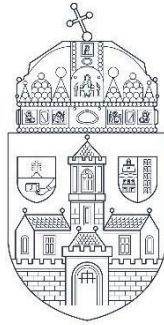


TUDOMÁNYOS DIÁKKÖRI DOLGOZAT



ÓBUDAI EGYETEM
ÓBUDA UNIVERSITY

Étkezés detektor fejlesztése kombinált mélytanuló algoritmusokkal és IPC- kompatibilis nyomtatott áramkörrel

Szerző(k):

Szántó Marcell

mérnökinformatikus MSc II. évfolyam.

Konzulensek:

Dr. Eigner György

egyetemi docens

Dr. Kertész Gábor

egyetemi docens

Budapest, 2022.

Tartalomjegyzék

1. Bevezetés	5
1.1. A népbetegség - Diabetes Mellitus	5
1.1.1. Epidemiológiai megfigyelések	5
1.1.2. Élettani és kórélettani alapok	5
1.1.3. Mérés és szabályozás	7
1.2. AI és Machine Learning	8
1.2.1. Gépi tanulás	8
1.2.2. Mélytanulás	9
1.3. Az AI és a medicina	9
2. Irodalomkutatás, előzmények	11
2.1. Adatgyűjtő eszközök	13
2.2. Korábbi gépi tanulási technikák	14
2.2.1. VGG16, Saját készítésű CNN és PairNet	14
2.2.2. AutoML SKlearn 2.0	15
2.3. Szoftver környezetek	16
3. Módszertan	18
3.1. Adatgyűjtés	18
3.2. Adattisztítás és zajszűrés	19
3.2.1. Adattisztítás	19
3.2.1.1. IMU és hang adatok tisztítása	19

3.2.2.	Zajszűrés	19
3.2.2.1.	IMU adatok szűrése	19
3.2.2.2.	Hang adatok szűrése	19
3.2.3.	Adatnormalizáció és mérési fajták	21
3.2.3.1.	Gyorsulási és szögelfordulási adatok mérése	21
3.2.3.2.	Hang adatok mérése	21
3.3.	Gépi tanulási technikák	22
3.3.1.	VGG16	22
3.3.2.	Auto-SKlearn 2.0	22
3.3.3.	DS-CNN	22
3.4.	Felhasznált felhőszolgáltatás	23
3.4.1.	Docker	23
3.4.2.	Microsoft Azure	23
3.4.2.1.	Blob Storage	23
3.4.2.2.	CosmosDB	24
4.	Metrikák és eredmények	26
4.1.	Értékelési szempontok	26
4.1.1.	AUC	26
4.1.2.	ROC	26
4.1.3.	Konfúziós mátrix	27
4.1.4.	Precision, recall és accuracy	27
4.1.5.	F1 score	27
4.1.6.	Pipeline Profiler	27
4.2.	Eredmények	28
4.2.1.	VGG16 eredmények	28
4.2.2.	AutoML-SKlearn 2.0 eredmények	29
4.2.3.	DS-CNN eredmények	29

4.2.4.	Háló kombinációk eredményei	29
4.2.4.1.	VGG16 + DS-CNN	30
4.2.4.2.	SKlearn 2.0 + DS-CNN	30
5.	Fizikai eszköztervezés	31
5.1.	Az IPC szabvány	31
5.2.	PCB tervezés	33
5.3.	Alkatrész választás	35
5.4.	Tesztelés	35
6.	Jövőbeni tervek és továbbfejlesztés	37
7.	Összefoglalás	38

Absztrakt

A társadalmunk jelentős százalékát érintő diabetes mellitus illetve az általa kialakuló patofiziológias tünetek és szövődmények okozta komplikációk megnehezítik a benne szenvedők életét. Magyarországon 1980 óta több mint 314 millióval nőtt az érintettek száma. Egy 2022-es felmérés szerint hazánkban a prediabéteszes és diabéteszes esetek száma eléri az 1,1 milliót. Az ultrafeldolgozott élelmiszerek, mozgásszegény, egészségtelen életmód és a genetikai hajlam gyakran vezethet cukorbetegség kialakulásához. A betegség az inzulin nevű hormonhoz köthető és annak hiányában vagy a sejtek inzulinérzékenységének nagyfokú romlásában, tehát az inzulinhatás elmaradásában és adott esetben az inzulinelválasztás hiányában nyilvánul meg. A vércukorszintet több élettani folyamat mellett a szénhidrátbevitel befolyásolja a legjobban. Az inzulinterápia, mint szabályozási folyamat a vércukorszint megadott határértékek között tartását hivatott elvégezni. Ebből a rendszerből az ellenőrző jelet a vércukorszint mérésével kaphatjuk meg, beavatkozni pedig inzulin vagy antidiabetikum adásával tudunk. A jelenleg forgalomban lévő inzulinpumpák nem alkalmasak vércukorszint mérésre, a páciensnek kell elvégeznie napi több alkalommal. A gyakran alkalmazott alapbólus terápia hatékonyabbá tételével javítható a diabetes mellitusban szenvedő emberek életminősége és a szénhidrátbevitelük automatikus naplózásával csökkenthető a rájuk nehezedő adminisztratív terhelést. Munkám célja olyan hardver és szoftver készítése, amely alkalmas a fent említettek elvégzésére, elhanyagolható hibázási valószínűséggel. Korábbi tevékenységem során több eszközt is teszteltem, illetve saját adatgyűjtőt készítettem, amivel az étkezés mozdulatsorát tudtam azonosítani. Mostani projektben az eddigi algoritmusok megbízhatóságát növeltem olyan módon, hogy a gyorsulás és elfordulásmérő által szolgáltatott adatokat kiegészítettem mikrofonnal rögzített hangokkal. Ezek a minták az étkezés leggyakoribb ismertetőjegyét tartalmazzák, amik az evőeszköz és tányér összeütközésekor keletkeznek. A két különböző bemenetet feldolgozó neurális háló kombinációja jobb hatékonyságot eredményezett. Továbbá kíváncsiságtól vezérelve a nyak első részére tapasztott mikrofonnal a falat nyelésének hangját rögzítettem, ami szintén egy specifikus jellemző. Egy DS-CNN modell segítségével teszteteket végeztem az adathalmaz tanulhatóságát illetően. Munkám részét képezte még egy IPC szabványnak megfelelő nyomtatott áramkör elkészítése, ami az egészségügyi eszközök esetében nélkülözhetetlen sztenderd.

1. fejezet

Bevezetés

Napjaink népbetegségeként is emlegetik a cukorbetegséget. Világszerte több millió ember életét nehezíti meg [40]. A kór patofiziológiás tünetei mellett a vércukorszintmérésre való odafigyelés, az inzulinpumpa megfelelő használata illetve a szénhidrátbevitel naplózása is terhet jelent a benne szenvedők számára. Munkám során olyan eszköz készítését tűztem ki célul, amely segít levenni az adminisztratív terhelést a diabéteszes emberek válláról és segít biológiailag kompatibilisebbé tenni az inzulinpumpás kezelést.

1.1. A népbetegség - Diabetes Mellitus

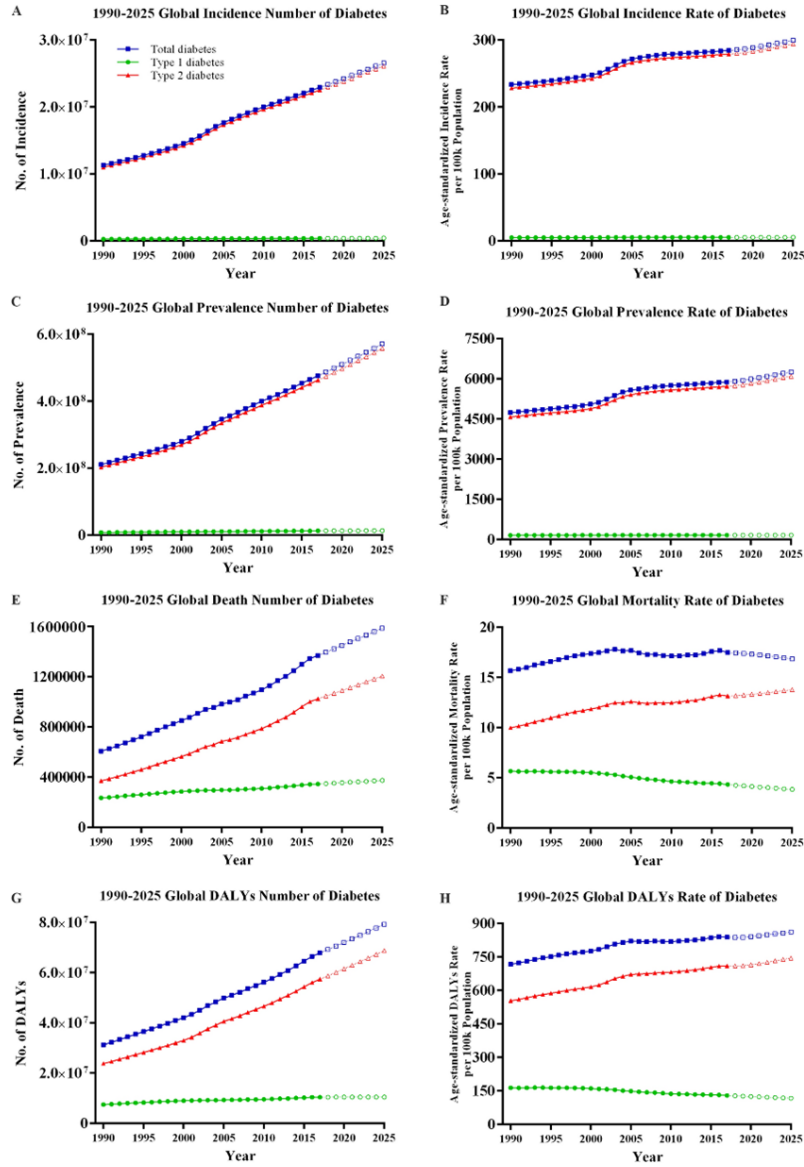
1.1.1. Epidemiológiai megfigyelések

Hazánkban a cukorbetegek száma az utóbbi 40 évben több mint 320 millióval nőtt [20]. Egy 2022-es felmérés adatai alapján jelenleg Magyarországon a diabéteszes és prediabéteszes állapotban lévők száma eléri az 1,1 millió főt [36]. Előrejelzés alapján ez a szám sajnos tovább fog növekedni [32].

A kór kialakulásának okai között feltűnik a mozgásszegény életmód, a túlzott cukorbevitel, a feldolgozott és ultrafeldolgozott ételek nagy mértékű fogyasztása, illetve a genetikai hajlam sem elhanyagolható.

