Richard M. Millis, Advances in ECG methods and analysis, 2012 Wilkins, L., „European Cardiovascular Disease Statistics 2017, https://researchportal.bath.ac.uk/en/publications/european-cardiovascular-disease-statistics-2017 N. Townsend, L., „European Heart Journal, Cardiovascular disease in Europe: epidemiological update 2016,” 14 August 2016. [Online]. Available: https://academic.oup.com/eurheartj/article/37/42/3232/2536403 . r. 2. Causes of death - diseases of the circulatory system. [Online]. Available: https://ec.europa.eu/eurostat/statistics-explained/images/6/67/Causes_of_death_%E2%80%94_diseases_of_the_circulatory_system%2C_residents%2C_2015_HLTH18.png . D. Dumitrascu, S. Hopulele és A. Baban, „Cardiovascular complaints following the uprising of December 1989 in Romania,” [Online]. Available: https://www.tandfonline.com/doi/abs/10.1080/07488009308409076 . A. H. A. (. C. o. H. Disease. [Online]. Available: https://www.heart.org/en/health-topics/consumer-healthcare/what-is-cardiovascular-disease/causes-risks--prevention-of-cardiovascular-disease .		
f) Termene obligatorii de consultații: săptămânal		
g) Locul și durata practicii: Universitatea „Sapientia” din Cluj-Napoca, Facultatea de Științe Tehnice și Umaniste din Târgu Mureș Primit tema la data de: 10.05.2022 Termen de predare: 27.06.2023 <div style="display: flex; justify-content: space-between;"> <div> Semnătura Director Departament  </div> <div> Semnătura coordonatorului  </div> </div> <div style="display: flex; justify-content: space-between; margin-top: 20px;"> <div> Semnătura responsabilului programului de studii  </div> <div> Semnătura candidatului  </div> </div>		

Declarație

Subsemnata/ul Kántor Zsuzsa-Krisztina, absolvent(ă) al/a specializării CALCULATOARE, promoția 2019-2023 cunoscând prevederile Legii Educației Naționale 1/2011 și a Codului de etică și deontologie profesională a Universității Sapientia cu privire la furt intelectual declar pe propria răspundere că prezenta lucrare de licență/proiect de diplomă/disertație se bazează pe activitatea personală, cercetarea/proiectarea este efectuată de mine, informațiile și datele preluate din literatura de specialitate sunt citate în mod corespunzător.

Localitatea, CORUNCA

Data:

Absolvent

Semnătura:



Extras

Metodele de diagnostic de astăzi în medicină se dezvoltă în mod constant, iar acum putem aplica tehnici pe care nu le puteam concepe înainte. În cadrul noilor posibilități și provocări tehnologice, îmi doresc să găsesc soluții prin care să putem recunoaște și trata mai eficient afecțiunile comune ale sistemului cardiovascular, evitând astfel decesul prematur sau evitabil.

Scopul principal al cercetării mele este dezvoltarea unor sisteme de diagnostic noi și avansate, care se concentrează specific pe caracteristicile afecțiunilor sistemului cardiovascular. Aceste sisteme ar permite efectuarea unor diagnostice mai precise și eficiente în identificarea și tratarea afecțiunilor. Fundamentul abordării mele îl reprezintă utilizarea semnalelor electrocardiografice (EKG) în medicină. Aceste semnale oferă informații precise și detaliate despre activitatea electrică a inimii, ceea ce permite detectarea prezenței diferitelor afecțiuni cardiace sau a altor abateri.

Primul pas implică prelucrarea preliminară a semnalelor, unde aplic filtrare și alte proceduri de normalizare. Apoi, creez o imagine cu canale de culoare RGB, un spectrogramă. Spectrograma este o reprezentare vizuală care arată spectrul de frecvență al semnalului în funcție de timp. Nuanțele diferite de culoare în canale indică intensitatea frecvenței, ceea ce facilitează analiza structurii și schimbărilor semnalului. Acest proces ajută la extragerea informațiilor esențiale din semnale și reduce distorsiunile cauzate de zgomot.

Aceste imagini servesc ca set de date de intrare pentru software-ul meu de instruire bazat pe învățarea profundă, a cărei sarcină principală este recunoașterea a patru boli specifice. Prin urmare, sistemul meu va fi capabil să identifice modele sau detalii minuțioase care nu sunt vizibile cu ochiul liber, oferind diagnoze mai precise persoanelor. Acest lucru poate contribui semnificativ la îmbunătățirea rezultatelor în domeniul sănătății și la creșterea calității vieții.

Cuvinte cheie: învățare profundă, afecțiuni cardiovasculare, spectrogramă, filtrarea semnalului, diagnosticare a bolilor.

**UNIVERSITATEA „SAPIENTIA” DIN CLUJ-
NAPOCA
FACULTATEA DE ȘTIINȚE TEHNICE ȘI UMANISTE, TÎRGU
MUREȘ
SPECIALIZAREA AUTOMATICĂ ȘI INFORMATICĂ
APLICATĂ**

**SPECTOGRAM ALAPÚ EKG
JELFELDOLGOZÁS ÉS
MÉLYTANULÁSÚ NEURONHÁLÓVAL
VALÓ ARITMIA DETEKTÁLÁSA
DIPLOMADOLGOZAT**

**Témavezető:
Prof. Dr. Ing. Dávid László**

**Végzős hallgató:
Kántor Zsuzsa-Krisztina**

**Orvosi konzulens:
Dr. Heidenhoffer Erhard**

Kivonat

A mai diagnosztikai módszerek az orvoslásban folyamatosan fejlődnek, és ma már olyan technikákat alkalmazhatunk, amelyeket korábban elképzelni sem tudtunk. Az új lehetőségek és technológiai kihívások között olyan megoldásokat szeretnék találni, amelyek révén hatékonyabban tudunk felismerni és kezelni gyakori szív- és érrendszeri betegségeket, elkerülve ezáltal a korai vagy az elkerülhető elhalálozást.

A kutatásom fő célja az, hogy új és fejlettebb diagnosztikai rendszereket fejlesszek ki, amelyek specifikusan a szív- és érrendszeri betegségek jellegzetességeire fókuszálnak. Ezen rendszerek segítségével pontosabb és hatékonyabb diagnózisokat lehetne elvégezni a betegségek azonosítása és kezelése során. A módszerem alapját az orvosi elektrokardiográfia (EKG) jeleinek felhasználása képezi. Ezen jelek precíz és részletes információkat nyújtanak a szív elektromos aktivitásáról, amelyek alapján lehetséges különböző szívbetegségek jelenlétét vagy egyéb eltéréseket kimutatni. Az első lépés a jelek előzetes feldolgozása, amely során szűrést és egyéb normalizáló eljárásokat alkalmazok. Ezután egy RGB színcsatornás képet, spektrogrammot hozok létre. A spektrogramm egy olyan vizuális reprezentáció, amely bemutatja a jelzés frekvencia-spektrumát az idő függvényében. A színcsatornák különböző árnyalatai jelzik a frekvencia intenzitását, így könnyebbé válik a jelzés struktúrájának és változásainak vizsgálata. Ez a folyamat segít kivonni a lényeges információkat a jelekből, és csökkenti a zajok által okozott torzításokat. Ezek a képek pedig bemeneti halmazként szolgálnak a mélytanulás alapú tanító szoftveremhez, melynek fő feladata az lesz, hogy felismerjen négy konkrét betegséget.

Tehát a rendszerem képes lesz felismerni olyan mintákat, vagy apró részleteket, amelyek emberi szemmel nem láthatóak, így pontosabb diagnózist biztosíthat az emberek számára, ami jelentősen hozzájárulhat az egészségügyi eredmények javulásához és az életminőség növekedéséhez.

Kulcsszavak: mélytanulás, szív- és érrendszeri betegségek, spektrogram, jel szűrés, betegség diagnózis.

**UNIVERSITATEA „SAPIENTIA” DIN CLUJ-
NAPOCA
FACULTATEA DE ȘTIINȚE TEHNICE ȘI UMANISTE, TÎRGU
MUREȘ
SPECIALIZAREA CALCULATOARE**

**ECG SIGNAL PROCESSING WITH
SPECTOGRAM IN ORDER TO DETECT
ARRYTHMIAS WITH DEEP
LEARNING NEURAL NETWORKS**

Supervisor:
Prof. Dr. Ing. Dávid László

Student:
Kántor Zsuzsa-Krisztina

Medical consultant:
Dr. Heidenhoffer Erhald

Abstract

Today, diagnostic methods in medicine are constantly evolving, and we can now apply techniques that were unimaginable before. Among the new possibilities and technological challenges, I aim to find solutions that can help us recognize and treat common cardiovascular diseases more effectively, thereby preventing premature or avoidable deaths.

The main goal of my research is to develop new and advanced diagnostic systems specifically focused on the characteristics of cardiovascular diseases. These systems would enable more accurate and efficient diagnoses during the identification and treatment of these diseases. The foundation of my approach is the utilization of electrocardiography (ECG) signals in medicine. These signals provide precise and detailed information about the heart's electrical activity, making it possible to detect various heart diseases or other abnormalities.

The first step involves preprocessing the signals, applying filtering and other normalization techniques. Then, I create an RGB color-channel image, a spectrogram. The spectrogram is a visual representation that shows the frequency spectrum of the signal over time. Different shades of color in the channels indicate the intensity of the frequencies, making it easier to analyze the structure and changes of the signal. This process helps extract essential information from the signals and reduces distortions caused by noise.

These images serve as the input dataset for my deep learning-based training software, whose main task is to recognize four specific diseases. Therefore, my system will be able to identify patterns or subtle details that are not visible to the human eye, providing more accurate diagnoses for people. This can significantly contribute to improving healthcare outcomes and enhancing the quality of life.

Keywords: deep learning, cardiovascular diseases, spectrogram, signal filtering, disease diagnosis.

Tartalom

1.	Bevezető	11
2.	Elméleti alapok	14
2.1.	A szív	14
2.2.	Elektrokardiográfia (EKG)	16
2.2.1.	Meghatározás	17
2.2.2.	Az EKG célja	17
2.2.3.	Az EKG-vizsgálat	18
2.2.4.	Az elektrokardiográfiai görbe	19
2.2.5.	A szívritmus, sinusritmus	20
3.	Gyakorlati megvalósítás	21
3.1.	Projekt leírása	21
3.2.	Felhasználói követelmények	24
3.3.	Funkcionális követelmények	26
3.4.	Választott programozási nyelv, környezet	27
4.	Gyakorlati megvalósítás	28
5.	Felhasznált adatbázisok	29
5.1.	Maros Megyei Klinikai Kórház adatbázisa	30
5.2.	Champman University és a Shaoxi Klinika közös adatbázisa	31
6.	Zajok szűrése és normalizálása	33
7.	Spektrogram	37
7.1.	EKG jelek és a Fourier transzformáció alkalmazása	37
7.2.	A Gyors Fourier - Transzformáció (FFT)	38

7.3. A Spektrogram készítése	39
8. Adatok előkészítése a mély tanuláshoz	44
9. A mélytanulású neuronhálózat	47
10. A program felépítése	50
10.1. Tanítási paraméterek hangolása.....	63
10.2. Tesztelés, eredmények és célok	66
11. Hivatkozások	67
12. Ábrajegyzék.....	71

1. Bevezető

A szív- és érrendszeri megbetegedések világ szinten az elhalálozások fő okai között van számon tartva [1], ezért kiemelt fontossággal bír a hatékony diagnosztika és ezeknek a megfelelő kezelése. Az elektrokardiográfia (EKG) az egyik legelterjedtebb és alapvető diagnosztikai módszer a kardiológiában és egyben nélkülözhetetlen is egy összetett vizsgálat helyes kiértékelése érdekében. Az EKG lehetővé teszi a szív elektromos tevékenységének mérését és értelmezését aminek jelentőségét az adja, hogy az általa nyújtott információk segítségével az orvosok képesek felismerni a szívbetegségeket, azonosítani az eltéréseket a normálistól, valamint követni a betegség lefolyását és hatékonyságát.

Az egyre inkább felgyorsult, sok impulzussal telítődött világ, sok pozitívumot rejt magában, de emellett számolnunk kell a negatív hatásaival is. Ez gyakran stresszel jár, rossz táplálkozással, és mozgásszegény életmóddal. Ezek az életmódbeli tényezők jelentős hatással vannak a szív egészségére, és hajlamosabbá teszik az embereket a szívbetegségekre. Az elhízás, a magas vérnyomás, a szívritmuszavarok és az érlemezsedés csak néhány példa ezekre a betegségekre, amelyek komoly kockázatot jelentenek az emberi szervezet számára.

Azonban a megelőzés kiemelt fontossággal bír a szívbetegségek terén. A tudomány és az orvostudomány fejlődése révén ma már számos módszer áll rendelkezésünkre a szívbetegségek megelőzésére és kezelésére. Az egészséges életmód, az étrend, a rendszeres testmozgás és a stresszkezelés mind olyan tényezők, amelyek kulcsszerepet játszanak a szívbetegségek kockázatának csökkentésében.

Számos tanulmány beszámól arról, hogy a halálesetek igen csak nagy százaléka szívbetegségekből származik. A Covid-19 világjárvány kitörése óta a 22 és 44 év közötti korosztályban a szívinfarktus okozta halálozás közel 30 százalékkal emelkedett. Emellett azt tapasztaljuk, hogy a fiatalok talán nincsenek kellően tájékozottak a kevésbé ismert szívbetegségek kockázati tényezőivel kapcsolatban. Például nem tudják, hogy az elektromos cigaretta ugyanúgy terhelheti a szívet, mint a hagyományos cigaretta. Ugyancsak káros hatással lehetnek a kábítószer-használat, az alvási zavarok és a krónikus betegségek, mint például a reumás ízületi gyulladás, a lupus a és a HIV-fertőzés. [2].

Egy 2017-es Európai szívbetegség jelentés szerint a szív- és érrendszeri betegségek

felelősek 1,8 millió halálesetért az Európai Unióban (EU), ami az összes haláleset 37%-át teszi ki az EU országában [3]. Elérhető bizonyítékok szerint mind az iszkémiás szívbetegség (IHD), mind pedig az agyi érkatasztrófák halálozási arányai általában magasabbak Közép- és Kelet-Európában, mint Északon, Délen és Nyugaton. Összességében az EU-ban 210 milliárd euróra becsülték a szívbetegségek éves költségét [4].

Az európai egészségügyi statisztikák szerint Romániában 153 953 haláleset köthető szív- és érrendszeri betegségekhez, és ezek az összes haláleset 59,3%-át teszik ki az országban (2015). A szív- és érrendszeri betegségekhez kapcsolódó összes haláleset 52,8%-a a férfi népességben és 66,4%-a a női népességben fordul elő. A szív- és érrendszeri betegségekből eredő szabványosított halálozási arány Romániában 954,8 mindkét nemre vonatkozóan, 1103,0 a férfiaknál és 841,3 a nőknél (100 000 lakosra vonatkoztatva). Másrésről, az 65 év alatti személyeknél a szabványosított halálozási arány 114,6, míg az 65 év felettiek esetében 140,423 (100 000 lakosra vonatkoztatva). Általában véve Romániának a legrosszabb mutatói vannak ebben a tekintetben az EU28 tagállamai között [5] .

Románia a halálozási ráta alapján az Európai Unió második legrosszabb országa Bulgária után. Az életkorra visszavezethető várható élettartam Romániában 73,8 év, a nőknél magasabb (77,5 év) a férfiakhoz képest (70,1 év). A legfrissebb adatok szerint a szív- és érrendszeri betegségek Romániában a halálozások fő okozói (az összes haláleset 35%-a), amit a daganatok (12% halálozás), a légzőrendszeri betegségek, emésztőrendszeri betegségek, fertőző betegségek, balesetek, mérgezések követnek. A nőknél a szív- és érrendszeri betegségekből fakadó halálozás magasabb, mint a férfiaknál (a szív- és érrendszeri betegségekből fakadó összes haláleset 53-56%-a nőkre esik), ugyanez elmondható az endokrin betegségekről, a táplálkozás és anyagcsere zavarokról. A férfiaknál a daganatok, mérgezések, emésztőrendszeri és légzőszervi betegségek okozta halálozási arány magasabb [6].

A megfelelő és az időben történő diagnosztizálás rendkívül nagy jelentőséggel bír. Ha időben fel tudjuk ismerni a betegségeket és rendellenességeket, és a szükséges intézkedéseket meg tudjuk tenni, számos élet megmenthető lenne. Egy tanulmány szerint Amerikában évente csaknem 200.000 ember élete megmenthető, az időben történő diagnosztizálással. [7]

Az elmúlt évtizedekben az EKG jel feldolgozása és elemzése terén jelentős fejlődés történt. Az algoritmusok és a számítógépes technológiák fejlesztése lehetővé tette, hogy az EKG

adatokat automatikusan kiértékeljék és részletes információkat nyújtsanak a szív állapotáról. Ez nemcsak időt takarít meg az orvosoknak, hanem hozzájárul a diagnózis pontosságához és a betegség kezelésének hatékonyságához.

Dolgozatom fő célja az orvosi diagnosztikai módszerek továbbfejlesztése, jelfeldolgozást és mesterséges intelligenciát alkalmazva. Célom, hogy időben felismerjük és diagnosztizáljuk azokat a betegségeket és rendellenességeket, amelyek veszélyeztethetik az életünket és a szeretteink életét.

A jelfeldolgozás lehetővé teszi az adatok alapos elemzését és a jelentős információk kiszűrését a zajos és komplex adatok között. Ez segít az orvosoknak az adatok átlátásában és a releváns mintázatok azonosításában, amelyek fontosak a betegségek korai felismeréséhez és hatékonyabb kezeléséhez.

A mesterséges intelligencia alkalmazása tovább erősíti ezt a folyamatot, és segítséget nyújt az adatok összefüggéseinek és mintázatainak azonosításában. A mesterséges intelligencia algoritmusainak segítségével képesek vagyunk olyan szinten nagy mennyiségű adatok feldolgozására, ami már messze átlépi az emberi kapacitás felső határát. Képesek vagyunk egy magasabb szinten felismerni azokat a jellemzőket, amelyek segítségével még megbízhatóbb diagnózist tudunk felállítani. Ezáltal az orvosok célzottabb kezeléseket és megelőző intézkedéseket alkalmazhatnak, elősegítve a szívbetegségek és más súlyos állapotok kialakulásának megelőzését.

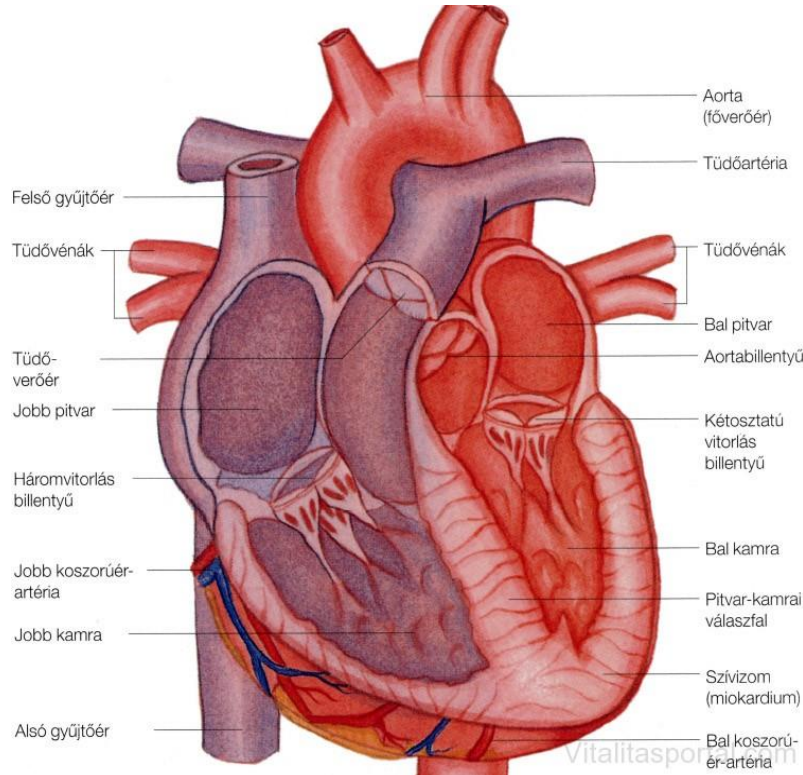
Projektem során konkrétan a négy fontos szívbetegség, a pitvarfibrilláció, pitvarlebegés, bal kamra magas feszültség (LVHV) és a TWC betegség felismerésére összpontosítok. Az ilyen betegségek időben történő diagnosztizálása kiemelkedően fontos, mivel megelőző intézkedésekhez folyamodhatunk és ezáltal elejét vehetjük a komolyabb szövődmények kialakulásának. A következőkben 12 elvezetési elektrokardiogrammok (EKG) kerülnek elemzésre és feldolgozásra, melyek a szív elektromos tevékenységét rögzítik. Ezeket az EKG jeleket használok fel a rendszer tanításához és a mesterséges intelligencia algoritmusokkal való modellezéshez. A kutatásomhoz két megbízható adatbázist használtam fel, a Maros Megyei Klinikai Kórház és a Chapman Egyetem, valamint a Shaoxi Klinika által közösen létrehozott adatbázisát.