1.1.2. Élettani és kórélettani alapok

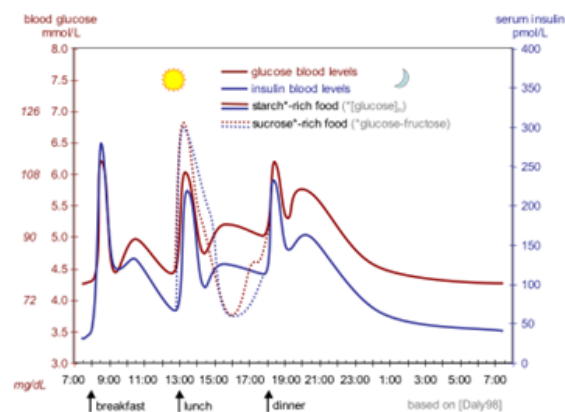
A betegségnek több típusa is létezik. Leggyakoribb az 1-es és a 2-es típusú diabetes mellitus, de találkozhatunk még gesztációs, gyógyszer okozta, hasnyálmirigy gyulladás által kiváltott és kevert típusú cukorbetegséggel is [31]. Ezek kialakulásának oka különbözik, de az élettani hatásuk nagyon hasonló. Míg az egészséges emberek szervezetében a vércukorszint a nap folyamán megadott határértékek között mozog, addig a diabéteszesek esetében ez nem mondható el. Normális esetben az első étkezés előtt mért érték 4,0 - 5,4 mmol/L közt van, míg étkezés után 120 perccel ugyanez 7,8 mmol/L körül mozog [13]. Amennyiben az egészséges tartománytól pozitív irányba jelentős eltérést



1.1. ábra. A diabéteszesek számának változása az idő függvényében [32].

mérünk, hiperglikémiás állapotról beszélhetünk. Ha az elvártnál jóval alacsonyabb vércukorszintet mérünk, hipoglikémiáról van szó. Ezt a vérben lévő glükóz paramétert az inzulin és glukagon nevű hormonok szabályozzák [[12].

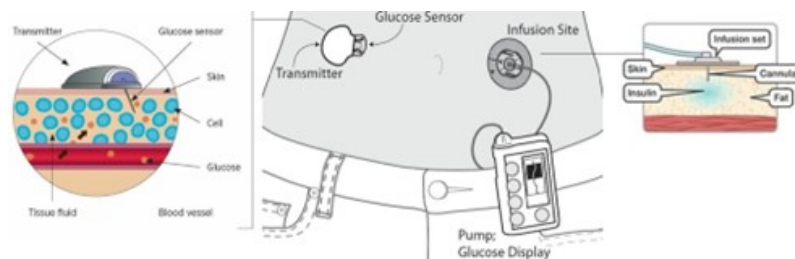
Az inzulint a hasnyálmirigy Langerhans-szigeteinek béta sejtjei termelik. Hatására a vércukorszint csökken azért, hogy az inzulin a sejteken lévő inzulinreceptorokhoz kapcsolódva glükózcsatornát nyit ki, amelyen aztán a cukor beáramlik a cellulába és ott keményítő formájában tárolódik, vagy felhasználódik. A glukagon az inzulin antagonistája, hiszen hatására a vércukorszint emelkedik. A máj glukagon hatására képes szőlőcukor előállítására, amire akkor van szükség ha a vércukorszint nagyon alacsony, de nem történik szénhidrátbevitel.



1.2. ábra. Az inzulin és vércukorszint változása a nap folyamán [18].

1.1.3. Mérés és szabályozás

A vércukorszint pontos beállításához szükség van annak megadott gyakorisággal történő mérésére. Ennek leggyakrabban használt eszköze a vércukorszint mérő készülék. Ehhez tartozik egy lándzsa, ami a vér kiserkentésére szolgál. A vércsepp a tesztsíkra kerül, aminek a vércukorszintmérő eszközhöz való csatlakoztatása után megtörténik a mérés. A cukortartalom meghatározása két elven szokott működni. Az első az amperometria. Ebben az esetben a tesztsík enzim alapú reagenssel van ellátva, amiből a vér hatására elektronok szabadulnak fel. Az ott lévő elektródán áthaladó áram nagysága a vér glükózkoncentrációjával arányos. A másik esetben a tesztsík színintenzitását vizsgálja az eszköz különböző fotometriai eszközökkel. Ezt kolorimetriának nevezzük. A vércukorszint mérésének egy másik eszköze a CGM szenzor. Ez egy olyan testre tapasztható apró eszköz, ami a benne lévő tű segítségével szövetközi vércukor monitorozásra alkalmas. Egy szenzor körülbelül 2 hétig használható és 15 percnként végez mérést, aminek eredményét egy erre a célra alkalmas eszközzel kiolvashatjuk.



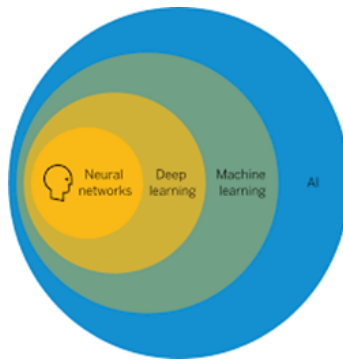
1.3. ábra. A CGMS és az inzulinpumpa elhelyezkedése [39].

A vércukorszint szabályozási folyamatába külső inzulin vagy antidiabetikum adásával lehet beavatkozni [39]. Az inzulin beadására régen inzulinfejskendőt használtak. Ma leggyakrabban inzulintollal találkozhatunk. Orvosi javaslatra az inzulinfüggő páciensek kaphatnak inzulinpumpát, ami az előző két eszközhöz képest leginkább tudja a hasnyálmirigy Langerhans-szigeteinek működését utánozni [31]. Ez a készülék magában alkalmas a bázis és bólus inzulin adagolására, amely előre kiszámított mennyiségeket foglal magába. Segítségével sokkal finomabban lehet szabályozni a vér glükózzintjét.

1.2. AI és Machine Learning

A mesterséges intelligencia napjainkban igen kutatott és nagy reményekkel kecsegtető terület. Leggyakoribb felhasználási területe az előrejelzéseket nyújtó rendszerek illetve terjedelmes adathalmazokban való mintafelismerést végző alkalmazások [44]. Több típust is megkülönböztetünk a mesterséges intelligencia algoritmusok közt az alapján, hogy mennyire tág területen képes feladatokat elvégezni úgy, hogy az akár az emberi teljesítményt is meghaladja. Ezen csoportosítás alapján megkülönböztetünk három típust [22]:

- szűk mesterséges intelligencia
- általános mesterséges intelligencia
- mesterséges szuperintelligencia



1.4. ábra. Az AI típusok felosztása [22].

A szűk MI (weak AI) annyit takar, hogy az algoritmus egy adott feladatot el tud végezni igen nagy, adott esetben akár az embernél is nagyobb pontossággal. Az általános MI (strong AI) már képes az emberi eredményeken túlmutatni gyakorlatilag bármilyen intellektuális területen. Ekkor a gép már emberi szintű kreativitással rendelkezik. A mesterséges szuperintelligencia (ASI) az emberi képességeket minden területen képes lenne felülmúlni. A kutatási eredmények azt mutatják, hogy ez még nem valósult meg, de a terület magasán kutatottsága és egyre mélyebb ismereteink idővel lehetővé teszik az ASI elkészítését is. A mesterséges intelligencia elkészítését tanulási folyamatnak szoktuk nevezni. Ennek két fajtáját különböztetjük meg [22]:

- gépi tanulás
- mélytanulás

1.2.1. Gépi tanulás

A gépi tanulás célja, hogy az algoritmus képes legyen explicit feladatok végrehajtására azok leprogramozása nélkül. Több részkategóriákról is beszélhetünk a kategóriában. Az első a felügyelt tanulás. Ebben az esetben jelentős emberi beavatkozás szükséges. Lényegében az ember tanítja

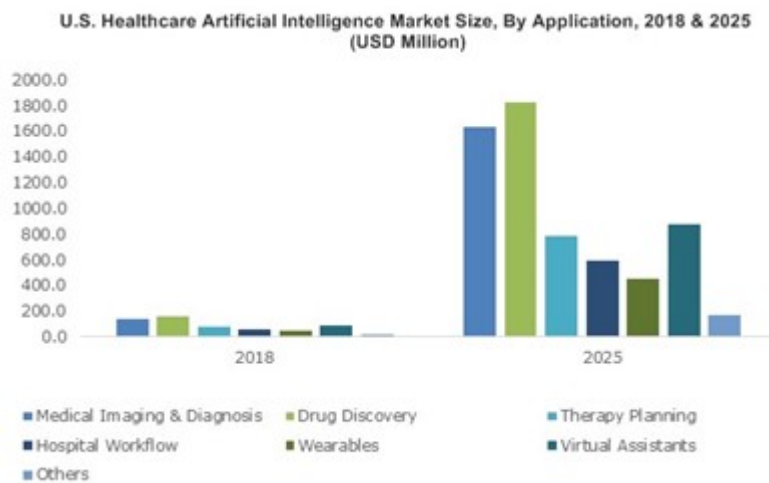
meg a modellt arra, hogy hogyan reagáljon a bemenő adatokra. A második kategória a félig felügyelt tanulás, amikor a modellt címkézett és címkézetlen adatok keverékével tanítják. Itt már kevesebb emberi beavatkozás szükséges. A harmadik kategória a felügyelet nélküli tanulási mód, amiben az algoritmus saját maga rendezi megadott kategóriákba a bemenő adatokat és önállóan keres mintázatokat. A negyedik kategória a megerősítéses tanulás, amikor az algoritmus a tanulás és működés során folyamatosan értékelésre kerül, amikor döntést hoz. Ha a döntése helyes, akkor pozitív megerősítést (jutalmat) kap, ha helytelen, akkor negatív megerősítést (büntetést). Ez a módszer eredményezi a leginkább rugalmas modelleket, amelyek a működésük során bármilyen bemeneti adathoz a leginkább tudnak alkalmazkodni.

1.2.2. Mélytanulás

A mélytanulás az emberi agy működéséhez hasonlóan szerkezetet használ, ezt neurális hálónak nevezzük. A tanításához sokkal nagyobb adatmennyiség szükséges mint a gépi tanulás esetében, de jóval rugalmasabb és kétértelműségeket is könnyebben képes kezelni. A mélytanulás esetében több algoritmust is használhatunk, mint például a konvolúciós neurális hálózatok, az ismétlődő (recurrent) neurális hálózatok, stb. A neurális hálózatok több rétegből állnak, amely rétegek között haladva a bementi adatok egyre absztraktabbak lesznek.

1.3. Az AI és a medicina

A mesterséges intelligencia alkalmazása a tudományok számos területén megfigyelhető, köztük az orvostudományokban is. Az egészségügy egy olyan terület, ahol nap mint nap rengeteg digitális adat képződik, ami az MI tanításához nélkülözhetetlen. Jelenleg ezen adatok feldolgozásának a nagy része emberi erővel történik. Ez amellett, hogy sok időt és energiát igényel, magában hordozza a hibázás lehetőségét. A leletek kiértékelése leginkább hosszadalmas és odafigyelést igénylő folyamat illetve az ebben történő hibázás akár emberi életet is követelhet. Szerencsére a mesterséges intelligencia kiválóan hasznosítható ezen a területen. Ma már számos helyen találkozhatunk vele, mint például a tüdőszűrés eredményeit vizsgáló neurális hálók, a patológia szöveteket analizáló algoritmusok, a röntgen és CT felvételeket kielemező MI-k, vagy akár olyan technológiák, amik a daganatos betegséggel küzdők kemoterápiás kezelésének optimalizálását hivatottak elvégezni [11]. Ezek mellett a cukorbetegség terápiájában is van szerepük, például létezik olyan fél-automata rendszer, ami a mesterséges hasnyálmirigy koncepciója szerint működik [24]. Jelen pillanatban folynak olyan irányú kutatások, amik a megerősítéses tanulás segítségével a vércukorszint határértékek között tartását biztosítják [37]. Ezen alkalmazások mellett még sok kisebb AI alapú alkalmazás létezik, amiknek egy nagy hányada a prevencióban betöltött szerepe által gyakorol hatást az életünkre [44]. 2018-ban végeztek egy felmérést illetve szakértők véleményét kikérve készítettek egy predikciót a mesterséges intelligencia egészségügyben betöltött szerepét illetően [11]. Ennek eredménye azt mutatja, hogy az AI felhasználása jelentősen növekedni fog a jövőben. Ezt szemlélteti az alábbi ábra.



1.5. ábra. Az AI medicinában betöltött szerepköre [37].

2. fejezet

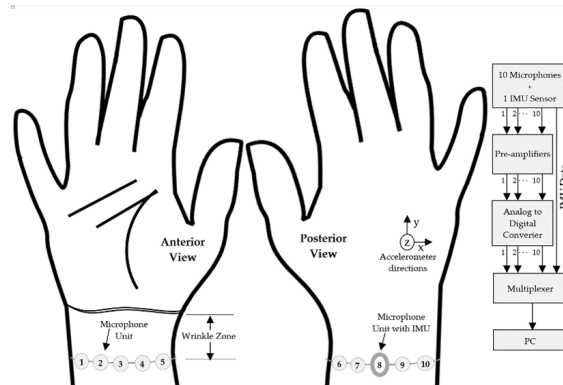
Irodalomkutatás, előzmények

A gesztusdetekció jelenleg egy intenzíven kutatott terület, hiszen a hétköznapiakban sok helyen alkalmazhatjuk. Például a HMI-k esetében egy teljesen új dimenziót nyithat meg a felhasználói élmény tekintetében. Emellett a pandémia is motiválta a kutatókat arra, hogy olyan érintés nélküli irányítási technikákat fejlesszenek ki, amik könnyen használhatók az átlag emberek számára. Munkám során én nem a gép-ember kommunikációra összpontosítottam, hanem magának egy mozgásformának a felismerését valósítottam meg. Korábbi kutatásaim során már értem el kisebb nagyobb sikereket a gesztusdetekcióban, de szerettem volna egy megbízhatóbb algoritmust készíteni [34]. Emiatt úgy döntöttem, hogy a korábbi IMU alapú mozgásfelismerő algoritmusokat a gyorsulásmérő 3 tengelyén mért adatok mellett a szögelfordulási értékekkel is kiegészítem. Reményeim szerint és később már tapasztalataim alapján is ez a megoldás javított az algoritmusok pontosságán. Több szakirodalomban találkoztam olyan jelenséggel, hogy a kutatók kiegészítették az IMU adatait más szenzorokból nyert értékekkel, ezzel javítva azok hatásfokát. Többen próbálkoztak az EMG technológia felhasználásával, ami az alkar könyökhöz végének alsó oldalán elhelyezett 32 elektróda segítségével valósult meg. Ez az izmok illetve inak megfeszüléséből nyert ki információt. Hidden-Markov modell készítésével nem egyénfüggő algoritmust készítettek a szakemberek, aminek pontossága 96,6% [35]. Ez igen jó érték, ugyanakkor azt észre kell vennünk, hogy a hétköznapiak tekintetében nem használható a fizikai kialakítás. Emiatt szerettem volna a hangot mint információforrást használni, hiszen ott a levegő az átvivő közeg és nem szükséges extra kábel.

Korábbi kutatásaim során egy MPU6050 6 tengelyű gyorsulás és elfordulásmérő szenzort használtam az adatgyűjtésre. Több helyen is találkoztam ezzel az eszközzel a szakirodalomkutatás során, ennek oka a megbízhatóság és pontosság mellett az olcsó ár és könnyű beszerezhetőség. Nem voltam egyedül azzal, hogy csupán ebből az egy szenzorból gyűjtött gyorsulás és elfordulás értékekkel próbáltam meg neurális hálót tanítani. A romániai Gheorghe Asachi Technikai Egyetem kutatói erre a dataset-re RCE (Restricted Coulomb Energy) és MLP (Multi Layer Perceptron) gépi tanulási algoritmusok segítségével próbáltak gesztusdetekciót készíteni. Munkájuk gyümölcse egy 85,4%-os és egy 88%-os modell lett [11]. Ezek mellett egy újabb technológiát is teszteltek, ami a DTW algoritmusban mutatkozott meg. Ez a Dynamic Time Warping rövidítése. Olyan idősor elemzéses algoritmust jelent, amely két időbeli sorozat közti hasonlóságot méri. Ezzel jól meghatározható a periodikusan ismétlődő mozgásforma, illetve a különböző étkezési sebességek sem jelentenek kihívást. A módszer eredménye 98,6% volt. A szakirodalomban leggyakrabban DTW,

MLP és CNN (Convolutional Neural Network) algoritmusokkal találkozhatunk.

Az általam használt mikrofonos IMU kiegészítés más kutatóknak is eszébe jutott, viszont a felhasználás kicsit különbözik. Míg én az étkezés hangjait próbálom felismerni mikrofonos adatgyűjtés révén, addig a hong kongi City University of Hong Kong egyetem kutatói az alkar bőrfelületén a nyomásváltozást vizsgálták. Minkét csuklóra 10-10 mikrofonból álló szenzoregységet helyeztek el, amely a bőrfelülethez tapadva alkalmassá vált az apró mozgások által keltett nyomásváltozások érzékeléséhez. A szakirodalomkutatás során számomra ez a megoldás tetszett a legjobban, úgy gondolom egy igen ötletes felhasználása volt ez a mikrofonoknak. Az IMU és mikrofontömb közös felhasználásából LDA (Latent Dirichlet Allocation) neurális hálót tanítva 80%-os pontosságú, míg SVM (Support Vector Machine) esetén 75%-os pontosságú modellt kaptak [8] A mérési elrendezés sematikus ábrája alább látható.



2.1. ábra. IMU és mikrofontömb segítségével megvalósított adatgyűjtés [8].

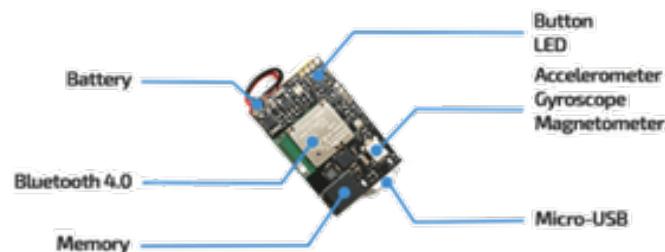
Mások olyan irányba indultak el, hogy a kéz mozdulatai helyett inkább a fej mozgását vizsgálták, amely egészen érdekes összefüggésekre világított rá a mintavételi frekvencia, a szabadságfokok és a tanított háló pontossága tekintetében. Munkájuk eredményéből jól látható, hogy a mozgás szabadságfokainak növelésével javult a háló teljesítménye. 3 szabadságfok esetén 75%, 9 szabadságfok esetén 87,56% volt a pontosság. A mintavételi frekvencia fontosságát ott lehet észrevenni, hogy míg a kéz mozdulatainak vizsgálata esetén sokszor elég a 20 Hz, addig a fej apróbb, gyorsabb mozgásainak érzékelésére a javasolt frekvencia minimum 50 Hz Ionut Cristian Severin és kutatótársai szerint [23].

Az Egyesült Királyság Sussexi Egyetemének kutatói az LSTM neurális hálókkal végeztek komoly munkát, mely során tizennyolc fajta kézmozdulat felismerésére tettek kísérletet. Az általuk választott MI algoritmus a finom motoros mozgások időbeni dinamikájára fókuszál és ez alapján próbál felismerni bizonyos mozgásokat. Az általuk publikált eredmények motiváltak engem arra, hogy a korábbi gyorsulási adatok mellett a szögelfordulási értékeket is adjam a tanítani kívánt algoritmus bemenetére, hiszen munkájuk kimenete azt mutatta, hogy a használt technológiák teljesítménye 15%-kal javult, amikor az accelerométer adatok mellett giroszkópból kapott értékeket is a bemenethez csatolták. Abban az esetben amikor a bemenő vektort tovább bővítették az előző két paraméter mellett mágneses érzékelők adataival, a kimeneten 20%-os javulás volt megfigyelhető [10].

2.1. Adatgyűjtő eszközök

Korábbi munkám során számos eszközt használtam adatgyűjtésre, hogy a saját adatbázisomat fel tudjam építeni. Először egy Arduino alapú MPU6050-et tartalmazó eszközt készítettem, melyet a csuklóra lehet rögzíteni és 12 órán keresztül képes adatokat gyűjteni, melyeket egy memóriakártyára ír ki. Az eszköz 75 Hz-es mintavételi frekvenciával működik, érzékelési tartománya pedig $250^\circ/\text{sec}$ -tól $2000^\circ/\text{sec}$ terjed. Gyorsulási értékek tekintetében a működési tartománya a ± 2 - ± 16 között van. A szenzor 16 bites A/D átalakítóval van felszerelve, amely bőven elég pontosságot szolgáltat az általam támasztott igényekhez [38]. Az mérőkarpánt elkészítésére a motiváció a költséghatékonyság, a technológia mélyebb megismerése és a szabadabb konfigurálhatóság volt.

Az MbientLab nevű cég készített egy MetaMotionS nevű eszközt, amely az általam készített adatgyűjtőnek egy sokkal kisebb és több szenzorral rendelkező tömeggyártott változata. Az készülék 48 órán keresztül képes adatot gyűjteni a benne lévő 100mAh-s akkumulátor segítségével. Legkisebb mintavételi frekvenciája 0,78 Hz, a legnagyobb 1600 Hz a gyorsulásmérő esetében. A szögelfordulásmérő tekintetében a 25 Hz és a 3200 Hz közötti tartományban képes dolgozni az eszköz. Lehetőség van az adatok real-time kinyerésére Bluetooth-on keresztül. Ez a karpánt kiválóan alkalmas étkezési gesztusok mozgási adatainak gyűjtésére, hiszen a Bosch BMI270 IMU szenzor a gyártó szerint erre a leginkább alkalmas eszköz, illetve az MbientLab specifikációjában találkozhatunk a naplózás és gesztusfelismerés kulcsszavakkal [25]. Az eszköz tesztelése során megvizsgáltam, hogy milyen pontossággal képes tartani a beállított mérési frekvenciát, hiszen idő alapon működő neurális hálók esetében ez kulcsfontosságú paraméter. Tapasztalataim szerint 0,7%-os hibával dolgozik a rendszer.