2. Elméleti alapok

2.1.A szív

A szív rendellenes viselkedésének megértéséhez alapfeltétel, hogy – legalább alapfogalmak szintjén – ismerjük strukturális összetevőit, valamint funkcióját. A szív megfelelő működése alapfeltétele az élet fennmaradásának, hiszen az egész életünk során a vérkeringést a szív biztosítja. A szív egy elektromechanikus pumpa, amelyben spontán ritmikusan keletkező akciós potenciálok vezérlik a szisztolét (összehúzódás) és diasztolét (ellazulás) váltakozását. A szív megállása hacsak nem történik azonnal kardiopulmonális újraélesztés - a személy életének végét jelenti [8].

Ezen fontos szerv megfelelő működése számos feltételt von maga után – szükséges, hogy egy megfelelő és sértetlen felépítésű, kompetens izomstruktúrával rendelkező többüregű pompa minden fala és ürege a helyes sorrendben, a megfelelő ritmussal és frekvenciával húzódjon össze úgy, hogy a vér áramlása nem akadályozott, és hatékonyan tudjon zajlani (helyes



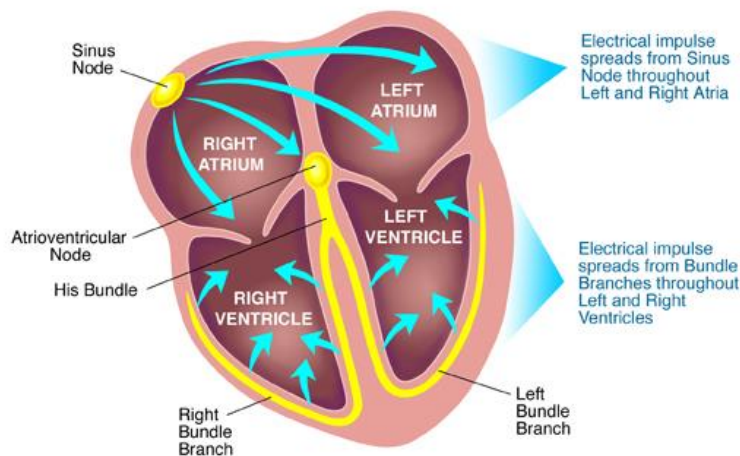
Ábra 1 : A szív szerkezete [9]

billentyűműködés) -, mely minden élethelyzethez a megfelelőképpen tud alkalmazkodni, a szükségleteket kielégítően ellátni.

A felső ábrán látható a szív sémás szerkezete. Dolgozatomban ezt csak érintőlegesen tárgyalom, lévén, hogy kizárólag a szív elektromos tevékenységét vizsgáltam.

A szív négy üregből áll (két pitvar, két kamra), melyek jobb- és balszívfélre oszlanak a sötvények (septumok) által. A pitvarokba érkezik a vér a vénás keringésből: a jobb pitvarba a teljes testből gyűjtött oxigénszegény, széndioxidban gazdag vér; a bal pitvarba a tüdőből érkező, oxigéndús vér. A pitvarokból a vitorlás billentyűkön keresztül (bal oldalon a mitralis, jobb oldalon a tricuspidalis billentyű) a kamrafalok ellazulása átszívja a vért a kamrák üregébe, ahonnan pedig: a jobb kamrából a pulmonalis billentyűn keresztül a tüdőbe jut az oxigénszegény vér gázcserére a tüdőverőereken keresztül; a bal kamrából az aortabillentyűn keresztül a főverőérbe (aortába) toódik a vér, ami a teljes szervezetben osztja szét az oxigéntelített vért. A vér pumpálásának ütemét egy speciális szövet és sejtszisztéma biztosítja, melyet a következő szegmensben tárgyalok.

Az emlős – és így az ember szívizomsejteket (myocytákat) két nagyobb csoportra osztjuk elektromos jellemzőik alapján. Az egyik csoportot a nodális szövet alkotja, amely magában foglalja a sinuscsomó (SA, sinoatriális csomó) és az atrioventricularis (pitvarkamrai, AV-csomó) rész sejtjeit. A nodális szövet sejtjeire jellemző, hogy spontán depolarizációval rendelkeznek, vagyis nincs valódi, tartós nyugalmi potenciál, a membránpotenciáluk folyamatosan fluktuál. A másik csoportba tartoznak a pitvari és kamrai izomsejtek, valamint az ingerületvezető rendszer azaz a His-Purkinje rendszer: His-köteg, Purkinje-rostok, Tawara-száraz. Ezen sejtekből álló rendszer felel a szív ritmikus kontrakciójáért és elernyedéséért [8].



Ábra 2 : A szív vezető és ingerület képző rendszerként [10]

Az felső ábrán látható a különböző részegységeinek elhelyezkedése a nodális szövetnek. A főbb ingerképző egység a szinuszcsomó (sinoatrialis csomó), melynek sejtjei generálják a vezérlőideg-ingerületet, amelyet még pacemaker potenciálnak is nevezünk. Az AV-csomó (atrioventricularis csomó), a His-köteg és a Tawara-szárok (Bundle Branch) egység is képesek vezérlőideg-ingerületet generálni, azonban ezek frekvenciája jóval kisebb, mint a szinuszcsomó frekvenciája, így normál működés során ezek a rendszerek nem adnak vezérlőjelet [11].

A szinuszcsomó általában percnként átlagosan 70 impulzust generál nyugalmi állapotban, mely impulzusok először haladnak végig az atrium falán, ösztönözve a munkaszövet összehúzódását, majd eljutnak az AV-csomóhoz, amely átvezeti az ingerületet a kamrák ingerületvezető rendszeréhez. Ez a rendszer viszonylag lassú, 30-40 impulzust generál percnként, és a lassabb működés biztosítja, hogy a kamrák csak a pitvarok aktiválódása után húzódnak össze.

2.2.Elektrokardiográfia (EKG)

A biometria olyan technológia, amely az emberek személyazonosságát automatikusan azonosítja fizikai és viselkedési jellemzőik alapján. Azonban hasonló módon alkalmazható az orvosi diagnosztikában is, mivel a mért jelek jellemzői bizonyos betegségek vagy elváltozások

esetén különböző kategóriákba sorolhatók. A biometrikus vizsgálatok során alkalmazott jellemzők közé tartozik az arcfelismerés, az írisz azonosítás, a járás elemzése, a verejtékpórusok vizsgálata, az ujj-érzékelés, a tenyérlenymat és tenyérérzékelés, a hanganalízis, a kézzel írt aláírás vizsgálata, valamint az elektrokardiogram (EKG).

Az elmúlt időszakban számos biometriai alkalmazást fejlesztettek az elektrokardiogram felhasználásával, amelyek viszonylag jó teljesítményt mutattak. A mérési folyamat bonyolultsága jelentősen csökkent, és a mérőeszközök kis méretű, könnyű és viselhető formában elérhető termékké váltak.

2.2.1. Meghatározás

Az EKG (elektrokardiográfia) egy non-invazív eljárás a diagnosztikában, amely hasznos információkat nyújt a szív állapotáról, működéséről. Az EKG az elektromos jelenségeket vizsgálja, amelyek a szívizom összehúzódásakor bekövetkező elektromos feszültség változásaként jelennek meg. Ez a változás egyedi, mivel minden embernek eltérő testalkata van. A szív összehúzódása elektromos impulzus hatására történik, amely általában a szinuszcsomóból ered, majd a szív sajátos ingerületvezető rendszerén keresztül jut el a szívizomsejtekhez. Ezt az elektromos impulzust fel lehet jegyezni a test felszínén elhelyezett elektródok segítségével. Az elektrokardiográfiai hullám egy szabályos görbe, amelynek sajátos jellemzői vannak [12].

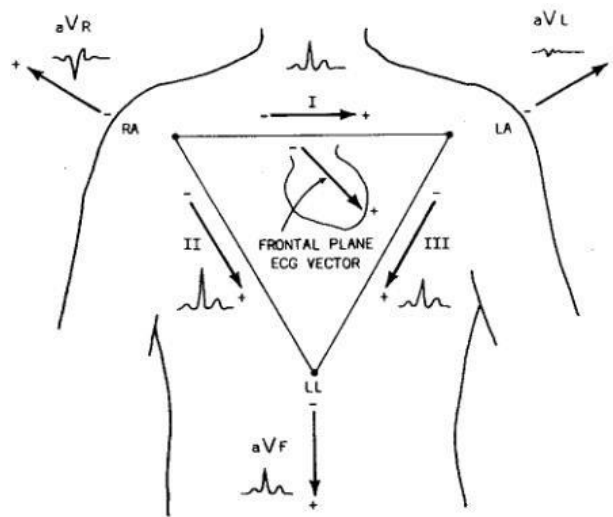
2.2.2. Az EKG célja

Az elektrokardiográfia (EKG) segítségével értékelni lehet a szívritmust, az akciós potenciál normál terjedését és más fontos paramétereket. Az EKG analízisével meg lehet állapítani a szívritmust, az ingerületvezetés sebességét, az impulzus kezdeti helyét, a szívizom vastagságát, a szívizom vérellátásával kapcsolatos lehetséges problémákat, valamint következtetéseket tudunk levonni az esetleges tüdőembóliával kapcsolatosan [12].

2.2.3. Az EKG-vizsgálat

A szív elektromos aktivitását vizsgálva következtethetünk a szív állapotára [12]. A szív jeleinek méréséhez nem elegendő egy vezetéket használni, mivel ez egy komplex, háromdimenziós szerv, amelynek elhelyezkedése és mérete egyénenként változik (egy adott tartományon belül). A teljes működés vizsgálata érdekében teljes képet kell kapnunk a szívizom polarizációjáról és depolarizációjáról. Így végső kiértékeléshez szükség van a szív elektromos működésének háromdimenziós térképezésére, amelyet a hagyományos, 12 vezetékes EKG mérés és modern jellemzőfeldolgozási és szűrőalgoritmusok valósítanak meg.

Az EKG-gép mellkasi és végtagi elektródákból gyűjti össze az elektromos jeleket. A mellkasi elektródák a horizontális sík tengelyeit, míg a végtagi elektródák a frontális sík elektromos tengelyeit rögzítik. A vizsgálathoz általában négy végtagi és hat mellkasi elektróda használatos, amelyekből az EKG-készülék hat alapvető elvezetést (I, II, III, aVF, aVR és aVL) hoz létre a végtagokról, valamint hat különböző elvezetést (V1, V2, V3, V4, V5 és V6) a mellkasról. Mindegyik elvezetés a szívet más szögből vizsgálja [12].



Ábra 3 : Einthoven háromszög, mérési vektorok és a normál jelformák

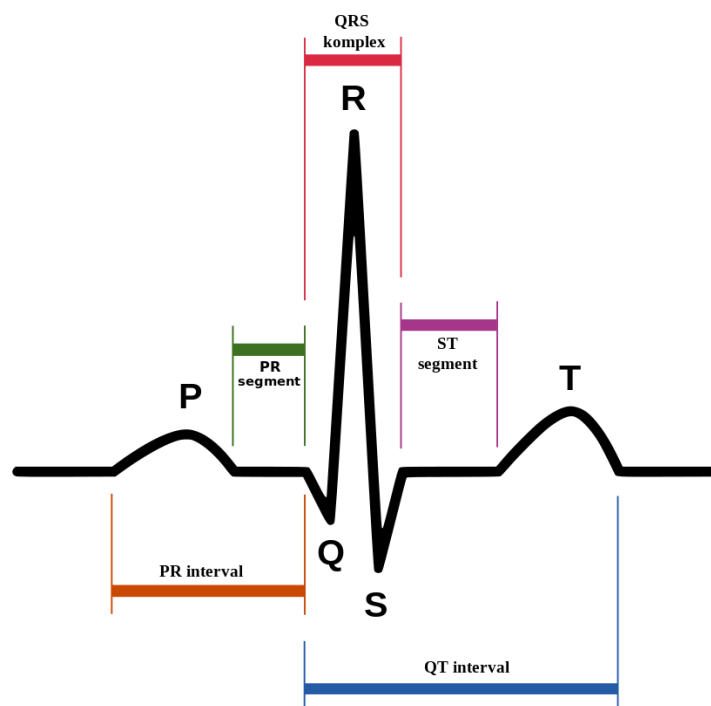
A modern EKG felvételek elkészítéséhez a szívet bipólusként kezeljük, ahol a depolarizált állapot az aktív, míg a polarizált állapot a passzív állapotot jelenti. A mérés lényege az áthaladó elektromos impulzusok összesítése a szívben. Ennek eredményeként követhetjük nyomon a szinuszcsomóból induló jelet az ingerületvezető pontokon keresztül, és információt kaphatunk a szív kamráinak időbeli helyzetéről, a kontrakciók és relaxációk intenzitásáról, ritmusáról és szinkronizációjáról.

A végső jel kialakulása az ezekből az jelekből származó összegzés eredményeként történik, ahogy az ábrán is látható. A nodális és az ingerületvezető szövet megfelelő pontjain áthaladó polaritásváltozások összessége alkotja azokat a jeleket, amelyeket az orvostudományban P-QRS-T hullámokként ismernek.

2.2.4. Az elektrokardiográfiai görbe

Az elektrokardiográfián észlelhető hullámok neveit, **P, Q, R, S, T** és **U** betűkkel jelöljük, melyeket egységesen használunk nemzetközi szinten. Minden hullám az EKG-on szív meghatározott részének depolarizációját (elektromos kisülését) vagy repolarizációját (elektromos újratöltődését) jelképezi [12].

A **Pitvari hullám** (P-hullám) pozitív amplitudóval rendelkezik, az impulzus pitvari terjedésének felel meg, melynek időtartama 0,06 - 0,11 másodperc. A P-Q távolság a kamra és pitvar közötti átvezetési idő, ami 0,12 - 0,2 másodperc. A **Q R S - komplex** a kamrák depolarizációját mutatja, tartalmaz egy kis negatív Q-hullámot, magas pozitív R-hullámot és negatív S-hullámot. Időtartama 0,06 - 0,12 másodperc, és a kamrák teljes munkaizomzatának depolarizációját jelenti. Az S T - szakasz a lassú repolarizációs szakasza a kamráknak. A **T - hullám** közepes amplitudóval rendelkezik és hosszabb ideig tart, jelzi a teljes repolarizációját a kamráknak, ennek időtartama 0,20 másodperc. A Q - T távolság a depolarizációjának és repolarizációjának együttes időtartama a kamraizomzatnak és. Az **U - hullám** forrása bizonytalan, viszont lehet az interventricularis septum repolarizációjának vagy a lassú újratöltődése a kamráknak jele [13] [12].



Ábra 4: EKG görbe [13]

2.2.5. A szívritmus, sinusritmus

A biztonságos szívritmus meghatározásához elengedhetetlen egy hosszabb elektrokardiográfiai-csík alapos elemzése a választott elvezetésben, amely általában a II-es elvezetés (amelyet a dolgozatomban is használni fogok). A szívritmus rendellenességei általában akkor válnak észrevehetővé, amikor legalább 12 vagy ennél több egymást követő komplexust vizsgálunk. [12].

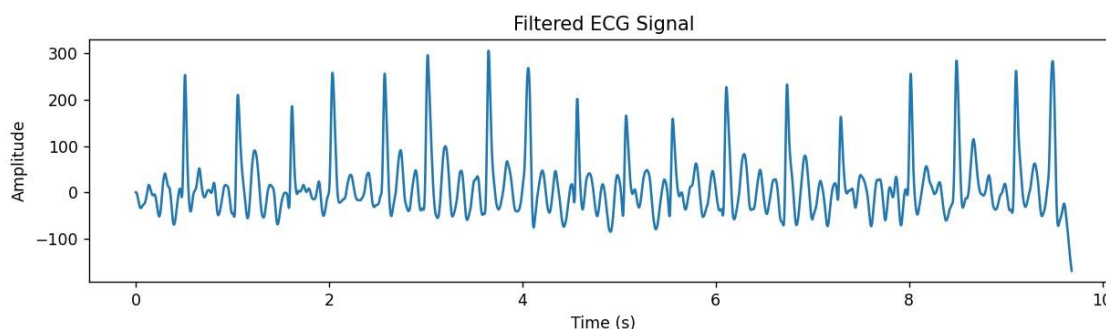
A szinuszritmus az a normális szívritmus, amelyben a sinuatrialis csomó azaz a szinuszcsozó, SA-csozó, Keith-Flack-csozó természetes pacemakerként működik, és a szívfrekvencia 60 - 100 ütés/ perc között van. Jellemzője, hogy ilyenkor a szívverések száma 60 -100/perc, a P-hullám felfelé mutat a II-es elvezetésben és lefelé aVR-ben, valamint mindegyik P - hullámot Q R S - komplexus követi. [12].

3. Gyakorlati megvalósítás

3.1. Projekt leírása

A projektben szeretnék egy olyan szoftvert létrehozni, amely mélytanulású neuronhálót alkalmazva segítséget nyújt az orvosi elektrokardiológiai (EKG) jelek kiértékelésében. Ennek révén a rendszer képes lesz konkrét szívbetegségek felismerésére a betegek EKG jeleinek elemzése alapján. Az alábbiakban bemutatásra kerülnek az általam vizsgált betegségek típusai:

- **Atrial Flutter (Pitvarlebegés)** : egy olyan szívritmuszavar, amelyben a pitvarok rendszertelenül és gyorsabban vernek, mint a szokásos. Ez a betegség a pitvarok

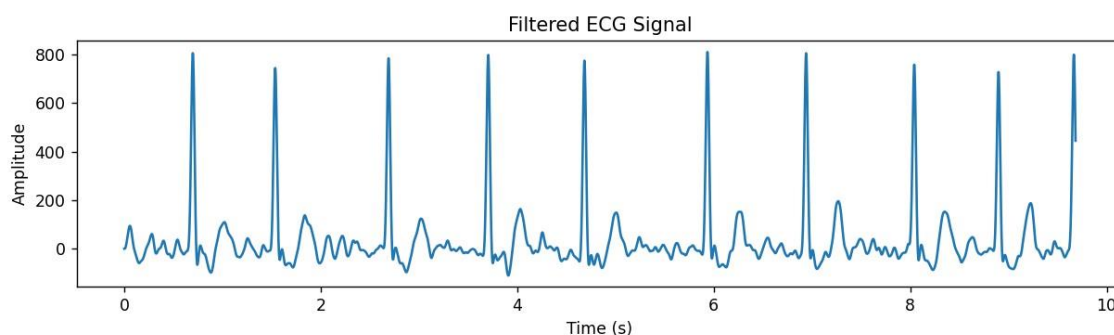


Ábra 5: Atrial Flutter

- elektromos rendszerének zavarai miatt alakul ki, amelyek a szív normális elektromos impulzusát irányítják. A pitvarlebegés során a pitvarokban létrejön egy körkörös elektromos impulzus, ami megzavarja a szív szabályos működését. Ennek eredményeként a szív kamrái is rendszertelenebben vernek, és csökkenthetik a szív teljesítményét. A pitvarlebegés tünetei változóak lehetnek, és lehetnek olyan esetek, amikor a beteg nem is észleli a ritmuszavart. Azonban más esetekben a betegek panaszokat tapasztalhatnak, mint például szapora szívverés, szívritmuszavar érzete, szédülés, légszomj, mellkasi fájdalom vagy gyengeség. Fontos az orvosi kezelés és rendszeres orvosi ellenőrzés a pitvarlebegés esetén, mivel a betegség hosszú távon szövődményeket okozhat, például

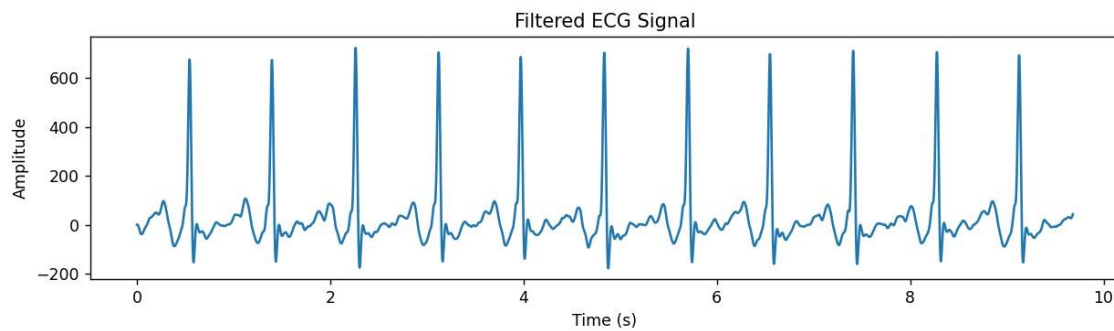
vérrögök kialakulását, szívelégtelenséget vagy stroke-ot.