2.2. ábra. MetaMotionS karpánt nyomtatott áramköre [25].

Egy harmadik eszköz, aminek segítségével adatokat gyűjtöttem, a telefonom volt. Ma már minden okostelefon rendelkezik beépített gyorsulásmérővel, melyet az AndroSensor nevű alkalmazás segítségével egyedi mérésre is tudunk használni [1]. Az adatgyűjtésbe való bevonása által így egy egyedi orientációban működő szenzorral is sikerült bővíteni a készletet, ugyanis a mérés során a telefont a csukló felső részére rögzítettem, de méretéből adódóan hátrébb került mint egy karpánt vagy óra. A program segítségével lineáris gyorsulási, gyorsulási és szögelfordulási adatokat, valamint a hozzájuk tartozó időbélyeget rögzítettem. A timestamp-ek segítségével utólag tudtam címkézni a méréseket másodpercre pontosan, hogy mikor történt étkezés. Az eredményeket a program .csv kiterjesztésű fájlban tárolja, ami megegyezik az általam is használt formátummal.

2.2. Korábbi gépi tanulási technikák

A kutatás korábbi szakaszában konvolúciós neurális hálókat és automatizált gépi tanulási folyamatot is teszteltem. Ezek rövid leírása és a hozzájuk tartozó eredmények alább találhatóak. Minden esetben .csv fájlokból történt az adatok beolvasása. Bementi halmazként a saját magam végzett mérési eredményeket használtam, mely 71 rövid és hosszabb étkezés gyorsulás és szögelfordulás értékeit tartalmazza összesen 310.000 rekordban. A tanítást különböző dataset felosztásokkal végeztem, melyekben sor került önvalidációra, a 75/25-ös, 80/20-as felosztásra, illetve egész napos mérések esetén 3/1 és 4/4-es részekre bontásra is. Az adatokon minden esetben adattisztítást végeztem, ami magába foglalta a hiányzó és a kilógó értékek helyreállítását. A CNN esetében a zajszűrés negatívan befolyásolta a háló teljesítményét, de az AutoML algoritmusoknál az IIR (Infinite Impulse Response) filter pozitív hatással volt a végeredményre. Az adathalmaz mintavételi frekvenciája 5 Hz volt [30].

2.2.1. VGG16, Saját készítésű CNN és PairNet

A VGG16 egy olyan konvolúciós neurális háló, ami 16 réteget tartalmaz. Ez egy igen nagy paraméterszámú algoritmus, ami az egyszerű architektúrája révén sok feladat megoldására használható. Alapvetően képekkel való munkára tervezték, de gyakran használják más témakörökben is rugalmassága miatt [19]. A VGG16 CNN esetén 20 másodperces időablakkal dolgoztam és a bemenő adatok nem voltak megsűrűve.

VGG16 eredmény	Accuracy	Precision	Recall	AUC
Pontosság	0.9497	0.8970	0.9320	0.9800

2.1. táblázat. A VGG16 háló legjobb eredménye

A saját készítésű CNN alapkonceptiója az volt, hogy a gyorsulásmérő adatait tengelyspecifikusan külön – külön kezelem. Ezzel csatornaspecifikus információkat tudtam kinyerni, melyek sokkal mélyebb összefüggésekre is fényt vethetnek. A csatornából származó tenzorokat végül egy concatenate réteg fűzte össze a feature tengely mentén. A háló legvégén egy szigmoid aktivációs függvénnyel ellátott fully-connected réteg van, ami 0 és 1 kimenettel jelzi azt, hogy étkezés tartalmazott-e a bemenő adatvektor. A saját készítésű CNN esetén 20 másodperces időablakkal dolgoztam és a bemenő adatok nem voltak megsűrűve.

Saját háló eredmény	Accuracy	Precision	Recall	AUC
Pontosság	0.9545	0.9627	0.8720	0.9870

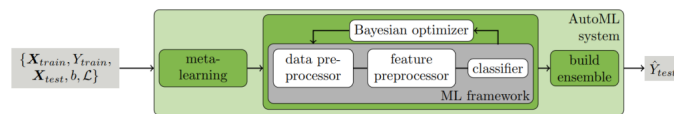
2.2. táblázat. A saját készítésű háló legjobb eredménye

A PairNet egy nemrég publikált architektúra, ami egy relatív sekély hálót jelent. Emiatt a működése gyors, sokkal kisebb a memóriaigénye, de nem képes igazán mély összefüggések felismerésére [43]. Lényege, hogy a konvolúciós rétegek lépésközei a filterek méretével együtt kettes értékre van állítva. A bemeneti adatvektor egy (50,3)-as vektor volt, amiben az 50 a tíz másodper-

ces időablak értékeit tartalmazza, a 3 pedig a gyorsulásmérő X,Y és Z tengelyeinek adatait jelenti. Végeredményben a PairNet nem tudott rátanulni a bemeneti adathalmazra, melyre a 0,51-es ROC értékből következtettem.

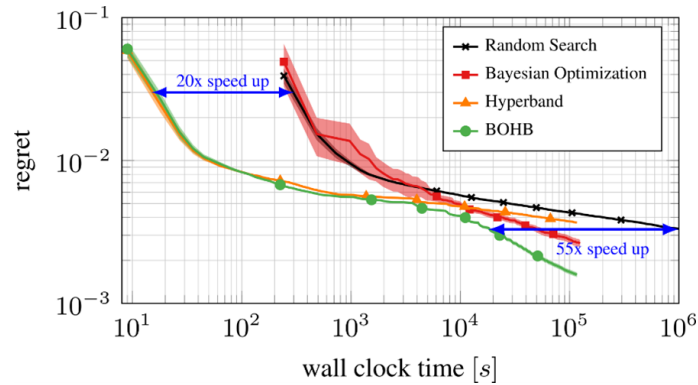
2.2.2. AutoML SKlearn 2.0

Az AutoML SKlearn a Scikit-learn könyvtárat használja az adatok megfelelő átalakításához és a gépi tanulási algoritmusokhoz. Előnye, hogy segít kiválasztani több modell közül, hogy melyik a legjobban teljesítő, ehhez pedig egy Bayes-optimalizálást használ. Végeredményként megkapjuk a legjobb AI modellt. az ahhoz tartozó hiperparaméterekkel, illetve egyéb osztályozókat is kapunk a hozzájuk tartozó paraméter javaslatokkal, amelyek közül aztán szabadon tudunk választani.



2.3. ábra. Az AutoML algoritmus felépítése [16].

Az AutoML-SKlearn három fő komponensből tevődik össze. Ezek a metatanulás, a Bayes-i optimalizálás és az együttes modellkialakítás [17]. Az AutoML-SKlearn 2.0 már több fejlesztést is tartalmaz az elődjéhez képest. Például a Bayes-optimalizálás helyett BOHB (Bayesian Optimizer and HyperBand) módszert alkalmaztak, ami egy multi-fidelity optimalizálási módszer, ami sebességben sokkal jobb az elődjéhez képest [boh]. Emellett a fejlesztők javítottak az algoritmus rugalmasságán, automatizálták a meta-optimalizálás problémáját, valamint új funkciók is megjelentek [2].



2.4. ábra. BOHB optimalizálás versus Bayes-optimalizálás sebessége [2].

AutoML-SKlearn 1.0	Nem filterezett	Savgol	Wiener	IIR	Medián
Pontosság	0.8799	0.9433	0.9704	0.9655	0.9447
F1-score	0.9102	0.9549	0.9758	0.9786	0.9534

2.3. táblázat. Az SKlearn 1.0 eredménye

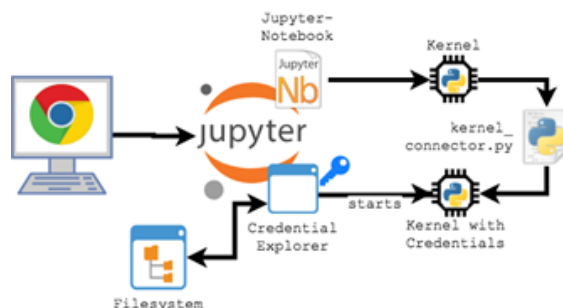
AutoML-SKlearn 2.0	Nem filterezett	Savgol	Wiener	IIR	Medián
Pontosság	0.9089	0.9456	0.9621	0.9822	0.9187
F1-score	0.9134	0.9675	0.9737	0.9876	0.9210

2.4. táblázat. Az SKlearn 2.0 eredménye

Az SKlearn 1.0 és 2.0 algoritmus esetében is több adathalmaz felosztással próbálkoztam. A hálók egyértelműen jobb eredményeket adtak szűrt adathalmazon, azon belül is az IIR filter szolgáltatta a legjobb hatásosságot. Az SKlearn 1.0 Gradient Boosting algoritmust javasolt az adathalmazhoz, ami jellemzően döntési fákból épül fel. A működése során egyszerű fákból kiindulva azok hibáit javítja és ezzel új fát hoz létre. Ezt addig folytatja, amíg a lehető legjobb eredményt nem kapja [7]. Az SKlearn 2.0 által javasolt algoritmus a Random Forest volt, ami hasonló a Gradient Boosting-hoz annyiban, hogy szintén döntési fákból építkezik. A folyamat során egyre több át von be, amivel egyre nagyobb lesz az erdő és így egyre komplexebb feladatok megoldására lesz képes [42]. A Gradient Boosting nagy hátránya a Random Forest-hez képest, hogy hajlamos a túltanulásra.

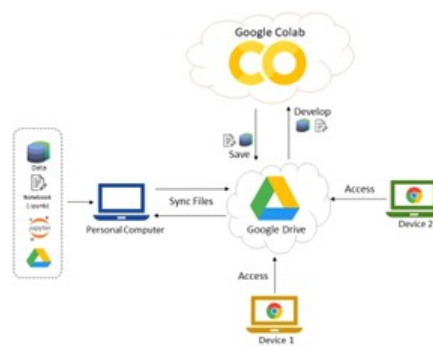
2.3. Szoftver környezetek

Az algoritmusok elkészítéséhez, tanításához és teszteléséhez Jupyter Notebook App-ot, Google Colaboratory-t és PyCharm fejlesztői környezetet használtam. A Jupyter egy kliens-szerver alkalmazás amiben notebook dokumentumokat lehet szerkeszteni és futtatni böngészőn keresztül. Lehetőség van távoli és lokális futtatásra is [5].



2.5. ábra. A Jupyter Notebook architektúrája [5].

A Colab egy Jupyter Notebook szolgáltatás, melyet a Google hozott létre kutatási célból, de bárki számára elérhető. Használatához nem szükségesek komoly beállítások, ugyanakkor lehetőséget nyújt távoli számítási erőforrások használatára (CPU + GPU) [4]. A használatának nagy előnye, hogy a tanításhoz használt fájlok tárolására lehetőség van Google Drive-ban, ami megkönnyíti a beolvasási folyamatot és bárholon elérhető illetve futtatható a modell.



2.6. ábra. A Jupyter Notebook architektúrája [4].

3. fejezet

Módszertan

Munkám során adatgyűjtést végeztem mind a mozgás mint a hang tekintetében. A kéztől származó információkat a korábban bemutatott saját készítésű adatgyűjtővel végeztem. Korábban már rendelkeztem 310.000 rekordnyi adattal, de az általam megvalósított kombinált neurális hálókhoz új adatokra volt szükségem, mert a régi méréseket nem tudtam időben szinkronizálni a hangfelvételhez. A hang rögzítésére Keyes KY-038 hangdetektort [9] használtam, amit hozzáépítettem a régi adatgyűjtőhöz. A mérések szinkronizálásának segítése végett minden adatrögzítést egy határozott koppintással kezdtem, mert ez az IMU adatokban és a hangfüggvényen is egy rövid de magas csúcsnak felel meg.

3.1. Adatgyűjtés

Jelen dolgozat elkészítéséhez 31 mérést végeztem, amiben már mindkét információs csatorna adatai rögzítésre kerültek. Teljes napos mérésekre nem került sor az idő hiányában, ám mindenképp szükséges a jövőben, hiszen a nap során igen sok hanghatással találkozunk. A 31 mérés összesen 368 percnyi étkezést tartalmaz. A hang rögzítésének mintavételi frekvenciája 44.1 kHz volt. Emellett külön készítettem olyan hangrögzítéseket, amikben az evőeszközöket ütöttem különböző anyagú tárgyakkal, edényekkel. Az alap elképzelésem az volt, hogy a korábbi algoritmust kiegészítem egy olyan neurális hálóval, ami az tárgy és evőeszköz ütközésének hangjára triggerelődik. Később ezt még kiegészítettem egy nagyon jellegzetes hang felismerésével, ami a fémdobozos italok nyitókupakjának a reccsenését és szisszenését jelenti. Ezzel nem lehet minden étkezésfajtát felismerni, ugyanakkor a tárgyból való evés illetve az energiatartalmú és sör fogyasztásának detektálása megbízhatóbb lett. A mérések közt kettő olyan található, amely étteremben került rögzítésre. Ennek fontosságát az alapzaj megfelelő kiküszöbölésének igénye adta. A sörös és energiatartalmú italok nyitásától 15 mérés készült.

3.2. Adattisztítás és zajszűrés

Az korábbi munkám során megtapasztaltam, hogy a gyorsulásmérővel rögzített adatok nagy mennyiségű zajt tartalmaznak, ami a kéz és asztal remegéséből fakad, de létrejöhet egy jármű, például vonat vagy villamos mozgásából adódóan is, amennyiben az étkezés járművön történik. A hang esetében sincs ez másképp, a helyzet inkább komplikáltabb, mert a környezetünkben élő más emberek gyakorlatilag folytonos zajforrásnak tekinthetők igen sokszínű és erősségű zavaró tényező kibocsátásával.

3.2.1. Adattisztítás

Az adattisztítás során a hiányzó vagy kilógó adatok eltávolításával és odaillő értékek pótlásával foglalkoztam.

3.2.1.1. IMU és hang adatok tisztítása

A gyorsulás és szögelfordulás szenzor esetében ennek nagyobb relevanciája volt, mert a mintavételi frekvencia 75 Hz volt és az egységnyi pillanatig tartó extrém értékek nem jelenthetnek valószínű elmozdulást. A hang esetében nem végeztem adattisztítást, mert a környezeti zajok túl nagy mennyiségben voltak jelen. Emiatt ha a kilógó értékeket normalizáltam volna, akkor lehet hogy számomra fontos információkat veszítek el. Az IMU adatok .csv fájlokban vannak tárolva, így egy egyszerű Python script segítségével el tudtam végezni az adattisztítást. Az egységnyi outliereket helyettesítettem a környező értékek átlagával, valamint a hiányzó rekordok esetében is ugyanígy tettem.

3.2.2. Zajszűrés

A zajszűrés korábbi tapasztalataim szerint nagyon fontos, mert a mért jelre sok zaj szuperponálódik, amely aztán negatívan befolyásolhatja a mesterséges intelligencia algoritmus tanulási képességét.

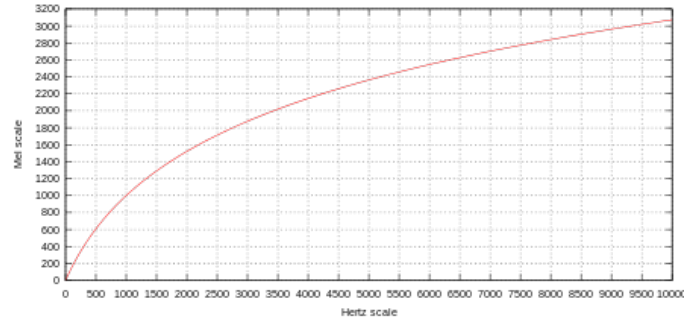
3.2.2.1. IMU adatok szűrése

Az IMU adatok tekintetében a korábbi tapasztalatok miatt az IIR filter használatára szorítkoztam. A végtelen impulzusválaszú szűrő működése folyamán autoregresszív mozgóátlagot használ [30]. Esetemben igen jó választás, mert a sztochasztikus idősorlevezésekhez hasonlóan itt is kiszámíthatatlan zajokkal terhelt jel volt az alap.

3.2.2.2. Hang adatok szűrése

A hangok esetében más zajszűrést kell alkalmazni. Erre a mel-spektrogram jelenti a megoldást, ami lényegében egy jel frekvenciájának logaritmikus transzformációját tartalmazza [33]. Ez lényegében

egy olyan skála, amin az egyes értékek olyan távolságra vannak megjelenítve mint ahogyan azt az emberi fül is hallaná. Tehát a mel skála segítségével a különböző frekvenciájú hangokat az emberi szám által is jól értelmezhető ábrán tudjuk megjeleníteni. Nem létezik egy univerzális mel skála, viszont a legtöbb képlet úgy határozza meg, hogy 1000 Hz-hez 1000 mel-t rendel a hallható küszöb fölött 40dB-nél és a frekvencia növekedésével csökken az együttható amivel növeljük a frekvenciához tartozó mel értéket [41]. A mel skálát szemlélteti az alábbi ábra.

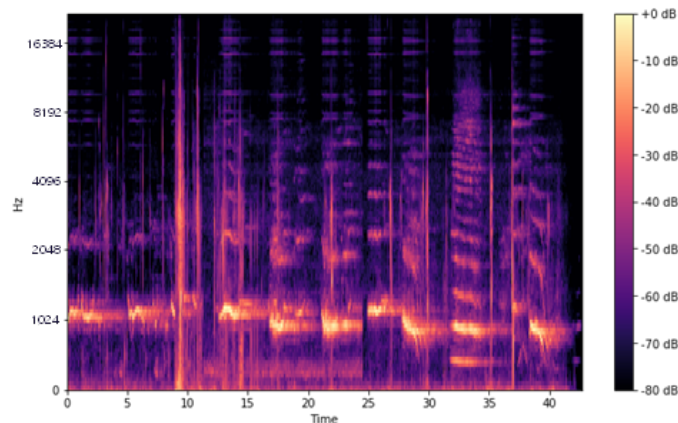


3.1. ábra. A mel skála és a frekvencia skála közti függvénykapcsolat [33].

A frekvencia és a mel közti átváltás képlete a következő [41]:

$$m = 2595 \cdot \log_{10} \left(1 + \frac{f}{700} \right) \quad (3.1)$$

A mel spektrogram segít abban, hogy csak egy meghatározott részére összpontosítsunk egy hangrészletnek. A következő képen látható 45 másodpernyi étkezés mel spektrogramja mely során levest fogyasztottam el. Jól látható ahogyan 1kHz környékén kirajzolódik a kanál és tányér ütközésének hangja. A mérést ábrázoló esetben csekély volt a háttérzaj, emiatt látható relatíve tiszta spektrogram.



3.2. ábra. Leves kanalizásának mel spektrogramja.

Azokban az esetekben, amikor a háttérben egyéb zajforrások voltak, szükség volt a spektrogram szűrésére. Ehhez sáváteresztő szűrőt használtam, amit az evőeszköz és tányér koccanásakor keletkező hang illetve a fémdobozos üdítőital nyitásakor létrejövő hang frekvenciájának áteresztésére kalibráltam. Empirikus tapasztalataim alapján az első esetben a 900 Hz – 1200 Hz közti sávot, míg a második esetben 450 Hz - 650 Hz közötti tartományt kell áteresztetni. A spektrogrammal

való munka a Python librosa könyvtárának használatával került megvalósításra, ami jelentősen meggyorsította a munkafolyamatot.

3.2.3. Adatnormalizáció és mérési fajták

Az adatnormalizációra első sorban az IMU adatok esetében volt szükség, ugyanis a korábbi adathalmazom több fajta eszközzel végzett méréssel készült, melyeknek az érzékenysége különböző volt.

3.2.3.1. Gyorsulási és szögelfordulási adatok mérése

A gyorsulási és szögelfordulási adatok rögzítésére a magam készítette adatgyűjtő eszközt használtam. A mintavételi frekvenciát 75 Hz-re állítottam be, a felhasznált szenzor pedig egy TDK gyártmányú MPU6050 volt. A mérés során kapott legnagyobb érték 19841 volt, a legkisebb pedig -6116 volt, a gyorsulás tekintetében és 311 illetve -204 a szögelfordulás esetében, melyeket a korábbiakhoz hasonlóan 0 és 1 közötti tartományba normalizáltam. Korábbi tapasztalataim alapján a normalizáció 5% - 8%-os javulást eredményezhet. Az adatgyűjtő a domináns kéz csukló fölötti részének felső oldalához volt rögzítve, a mért értékeket .csv formátumban kaptam meg.

3.2.3.2. Hang adatok mérése

A hangok mérésére egy Arduino-hoz gyakran használt Keyes KY-038 analóg kimenetű mikrofont használtam. A mintavételi frekvencia 44.1 kHz volt. Az eszköz használata nem igényelt módosítást a korábban elkészített eszközön, viszont az érzékenység állítására szolgáló potenciométer sok kísérletezést igényelt. A rögzített adatokat .mp3 formátumban kaptam meg, majd a feldolgozásra saját Python scriptet készítettem.



3.3. ábra. Keyes KY-038 mikrofon modul

[9]

3.3. Gépi tanulási technikák

A korábbiakhoz hasonlóan VGG16, és SKlearn 2.0 algoritmusokat használtam az IMU adatok feldolgozására, annyi különbséggel hogy a bemeneti adatvektort a gyorsulás értékek mellett szögelfordulási adatokat is felhasználtam.