- **Atrial Fibrillation (Pitvarfibrilláció)** : egy olyan szívritmuszavar, amely során a szív nem rendszeresen húzódik össze, így a pulzus tapintásakor nem tapasztalható a szokásos lüktetés, hanem inkább ritmuszavaros pulzációt érezhetünk. Normális alap esetben a szív ritmikus működését a szinuszcsomó szabályozza. Pitvarfibrilláció esetében a szívpitvarban több elektromos impulzus keletkezik, amelyek a pitvar-kamrai csomóhálózaton keresztül eljutva a szívkamrákhoz, szabálytalan szívösszehúzódásokat váltanak ki. A pitvarfibrilláció lehet folyamatos, átmeneti vagy időszakos jellegű, és a tünetek súlyossága is változhat ezáltal. Ezen betegség kezelésének egyik legfontosabb célja a stroke (agyi érkatasztrófa) megelőzése. Ezért a legtöbb pitvarfibrillációban szenvedő személynek vérröggképződést megakadályozó gyógyszereket kell szednie. [14]



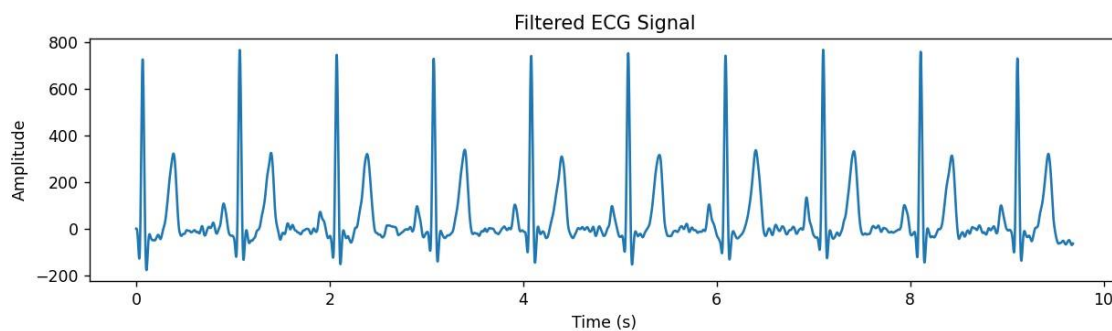
Ábra 6 : Atrial Fibrillation

- **Left ventricle high voltage (bal kamra magas feszültség)**: ez egy elektrokardiográfiai (EKG) jellemző, amely arra utal, hogy az EKG felvételen a bal kamra elektromos aktivitása fokozott vagy magas feszültséget mutat. Ez a jelenség lehet fiziológiás (normális variáció) vagy patológiás (egyéb szívbetegség jele). Az LVHV tehát inkább egy leíró jellemző, amely azt mutatja, hogy a bal kamra elektromos aktivitása erősebb vagy magasabb feszültséget mutat az EKG-n. A bal kamra a szív bal oldalán található, és felelős a vér kipumpálásáért a test számos területére.



Ábra 7 : Left ventricle high voltage

- TWC (T Wave Change)** : egy olyan elektrokardiográfiás (EKG) eltérés, amely a T hullám megváltozott alakját jelenti. A T hullám az EKG görbájén látható elektromos jelek egyik része, és az izmok elektromos aktivitását tükrözi a szívben. A TWC általában különböző szívbetegségekkel vagy rendellenességekkel társulhat, például szívizomkárosodással, elektrolit-egyensúlyhiánnyal vagy gyógyszerhatásokkal. A TWC fontos jelző lehet a szívbetegségek diagnózisában és a betegség progressziójának figyelemmel kísérésében. Az EKG vizsgálat segítségével az orvosok értékelhetik a TWC jelenlétét, mértékét és jellegét, amely segíthet a betegség azonosításában és a megfelelő kezelés kiválasztásában. A betegségben a T hullám, amely az EKG mérés során a szív elektromos aktivitásának egy részét mutatja, módosul vagy torzul. Ez a szívbetegség általában más szívrendellenességekkel vagy szívizom károsodással társulhat. A T wave Change szívbetegség jelei és tünetei változóak lehetnek, és az érintettek különböző szívproblémákkal küzdenek, mint például a szívritmuszavarok vagy a szívizom elégtelenség.



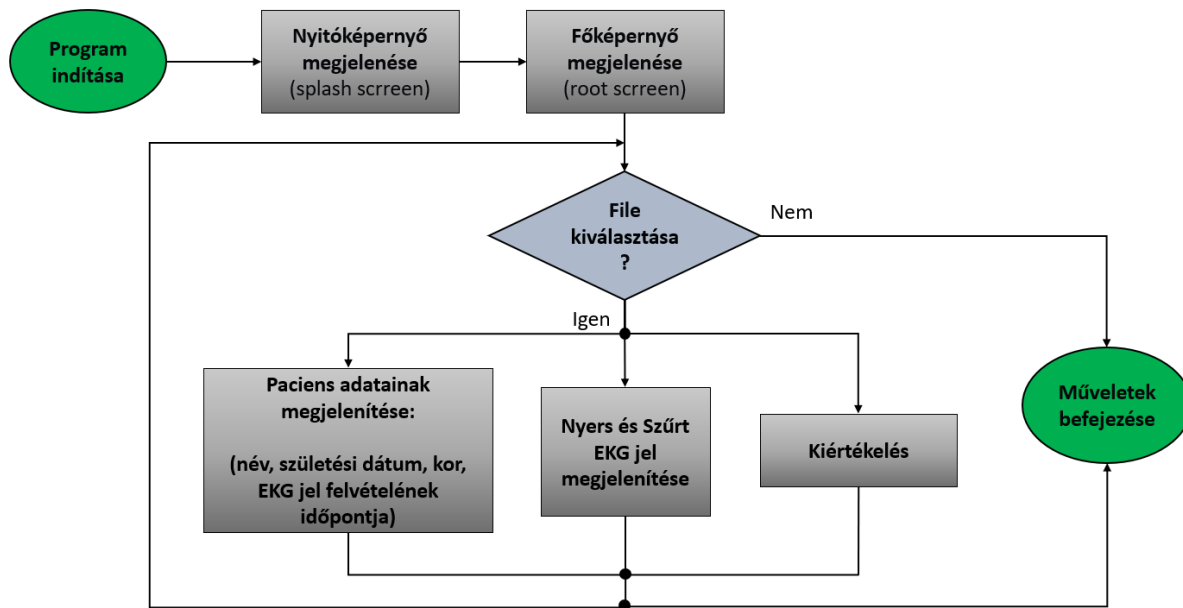
Ábra 8: T Wave Change

3.2. Felhasználói követelmények

A szoftverem olyan funkciókkal rendelkezik, amelyek megfelelnek az alábbi felhasználói követelményeknek:

- **Grafikus felület biztosítása:** A szoftver rendelkezik egy felhasználóbarát grafikus felülettel, ahol a felhasználó különböző funkciókat tud végrehajtani. Ez lehetővé teszi a könnyű navigációt és az intuitív használatot.
- **Állományok megnyitása:** A felhasználó képes megnyitni állományokat a szoftveren belül. Az állományok kizárólag .csv kiterjesztésűek lehetnek. Miután az állományt megnyitotta, az aktuálisan megnyitott fájl megjelenik a felületen.
- **Szüretlen és szűrt EKG jel megjelenítése:** Az állomány megnyitása után a szoftver megjeleníti a szüretlen és a szűrt EKG jelet. Az EKG jelek 12 elvezetéses EKG-ból származnak, és a második elvezetést jelenítik meg a felhasználónak.
- **EKG tulajdonosának adatai:** A szoftver megjeleníti az EKG felvétel tulajdonosának nevét, korát és pontos időpontját, amikor a felvétel készült. Ez segít az azonosításban és a betegadatok könnyű elérésében.
- **EKG tulajdonosának adatai:** A szoftver megjeleníti az EKG felvétel tulajdonosának nevét, korát és pontos időpontját, amikor a felvétel készült. Ez segít az azonosításban és a betegadatok könnyű elérésében.
- **EKG jel kiértékelése:** A szoftver képes kiértékelni az EKG jelet, és információkat nyújtania szívritmusról, szabálytalanságokról vagy más jellemzőkről. Ez segít a felhasználónak a diagnózis készítésében és az EKG adatok értelmezésében.

Ezen funkciók révén a szoftver megfelel a felhasználók által támasztott követelményeknek, lehetővé téve az EKG jelek elemzését, a betegadatok megjelenítését és a szívritmuszavarok detektálását



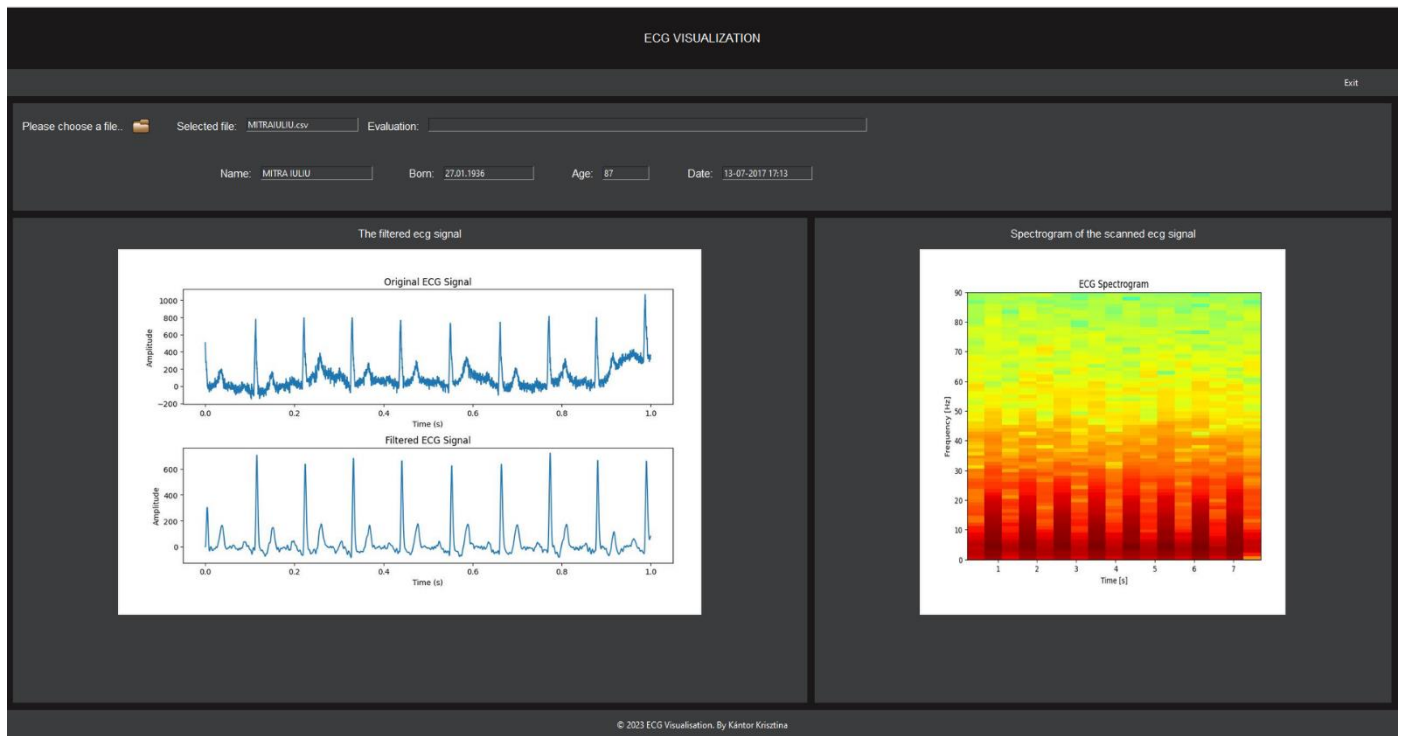
Ábra 9: Működést bemutató folyamatábra

Az elkészült szoftver jelenleg három fő ablakot és egy EXIT gombot tartalmaz. Az első ablak lehetővé teszi a felhasználó számára, hogy kiválassza a kívánt .csv fájlt, amelyet a rendszer feldolgoz és megjelenít. Az ablakban alapvető információkat kapunk az EKG jel tulajdonosáról, mint például a név, születési év, kor és a felvétel készítésének pontos ideje. Emellett az "Evaluation" szövegdoz is található az ablakban, amely a jel kiértékelését jeleníti meg.

A második ablak (a bal alsó sarokban található) lehetővé teszi a jel megtekintését a fájl beolvasása után, mind nyers (zajos), mind szűrt formában.

A harmadik ablak (jobb alsó sarok) pedig a szűrt jel spektrogramját jeleníti meg.

Ezen funkciók mellett még egy kilépő gomb található: az "Exit", amellyel kiléphetünk a programból.



Ábra 10: Felhasználói felület

3.3. Funkcionális követelmények

1. EKG jel feldolgozása: A rendszernek képesnek kell lennie az EKG jelek beolvasására és feldolgozására az adatbázisból vagy külső forrásokból.
2. Szűrők alkalmazása: A rendszernek képesnek kell lennie különböző szűrők alkalmazására az EKG jelek javítása és zaj csökkentése érdekében.
3. Spektrogram létrehozása: A rendszernek képesnek kell lennie a spektrogramok generálására az EKG jelekből, amelyek vizualizálják az idő-frekvencia tartományban megjelenő jellemzőket.
4. Mélytanulás alkalmazása: A rendszernek képesnek kell lennie a mélytanulás algoritmusok használatára az EKG jelek automatikus kiértékeléséhez és esetleges rendellenességek, szívbetegségek azonosításához.
5. EKG jel kiértékelése: A rendszernek képesnek kell lennie az EKG jelek elemzésére és

kiértékelésére, valamint az eredmények megjelenítésére vagy továbbítására az egészségügyi szakemberek számára.

3.4. Választott programozási nyelv, környezet

A szoftverem fejlesztése során a Visual Studio Code fejlesztői környezetet és a Python programozási nyelvet választottam. A Visual Studio Code egy népszerű és erőteljes fejlesztői eszköz, amely számos előnyös funkciót és integrációt kínál a fejlesztési folyamat során. A Python programozási nyelv pedig jól ismert a könnyű tanulhatósága és olvashatósága miatt, valamint a széleskörű támogatottsága és gazdag modulválasztéka miatt ezen kívül pedig világszerte a mesterséges intelligenciával kapcsolatos fejlesztések jelentős része Python nyelven történik.

A felhasználói felület kialakításához a Tkinter nevű beépített Python könyvtárat használtam. A "Tkinter" rövidítés a "Tk interface" kifejezésből ered, és arra utal, hogy a könyvtár az eredeti Tk (Toolkit) grafikus rendszert használja. A Tkinter segítségével könnyedén létrehozhatunk interaktív és felhasználóbarát felületeket, amelyek tartalmazhatnak gombokat, szövegdobozokat, címkéket, listákat és más elemeket. A Tkinter rendkívül elterjedt és népszerű a Python fejlesztők között, mivel könnyen használható és rendkívül rugalmas. Emellett a Tkinter támogatja az eseményvezérelt programozást, így könnyedén kezelhetjük a felhasználói interakciókat és reagálhatunk az eseményekre.

A Python és a Tkinter kombinációjának előnyei közé tartozik a könnyű tanulhatóság és a gyors fejlesztési idő. A Python nyelv és a Tkinter könyvtár széleskörű dokumentációval és közösséggel rendelkezik, így könnyen hozzáférhető forrás és segítség állt rendelkezésemre a fejlesztési folyamat során.

A tanítás során pedig a TensorFlow keretrendszert használtam, mely egy nyílt forráskódú, gépi tanulásra és mély tanulásra szolgáló keretrendszer, amelyet a Google fejlesztett ki. A TensorFlow kifejezetten hatékonyan támogatja a nagy adathalmazok kezelését és a bonyolult neurális hálózatok kialakítását [15]. Python nyelven íródott, így ezen nyelv rugalmassága és egyszerűsége lehetővé teszi a kódrészletek könnyű megírását és olvashatóságát, valamint a tudományos számításokhoz szükséges egyéb könyvtárak és eszközök könnyű

integrálását. Kiterjedt eszközkészletet biztosít a modell értékelésére és optimalizálására. Például lehetőség van a veszteségfüggvények és optimalizáló algoritmusok testreszabására, a hiperparaméterek hangolására és a teljesítmény mérésére. Emellett a TensorFlow tartalmaz eszközöket a modell kimentésére és újrahasznosítására későbbi felhasználásra, tehát igen nagy mértékben hozzájárult a tanítási modellem gyors felépítéséhez és használatához.

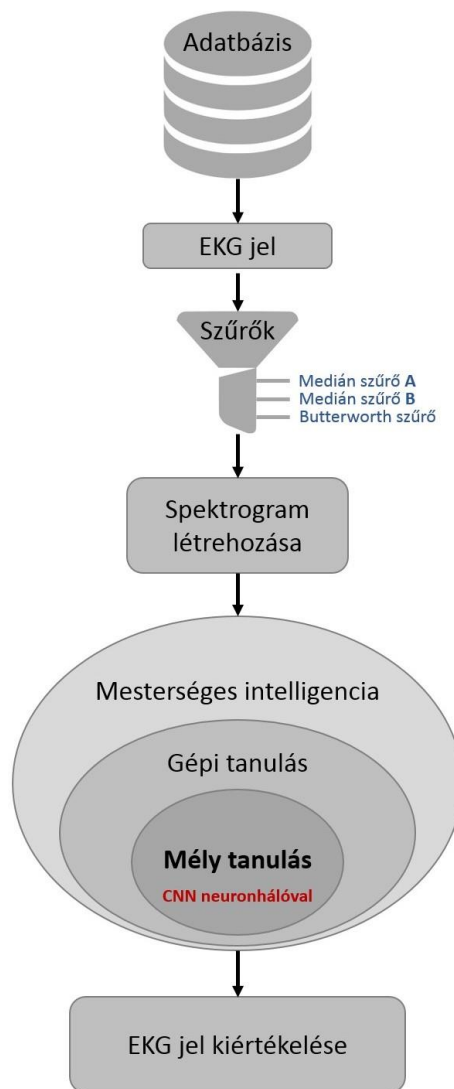
4. Gyakorlati megvalósítás

A rendszerem gyakorlati megvalósítása több komponensből áll (12. Ábra), amint azt a rendszer tömbvázlatán is láthatjuk. Az első komponens az adatbázis. A projektem során két forrást is használtam, a Maros Megyei Klinika és a Champman University és a Shaoxi Klinika közös adatbázisát. Ezek a források biztosították az alapvető EKG adatokat, amelyekre a rendszerem felépült és amelyeket a további feldolgozási lépések során használtam.

Ezután következik az EKG jelek kinyerése, amely során az EKG jelet tartalmazó fájlból kiválasztom minden esetben a második elvezetést és előkészítem a további feldolgozásra.

A következő lépés a szűrők alkalmazása, amely az én esetemben a Median és Butterworth szűrők. Ezek a szűrők segítenek majd kiszűrni a zajt és javítani az EKG jelek minőségét.

Ezután a spektrogram létrehozása következik, amely egy olyan vizualizáció, amely megjeleníti az EKG jel időbeli és frekvenciabeli jellemzőit. Majd az EKG spektrogramjait felhasználva bemenetként, mélytanulást alkalmazva tanítjuk a rendszert, különböző betegségek felismerésére. Végül következik az EKG jel kiértékelése, amely magában foglalja az EKG jelek spektrogramjának elemzését és értelmezését az esetleges rendellenességek vagy betegségek felismerése érdekében.



Ábra 11 : A rendszer egyszerűsített tömbvázlata

5. Felhasznált adatbázisok

A projekt elkészítése során két adatbázist használtam, amelyek hozzájárultak a szoftver fejlesztéséhez és teszteléséhez. Az első adatbázis a Maros Megyei Klinikai Kórház által biztosított adatbázis volt, amely jelentős mennyiségű valós betegadatot tartalmazott. Ez az adatbázis lehetővé tette számomra, hogy valós klinikai eseteket vizsgáljak és a szoftveremet a

valós körülmények között teszteljem. A második adatbázis pedig a Champman University és a Shaoxi Klinika közös erőfeszítésével létrehozott nyílt adatbázis volt. Ez az adatbázis speciális kutatási célokra került létrehozásra. Ennek az adatbázisnak az használata lehetővé tette számomra, hogy még több adatot tudjak feldolgozni és még több összehasonlítást végezzek a szoftveremben alkalmazott algoritmusok segítségével.

5.1. Maros Megyei Klinikai Kórház adatbázisa

Dolgozatom során első részben a Maros Megyei Klinikai Kórház adatbázisából származó 12 elvezetési EKG jeleket dolgoztam fel.

Az adatok beszerzése és feldolgozása időigényes feladat volt, mivel minden egyes EKG jelet külön meg kellett keresnünk a klinika adatbázisaiban a betegség alapján, majd végső diagnózist is végrehajtottunk annak érdekében, hogy biztosan pontos és megbízható jeleket dolgozzunk fel. Ez a folyamat jelentős időt és erőforrásokat igényelt, de végül egy értékes adatbázist kaptunk.

Az EKG jelek mintavételezési frekvenciája 500 Hz volt, ami azt jelenti, hogy másodpercenként 500 mintapontot rögzítettek az EKG felvétel során. Az EKG jelben összesen 4000 mintapont található, amelyek a szív elektromos aktivitásának változásait tükrözik. Ennek alapján könnyen kiszámítható, hogy a felvétel időtartama hány másodpercet jelent. Az időtartam meghatározása a mintavételezési frekvencia és a mintapontok száma alapján történik, és minden felvétel esetén 8 másodpercet eredményez.

Minden esetben a második elvezetésre összpontosítottam és azt elemeztem. Az elvezetés kiválasztásához egy algoritmust használtam, amely kinyeri a kiválasztott .cs fájl teljes második elvezetését. Az algoritmus eredményét egy új fájlba mentettem, így lehetőséget teremtve a további műveletek elvégzésére, amelyeket ezen az elvezetésen kell végrehajtanom.

Az első 19 sor minden ilyen .csv fájl esetében fontos információkat tartalmaz, mint például a mintavételezési frekvenciát, a mintapontok számát, valamint az EKG tulajdonosának nevét, korát és születési évét.

```

def process_data(input_file, output_file):
    if not os.path.exists(input_file):
        print("Error: The input file does not exist!")
        return

    try:
        with open(input_file, "r") as f:
            lines = f.readlines()

        results = []
        for i, line in enumerate(lines):
            if i >= 19:
                line = line.strip()
                values = line.split(";")
                if len(values) > 1:
                    results.append(values[1])

        with open(output_file, "w") as f:
            f.write("\n".join(results[:4000]))

        print("Processing completed. Results saved to:", output_file)
    except Exception as e:
        print("An error occurred during processing:", str(e))

```

Ábra 12: Adatfeldolgozás - kódrészlet

5.2. Champman University és a Shaoxi Klinika közös adatbázisa

Ez az újonnan létrehozott kutatási adatbázis a Champman Egyetem és a Shaoxing Kórház (Shaoxing Hospital Zhejiang University School of Medicine) közreműködésével jött létre, célja, hogy lehetővé tegye a tudományos közösség számára az aritmia és más kardiovaszkuláris állapotokkal kapcsolatos új tanulmányok végzését.

Ez az új adatbázis, amely 10,646 páciens 12-vezetékes EKG-jét tartalmazza, 500 Hz mintavételezési frekvenciával, 11 gyakori ritmus és 67 egyéb kardiovaszkuláris állapot címkézésével, szakértők által. Ebben az esetben is az EKG jelek mintavételezési frekvenciája 500 Hz, viszont EKG jelben összesen 5000 mintapont található, tehát egy 10 másodperces felvételeket tartalmaz az adatbázis (12 dimenziós, 5000 mintát tartalmazó minta). Az adathalmaz használható aritmia és más kardiovaszkuláris állapotokra fókuszáló tanulmányokban új és

klasszikus statisztikai és gépi tanulási technikák tervezésére, összehasonlítására és finomhangolására. [16]

Ez az adatbázis már nem tartalmaz betegadatokat, csupán a 12 elvezetést. Ugyanúgy kiválasztottam a második elvezetést, ehhez is külön algoritmust írtam, ami csak kissé volt másabb, mivel az adatok nem szerepelnek a kezdetén.

Ahogy korábban említettem, ez az adatbázis hatalmas mennyiségű adatot tartalmaz. Az én fókuszom a négy fő szívbetegségre irányult, így ezekre koncentráltam az adatok kiválasztásakor. Ez egy hosszabb folyamat volt, mivel gondosan meg kellett válogatnom, mi az, ami számomra releváns információt nyújt.

A fő mappában, amelyben ezek az EKG jelek voltak, nem volt semmilyen előre meghatározott csoportosítás. Ezért először is csoportosítanom kellett ezeket a betegségeket. Ehhez a Diagnostics.xlsx nevű fájlt használtam, amely tartalmazta az EKG fájlok nevét, néhány adatot róluk, valamint a betegségek nevét. Aztán összehasonlítottam ezeket a fájlneveket az oldalról letöltött mappa fájlneveivel annak érdekében, hogy megbizonyosodjak arról, hogy léteznek. Majd külön mappába rendeztem azokat a fájlokat, amelyek számomra relevánsak voltak.

Ez a csoportosítási és rendezési folyamat segített nekem megtalálni és rendszerezni azokat az adatokat, amelyekre szükségem volt a további elemzéshez és vizsgálatokhoz. Ezeket a "Diseases" mappában tároltam el. Annak ellenére, hogy az adatbázis nagy és viszonylag sokféle betegséget tartalmaz, nagyobb az eloszlás is. A számomra szükséges EKG jeleket nézve, betegségenként ez 100-100 jelet jelent.

inputs > MUSE_20180111_155710_23000.csv

```
1 I,II,III,aVR,aVL,aVF,V1,V2,V3,V4,V5,V6
2 -102.48,-156.16,-53.68,131.76,-24.4,-107.36,112.24,48.8,58.56,-97.6,-126.88,-131.76
3 -82.96,-112.24,-29.28,97.6,-29.28,-73.2,122.48,8.58,56,-87.84,-122,-117.12
4 -58.56,-78.08,-19.52,68.32,-19.52,-48.8,131.76,48.8,58.56,-78.08,-112.24,-107.36
5 -73.2,-82.96,-9.76,78.08,-34.16,-48.8,126.88,29.28,53.68,-87.84,-117.12,-117.12
6 -73.2,-68.32,4.88,73.2,-39.04,-34.16,122.29,28.48,8,-87.84,-122,-117.12
7 -68.32,-63.44,4.88,68.32,-39.04,-29.28,126.88,43.92,43.92,-78.08,-112.24,-102.48
8 -73.2,-68.32,4.88,73.2,-39.04,-34.16,122.43,92.39,04,-82.96,-122,-102.48
9 -63.44,-68.32,-4.88,68.32,-29.28,-39.04,122.53,68.48,8,-82.96,-112.24,-92.72
10 -53.68,-82.96,-29.28,68.32,-14.64,-58.56,122.48,8.48,8,-78.08,-112.24,-102.48
11 -24.4,-68.32,-43.92,48.8,9.76,-58.56,122.43,92.68,32,-73.2,-107.36,-102.48
12 -34.16,-58.56,-24.4,48.8,-4.88,-43.92,122.48,8.73,2,-58.56,-92.72,-92.72
13 -68.32,-53.68,14.64,63.44,-43.92,-19.52,122.58,56.68,32,-48.8,-92.72,-82.96
14 -58.56,-43.92,14.64,53.68,-39.04,-14.64,107.36,58.56,73.2,-48.8,-87.84,-78.08
15 -53.68,-48.8,4.88,53.68,-29.28,-24.4,92.72,53.68,73.2,-43.92,-87.84,-82.96
16 -58.56,-53.68,4.88,58.56,-34.16,-24.4,82.96,48.8,73.2,-43.92,-87.84,-82.96
17 -48.8,-29.28,19.52,39.04,-34.16,-4.88,82.96,39.04,73.2,-39.04,-82.96,-73.2
18 -48.8,-24.4,24.4,39.04,-39.04,0,92.72,39.04,53.68,-43.92,-82.96,-73.2
19 -34.16,-24.4,9.76,29.28,-24.4,-9.76,102.48,34.16,53.68,-48.8,-87.84,-73.2
20 -29.28,-19.52,9.76,24.4,-19.52,-4.88,107.36,34.16,63.44,-43.92,-87.84,-68.32
21 -19.52,-19.52,0,19.52,-9.76,-9.76,97.6,39.04,68.32,-43.92,-97.6,-78.08
22 -14.64,-14.64,0,14.64,-9.76,-9.76,82.96,39.04,68.32,-43.92,-87.84,-63.44
23 -43.92,-39.04,4.88,43.92,-24.4,-19.52,82.96,29.28,48.8,-48.8,-92.72,-68.32
24 -58.56,-34.16,24.4,48.8,-43.92,-4.88,87.84,29.28,43.92,-48.8,-87.84,-63.44
25 -107.36,-48.8,58.56,78.08,-82.96,4.88,97.6,24.4,48.8,-48.8,-87.84,-63.44
26 -97.6,-48.8,48.8,73.2,-73.2,0,117.12,19.52,39.04,-53.68,-97.6,-73.2
27 -58.56,-19.52,39.04,39.04,-48.8,9.76,126.88,29.28,43.92,-58.56,-87.84,-68.32
28 -24.4,4.88,29.28,9.76,-29.28,14.64,117.12,19.52,48.8,-48.8,-78.08,-48.8
29 -29.28,9.76,39.04,9.76,-34.16,24.4,82.96,19.52,39.04,-43.92,-73.2,-48.8
30 -29.28,-4.88,24.4,19.52,-29.28,9.76,58.56,14.64,34.16,-34.16,-63.44,-43.92
31 -9.76,14.64,24.4,0,-19.52,19.52,48.8,24.4,53.68,-14.64,-48.8,-34.16
32 9.76,34.16,24.4,-19.52,-9.76,29.28,43.92,39.04,58.56,-14.64,-43.92,-29.28
```

Ábra 13: Egy EKG jel a Chapman University és a Shaoxi Klinika közös adatbázisából

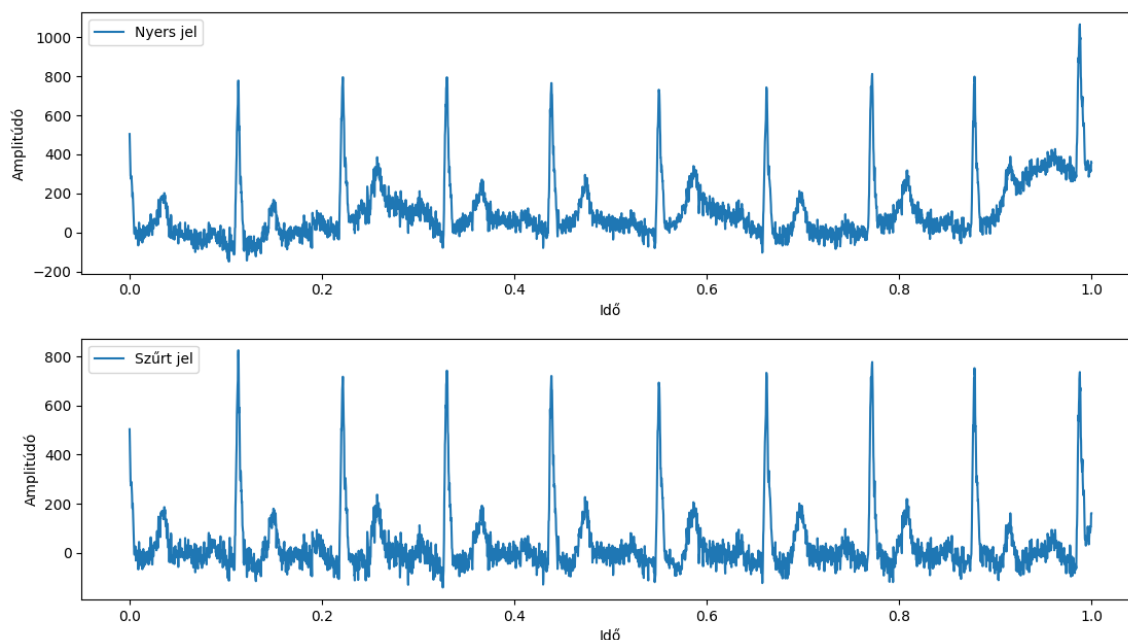
6. Zajok szűrése és normalizálása

A zajok jelenléte az EKG jelekben gyakran nehezítheti a szív elektromos aktivitásának pontos értelmezését. Ezeknek a kiszűrése segít kiemelni a szív valódi elektromos jeleit és javítja az EKG jelminőségét, igen fontos lépés a pontosabb diagnózis elérése érdekében. A zajok lehetnek külső elektromágneses interferenciák, mozgásokból eredő artefaktumok vagy más elektromos forrásokból származó interferenciák, ezt alapjel eltolódásnak is nevezzük. A szakirodalomban számos példát is találhatunk ennek a jelenségnek orvosolására, számos helyen alul áteresztő szűrővel küszöbölik ki ezt, aminek a vágási frekvenciája 0.5Hz, vagy akár **FIR** (Finite Impulse Response) szűrővel, amely egy digitális szűrő típus, lineáris időinvariáns rendszer, amit gyakran alkalmaznak zajszűrésre és jelek frekvenciatartománybeli alakítására.

Én viszont az alapjel eltolódást két **medián** szűrővel küszöbölttem ki, amire szintén találunk példát a szakirodalomban [17]. A medián szűrő egy nemlineáris digitális szűrés technika, amely gyakran használatos zaj eltávolítására képekről vagy jelekről. Az ilyen zajcsökkentés tipikus előfeldolgozási lépés, amely javítja a későbbi feldolgozás eredményeit

[18]. Ez egy hatékony módszer a digitális jel-feldolgozásban, amely segít kiszűrni az adatok közötti szélsőséges értékeket és hirtelen változásokat. A Python programozási nyelvben a SciPy könyvtárban található beépített `signal.medfilt` függvényt használtam a medián szűrés implementálásához. A `signal.medfilt` függvény paramétereiként a jelet és az ablak méretét kellett megadnom. Az ablak mérete általában páratlan szám, hogy a középső érték (medián) könnyen kiszámítható legyen (Pythonban az ablak mérete kizárólag páratlan szám lehet).

A kódomban két különböző ablakméretű medián szűrőt alkalmaztam. Az egyik ablak mérete 201, míg a másiké 601. A különböző ablak méretek lehetővé tették a jelem különböző aspektusainak és frekvenciáinak kezelését. A kisebb ablak méretek általában finomabb szűrést eredményeznek, míg a nagyobb ablak méretek hatékonyabban szűrik ki a hosszabb távú változásokat.



Ábra 14: Medián szűrővel szűrt jel

A medián szűrő alkalmazása fontos lépés volt a jel nagyobb zajainak kiszűrésében és a jel normalizálásában. A fent látható ábrán (15. Ábra) jól látszik, hogy a medián szűrő alkalmazása után az artefaktum eltűnt, és a jel egyenes vonalba került. Ez eredményeként sikeresen

kiküszöböltük a hibás zajt, ami javította a jelminőséget. A további zajszűrés érdekében alkalmaztam a **Butterworth-szűrőt** a jelre, amely egy sima átviteli karakterisztikával rendelkező szűrő típus. A Pythonban található SciPy könyvtárban elérhető butter és lfilter függvényeket használtam a Butterworth-szűrő létrehozására és alkalmazására. A függvény paraméterei között a filter_order meghatározza a szűrő rendjét, vagyis a frekvenciamenet lejtését. A cutoff_freq a vágási frekvenciát jelenti, ami meghatározza, hogy milyen frekvenciánál történjen az átvitel lecsökkentése. A sampling_freq a jel mintavételi frekvenciáját reprezentálja, azaz az adott jel mintavételezési sebességét. A nyquist_freq változó az ún. Nyquist-frekvencia, ami a mintavételi frekvencia fele. Ez az érték kulcsfontosságú a szűrés során, mivel meghatározza a legnagyobb frekvenciát, amely helyesen reprezentálható a mintavétellel. A normalized_cutoff_freq a normalizált vágási frekvencia, amely az átviteli frekvencia és a Nyquist-frekvencia hányadosa.

```
'''Application of median filters'''
window_size = 601
correction_signal = signal.medfilt(y, kernel_size=window_size)
filtered_signal = y-correction_signal

window_size = 201
correction2_signal = signal.medfilt(filtered_signal, kernel_size=window_size)
filtered_signal2 = filtered_signal-correction2_signal

filter_order=4
cutoff_freq=15
sampling_freq=500
nyquist_freq = 0.5 * sampling_freq
normalized_cutoff_freq = cutoff_freq / nyquist_freq
b, a = butter(filter_order, normalized_cutoff_freq, btype='low')
y_filtered_butter = signal.lfilter(b, a, filtered_signal2)
```

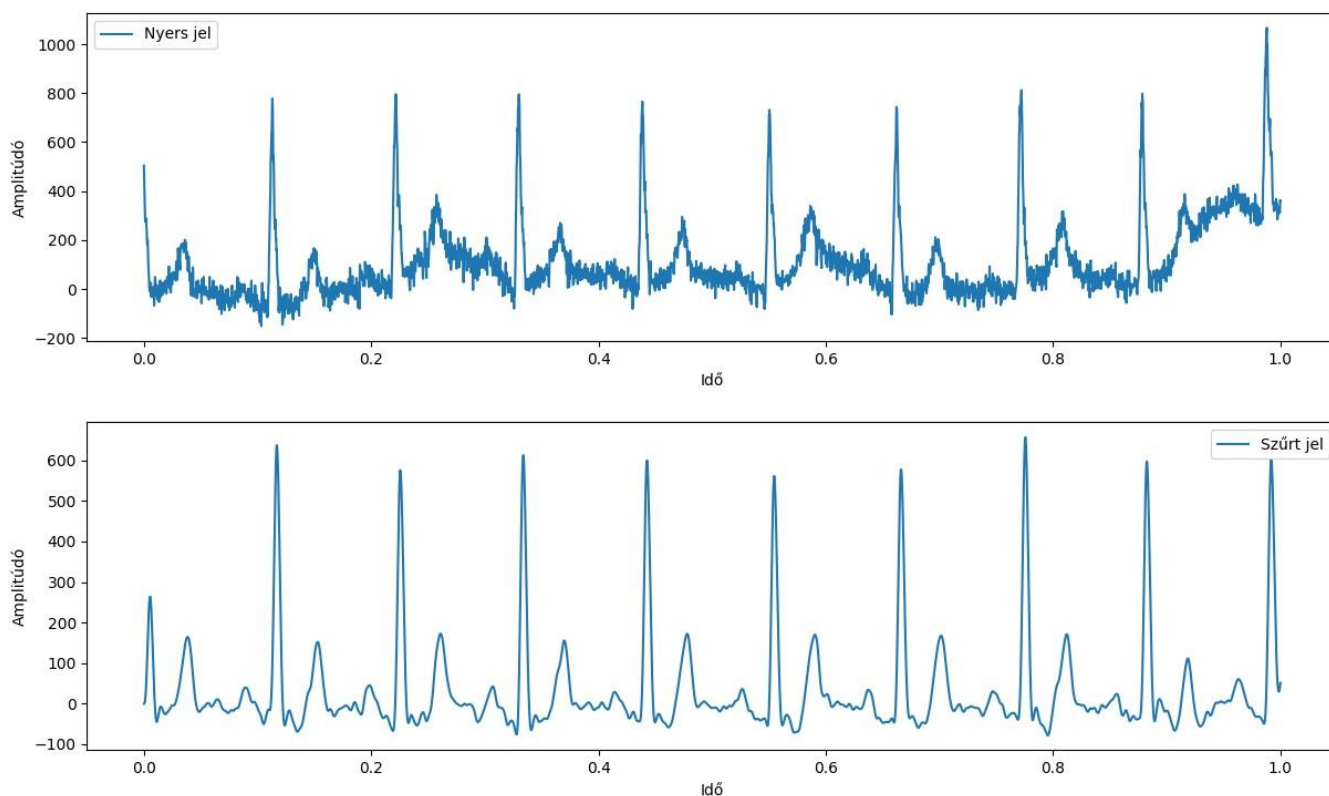
Ábra 15: Szűrés Medián és Butterworth szűrővel

A frekvencia normálása fontos lépés a szűrő tervezése során, mivel gyakran a frekvenciát normáljuk a mintavételi frekvenciához képest. A butter függvény a filter_order, a normalized_cutoff_freq és a btype paraméterek alapján kiszámítja a **Butterworth-szűrő** átviteli függvényének együtthatóit. Ezek az (a ,b) együtthatók meghatározzák a szűrő viselkedését és

jellemzőit. Ez az [a] és [b] egy-egy 1 dimenziós tömb, amely tartalmazza a szűrő bemeneti együtthatóit. A szűrő átviteli függvénye a b és a együtthatók segítségével kerül létrehozásra és alkalmazásra a `signal.lfilter` függvényben.

Az `lfilter` függvény a `signal` modulban található szintén a SciPy könyvtárban. Ez a függvény egy lineáris szűrőt hajt végre a bemeneti jelen a megadott átviteli függvény együtthatóival. A lenti ábrán már megtekinthetjük a szűrt jelet. Ez a végső szűrt jel, amelyet további feldolgozásra fogok alkalmazni. Ezen jel alapján készítek majd egy spektrogramot.

A spektrogram egy olyan vizualizációs eszköz, amely ábrázolja a jel frekvenciájának változását az idő függvényében. A jel időtartama vízszintes tengelyen jelenik meg, míg a frekvencia függőlegesen. A spektrogram segítségével könnyen megfigyelhetők az időben változó frekvenciasávok, amelyek fontos információkat hordozhatnak a jel jellemzőiről, például a ritmusáról vagy a frekvenciaváltozásokról.



Ábra 16: Szűrt jel, a szűrők alkalmazása után

7. Spektrogram

7.1. EKG jelek és a Fourier transzformáció alkalmazása

Jelenleg az egyik legnépszerűbb analitikus módszer a jelek és adatok elemzésére a Fourier - transzformáció, hiszen lehetővé teszi a jelek átváltását az időtartományból a frekvenciatartományba, megmutatva a jel frekvenciáit és annak amplitúdóját. Alapja az, hogy bármely időfüggő jel felbontható olyan egyszerűbb, harmonikus hullámformákra, amelyek különböző frekvenciákkal rezegnek. Az eljárás az eredeti jel időbeli változásait elemzi, és kinyerhetők belőle a jellemző frekvenciakomponensek.