3.3.1. VGG16

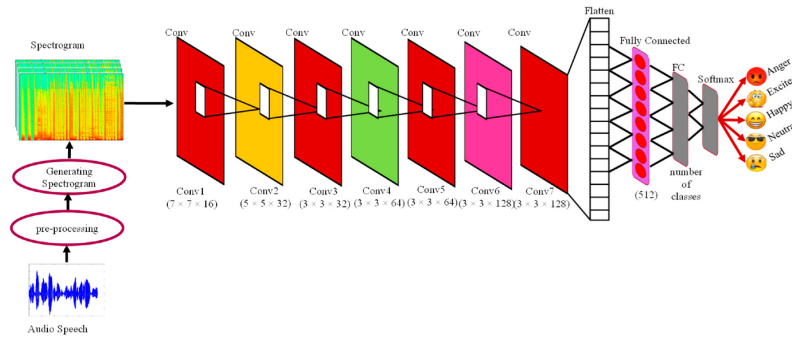
A VGG16 konvolúciós neurális hálót leggyakrabban képek feldolgozására használják [19]. A 16 réteg lévén mély összefüggések megtalálására is képes. A korábbiakhoz hasonlóan most is 20 másodperces időablakot alkalmaztam, az adatokon pedig nem végeztem szűrést. Az előzőekhez képesti változás abban nyilvánul meg, hogy az eddigi bemeneti vektort kibővítettem. A háló az eredeti 2 dimenziós konvolúcióról már korábban 1 dimenziós konvolúcióra lett átállítva. A (100,6)-os bemeneti vektor mérete függ a mintavételi frekvenciától. Jelen esetben az adathalmaz alulmintavételezésen esett át, mely után a mintavételi frekvencia 5 Hz-nek felelt meg. Ennek oka az volt, hogy igen nagy volt a bemenő adat mérete és ez az erőforráskorlátokba ütközött. Ezzel a háló megkapta a szögelfordulási adatokat is. Fontos megjegyezni, hogy a tanítást normalizált és nem normalizált adatverzióval is kipróbáltam, de végülis a korábbi tapasztalatok miatt a nyers adatokat használtam a legtöbb esetben.

3.3.2. Auto-SKlearn 2.0

Az SKlearn 2.0-át használva nem csak egy modellt kaptam, hanem javaslatot annak hiperparamétereire, illetve az azzal elért eredményeket. Az adatok beolvasása esetében kellett módosítanom a korábbi algoritmusomat. A korábbi tanító és tesztalmaid lecseréltem egy olyan dataset-re, amiben benne vannak a normalizált gyorsulási és szögelfordulási adatok is.

3.3.3. DS-CNN

A DS-CNN (Deep Stride Convolutional Neural Network) a hang alapú detekció területén gyakran használt algoritmus [27]. Az én választásom is a szakirodalomkutatás során olvasottak miatt esett erre az algoritmusra. A bemeneti paraméter egy két dimenziós jellemzőtérkép, ami a szűrt mel spektrogram. A vizsgálni kívánt jellemzők alapján 1 másodperces időablakokra bontottam a hangfájlt és azokból készítettem spektrogramokat. A DS-CNN felépítése alapján először egy átlagos konvolúciós rétegen megy keresztül a bemenő adat, miközben különböző szűrőket is átfuttat rajta az algoritmus. A konvolúciós rétegek után következik egy szakaszos normalizációs réteg ReLu aktivációs függvénnyel. Ennek szerepe leginkább az algoritmus gyorsításában illetve a túltanulás elkerülésében nyilvánul meg. A legvégén egy fully-connected softmax réteg van, ami megmondja hogy a hallottak alapján a vizsgálni kívánt esemény történt-e, vagy valamilyen más hanghatás szerepel az adatokban [28]. A DS-CNN architektúrája a következő ábrán látható:



3.4. ábra. Egy DS-CNN felépítése [28].

3.4. Felhasznált felhőszolgáltatás

3.4.1. Docker

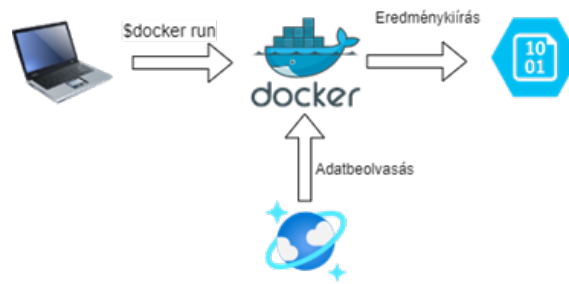
A Docker jelenleg az egyik legelterjedtebb container framework, amelynek implementációja a Linux kernel nyújtotta szeparációs lehetőségekre támaszkodik. Komoly előnye a sebesség. A technológiából fakadóan a Docker containerok kicsik illetve gyorsak, hiszen tulajdonképpen csak egy, a kernelen futó sandbox környezetről van szó, aminek kicsi az erőforrás igénye. A Docker alapvetően három részből áll: a daemon, a kliens, és egy REST API. A daemon a host gépen fut és parancsokat fogad a kientől. Az API segítségével közvetlenül tudunk kommunikálhatunk a daemonnal. A containerok definiálására a dockerfile szolgál. Ez instrukciókat tartalmaz arra vonatkozóan, hogy hogyan épüljön fel a container [14]. Ezekkel a parancsokkal például előírhatjuk az operációs rendszer verzióját, bizonyos Linux packagek installációját, portok megnyitását, és még rengeteg egyéb dolgot, mint esetenben a telepítendő python packagek-et. A Docker konténernek egy Python 3.10-et kellett használni, mert korábbi tapasztalataim szerint ez az a python verzió, amivel minden AutoML komponens rendesen működik. A req.txt tartalmazza azokat a packagek-et, amelyeket telepíteni kell a kód futtatásához.

3.4.2. Microsoft Azure

A Microsoft Azure egy olyan felhőszolgáltatási gyűjtemény, amely rendkívül sok eszközt biztosít a felhasználóknak a fejlesztéseikhez. Számomra a tárhelyek voltak a leginkább hasznosak, amiknek segítségével könnyebben tudtam az AutoML SKlearn 2.0 algoritmus tanítását Docker konténerben elvégezni. Az megvalósított struktúra a következő képen látható.

3.4.2.1. Blob Storage

A Blob Storage a Microsoft Azure olyan funkciója, ami lehetővé teszi a fejlesztők számára, hogy strukturálatlan adatokat tároljanak egy felhőplatformon. Ezek az adatok ezután bárhol elérhetők lesznek. Igen sokféle tartalmat lehet feltölteni: hangot, videót és szöveget. A blobok „tárolókba” vannak csoportosítva, amik felhasználói fiókokhoz vannak csatolva. A blobokat .NET kód segítségével lehet módosítani [15]. A blob-tárolás előnyeit (az objektumtárolás egy típusa)



3.5. ábra. A program logikai rendszerterve.

hasznosító alkalmazások a következők:

- Online médiamegosztás
- Média streamelés
- Elosztott fájl hozzáférés
- Naplófájlok tárolása és frissítése
- IoT (Internet of Things) alkalmazások

A blobok tárolókban vannak elhelyezve – ez a Microsoft Azure Storage-fiók alapvető tárolóegysége. A tárolók meghatározott fájl típusokat tartalmazhatnak – például az egyikben képeket, a másikban videofájlokat tárolhat a felhasználó. A konténerek olyanok mint a könyvtárszerkezeten belüli mappák. Három fő blob-fajta létezik: hozzáfűzés, blokkolás és oldalblob [15].

- Blobok hozzáfűzése Ezeket kifejezetten hozzáfűzési műveletekhez tervezték. A hozzáfűzött blob leggyakrabban a naplófájlok tárolására illetve frissítésére szolgál. Az új blokkokat hozzá tudjuk fűzni egy blob végéhez, ugyanakkor a korábban már elkészített blokkokat nem tudjuk később módosítani vagy törölni.
- Blobok blokkolása A blokkblobok főként médiafájlok, dokumentumok, szövegfájlok illetve bináris adatok tárolására szolgálnak.
- Oldalblobok Az oldalblobok alkalmasak olvasási illetve írási műveletekre. Az oldalblobokat leggyakrabban az operációs rendszerek és a katasztrófa-helyreállítási adatok tárolására használjuk.

Az automl algoritmus egyik végeredményét, ami egy modelleket és azok paramétereit tartalmazó táblázat (Pipeline Profiler) sajnos Blob storage-ben nem sikerült megjeleníteni. Lementés után VS Code-ban sikerült csak megnyitnom.

3.4.2.2. CosmosDB

Az Azure Cosmos DB a Microsoft globálisan elosztott, több modelles adatbázis-szolgáltatása. Séma-agnosztikus és vízszintesen méretezhető. Belsőleg a Cosmos DB az elemeket tárolókba rendezi.

A tárolók adatbázisokba vannak csoportosítva, amelyek a tárolók feletti névterekhez hasonlítanak. A tárolók séma-agnosztikusak, ami annyit tesz, hogy a rendszer nem kényszerít ki sémát az elemek hozzáadásakor [3]. Alapértelmezés szerint az elemek minden mezője automatikusan indexelve van, ez jobb teljesítményt biztosít anélkül, hogy konkrét lekérdezési mintákat használnánk. A tárolók egyedi kulcskényszereket is használhatnak ha az adatok integritása ezt megkívánja. A belső adatmodell a következőkön keresztül érhető el:

- egy szabadalmaztatott SQL API
- öt különböző kompatibilitási API, amik olyan végpontokat tesznek közzé, amik részben kompatibilisek a MongoDB, avGremlin, a Cassandra, az Azure Table Storage protokolljaival. Ezek az API-k lehetővé teszik bármely kompatibilis alkalmazás számára, hogy szabványos illesztőprogramokon vagy SDK-kon keresztül csatlakozzon a Cosmos DB-hez [3].

4. fejezet

Metrikák és eredmények

A következő fejezetben az elért eredményeket és azok értékelési paramétereit mutatom be.

4.1. Értékelési szempontok

A neurális hálók eredményein értékelésére számos paramétert használunk. Semmiképp nem elég egy szempont értékét megvizsgálni, mert lehet hogy a pontosság jónak tűnik, de az AUC alapján kiderül, hogy háló nem tudott rátanulni az adathalmazra, csak tippelt. A leggyakrabban használt mérőszám az osztályozási pontosság.

4.1.1. AUC

Az AUC értéke a ROC görbe alatti terület méretével egyezik meg. Segítségével megkapjuk a teljesítmény összesített mértékét az összes lehetséges besorolási küszöböt figyelembe véve. Leginkább úgy lehet ezt elmagyarázni, hogy az AUC azt mutatja meg, mekkora a valószínűsége, hogy a modell egy random pozitív példát jobban rangsorol mint egy random negatív példát. Az AUC értéke egy 0 és 1 közötti szám. Abban az esetben ha a modell 100%-osan teljesít, az AUC értéke 1.

4.1.2. ROC

A ROC görbe grafikusán jeleníti meg a modell teljesítményét az összes osztályozási küszöb figyelembe vételével. Két paramétert találunk rajta, a TPR-t (True Positive Rate) és FPR-t (False Positive Rate). Amennyiben az osztályozási threshold-ot csökkentjük, a modell több elemet fog pozitívnak minősíteni. Ezzel nőhet az FPR mennyisége.

4.1.3. Konfúziós mátrix

A konfúziós mátrix grafikusan jeleníti meg az osztályozó predikcióinak eredményeit. Leolvashatjuk róla, hogy a modell milyen mennyiségben és hogyan tévedett, ugyanis ábrázolja az igaz pozitív, a hamis pozitív, az igaz negatív és a hamis negatív eshetőségek száma is. Az ideális eset, amikor csak a főátlóban vannak 0-tól eltérő értékek, ami azt jelenti, hogy a modell egyszer sem tévedett.

4.1.4. Precision, recall és accuracy

A precision egy olyan arányszám, ami azt mutatja meg, hogy a pozitívnak mondott érték mennyire biztosan pozitív és mennyire lehetne ugyanez negatív. Kiszámítása az alábbi képlet szerint történik:

$$Precision = \frac{TP}{TP + FN} \quad (4.1)$$

A recall azt adja meg, hogy az osztályozó milyen arányban rontja el a helyes osztály felismerését. Kiszámítása az alábbi módon történik:

$$Recall = \frac{TP}{TP + FP} \quad (4.2)$$

Az accuracy egyszerűen azt adja meg, hogy az osztályozó milyen pontossággal működik. Képlete:

$$Accuracy = \frac{TP + TN}{TP + TN + FP + FN} \quad (4.3)$$

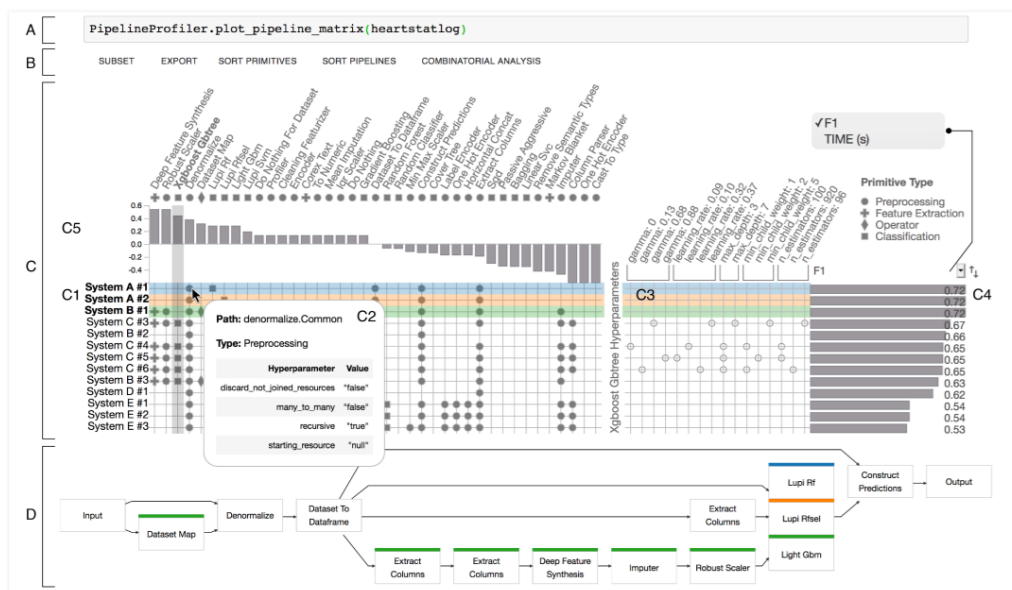
4.1.5. F1 score

Az F1 score egy olyan paraméter, amit akkor érdemes használni, ha vagy negatív vagy pozitív példából nagyon sok van az adathalmazunkban. Ebben az esetben a precision magas értéket szolgáltat, amiből arra a következtetésre juthatunk, hogy a modell jól működik, miközben teljesen rosszul teljesít. Kiszámítása módja az alábbi módon történik:

$$F1score = 2 \cdot \frac{Precision * Recall}{Precision + Recall} \quad (4.4)$$

4.1.6. Pipeline Profiler

Ez egy AutoML SKlearn specifikus elemző eszköz. Valójában egy grafikus elemző eszköz, ami vizuális reprezentációkat és interakciókat tartalmaz. Részét képi a Pipeline Mátrix, ami az AutoML rendszer által elkészített folyamatot összegzi, illetve tartalmazza a hiperparamétereket és primitíveket. A grafikonok összehasonlító nézete strukturális hasonlóságok és különbségek kiemelésére alkalmas. A következő ábrán látható a Pipeline Profiler vizuális eredménymegjelenítő.



4.1. ábra. Pipeline Profiler eredménymegjelenítő [26].

4.2. Eredmények

A különböző modelleket számos tanító/tesztalmaz aránnyal illetve beállítással kipróbáltam. A korábbi munkám eredményeként már rendelkeztem 310.000 rekordot tartalmazó adathalmazzal, ami gyorsulás és elfordulás értékeket tartalmaz, mostani kutatásomhoz pedig 368 percnyi mérést rögzítettem. Ez a gyorsulási értékek szempontjából 1.656.733 rekordot jelent, a hangfeldolgozás esetében pedig 22.089 másodpercnyi mp3 fájlt. Az adatgyűjtésben segítségemre volt a menyasszonyom, aki összesen 12 mérést végzett a 31-ből.

4.2.1. VGG16 eredmények

A neurális háló tanításához a korábban és az újonnan gyűjtött adatokat egyaránt felhasználtam. Ez összesen 1.966.733 rekordot jelent. Az általam beállított időablak 20 másodperc volt.

Szepont \ Felosztás	Önvalidáció	75/25	80/20
Accuracy	96,56	98,01	98,04
Precision	94,13	97,69	97,53
F1-score	89,82	90,12	98,70

4.1. táblázat. VGG16 CNN eredménye.

Az eredményeket vizsgálva azt tapasztaltam, hogy a gyorsulási és szögelfordulási adatokat együttesen felhasználó konvolúciós neurális háló jobb eredményt szolgáltat mint a csak gyorsulási értékeken tanuló verziója 80/20 tanuló/teszt halmaz esetében. Ez a várakozásokkal egybevágó eredmény.

4.2.2. AutoML-SKlearn 2.0 eredmények

Az SKlearn 2.0 esetében is felhasználtam mind a frissen mért 31 étkezés adatait, mind a korábbi dataset értékeit.

Szempont \ Felosztás	Önvalidáció	75/25	80/20
Accuracy	88,50	95,19	94,88
Precision	87,89	94,00	94,14
F1-score	86,34	94,32	93,76

4.2. táblázat. AutoML SKlearn 2.0 eredménye.

Az remélt értékekkel sajnos ellenkező eredményt kaptam a vizsgálati szempontok értelmezése után. A szögelfordulási értékeket is felhasználó SKlearn 2.0 algoritmus minden tanító és tesztal-maz arány esetében rosszabbul teljesített mint a csak gyorsulási értékeket felhasználó verziója. A javasolt algoritmus jelen esetben is a Random Forest volt. Véleményem szerint további algoritmu-sok kipróbálása is szükséges, illetve az idő rövidsége miatt nem volt időm az IIR filteren kívül más szűrők által generált adathalmazokon is futtatni az algoritmust.

4.2.3. DS-CNN eredmények

A hangok feldolgozására használt neurális háló tanításához csak az újonnan rögzített adathalmazt használtam. A hangfelvételeket 1 másodperces ablakokra bontottam, mert számításaim alapján a számomra fontos hangeffektek egy 1 másodperces időablakba beleférnek. Az így kapott kereteken elvégzett megfelelő transzformációk után a háló bemenetére kerültek az adatok.

Szempont \ Felosztás	Önvalidáció	75/25	80/20
Accuracy	90,32	84,23	86,49
Precision	88,82	81,09	86,33
F1-score	89,33	83,76	85,17

4.3. táblázat. DS-CNN eredménye.

A kapott eredmények alapján kijelenthetem, hogy a neurális háló sikeresen rátanult az adat-halmazra és megtalálja a releváns hangeffekteket. A pontosság sajnos nem éri el a 87%-ot, ami miatt a háló rosszabbul teljesít mint a mozgási adatokat feldolgozó algoritmusok. Véleményem szerint további adatok gyűjtésével és az adathalmaz bővítésével illetve a hanghatások hosszához optimalizált ablakmérettel jobb hatásfok érhető el.

4.2.4. Háló kombinációk eredményei

A tanított neurális hálók legjobban teljesítő modelljeit együttesen is teszteltem. Összesen két teszt esetet végeztem el. Az egyikben a két konvolúciós háló együttes működését vizsgáltam, a másikban az AutoML algoritmus és a DS-CNN összegzett hatásfokát derítettem fel. Az együttes vizsgálat azon az elven alapszik, hogy amennyiben mindkét háló étkezést ismer fel ugyan abban

az időintervallumban, akkor biztosan mondhatjuk, hogy szénhidrátbevitel történik. Viszont ha valamelyik háló nem detektál semmit, akkor kizárom annak lehetőségét, hogy a vizsgált alany étkezik. A hálók tesztelése külön – külön történt és később, manuálisan végeztem el az időpontok szinkronizációját.

4.2.4.1. VGG16 + DS-CNN

A háló tanítására egyaránt használtam a régi és az új adatbázist, viszont a teszteléshez kizárólag azt az adathalmazt használtam, ami mozgás és hang adatokat is tartalmaz. Ez biztosította a lehetőséget a hang alapú és a mozgás alapú étkezés detekció együttes működésének vizsgálatára.

Szempont \ Felosztás	80/20
Accuracy	97,38
Precision	97,11
F1-score	93,57

4.4. táblázat. VGG16 és DS-CNN kombinációjának eredménye.

A tesztelés után arra a következtetésre jutottam, hogy a két neurális háló együttesen rosszabbul teljesít mint a VGG16 konvolúciós algoritmus önmagában. Ennek okát abban vélem felfedezni, hogy a hangfelismerés még nem működik elég pontosan és ez lerontja a másik háló végeredményét.

4.2.4.2. SKlearn 2.0 + DS-CNN

A háló tanítására az előző esethez hasonlóan mindkét adathalmazt felhasználtam, viszont tesztelni csak az új dataset-en teszteltem. Egyedül így lehetett vizsgálni azt, hogy a hang és mozgás felismerés is működik egy időben.

Szempont \ Felosztás	80/20
Accuracy	93,19
Precision	91,98
F1-score	91,02

4.5. táblázat. SKlearn 2.0 és DS-CNN kombinációjának eredménye.

Az eredmény a másik hálókombinációhoz hasonló. A két háló együttes működése minden vizsgált paraméter esetén rosszabb eredményt hozott mint az SKlearn 2.0 önmagában. Ennek oka itt is abban rejlik, hogy a pontatlanul működő hangfelismerési algoritmus lerontja az automatizált gépi tanulási folyamat kimenetét.