A Fourier-transzformáció alkalmazása az EKG jelek elemzésére lehetővé teszi a frekvenciaelemek kinyerését és azok részletes vizsgálatát. Az EKG jelek átváltása a frekvenciatartományba fontos információkat nyújthat a szív elektromos aktivitásának jellemzőiről, például a szívritmusról és a különböző hullámformákról (P, QRS, T hullámok).

Ezen jelek Fourier-transzformációjának eredménye egy olyan frekvenciaspektrum, amelyben a frekvenciakomponensek amplitúdója és fázisa jelenik meg. Ez a spektrum lehetővé teszi a jellemző frekvenciák azonosítását és vizualizációját, ami segíthet a szívbetegségek diagnózisában és a kezelés hatékonyságának monitorozásában.

A digitális jelfeldolgozásban gyakran használt módszer a diszkrét Fourier-transzformáció (DFT). A DFT segítségével átválthatjuk a diszkrét időben mintavételezett jelet a frekvenciatartományba. Azonban a DFT sok számítási lépést igényel, és ezáltal lassú lehet a nagy adatmennyiségek esetén [19]. A gyakorlatban az ilyen számítási feladatokhoz általában a Gyors Fourier - Transzformációt (Fast Fourier Transform - FFT) használják [19]. Ez algoritmus, lehetővé teszi a hatékony DFT-számítást. Gyorsabb működése azzal magyarázható, hogy kihasználja a jel sajátosságait és csökkenti a számítási lépések számát. Alkalmazása jelentős előnyöket nyújt a valós idejű jelfeldolgozásban és az adatgyűjtésben, mivel lehetővé teszi a nagy adatmennyiségek gyors és hatékony feldolgozását, ami különösen fontos például a hangfeldolgozásban, a képfeldolgozásban és a kommunikációtechnológiában.

7.2. A Gyors Fourier - Transzformáció (FFT)

A dolgozatomban során Gyors Fourier - Transzformációt (FFT - Fast Fourier - Transform) használok, mely egy algoritmus, amelyet a diszkrét Fourier-transzformáció (DFT) hatékony kiszámítására használnak. Az FFT segítségével a jel transzformáltját gyorsan és hatékonyan lehet meghatározni.

Az FFT megfelelő működéséhez az N mintavétel szükséges, amelynek értéke 2^n legyen, ahol n legalább 6. Ez azt jelenti, hogy a mintavételezés során elegendő mennyiségű adatot kell rögzíteni a jelről, hogy pontos transzformációt lehessen végrehajtani. Úgy kell mintavételezés frekvenciáját megválasztani, hogy legalább kétszer nagyobb kell legyen, mint a feldolgozásra váró jel maximális frekvenciája. Enélkül a transzformáció torz képet eredményezhet. Ezért fontos, hogy a mintavételezési frekvencia megfelelően magas legyen, hogy a jel frekvenciájának információtartományát megfelelően lehessen reprezentálni. Az algoritmus műveletigénye $N \log N$.

$$\text{A sor} \quad y(\omega) = \frac{T}{\sqrt{2\pi}} \sum_{n=-\infty}^{\infty} x_n e^{-i\omega t_n}$$
$$\text{ahol} \quad x_n = \frac{1}{\sqrt{2\pi}} \int_{-\pi F}^{\pi F} y_n e^{i\omega t_n} d\omega$$

Ábra 17 : Gyors Fourier- transzformáció

A gyors Fourier - Transzformáció (FFT) egy algoritmus, amely rekurzívan működik és a "Divide et Impera" (oszd meg és uralkodj) elvén alapul. Az alap kiindulási képlet a mintavételek felosztásával nyerhető. A mintavételeket páros és páratlan indexű mintavételekre osztjuk, majd ezeket külön-külön Fourier-transzformáljuk. Ezt követően az eredményeket kombináljuk a komplex exponensekkel való szorzással és összeadással. Ez a műveletsorozat rekurzívan

ismétlődik, amíg el nem érjük a végső eredményt, ahogy a lenti ábra is mutatja (Ábra).

$$\begin{aligned}
 f_j &= \sum_{k=0}^{n-1} x_{2k} e^{-\frac{2\pi i}{2n} j(2k)} + \sum_{k=0}^{n-1} x_{2k+1} e^{-\frac{2\pi i}{2n} j(2k+1)} \\
 &= \sum_{k=0}^{n-1} x'_k e^{-\frac{2\pi i}{n} jk} + e^{-\frac{\pi i}{n} j} \sum_{k=0}^{n-1} x''_k e^{-\frac{2\pi i}{n} jk} \\
 &\begin{cases} f'_{j-n} - e^{-\frac{\pi i}{n}(j-n)} f''_{j-n} & \text{ha } j < n \\ f'_j + e^{-\frac{\pi i}{n} j} f''_j & \text{ha } j \geq n \end{cases}
 \end{aligned}$$

Ábra 18: Gyors Fourier Transzformáció képlet levezetése [19]

7.3. A Spektrogram készítése

A spektrogram és a scalogram két elterjedt módszer az EKG jelek tudományos feldolgozására és elemzésére. Ezek a módszerek lehetővé teszik a jel frekvenciaelemzését és időbeli változásainak részletes megfigyelését, amelyek fontos információkat nyújtanak a szívbetegségek azonosításához és monitorozásához

A scalogramok a skálázott reprezentációját jelentik a Continuous Wavelet Transformation (CWT) eredményének, amelyet az eredeti jelre alkalmaznak. Tehát lehetővé teszi egy jel legreprezentatívabb skáláinak (vagy frekvenciáinak) az azonosítását, azaz azokat a skálákat, amelyek a legnagyobb mértékben hozzájárulnak a jel teljes energiájához. A scalogram általában kép formájában jelenik meg, ahol az y tengelyen a frekvencia, az x tengelyen pedig az idő van ábrázolva, és a színek vagy árnyalatok jelzik a jel energiaszintjét a különböző frekvenciaskálákon és időpillanatokban.

Ez a vizualizáció lehetővé teszi a jel frekvenciai összetevőinek és időbeli változásainak vizsgálatát, valamint a jel jellemző skáláinak azonosítását. A scalogram alkalmazása különösen hasznos a valós világbeli jelek elemzésében, ahol különböző skálákon jelentkező tulajdonságok észlelhetők.

A dolgozatomban a spektrogramot használom az EKG jelek elemzéséhez és a szívbetegségek diagnosztizálásához. A spektrogram segítségével részletesen megvizsgálom a jel frekvenciaelemzését és időbeli változásait, ami lehetővé teszi a különböző hullámformák és frekvenciakomponensek azonosítását. Ez a módszer hozzájárul a megbízhatóbb és hatékonyabb szívbetegség diagnózisához és monitorozásához.

A spektrogram egy rendszer, amely lehetővé teszi egy adott jel részeinek megjelenítését a hullámhosszuk szerinti felbontásban. Az EKG jelek feldolgozása során spektrogramot kapunk, amikor például Fourier-transzformációt alkalmazunk a jelen, de más transzformációs módszerek is alkalmazhatók a spektrogram létrehozására. Ezek közé tartoznak például a Wavelet-transzformáció, a Short-Time Fourier-transzformáció (STFT), a Gabor-transzformáció és a Stockwell-transzformáció.

A spektrogram tehát egy olyan vizuális reprezentációja egy jel frekvenciaspektrumának, amint az idővel változik. Egy kép, amelyen az idő a vízszintes tengelyen, a frekvencia pedig a függőleges tengelyen van ábrázolva. Azt ábrázolja, hogy a jel milyen frekvenciájú összetevőkkel rendelkezik és ezek hogyan változnak az idő függvényében.

Ezen kép készítéséhez Gyors Fourier-transzformációt alkalmaztam az időablakokra. Ez a transzformáció lehetővé teszi, hogy az időben változó jelek frekvenciabeli tartalmát vizsgáljuk. A képpontok színe vagy intenzitása jelzi az adott frekvenciának a jelben betöltött szerepét vagy energiaszintjét. A sötétebb vagy intenzívebb színek általában a nagyobb energiaállományokat jelölik. A spektrogram nem tartalmaz információt az ábrázolt jel pontos vagy akár közelítő fázisáról. Ezért nem lehetséges visszafordítani a folyamatot és létrehozni egy másolatot az eredeti jelről egy spektrogramból [20].

A mint említett Fourier-transzformáció alkalmazása, a spektrogram létrehozásához az alábbi képlettel valósul meg (19. Ábra):

$$S(f, t) = |F \{x(T)\} (f, t)|^2$$

Ábra 19: Spektrogram létrehozásának a képlete

Ahol: $S(f, t)$ a spektrogram értéke a frekvencia (f) és az idő (t) függvényében. $F\{x(\tau)\}(f, t)$ a Fourier-transzformáció eredménye a bemeneti jel ($x(\tau)$) időbeli ablakára alkalmazva.

A Fourier-transzformáció kiszámolja a bemeneti jel frekvenciadomináns reprezentációját, és az abszolút érték négyzetét veszi, hogy az eredményt a spektrogram intenzitásának megfelelően reprezentálja.

Ez a képlet tehát leírja, hogyan alakítjuk át a jel időbeli reprezentációját frekvenciadomináns formára, és hogyan kapjuk meg a spektrogram értékeit a frekvencia és az idő dimenziók mentén.

Amikor egy EKG jel spektrogramját vizsgáljuk, számos hasznos információt nyerhetünk az adott jel jellemzőiről. A spektrogram segítségével megismerhetjük az EKG jelben jelen lévő frekvenciakomponenseket. Az EKG jel tartalmazhat különböző frekvenciájú hullámokat és ritmusokat, amelyek információt adnak az egészségi állapotról és a szív működéséről. Például az alacsony frekvenciájú komponensek a szív alapvető működését jelzik, míg a magas frekvenciájú komponensek az esetleges zavarokat vagy szívritmuszavarokat mutathatják.

A spektrogram segítségével azonosíthatjuk az EKG jelben lévő jellegzetes eseményeket, például a QRS-komplexumokat, a P-hullámokat vagy a T-hullámokat. Ez lehetővé teszi az automatikus detektálást és a jellemzők mérését az adott jelentős pontokon.

A lenti ábrán bemutatott kódrészletben a jel feldolgozása után, azaz a szűrés után, használtam a `specgram()` beépített függvényt a `matplotlib.pyplot` modulból. Ez függvény létrehoz egy spektrogramot, amely a jelsorozat vagy jelrendszer frekvencia-tartalmának vizualizációját nyújtja. Több fontos paramétere van, amelyek befolyásolják a spektrogram tulajdonságait és felbontását. Az alábbiakban részletesen bemutatom ezeket a paramétereket:

```

def plot_ecg_spectrogram(file):
    '''Data Processing'''
    inputfile = file
    processed_file = config_data['processed_file']
    process_data(inputfile, processed_file)

    '''Data extraction from the file'''
    dataset = pd.read_csv(processed_file)
    y = dataset.iloc[:, 0].tolist()
    y_filtered = ecg_filtering(y)

    '''specifying the variables'''
    fs = 500
    fig, ax = plt.subplots(figsize=(8, 6))
    ax.set_title('ECG Spectrogram')
    ax.set_xlabel('Time [s]')
    ax.set_ylabel('Frequency [Hz]')
    Pxx, freqs, bins, im = ax.specgram(y_filtered, NFFT=440, Fs=fs, noverlap=220, cmap='jet')
    plt.ylim(0,90)

    '''Save spectrogram as an image'''
    output_file= config_data['spektogram_file']
    fig.savefig(output_file, bbox_inches='tight', pad_inches=0.5)
    plt.close(fig)

    return output_file

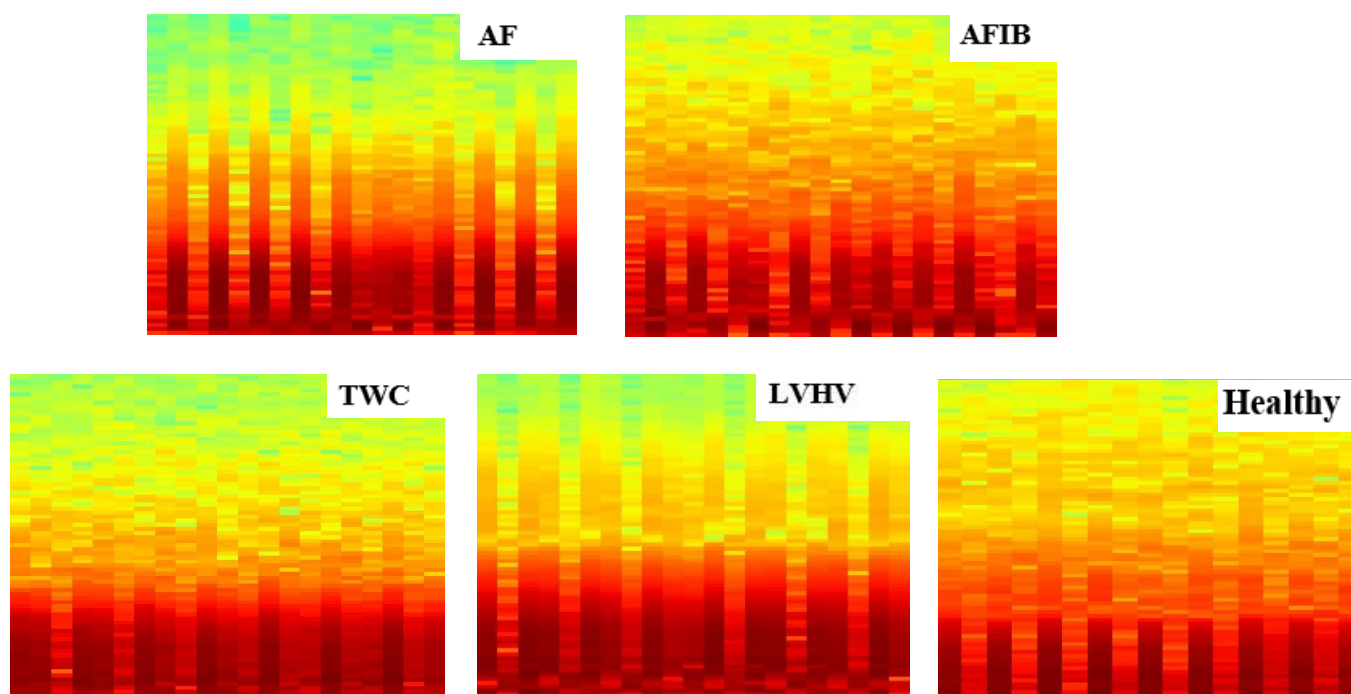
```

Ábra 20 : Spektrogram elkészítése majd lementése

- **x** : Ez az input jelsorozat vagy jelrendszer, amelyre a spektrogramot szeretném alkalmazni. Az én esetemben ez az **y_filtered** változó. Amely a már szűrt jelet tartalmazza.
- **NFFT**: Ez a paraméter határozza meg az FFT (Gyors Fourier-transzformáció) ablak hosszát. Az FFT egy matematikai eljárás, amely a jel frekvencia-tartalmát elemzi. Az NFFT értéke meghatározza az ablak méretét, és befolyásolja a spektrogram felbontását. Nagyobb érték finomabb frekvenciafelbontást eredményez, azaz részletesebb képet ad a frekvencia tartalomról, ugyanakkor növeli a számítási terhelést is. Az én esetemben ez 440.
- **Fs**: Ez a mintavételi frekvencia, vagyis az adatok mintavételezési sebessége. A spektrogram létrehozásakor szükséges tudni, hogy az adatok milyen gyorsan vannak

mintavételezve az időben. Az **Fs** értéke nálam 500-ra van állítva, mint már említettem a klinikai adataimnak ennyi volt a mintavételezési sebessége, ez mindkét adatbázisra érvényes.

- **noverlap**: Az átfedési tényező az FFT ablakok átfedését határozza meg. Az átfedés növelése simább spektrogramot eredményez, mivel finomabb időbeli változásokat mutat, de ezzel együtt növeli a számítási terhelést. Az általam használt átfedési tényező értéke 220, de ez nem lehet nagyobb, mint az ablak mérete (NFFT). Fontos megérteni, hogy az átfedés növelése több FFT számítását jelenti, ami növeli a számításigényt. Ezért az átfedés mértékét korlátozni kell az adott adatsor és a spektrogram felbontásával összhangban, hogy a megfelelő eredményt kapjuk.
- **cmap**: Ez a paraméter meghatározza a színskálát, amelyet a spektrogramban szeretnénk használni. Például a 'jet' érték egy olyan színskálát jelent, amely fokozatosan változik a kékes, zöldes, sárgás és vörös árnyalatok között, és segít a frekvencia tartalom vizuális megjelenítésében.



Ábra 21: Spektrogramok betegségek szerint

8. Adatok előkészítése a mély tanuláshoz

Az előzőleg említett jel szűrő, és spektrogram előállító algoritmusaim automatizálnom kellett, hogy hatékonyan és rövid időn belül feldolgozhassak akár több száz adatot is. Ennek érdekében továbbfejlesztettem az algoritmust, amely képes volt kezelni a nagy mennyiségű adatot és a végrehajtást. Különösen a nyílt adatbázissal kapcsolatban végeztem további munkát. Itt nagyobb adatmennyiséggel dolgoztam, és a folyamatok végén a rendezést is elvégeztem. A cél az volt, hogy a spektrogramokat rendszerezetten tároljam a további elemzések és feldolgozások érdekében.

Kezdetben, ahogy már korábban említettem, a nyers EKG adatok eredeti formában egyetlen mappában voltak tárolva. Azonban felmerült egy kis probléma, amikor rájöttem, hogy a "Diagnostics.xlsx" nevű fájl, amely tartalmazta a mappában található fájlok neveit és azokhoz tartozó diagnózisokat, nem minden fájlját tartalmazta a mappa. Ezért szükség volt arra, hogy kiválasszam, melyik fájlok vannak valójában meg nekem, amiket ténylegesen tudnék használni.

Először is, a mappában található fájlok neveit kimentettem egy .txt fájlba. Ezután az "Diagnostics.xlsx" fájlt átalakítottam úgy, hogy csak a fájlneveket tartalmazza. Ezáltal könnyen szűrhetővé vált egy egyszerű összehasonlító algoritmussal, hogy melyik fájlokat rendelhetem hozzá a diagnózisokhoz. Így már tényleg csak a csoportosítás maradt hátra.

```
def compare_filenames(input_folder, csv_file, output_folder):
    file_path = os.path.join(input_folder, '..', csv_file)

    with open(file_path, 'r') as file:
        reader = csv.DictReader(file)
        data = list(reader)

    for filename in os.listdir(input_folder):
        file_path = os.path.join(input_folder, filename)

        for row in data:
            if filename == row['FileName'] and row['Rhythm'] == 'SR' and row['Beat'] == 'NONE':
                rhythm = row['Rhythm']

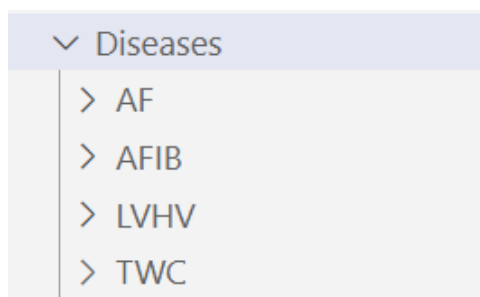
                destination_folder = os.path.join(output_folder, rhythm)
                os.makedirs(destination_folder, exist_ok=True)

                new_filename = f"{filename}_{row['Rhythm']}.{filename.split('.')[-1]}"
                shutil.move(file_path, os.path.join(destination_folder, new_filename))
                break

    print("The files have been successfully moved to the new folders..")
```

Ábra 22: Adatok csoportosítása, betegség szerint, ebben az esetben sima szinusz ritmus van kiválasztva

A fenti kódrészletben (18. ábrán) először ellenőrzöm, hogy az output_folder mappa létezik-e. Amennyiben nem létezik, létrehozom azt. Ezután végigiterálok minden CSV fájlra és elvégzem a feldolgozási lépéseket: beolvasom a fájlt, szűröm az adatokat, majd elkészítem a spektrogramot. A spektrogramot képként elmentem csoportosítva fájlnev szerint.



Ábra 23: A csoportosított mappa struktúrája

```

def create_spectrograms_and_sort_files(folder_path, output_folder):
    processed_file = config_data['processed_file']

    if not os.path.exists(output_folder):
        os.makedirs(output_folder)

    keywords = ["_AFIB.", "_AF.", "_LVHV.", "_RBBB.", "_TWC."]

    folder_name = os.path.basename(folder_path)
    folder_name = folder_name.replace(" ", "")

    if folder_name != os.path.basename(folder_path):
        new_folder_path = os.path.join(os.path.dirname(folder_path), folder_name)
        if not os.path.exists(new_folder_path):
            os.makedirs(new_folder_path)
    else:
        new_folder_path = folder_path

    for root, dirs, files in os.walk(folder_path):
        for file in files:
            if file.endswith(".csv"):
                file_path = os.path.join(root, file)

                with open(file_path, 'r') as csvfile:
                    process_csv_data(file_path, processed_file)
                    dataset = pd.read_csv(processed_file)
                    y = dataset.iloc[:, 0].tolist()
                    y_filtered = ecg_filtering(y)

                    fs = 500
                    fig, ax = plt.subplots(figsize=(8, 6))
                    ax.set_title('ECG Spectrogram')
                    ax.set_xlabel('Time [s]')
                    ax.set_ylabel('Frequency [Hz]')
                    Pxx, freqs, bins, im = ax.specgram(y_filtered, NFFT=440, Fs=fs, noverlap=220, cmap='jet')
                    plt.ylim(0, 90)

                    '''Save spectrogram as an image'''
                    new_file_name = file.replace(".csv", ".jpg")
                    output_file_path = os.path.join(output_folder, new_file_name)
                    plt.savefig(output_file_path)
                    plt.close()

                for keyword in keywords:
                    if keyword in file:
                        keyword_folder = os.path.join(new_folder_path, keyword)
                        if not os.path.exists(keyword_folder):
                            os.makedirs(keyword_folder)

                        new_file_path = os.path.join(keyword_folder, new_file_name)
                        shutil.move(output_file_path, new_file_path)
                        break

    print("The spectrograms have been created and saved in the new folder.")
    print("The files have been sorted into the new folders.")

```

Ábra 24 : Az alábbi függvény létrehozza a spektrogramokat minden egyes jelre, majd ezeket csoportosítja fájlokba a nevük alapján.

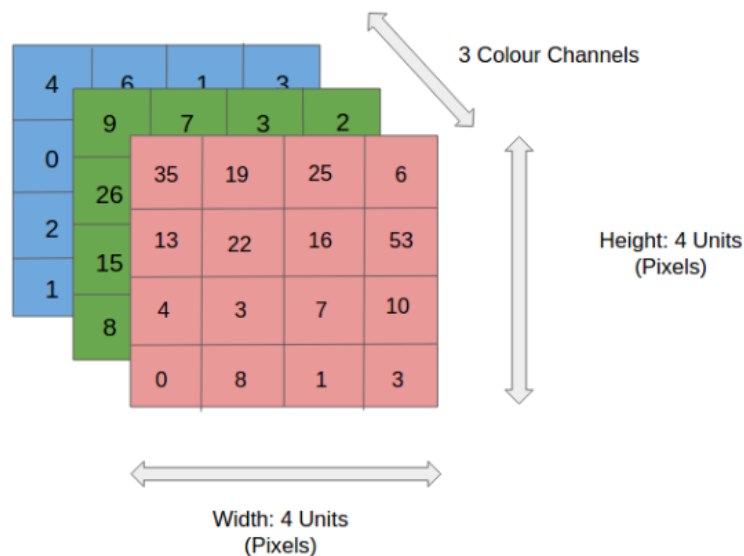
9. A mélytanulású neuronhálózat

A mélytanulás (Deep Learning), a többrétegű neurális hálózatok alkalmazásával nagy mennyiségű adatokon alapuló feladatok megoldását célozza meg. A mélytanulás azon alapszik, hogy a neurális hálózatok képesek automatikusan kinyerni és reprezentálni az adatokban rejlő jellemzőket és mintákat. Az ilyen hálózatok rendelkeznek bemeneti réteggel, rejtett rétegekkel és kimeneti réteggel.

Jellemzője, hogy nagyobb mélységgel és komplexitással rendelkeznek, mint a hagyományos gépi tanulási algoritmusok, az adatokat hierarchikusan és többszintű módon dolgozza fel. Az alsóbb szinteken a hálózat egyszerűbb tulajdonságokat és mintázatokat ismer fel, míg a felsőbb szinteken egyre absztraktabb és komplexebb jellemzőket tanul. Ez lehetővé teszi a mélytanulásnak, hogy magasabb szintű reprezentációkat és tudást építsen fel az adatokból, amelyeket aztán felhasználhat az adott feladat megoldására [21].

A Deep Learning területén számos hálózattípus található már, mint például az RNN (Rekurrens neurális hálózatok), LSTM (Hosszú rövid távú memóriával rendelkező hálózatok) és sok másik. Ezek a hálózatok különböző alkalmazási területeken használhatók. A kutatásomban azonban a konvolúciós neurális hálózatokat (CNN) alkalmaztam, mivel ezek a legalkalmasabbak a képfeldolgozásra.

Az alábbiakban részletesebben bemutatom a Convolutional Neural Network (CNN) működését és az alkotóelemeit. Majd a következő fejezetben ismertetem, hogy hogyan építettem fel saját modelletem. Mint ahogy a fenti képen látszik (19. Ábra), a hálózat bemenete egy kép, amelyet pixelértékek alkotnak. A kép lehet színes (RGB csatornákkal) vagy szürkeárnyalatos (egy csatornával). Az én esetemben háromcsatornás RGB képekről van szó, amelyek spektrogramokat ábrázolják. A lenti kép egy illusztráció, hogyan is kell elképzelni ezeket a képeket, jelen esetben egy 4x4x3-as RGB kép található.



Ábra 25: 4x4x3-as RGB kép [22]

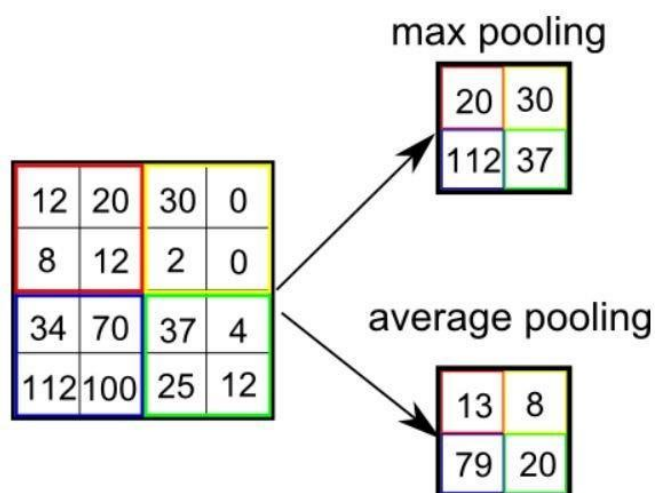
El lehet képzelni, hogy mennyire számításigényes lenne a folyamat, ha a képek elérnék akár a 8K (7680×4320) méretet. A konvolúciós neurális hálózat (ConvNet) feladata, hogy átalakítsa a képeket olyan formára, amely könnyebben feldolgozható, miközben megőrzi a fontos jellemzőket, amelyek szükségesek a pontos predikcióhoz. Ez fontos, amikor olyan architektúrát tervezünk, amely nemcsak jó a jellemzők tanulásában, hanem skálázható is masszív adatkészletek esetén [22].

A bemenetet máris követik a konvolúciós rétegek, melyek a CNN legfontosabb elemei. A feladata az input adatok (általában képek) jellemzőinek kinyerése és reprezentáció létrehozása. A konvolúciós rétegben számos kis méretű szűrő vagy kernel található, amelyeket a bemeneti adatokon végighúzva konvolúciós művelettel kapcsolódnak az adathoz. Működése során a szűrők (kernel) súlyozzák a bemeneti adatok elemeit. A súlyozott értékek összegezése révén új, részletesebb jellemzők jönnek létre. Ezáltal a konvolúciós réteg felismerheti az adatokban található vizuális mintákat és struktúrákat. Előnye, hogy az általa kinyert jellemzők lokális információkat tartalmaznak. Ez lehetővé teszi, hogy a réteg érzékeny legyen a kép különböző részein található mintázatokra. Emellett a konvolúciós rétegeknek a paraméterek megosztása is jellemző, ami azt jelenti, hogy ugyanazokat a súlyokat használják a kép különböző részein történő konvolúció során. Ezáltal csökkenthető a modell paraméterek száma és a számítási

költség. Azt még hozzá tenném, hogy hagyományosan az első konvolúciós réteg felelős a alacsony szintű jellemzők, például élek, szín, gradiens iránya stb. rögzítéséért. Az további rétegek hozzáadásával az architektúra alkalmazkodik a magas szintű jellemzőkhöz is, így kapunk egy hálózatot, amelynek átfogó megértése van a képekhez a adathalmazban, hasonlóan ahogy mi is megértenénk [22].

Az aktivációs rétegek a konvolúciós rétegek után következnek, és nemlinearitást adnak a hálózatnak. Gyakran használt aktivációs függvények például a ReLU (Rectified Linear Unit) vagy a szigmoid függvény. Az aktivációs rétegek aktiválják a neuronokat az alapján, hogy a konvolúciós rétegekből származó jelek meghaladják-e a küszöbértéket. (ezt a következő fejezetben jobban részletezem)

A Pooling réteg alkalmazása opcionális a konvolúciós rétegek után, ez a hálózat architektúrájától függ. Két típusú ilyen réteg létezik (21. Ábra): Average Pooling és Max Pooling. Az Average Pooling visszaadja a képben a kernellel lefedett terület összes értékének átlagát, míg a Max Pooling visszaadja a kernellel lefedett terület legnagyobb értékét én ezt az utóbbit használom. Ezek a kép méretének csökkentésére és az invariáns jellemzők kiemelésére szolgálnak. Mivel a legnagyobb értékeket vagy az átlagokat választják ki egy adott területről, ezzel csökkentik a dimenziót és a számítási terhelést.



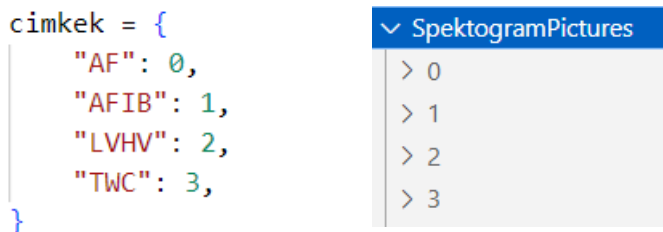
Ábra 26: Pooling réteg típusainak a szemléltetése [22]

A teljesen összekapcsolt rétegek (Dense) a CNN utolsó részét képezik. Ezek azok a hagyományosan tanult rétegek, amelyek minden neuronját minden előző réteg neuronjához kapcsolják. Ez azt jelenti, hogy minden neuron a rétegben kapcsolódik minden neuronhoz a következő rétegben, és minden kapcsolat rendelkezik súlyokkal. Ezek a rétegek segítenek az osztályozás vagy a végrehajtandó feladat típusától függő kimenet előállításában. A softmax aktivációs függvényt általában az utolsó Dense rétegben alkalmazzák osztályozási feladatoknál, ahol több kimeneti osztály van. A softmax függvény kimenete egy valószínűségi eloszlás, amely megmutatja, hogy a bemenet melyik osztályhoz tartozik a legnagyobb valószínűséggel.

10. A program felépítése

Az előző részben bemutattam a Konvolúciós Neuronháló általános felépítését és az alapvető komponenseit. Most részletesen bemutatom a szoftverem felépítését és működését, mely kifejezetten a tanításra összpontosít. Ez a következő lépésekből áll:

- Adatbetöltés: Kezdetben egy ciklus segítségével végigiteráltam a már előkészített mappán, amelyben a négy betegség külön mappákban volt csoportosítva, viszont mostmár számok szerint. Minden egyes szám, egy-egy betegséget jelöl. Így könnyedén hozzárendelhettem a címkéket a megfelelő betegségekhez.



Ábra 27 : Az adatkészletem struktúrája

Ezután egy belső ciklusban végigiterál a képek listáján, és megpróbálja betölteni és

feldolgozni azokat. A képek betöltése `Image.open()` függvénnyel történik, majd a kép átméretezése `resize()` metódussal 100x100 képméretre. Az átméretezett kép NumPy tömbbé alakítása után hozzáadjuk az adatok listához, és a képhez tartozó címkét (i) hozzáadjuk a címkek listához. Ha a kép betöltése vagy feldolgozása közben hiba lép fel, akkor az `except` blokkban található üzenetet írjuk ki a konzolra.

Végül a adatok lista NumPy tömbbé alakul, amely a betöltött és feldolgozott képek adatait tartalmazza.

```
adatok = []
cimkek = []
path = os.getcwd()
pictures = config_data['spectograms_folder']

for i in range(4):
    path = os.path.join(path, pictures, str(i))
    kepek = os.listdir(path)
    for j in kepek:
        try:
            kep = Image.open(os.path.join(path, j)).resize((100,100))
            kep = np.array(kep)
            adatok.append(kep)
            cimkek.append(i)
        except:
            print("The image could not be loaded.")

adatok = np.array(adatok)
cimkek = np.array(cimkek)

print(adatok.shape, cimkek.shape)
```

Ábra 28 : Adatbetöltés

- Adat előkészítése: az adatokat felosztottam tanító és validációs célra a `train_test_split` függvény segítségével. Így dataset két részre oszlik: `X_tanit` és `X_validal` az input adatokhoz, valamint `y_tanit` és `y_validal` a megfelelő címkekhez. A `test_size` paraméter értéke 0.2, ami azt jelenti, hogy az adatok 20%-a lesz használva validációra.

Az adatok felosztása után kiíratjuk a tanító és validációs halmazok alakját, hogy ellenőrizzük az adatok dimenzióját. Az `X_tanit` és `X_validal` tömbök az input adatokat

reprezentálják, míg a `y_tanit` és `y_validal` tömbök tartalmazzák a megfelelő címkéket. Ezután a címkéket one-hot kódolással alakítjuk át. Ez a kódolási séma bináris vektor formájában ábrázolja a címkéket, ahol az index, amely a címkét reprezentálja, 1-re van állítva, míg az összes többi index 0-ra van állítva. A `num_classes` paraméter értéke 4, ami az adathalmazomban található osztályok teljes számát jelenti.

```
"""Split the data for training and testing"""
X_tanit, X_validal, y_tanit, y_validal = train_test_split(adatok, cimkek, test_size=0.2)

print(f"Betanitasi adatok: {X_tanit.shape}, labels: {y_tanit.shape}")
print(f"Validalasi adatok: {X_validal.shape}, labels: {y_validal.shape}")

y_tanit = to_categorical(y_tanit, num_classes=4)
y_validal = to_categorical(y_validal, num_classes=4)
```

Ábra 29 : Adatok szétválasztása

Például, az én esetemben, az adathalmazom, amelyben a címkék kategóriák (pl. AF, AFIB, TWC betegségek), akkor az one-hot encoding során minden kategóriához egy bináris vektor tartozik. Például az AF kategóriához tartozó vektor [1, 0, 0,0], az AFIB kategóriához tartozó vektor [0, 1, 0, 0], és a TWC kategóriához tartozó vektor [0, 0, 1,0] és így tovább.

Ez az átalakítás azért fontos, mivel a tanulási modell vagy algoritmus általában számokkal dolgozik, és nem tud közvetlenül kezelni kategorikus vagy szöveges adatokat. Így one-hot encoding segítségével a címkék olyan formát kapnak, amelyet a modellek könnyen kezelhetnek és amelyre alkalmazhatók a numerikus algoritmusok [23].

- Modell kialakítása: Ezután létrehoztam a Konvolúciós Neuronháló modellt. A modell építéséhez a Keras Sequential modelljét használtam, amely lehetővé tette a rétegek egymás utáni sorrendben történő hozzáadását [24].

lehetnek konvolúciós rétegek, pooling rétegek, teljesen összekapcsolt rétegek, dropout rétegek, stb. A rétegek paramétereinek (pl. szűrők száma, aktivációs függvény, stb.) beállítása rétegenként történik [25].

```
'''Building the model'''
model = Sequential([
    Conv2D(filters=32, kernel_size=(5,5), activation='relu', input_shape=X_tanit.shape[1:]),
    Conv2D(filters=32, kernel_size=(5,5), activation='relu'),
    MaxPool2D(pool_size=(2, 2)),
    Dropout(rate=0.25),
    Conv2D(filters=64, kernel_size=(3, 3), activation='relu'),
    Conv2D(filters=64, kernel_size=(3, 3), activation='relu'),
    MaxPool2D(pool_size=(2, 2)),
    Dropout(rate=0.25),
    Flatten(),
    Dense(256, activation='relu'),
    Dropout(rate=0.5),
    Dense(5, activation='softmax')
])
```

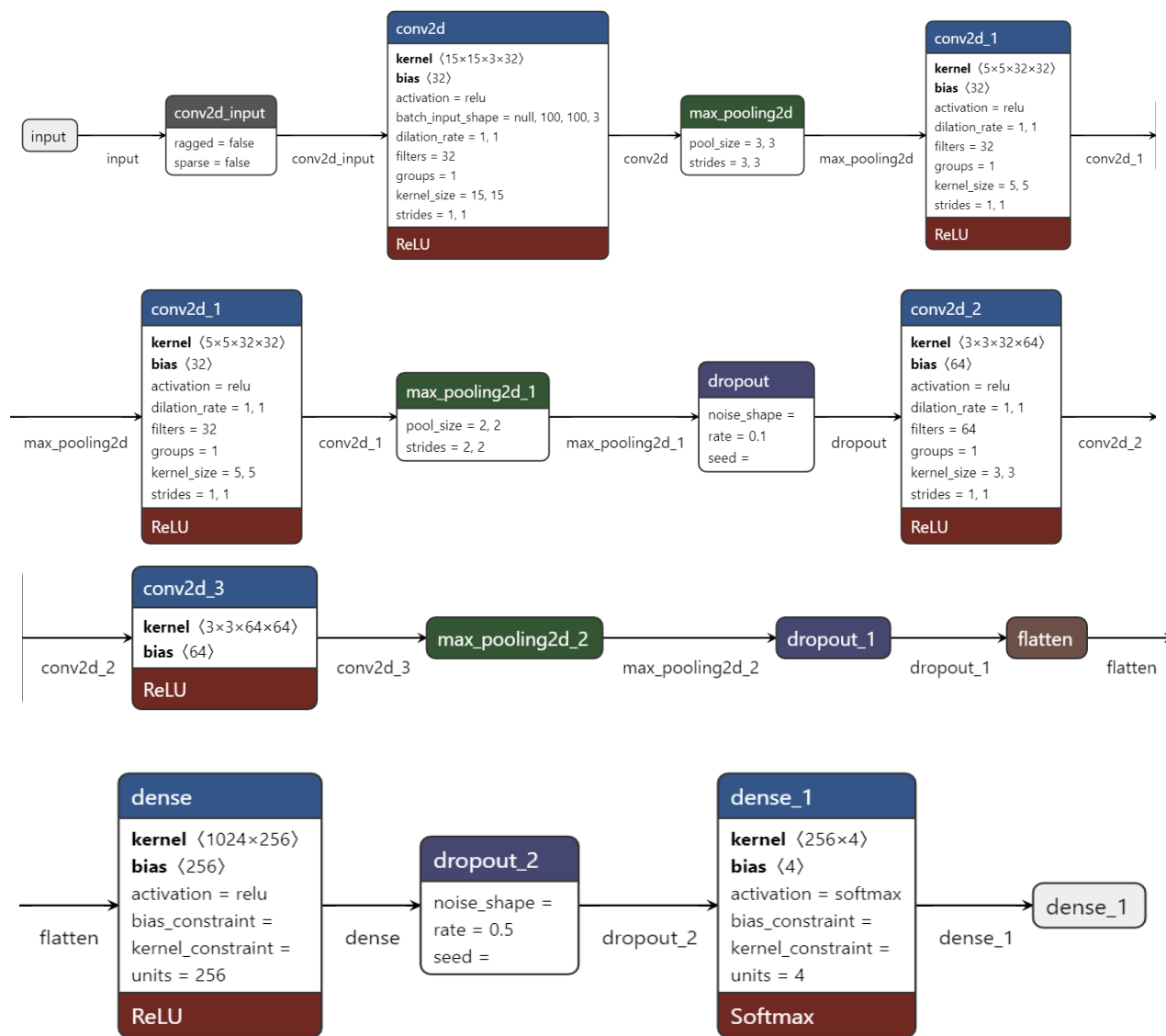
Ábra 30: A modell építése

A Keras Sequential modell egy lineáris, egymás utáni rétegekből álló neurális hálózatot jelent. A Sequential modellt egyszerűen létrehozhatjuk és konfigurálhatjuk a Keras könyvtár segítségével. Az egymás utáni rétegek lineárisan kapcsolódnak egymáshoz, és az adatok az egyik rétegből áramlanak a következőbe. Itt a rétegek

A modellemben a következő rétegek találhatók (27. Ábra):

- Két konvolúciós réteg (Conv2D), mindegyik 32 szűrővel és 5x5-ös kernel mérettel. Az aktivációs függvényük a ReLU.
- Ezt követi egy MaxPooling réteg (MaxPool2D), amely 2x2-es ablakkal végzi a down-samplinget.
- Dropout réteg (Dropout) a túltanulás elkerülése érdekében, ahol az eldobás aránya 0,25.
- Két további konvolúciós réteg, mindkettő 64 szűrővel és 3x3-as kernel mérettel, aktivációs függvényként a ReLU-t használva.
- Újabb MaxPooling réteg, szintén 2x2-es ablakkal.
- Újabb Dropout réteg, most már 0,25-ös eldobás aránnyal

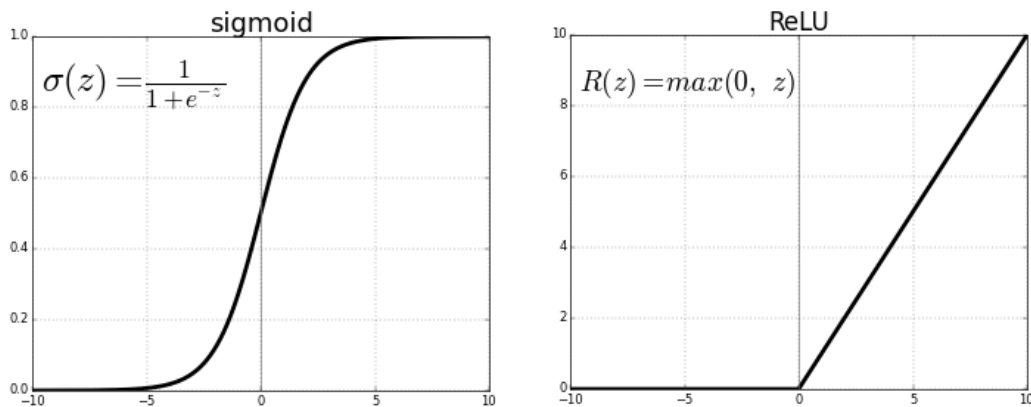
- Az adatok lapítása (Flatten), hogy a 2D tenzorokat 1D vektorrá alakítsa.
- Teljesen összekapcsolt réteg (Dense) 256 egységgel és ReLU aktivációs függvényvel.
- Dropout réteg a túltanulás további csökkentésére, ezúttal 0,5-ös eldobás aránnyal.
- Végül egy teljesen összekapcsolt réteg (Dense) 5 egységgel és softmax aktivációs függvényvel, amely a kimeneti osztályok valószínűségeit adja.



Ábra 31: Modellem felépítése

A Conv2D -ről azt kell tudni, hogy egy kétdimenziós konvolúciós réteg, mint a nevéből is látszik. A megadott számú szűrőt alkalmazza a bemeneti adatra, és konvolúciós műveletet hajt végre. Minden szűrő különböző jellemzőket detektál a bemeneten, például éleket, textúrákat vagy mintázatokat [26].

A szűrők száma meghatározza a kimeneti térfogat mélységét. A kernel méret meghatározza a szűrő térbeli dimenzióit. Például egy (3, 3) kernel méret azt jelenti, hogy egy 3x3-as szűrőt alkalmazunk a bemenetre. Az aktivációs függvény nem-linearitást vezet be a Conv2D réteg kimenetébe. A két leggyakrabban használt aktivációs függvények például az 'relu' (rectified linear unit) vagy a 'sigmoid', én relut használtam a konvolúciós rétegekben (33. Ábra).



Ábra 32 : Két leggyakrabban használt aktivációs függvény : sigmoid és relu

Tekintsük meg részletesebben a ReLU (Rectified Linear Unit) aktivációs függvény működését, amelyet én használtam. A ReLU az egyik leggyakrabban használt aktivációs függvény a neurális hálózatokban [27], mely az alábbi egyszerű matematikai képletet követi:

$$f(x) = \max(0, x)$$

Az x az adott bemeneti értéket jelöli. Az aktivációs függvény ebben az esetben a bemenetet továbbítja, ha az pozitív vagy nulla, vagyis a pozitív értékeket változatlanul

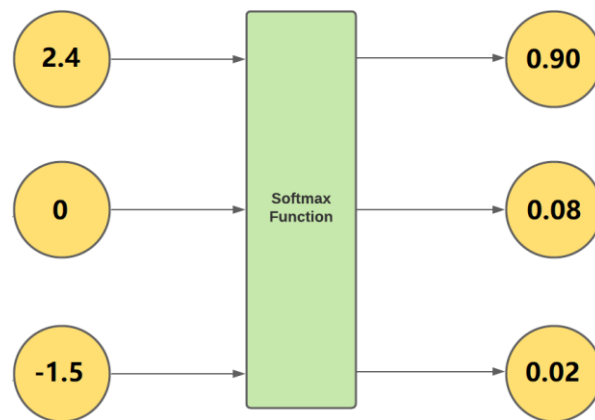
továbbítja. Ha a bemenet negatív, akkor az aktivációs függvény a kimenetet nullára állítja [27]. Ennek a függvénynek több előnye is van, mint már említettem nemlineáris aktivációt hoz létre, ez segít a neurális hálózatnak tanulni és modellezni a bemeneti adatokban rejlő nemlineáris kapcsolatokat. A másik előnye az a hatékonyság, mivel egyszerű matematikai

műveletekből áll, így gyorsan kiszámítható. Ezáltal hozzájárul a neurális hálózatok hatékonyabb és gyorsabb működéséhez. Másrészt ritkítja az aktivációkat. A negatív bemenetek nullára állítása miatt csak a pozitív jellemzők maradnak aktívak. Ez elősegíti a túlillesztés (overfitting) elkerülését és a modell általánosíthatóságának javítását.

Viszont van egy hátránya is, amit muszáj megemlíteni, az pedig nem más, mint a "dead relu" jelenség, azt jelenti, hogy a függvény az adott neuronban mindig nullát ad vissza a bemeneti tartomány egy adott részén. Ez azt jelenti, hogy minden bemeneti érték esetén a neuron kimenete mindig 0 lesz, és így a neuron "elakad" vagy "meghal".

Ez gyakran akkor fordul elő, amikor az adott neuron inicializálása során túl nagy súlyok vagy elfajult súlyok vannak jelen, amelyek miatt a ReLU aktivációs függvény mindig inaktív állapotban marad. Ez a jelenség problémát jelenthet a neurális hálózat hatékonyságára és tanulási képességére nézve, mivel a "meghalt" neuronok nem adnak hozzá semmilyen információt a hálózat működéséhez [28].

A másik aktivációs függvény, mint az ábrán is látszik (25. Ábra) az a softmax, melyet a kimeneti rétegen alkalmazva normalizálja a kimeneti értékeket olyan módon, hogy a kimenetek összege 1 legyen. Ez azért fontos, mert a softmax függvény a valószínűségi eloszlást reprezentálja az osztályok között. Tehát a kimeneti értékek interpretálható valószínűségeket jelentenek, és összegezve 1-et adnak [29].



ábra 33 : Softmax működési elve [30]

- Modell tanítása: **model.compile** (**loss='categorical_crossentropy'**, **optimizer='adam'**, **metrics=['accuracy']**) sor segítségével beállítjuk a modell tanítási konfigurációját.

A **compile** metódus meghatározza, hogy a modell hogyan fogja optimalizálni és kiértékelni a súlyokat a tanítás során. A **loss** paraméter megadja a veszteségfüggvényt, amelyet a modell minimalizálni fog. Én a **categorical_crossentropy** veszteségfüggvényt használom, amely a többosztályos osztályozásra alkalmas.

Az **optimizer** paraméter az optimalizáló algoritmust határozza meg, itt az 'adam' optimalizálót használom. Erről azt kell tudni, hogy az SGD-hez hasonlóan a gradiensek alapján frissíti a súlyokat, de az Adam beépít egy adaptív sebességillesztési módszert is. Ez lehetővé teszi az algoritmus számára, hogy a tanítás során változtassa az adatkészletekhez és a paraméterekhez való alkalmazkodást, így hatékonyabban optimalizálja a hálózatot. Az **metrics** paraméter meghatározza a kiértékelési metrikákat, amelyeket a modell a tanítás során és a kiértékeléskor követni fog. Itt az 'accuracy' metrikát használjuk.

```
model.compile(loss='categorical_crossentropy', optimizer='adam', metrics=['accuracy'])

epoch = 4
history = model.fit(X_train, y_train, batch_size=10, epochs=epoch, validation_data=(X_validates, y_validates))

plt.figure(0)
plt.plot(history.history['accuracy'], label='training accuracy')
plt.plot(history.history['val_accuracy'], label='val accuracy')
plt.title('Accuracy')
plt.xlabel('epochs')
plt.ylabel('accuracy')
plt.legend()
plt.show()

plt.figure(1)
plt.plot(history.history['loss'], label='training loss')
plt.plot(history.history['val_loss'], label='val loss')
plt.title('Loss')
plt.xlabel('epochs')
plt.ylabel('loss')
plt.legend()
plt.show()
```

Ábra 34: kompilálás, tanítás

Ezután az ábrák segítségével könnyen nyomon követhetők az **accuracy** és **loss** értékek változásai, és segítséget nyújthatnak a modell finomhangolásában és továbbfejlesztésében.

Az első ábra az "Accuracy" (pontosság) nevű címmel jelenik meg. Az x tengelyen az epochok (tanítási ciklusok) száma látható, míg az y tengelyen az **accuracy** értékek vannak ábrázolva. Az **accuracy** azt mutatja, hogy milyen mértékben sikerült a modellnek helyesen beazonosítani a bemeneti adatokat a tanítás során. Az ábra az **accuracy** értékek változását mutatja a tanítás során és a validáció során. Ha az ábra emelkedik, azt jelenti, hogy a modell jobban teljesít, ha pedig csökken, akkor a teljesítménye romlott.

A második ábra a "Loss" (veszteség) nevű címmel jelenik meg. Az x tengelyen az epochok (tanítási ciklusok) száma látható, míg az y tengelyen a **loss** értékek vannak ábrázolva. A **loss** érték azt jelzi, mennyire távol vannak a modell predikciói a valós értékektől. A veszteség értéke az egyik fő mértéke a modell teljesítményének, és a célja, hogy minimalizálja a predikció és a valós értékek közötti különbséget. Az ábra

az **accuracy** ábrához hasonlóan mutatja a **loss** értékek változását a tanítás és a validáció során.

Az 28. ábrát vizsgálva láthatjuk, hogy jelen van még a **history** objektum. A **history** az a Keras könyvtárban definiált objektum, amely a modell tanítási folyamata során rögzíti és tárolja az epochonkénti metrikákat és veszteségeket. Amikor a Keras modellt tanítjuk a **fit()** módszerrel, a visszatérési értéke egy **History** objektum lesz, amely tartalmazza azokat az adatokat, amelyek felhasználhatók az ábrázoláshoz.

- Modell kiértékelése: Végül kiértékelem a tanított modellt egy kiértékelő adathalmazon (teszthalmazon). Ez lehetővé teszi számomra, hogy megmérjem a modell pontosságát és hatékonyságát, valamint, hogy értékeljem a teljesítményét a valós világban.

Kezdetben meg kell adnom egy olyan fájlt, amely a tesztadatokat tartalmazza. Ez a "test.csv" (36. Ábra) nevű fájl, amelynek egy oszlopában (ClassId) a címkéket, a másik oszlopában pedig (Path) a képek elérési útvonalát tárolja. Az "y_test" változóban tárolom a címkéket (ClassId oszlop) a "values" attribútum segítségével, amely egy NumPy tömböt ad vissza. A "kepek" változóban tárolom a képek elérési útvonalát (Path oszlop), és hozzáadom a szükséges elérési útvonal előtagot.

```
teszt.csv
1 Path,ClassId
2 SpektogramPictures\0\MUSE_20180111_155542_84000_AF.jpg,0
3 SpektogramPictures\0\MUSE_20180112_072946_46000_AF.jpg,0
4 SpektogramPictures\0\MUSE_20180112_073547_67000_AF.jpg,0
5 SpektogramPictures\0\MUSE_20180112_123916_57000_AF.jpg,0
6 SpektogramPictures\0\MUSE_20180112_132814_57000_AF.jpg,0
7 SpektogramPictures\0\MUSE_20180112_134829_78000_AF.jpg,0
8 SpektogramPictures\0\MUSE_20180112_134948_67000_AF.jpg,0
9 SpektogramPictures\0\MUSE_20180112_165921_53000_AF.jpg,0
10 SpektogramPictures\0\MUSE_20180113_073124_07000_AF.jpg,0
11 SpektogramPictures\0\MUSE_20180113_074139_88000_AF.jpg,0
12 SpektogramPictures\0\MUSE_20180113_123125_78000_AF.jpg,0
13 SpektogramPictures\0\MUSE_20180113_171652_90000_AF.jpg,0
14 SpektogramPictures\0\MUSE_20180113_171713_58000_AF.jpg,0
15 SpektogramPictures\0\MUSE_20180114_071920_54000_AF.jpg,0
16 SpektogramPictures\0\MUSE_20180114_123606_04000_AF.jpg,0
17 SpektogramPictures\0\MUSE_20180114_124957_60000_AF.jpg,0
18 SpektogramPictures\0\MUSE_20180114_124958_64000_AF.jpg,0
```

Ábra 35: test.csv, mely 690 adatot tartalmaz jelenleg

Ezt követően betöltöm és előfeldolgozom a képeket: Az előző lépésben megszerzett kép elérési útvonalakat végigiteráljuk a "kepek" változó segítségével. A képet megnyitjuk a "Image.open(kep_path)" segítségével, majd méretezzük a képet a (100,100) méretre a "resize((100,100))" függvénnyel. Ezután a képet NumPy tömbbé konvertálom a "np.array(kep)" segítségével, és hozzáadom az előfeldolgozott adatokat a "test_adatok" listához.

A "model.predict(X_test)" segítségével előrejelzéseket készítek a teszt adatokra. Az előrejelzések egy valószínűségi mátrixot eredményeznek. A "np.argmax(jel, axis=1)" segítségével kiválasztom az előrejelzések legnagyobb valószínűségű osztályát az oszlopok mentén. Az előrejelzések általában olyan valószínűségi mátrix formájában vannak, ahol minden sor az egyes adatpontokhoz tartozó osztályok valószínűségeit tartalmazza. A **np.argmax(jel, axis=1)** függvényhívás pedig kiválasztja az osztályok közül azt az osztályt, amelynek a legnagyobb valószínűsége van az adott adatpontokhoz. Így kapok egy tömböt, amely az előrejelzett osztályokat tartalmazza a tesztadatokhoz.

Az "accuracy_score(cimkek, jel_ln)" segítségével kiszámítom a pontossági értéket az előrejelzések és a valódi címkék alapján. A "confusion_matrix(cimkek, jel_ln)" segítségével létrehozom a confusion matrixot, amely bemutatja a helyes és helytelen osztályozásokat.

A confusion matrix (zavaros mátrix) egy olyan táblázat, amely a predikciók és a valós osztályok közötti összefüggéseket mutatja be egy osztályozási feladatban. A mátrix oszlopai az előrejelzett osztályokat, míg sorai a valós osztályokat jelölik. Minden cella a táblázatban az adott osztály párosított előrejelzési eredményeit tartalmazza. [31]

		Actual Values	
		Positive (1)	Negative (0)
Predicted Values	Positive (1)	TP	FP
	Negative (0)	FN	TN

Ábra 36 : Confusion Matrix [32]

A confusion matrix segítségével könnyen megérthetjük, hogy milyen típusú hibák történtek az osztályozási feladat során. A főbb értékek a következők:

- True Positive (TP): A helyesen előrejelzett pozitív esetek száma.
- True Negative (TN): A helyesen előrejelzett negatív esetek száma.
- False Positive (FP): A hamisan pozitívnak előrejelzett esetek száma.
- False Negative (FN): A hamisan negatívnak előrejelzett esetek száma.

A lenti kódrészletben (38. Ábra) egy függvényt definiálok, amely a confusion matrix alapján számolja ki az egyes osztályokra vonatkozó értékeket, például a TP, TN, FN, FP, PPV, TPR, TNR, NPV, ACC és DS értékeket.

```
def tn_fn_fp_tp(cm):
    results = {}

    for i in range(4):
        TP = cm[i, i]

        TN = np.sum(np.delete(np.delete(cm, i, axis=0), i, axis=1))

        FN = np.sum(cm[i, :]) - TP

        FP = np.sum(cm[:, i]) - TP

        PPV = TP / (TP + FP)

        TPR = TP / (TP + FN)

        TNR = TN / (TN + FP)

        NPV = TN / (TN + FN)

        ACC = (TP + TN) / (TP + FP + FN + TN)

        DS = (2 * TP) / ((2 * TP) + FP + FN)

        results[f"{i+1}. osztaly:" ] = {"TP": TP, "TN": TN, "FN": FN, "FP": FP, "PPV": PPV, "TPR": TPR, "TNR": TNR, "NPV": NPV, "ACC": ACC, "DS": DS}

    return results

results = tn_fn_fp_tp(confusion_matrix)
```

Ábra 37: Kódrészlet: kiszámítjuk és tároljuk a confusion matrix alapján származtatott fontos metrikákat az egyes osztályokhoz.

A számítások után nem maradt más hátra, mint kiírni az eredményeket és tárolni (39. Ábra). Tehát végigiterálok az eredmények szótáron, amelyben az előző lépésben számított eredmények találhatók az egyes osztályokra vonatkozóan.

Kiíratom az osztály nevét (osztaly_neve) és az összes releváns metrikát az adott osztályhoz tartozó osztaly_dict dictionary-ből (TP, TN, FN, FP, PPV, TPR, TNR, NPV, ACC, DS). Majd létrehozok egy Pandas DataFrame-t (df) az eredmények dictionary-ből, ahol az osztálynevek lesznek az indexek. Hozzáadom a többi értéket is a DataFrame-hez, például PPV, TPR, TNR, NPV, ACC és DS, amelyeket az előző lépésben kiszámított.

Az adatokat exportálok ezután egy CSV-fájlba (StatisticalMetrics.csv), amely tartalmazza az összes osztályhoz tartozó statisztikai mutatókat. A modellt elmentem (model.save) egy H5 formátumú fájlba (kimentett_model.h5), így később újra felhasználható lesz.

Az .h5 kiterjesztés egy fájltypust jelöl, amelyet a Hierarchical Data Format (HDF) használ [33]. Az HDF5 egy adattárolási formátum és adatmodell, amely lehetővé teszi nagy mennyiségű adat hatékony tárolását és kezelését. Az .h5 fájlok gyakran használatosak a gépi tanulásban és a neurális hálózatokban, mivel lehetővé teszik a modell struktúrájának és súlyainak megőrzését.

```

for class_name, class_dict in results.items():
    print(f"{class_name}")
    print(f"TP: {class_dict['TP']}")
    print(f"TN: {class_dict['TN']}")
    print(f"FN: {class_dict['FN']}")
    print(f"FP: {class_dict['FP']}")
    print(f"PPV: {class_dict['PPV']}")
    print(f"TPR: {class_dict['TPR']}")
    print(f"TNR: {class_dict['TNR']}")
    print(f"NPV: {class_dict['NPV']}")
    print(f"ACC: {class_dict['ACC']}")
    print(f"DS: {class_dict['DS']}")
    print("*****")

df = pd.DataFrame.from_dict(results, orient='index')

df['PPV'] = df['TP'] / (df['TP'] + df['FP'])
df['TPR'] = df['TP'] / (df['TP'] + df['FN'])
df['TNR'] = df['TN'] / (df['TN'] + df['FP'])
df['NPV'] = df['TN'] / (df['TN'] + df['FN'])
df['ACC'] = (df['TP'] + df['TN']) / (df['TP'] + df['FP'] + df['FN'] + df['TN'])
df['DS'] = (2 * df['TP']) / ((2 * df['TP']) + df['FP'] + df['FN'])

df.to_csv('StatisticalMetrics.csv', index=False)

model.save('saved_model.h5')

```

Ábra 38 : Adatok kimentése

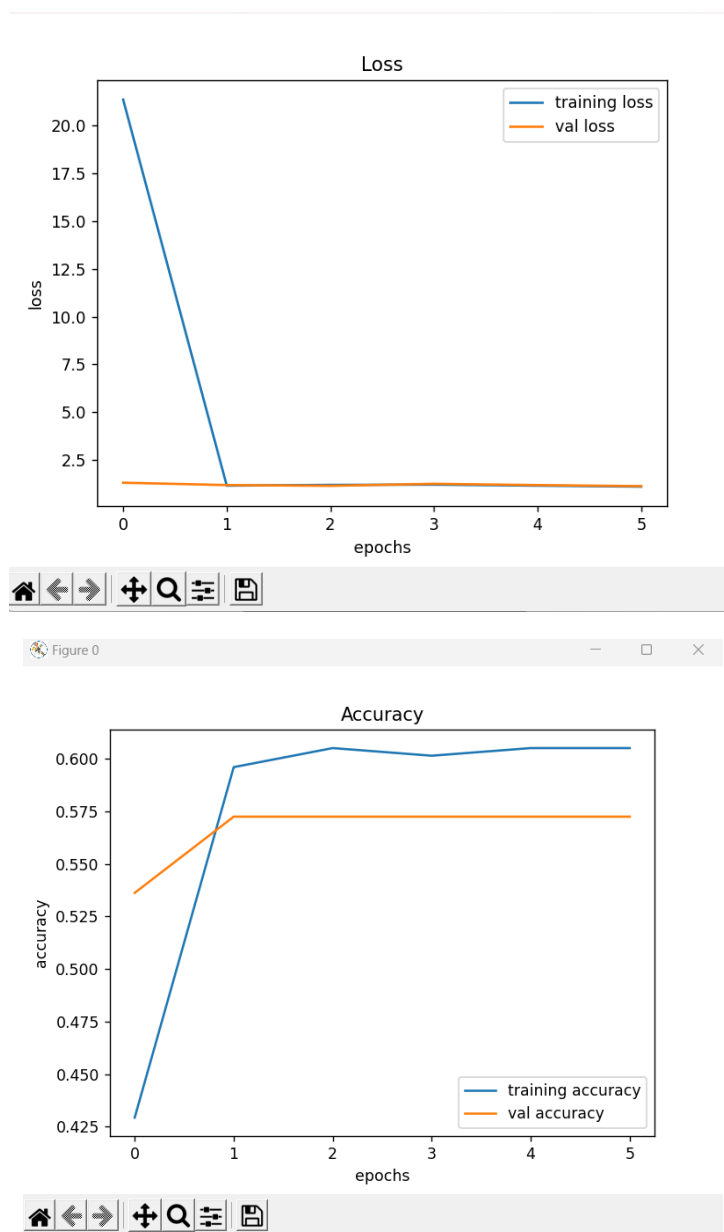
A .h5 fájlokban tárolt adatok könnyen olvashatók és írhatók Python-ban, és számos keretrendszer, például a Keras és a TensorFlow is támogatja ezt a fájlípust a modellmentés és a későbbi újratöltés céljából [33].

10.1. Tanítási paraméterek hangolása

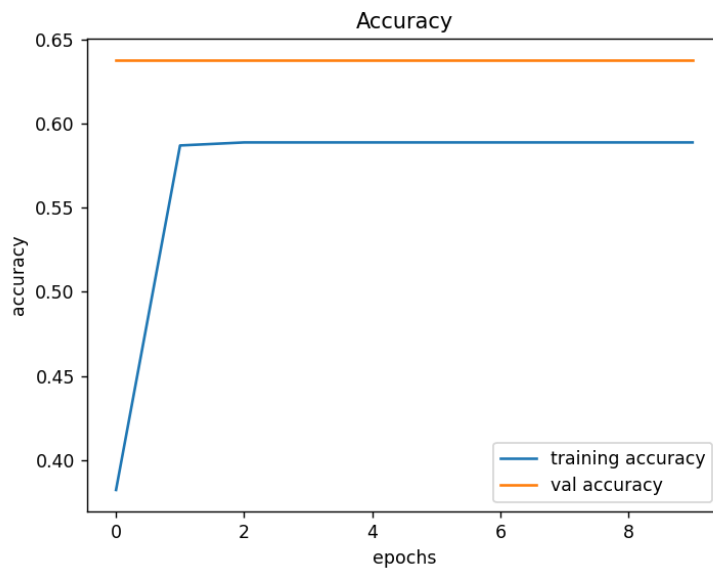
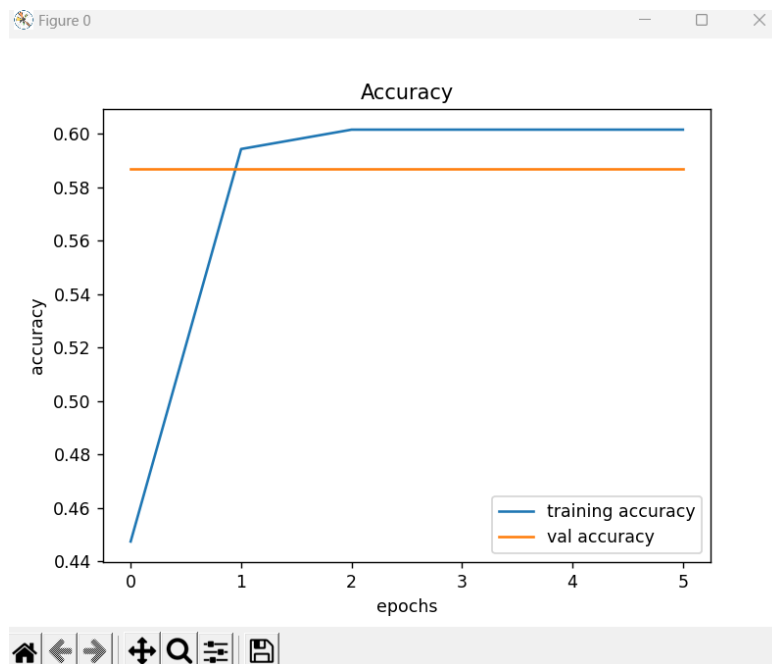
A modell felépítése után elengedhetetlen lépés a tanítási paraméterek finomhangolása annak érdekében, hogy a modell a lehető legjobb teljesítményt nyújtsa és a legmagasabb predikciós pontosságot érje el. A paraméterek hangolása során különböző beállításokat és kísérleteket végeztem annak érdekében, hogy optimalizáljam a modell viselkedését.

A tanítási paraméterek közé tartozik például az epochok száma, a batch mérete, a tanulási ráta, az aktivációs függvények típusa, szűrők méretének változtatása és még sok más.

A paraméterek hangolása időigényes folyamat volt, mivel több kísérletet kellett elvégezni és a modellt többször újra kellett tanítani. Az egyes változtatások hatását kiértékeltek a modell pontosságára és más teljesítménymutatókra. Az optimális paraméterbeállítások megtalálása érdekében több iterációra és finomhangolási lépésre volt szükség.



Ábra 39 : Accuracy és loss



Ábra 40: További tanítás

Mint ahogy látszik az első tanítási ciklusoknál még emelkedik a ‘training accuracy’, ezt követően miután eléri a 60%-ot, egy bizonyos számú ciklusnál, már csak stagnál. Következő lépésben a betanításhoz szükséges adatokon fogok még előfeldolgozást végezni, hogy még jobb képeken tudjam tanítani a modellem, ugyanakkor a modell paraméterei kerülnek még változtatásra, vagy a ciklusok száma.

```

Epoch 4/10
18/18 [=====] - 21s 1s/step - loss: 1.1030 - accuracy: 0.6196 - val_loss: 1.2067 - val_accuracy: 0.5145
Epoch 5/10
18/18 [=====] - 22s 1s/step - loss: 1.0694 - accuracy: 0.6196 - val_loss: 1.2312 - val_accuracy: 0.5145
Epoch 6/10
18/18 [=====] - 23s 1s/step - loss: 1.0855 - accuracy: 0.6196 - val_loss: 1.2277 - val_accuracy: 0.5145
Epoch 7/10
18/18 [=====] - 22s 1s/step - loss: 1.0924 - accuracy: 0.6196 - val_loss: 1.2341 - val_accuracy: 0.5145
Epoch 8/10
18/18 [=====] - 22s 1s/step - loss: 1.0986 - accuracy: 0.6196 - val_loss: 1.2304 - val_accuracy: 0.5145
Epoch 9/10
18/18 [=====] - 22s 1s/step - loss: 1.0805 - accuracy: 0.6196 - val_loss: 1.2498 - val_accuracy: 0.5145
Epoch 10/10
18/18 [=====] - 22s 1s/step - loss: 1.0816 - accuracy: 0.6196 - val_loss: 1.2008 - val_accuracy: 0.5145

```

Ábra 41: Tanítás korszakonként, itt 10 korszakon keresztül

10.2. Tesztelés, eredmények és célok

Jelenleg a modell 61%-os pontosságot ér el, de ez nem tántorít el, hanem inkább motivál, hogy tovább dolgozzak a jövőben és elérjem a célomat, hogy minimum 90%-os pontosságot érjek el.

Az eredmények elsőre talán kissé csalódást okoznak, de tudom, hogy a siker elérése időt és erőfeszítést igényel. Ezért fontos számomra, hogy ne adjam fel, hanem folytassam a modell további fejlesztését, finomhangolását és optimalizálását. Az első lépés az eredményesség növelése érdekében az, hogy alaposan áttekintem a jelenlegi modellt és az alkalmazott paramétereket. Lehet, hogy szükség lesz a modell architektúrájának módosítására, az adatok további feldolgozására vagy a tanítási paraméterek finomhangolására.

Továbbá, fontosnak tartom a minőségi adatok beszerzését és az adatkészlet bővítését. Minél változatosabb és reprezentatívabb adatokkal dolgozom, annál jobban felkészülhetek a valós környezetben előforduló esetekre.

Az előttem álló kihívásokkal tisztában vagyok, de motivált vagyok, hogy előrelépést érjek el a szív- és érrendszeri betegségek diagnosztikájában. A célom az, hogy fejlesszek egy olyan diagnosztikai rendszert, amely a legmodernebb technológiákat és eljárásokat alkalmazza a betegségek pontosabb felismerése és kezelése érdekében.

11. Hivatkozások

- [1] „Dagenais, G. R. et al. Variations in common diseases, hospital admissions, and deaths in middle-aged adults in 21 countries from,” *Lancet*, nr. 395(10226), pp. 785-794, 2020.
- [2] Juhász Marianna, 21 03 2023. [Interactiv]. Available: <https://divany.hu/testem/szivbetegseg-fiatal/>.
- [3] E. Wilkins, L. Wilson, K. Wickramasinghe, P. Bhatnagar, J. Leal, R. Luengo-Fernandez, R. Burns, M. Rayner și N. Townsend, „European Cardiovascular Disease Statistics 2017,” [Interactiv]. Available: <https://researchportal.bath.ac.uk/en/publications/european-cardiovascular-disease-statistics-2017>.
- [4] N. Townsend, L. Wilson, P. Bhatnagar, K. Wickramasinghe, M. Rayner și M. Nichols, „European Heart Journal, Cardiovascular disease in Europe: epidemiological update 2016,” 14 August 2016. [Interactiv]. Available: <https://academic.oup.com/eurheartj/article/37/42/3232/2536403>.
- [5] r. 2. Causes of death - diseases of the circulatory system. [Interactiv]. Available: https://ec.europa.eu/eurostat/statistics-explained/images/6/67/Causes_of_death_%E2%80%94_diseases_of_the_circulatory_system%2C_residents%2C_2015_HLTH18.png.
- [6] D. Dumitrascu, S. Hopulele și A. Baban, „Cardiovascular complaints following the uprising of December 1989 in Romania,” [Interactiv]. Available: <https://www.tandfonline.com/doi/abs/10.1080/07488009308409076>.
- [7] A. H. A. (. C. o. H. Disease. [Interactiv]. Available: <https://www.heart.org/en/health-topics/consumer-healthcare/what-is-cardiovascular-disease/causes-risks--prevention-of-cardiovascular-disease>.
- [8] A. Fonyó, „Az orvosi ételtan tankönyve,” p. 209.
- [9] T. Griffel . [Interactiv]. Available: <https://vitalitasportal.com/magasvernyomas-kezeles/a-verkeringes-feladata-a-sziv-felepitese/>.

- [10] UpToDate. [Interactiv]. Available:
https://www.uptodate.com/contents/image?imageKey=PI%2F71235&topicKey=PI%2F3424&source=see_link.
- [11] AmbuSkill, „A szív működés élettana,” 27 május 2021. [Interactiv]. Available:
<https://www.youtube.com/watch?v=OvDyDWepn14>.
- [12] Elektrokardiográfia. [Interactiv]. Available:
<https://hu.wikipedia.org/wiki/Elektrokardiogr%C3%A1fia>.
- [13] „Electrocardiogram-WikiLectures,” [Interactiv]. Available:
<https://www.wikilectures.eu/w/Electrocardiogram>.
- [14] b. Dr. Molnár Dóra - kardiológus, „Pitvarfibrilláció - Egy veszélyes szív működési zavar,” 28 10 2021. [Interactiv]. Available:
https://www.webbeteg.hu/cikkek/sziv_es_errendszer/14941/a-pitvarfibrillacio-tunetei-es-kezelese.
- [15] K. P. Dr. Buzáné, „Ismerkedés a TensorFlow rendszerrel,” [Interactiv]. Available:
http://biointelligence.hu/pdf/tf_bkp.pdf.
- [16] J. Zheng, „A 12-lead electrocardiogram database for arrhythmia research covering more than 10,000 patients,” [Interactiv]. Available:
<https://figshare.com/collections/ChapmanECG/4560497/2>.
- [17] „Median Based Method for Baseline Wander,” [Interactiv]. Available:
<https://openresearch.lsbu.ac.uk/download/f890f613e8768ec0265148ae2ab2df6bf0f8f26bd7718e41d31422f26ed422df/436069/Median%20Based%20Method.pdf>.
- [18] „Median filter,” [Interactiv]. Available: https://en.wikipedia.org/wiki/Median_filter.
- [19] „Fourier-transzformáció,” [Interactiv]. Available: <https://hu.wikipedia.org/wiki/Fourier-transzform%C3%A1ci%C3%B3>.
- [20] Wikipédia, „Spectrogram,” [Interactiv]. Available:
<https://en.wikipedia.org/wiki/Spectrogram>.
- [21] M. Nielsen, „neuralnetworksanddeeplearning-Deep learning,” [Interactiv]. Available:
<http://neuralnetworksanddeeplearning.com/chap6.html>.
- [22] S. Saha, „A Comprehensive Guide to Convolutional Neural Networks,” 15 December 2018.

- [Interactiv]. Available: <https://saturncloud.io/blog/a-comprehensive-guide-to-convolutional-neural-networks-the-eli5-way/>.
- [23] J. Brownlee, „Machine Learning Mastery,” 28 July 2017. [Interactiv]. Available: <https://machinelearningmastery.com/why-one-hot-encode-data-in-machine-learning/>.
- [24] fchollet, „The Sequential model,” 12 April 2020. [Interactiv]. Available: https://keras.io/guides/sequential_model/.
- [25] „Keras Sequential,” [Interactiv]. Available: <https://www.educba.com/keras-sequential/>.
- [26] „PyTorch,” [Interactiv]. Available: <https://pytorch.org/docs/stable/generated/torch.nn.Conv2d.html>.
- [27] S. Sharma, „Activation Functions in Neural Networks,” 6 Sep 2017. [Interactiv]. Available: <https://towardsdatascience.com/activation-functions-neural-networks-1cbd9f8d91d6>.
- [28] „What is the "dying ReLU" problem in neural networks?,” [Interactiv]. Available: <https://datascience.stackexchange.com/questions/5706/what-is-the-dying-relu-problem-in-neural-networks>.
- [29] T. Wood, „Softmax Function,” [Interactiv]. Available: <https://deeptai.org/machine-learning-glossary-and-terms/softmax-layer>.
- [30] „machinelearningtheory,” [Interactiv]. Available: <https://machinelearningtheory.org/docs/Unified-View/softmax-function/>.
- [31] „Confusion Matrix,” [Interactiv]. Available: https://en.wikipedia.org/wiki/Confusion_matrix.
- [32] U. C. Matrix. [Interactiv]. Available: <https://towardsdatascience.com/understanding-confusion-matrix-a9ad42dcfd62>.
- [33] „What is an H5 file?,” [Interactiv]. Available: <https://docs.fileformat.com/misc/h5/>.
- [34] J. Zheng, C. Rakovski, S. Danioko, J. Zhang, H. Yao și G. (. A. 1.-l. e. d. f. a. r. c. m. t. 1. p. Hangyuan, 2019. [Interactiv]. Available: <https://doi.org/10.6084/m9.figshare.c.4>.
- [35] S. a. O. M. c. t. -. M. r. B. D. J. Á. docens, 2002. [Interactiv].
- [36] „IBM-What is deep learning?,” [Interactiv]. Available: <https://www.ibm.com/topics/deep-learning>.

- [37] R. Holbrook și A. Cook, „Kaggle-A Single Neuron,” [Interactiv]. Available: <https://www.kaggle.com/code/ryanhobbrook/a-single-neuron/tutorial>.
- [38] E. L. Tudományegyetem, „A szív felépítése és működése,” 2017. [Interactiv]. Available: http://physiology.elte.hu/eloadas/bev_biol_3/Elettan4_szivmukodes_2017.pdf.
- [39] Doc.ro, „Timpul: Un român face infarct la fiecare 30 de minute,” [Interactiv]. Available: <https://doc.ro/stiri/timpul-un-roman-face-infarct-la-fiecare-30-de-minute>.
- [40] „Pitvarfibrilláció,” [Interactiv]. Available: <https://www.27sello.hu/kardiologia/pitvarfibrillacio>.

12. Ábrajegyzék

ÁBRA 1 : A SZÍV SZERKEZETE [9].....	14
ÁBRA 2 : A SZÍV VEZETŐ ÉS INGERÜLET KÉPZŐ RENDSZERKÉNT [10]	16
ÁBRA 3 : EINTHOVEN HÁROMSZÖG, MÉRÉSI VEKTOROK ÉS A NORMÁL JELFORMÁK.....	18
ÁBRA 4: EKG GÖRBE [14]	20
ÁBRA 5: ATRIAL FLUTTER.....	21
ÁBRA 6 : ATRIAL FIBRILLATION.....	22
ÁBRA 7 : LEFT VENTRICLE HIGH VOLTAGE	23
ÁBRA 8: T WAVE CHANGE.....	23
ÁBRA 9: MŰKÖDÉST BEMUTATÓ FOLYAMATÁBRA	25
ÁBRA 10: FELHASZNÁLÓI FELÜLET	26
ÁBRA 11 : A RENDSZER EGYSZERŰSÍTETT TÖMBVÁZLATA	29
ÁBRA 12: ADATFELDOLGOZÁS - KÓDRÉSZLET	31
ÁBRA 13: EGY EKG JEL A CHAMPMAN UNIVERSITY ÉS A SHAOXI KLINIKA KÖZÖS ADATBÁZISÁBÓL	33
ÁBRA 14: MEDIÁN SZŰRŐVEL SZŰRT JEL	34
ÁBRA 15: SZŰRÉS MEDIÁN ÉS BUTTERWORTH SZŰRŐVEL.....	35
ÁBRA 16: SZŰRT JEL, A SZŰRŐK ALKALMAZÁSA UTÁN	36
ÁBRA 17 : GYORS FOURIER- TRANSZFORMÁCIÓ	38
ÁBRA 18: GYORS FOURIER TRANSZFORMÁCIÓ KÉPLET LEVEZETÉSE [20].....	39
ÁBRA 19: SPEKTROGRAM LÉTREHOZÁSÁNAK A KÉPLETE	41
ÁBRA 20 : SPEKTROGRAM ELKÉSZÍTÉSE MAJD LEMENTÉSE.....	42
ÁBRA 21: SPEKTROGRAMOK BETEGSÉGEK SZERINT	43
ÁBRA 22: ADATOK CSOPORTOSÍTÁSA, BETEGSÉG SZERINT, EBBEN AZ ESETBEN SIMA SZINUSZ RITMUS VAN KIVÁLASZTVA.....	45
ÁBRA 23: A CSOPORTOSÍTOTT MAPPA STRUKTÚRÁJA.....	45
ÁBRA 24 : AZ ALÁBBI FÜGGVÉNY LÉTREHOZZA A SPEKTROGRAMOKAT MINDEN EGYES JELRE, MAJD EZEKET	

CSOPORTOSÍTTJA FÁJLOKBA A NEVÜK ALAPJÁN.	46
ÁBRA 25: 4x4x3-AS RGB KÉP [23]	48
ÁBRA 26: POOLING RÉTEG TÍPUSAINAK A SZEMLÉLTETÉSE [23]	49
ÁBRA 27 : AZ ADATKÉSZLETEM STRUKTÚRÁJA	50
ÁBRA 28 : ADATBETÖLTÉS	51
ÁBRA 29 : ADATOK SZÉTVÁLASZTÁSA	52
ÁBRA 30: A MODELL ÉPÍTÉSE.....	53
ÁBRA 31: MODELLEM FELÉPÍTÉSE	54
ÁBRA 32 : KÉT LEGGYAKRABBAN HASZNÁLT AKTIVÁCIÓS FÜGGVÉNY : SIGMOID ÉS RELU	55
ÁBRA 33 : SOFTMAX MŰKÖDÉSI ELVE [31]	57
ÁBRA 34: KOMPILÁLÁS, TANÍTÁS	58
ÁBRA 35: TEST.CSV, MELY 690 ADATOT TARTALMAZ JELENLEG.....	59
ÁBRA 36 : CONFUSION MATRIX [33].....	61
ÁBRA 37: KÓDRÉSZLET: KISZÁMÍTJUK ÉS TÁROLJUK A CONFUSION MATRIX ALAPJÁN SZÁRMAZTATOTT FONTOS METRIKÁKAT AZ EGYES OSZTÁLYOKHOZ.	62
ÁBRA 38 : ADATOK KIMENTÉSE.....	63
ÁBRA 39 : ACCURACY ÉS LOSS.....	64
ÁBRA 40: TOVÁBBI TANÍTÁS.....	65
ÁBRA 41: TANÍTÁS KORSZAKONKÉNT, ITT 10 KORSZAKON KERESZTÜL	66

UNIVERSITATEA SAPIENTIA DIN CLUJ-NAPOCA
FACULTATEA DE ȘTIINȚE TEHNICE ȘI UMANISTE, TÎRGU-MUREȘ
SPECIALIZAREA CALCULATOARE

Vizat decan
Conf. dr. ing. Domokos József



Vizat director departament
Ș.l. dr. ing. Szabó László Zsolt