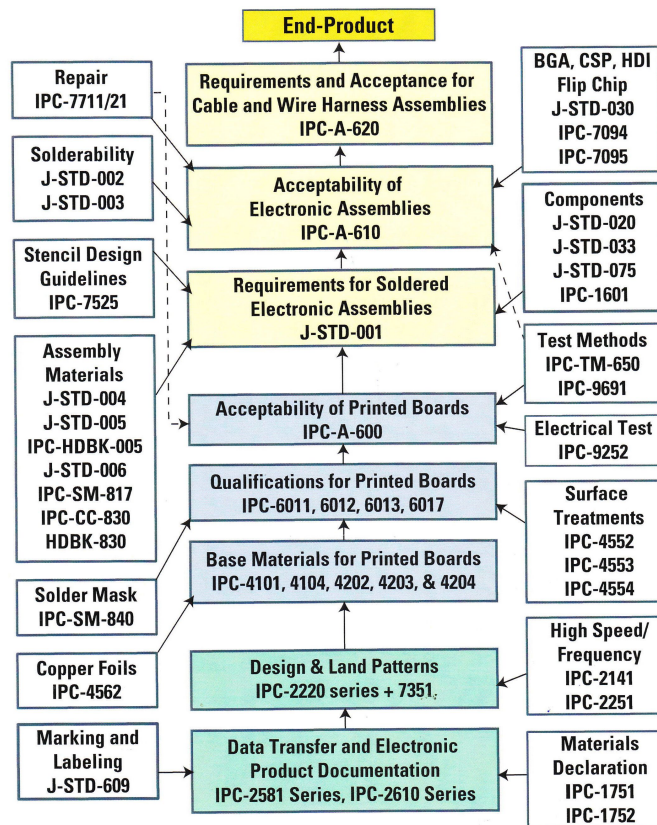
5. fejezet

Fizikai eszköztervezés

Munkám részét képezte egy olyan eszköz megtervezése, ami képes a korábban említett fizikai paraméterek gyűjtésére. Ehhez az eszköznek tartalmaznia kell egy IMU-t és egy mikrofont. A jelenlegi periódusban még nem koncentráltam arra, hogy az készülék képes legyen neurális hálók futtatására, de a tervezés folyamán igyekeztem olyan elv szerint dolgozni, hogy később csupán a mikrokontroller lecserélésével alkalmas legyen MI futtatására is. Amennyiben a karpánt eléri kívánt szerepét és az inzulinpumpával kommunikálva fogja segíteni a precízebb inzulin adagolást, egészségügyi terméknek fog minősülni. Emiatt teljesítenie kell az IPC szabvány kritériumait. Munkám során tanulmányoztam a szabványt és az általam megvalósított tervezési esetekben betartottam az előírt kritériumokat. Továbbá a felhasználni kívánt eszközök kiválasztása során is ezen elvek szerint jártam el. A beszerzett alkatrészeket teszteltem és a dolgozatban dokumentáltam, hogy milyen problémákba ütköztem, illetve milyen megoldási lehetőségeket javaslok azok megszüntetésére.

5.1. Az IPC szabvány

Az IPC (Institute for Printed Circuits) szabványt világszerte alkalmazzák elektronikai eszközgyártók mind a tervezés mind a gyártás folyamatában. Előnye, hogy megkönnyíti a megrendelők és a gyártók közötti kommunikációt annak közös nyelvre hozása által, valamint javítja a termék minőségét és megbízhatóságát. Az IPC az eszközök minőségének garantálás miatt a teljes munkafolyamatra készített szabványokat, melyet az alábbi szabványfa szemléltet [21]. Az IPC mellett van egy másik szabványcsalád is, amelyet az orvostechikai eszközöket gyártó cégeknek és vállalatoknak nagyon szigorúan be kell tartani. Ez az ISO 13485:2016 szabvány, amely a teljes minőségirányítási rendszert átfogóan kezeli. Azok a szervezetek akik ezeket betartják, jogosultak orvostechikai eszközök tervezésére és fejlesztésére, gyártására, tárolására, forgalmazására, telepítésére illetve szervizelésére. Az általam használt IPC-2221A szabvány ennél sokkal kevésbé szigorú elveket határoz meg több szempont esetén [29].



5.1. ábra. IPC szabványfa [21].

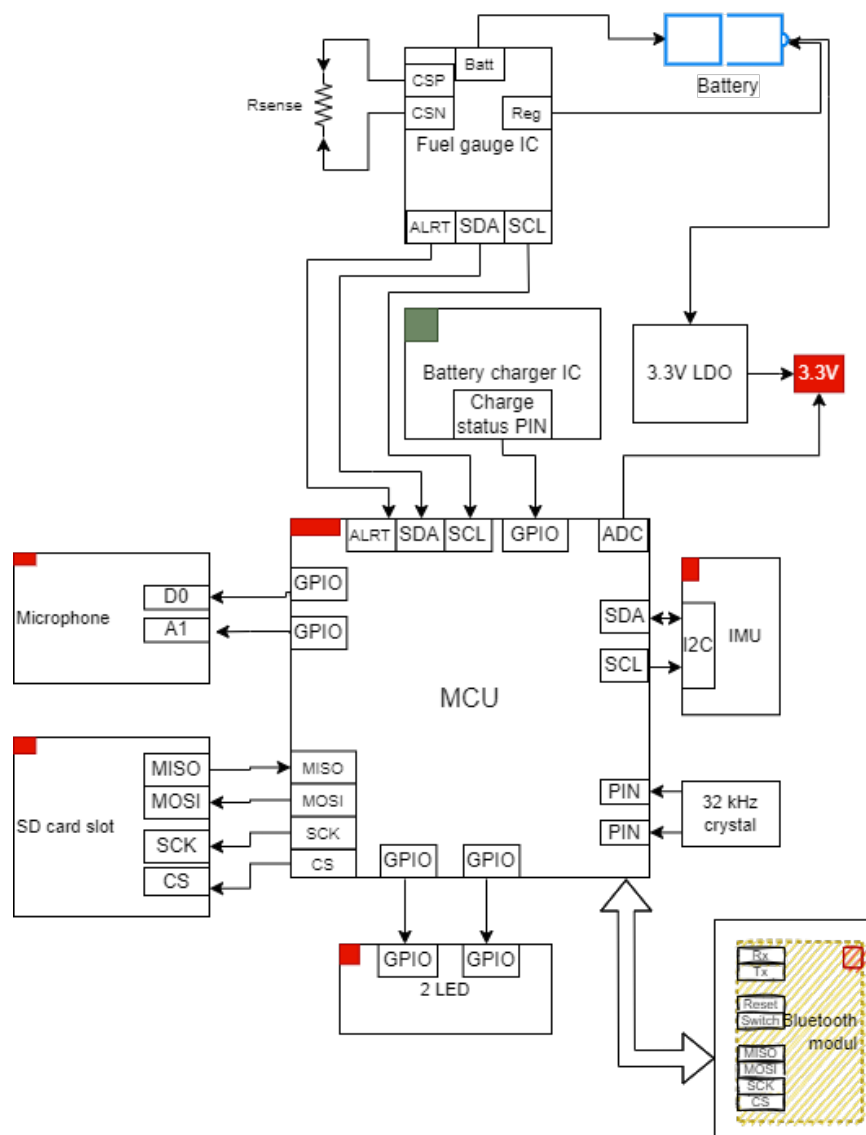
Jelen esetben számomra az IPC-2221A szabvány volt a legfontosabb, ami a nyomtatott áramkörök tervezésének elveit foglalja magában. A legfontosabb szempontok az alábbiak:

- Anyagválasztás
- Anyagvastagság választás
- ESD védelem
- Strukturális felépítés
- Méret és elrendezési elvek
- Merevítések alkalmazása
- Impedancia illesztés
- Vezetősávok tervezése
- Buszok tervezése
- Viák kialakítása
- Hődisszipációs elvek
- Hővezetés
- Kényszerek és toleranciák

- Alkatrész elhelyezések
- Alkatrész választási elvek
- Tesztelhetőség
- Gyárthatóság

5.2. PCB tervezés

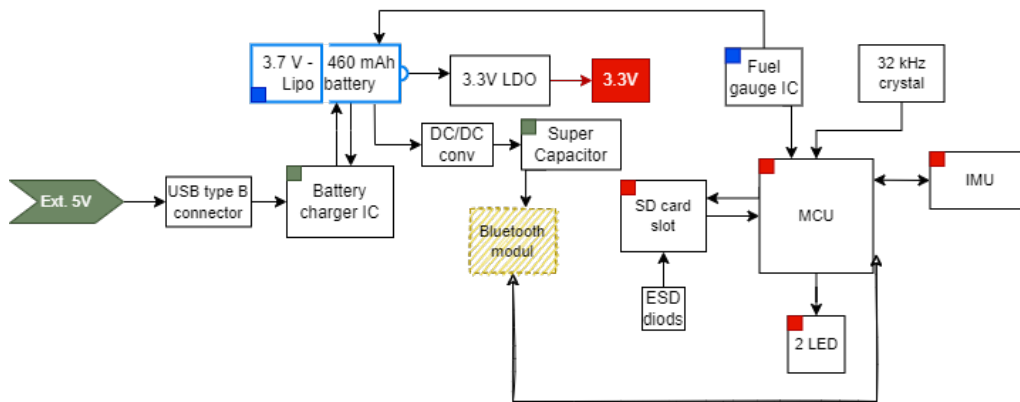
A tervezéshez az Altium Designer 19.1.8 tervezőprogramot használtam, mert ezt tudtam legálisan beszerezni. A készített áramkör logikai rendszertervét szemlélteti az alábbi ábra.



5.2. ábra. Logikai rendszerterv.

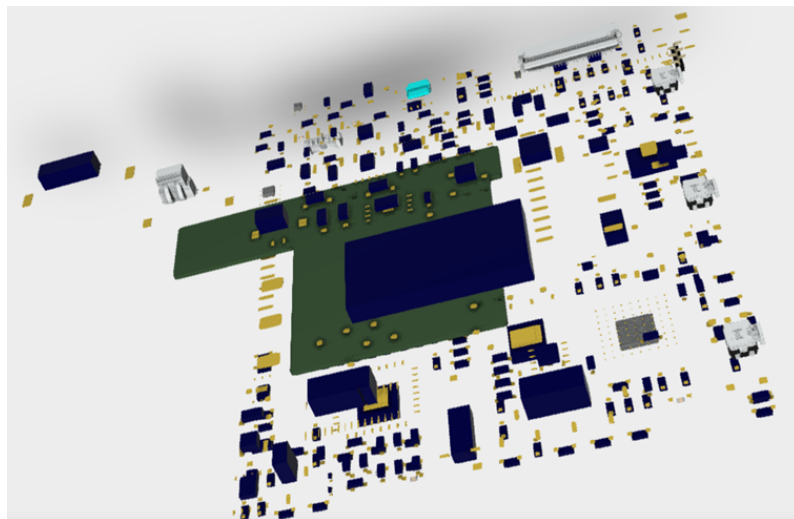
A képen nem alkalmaztam busz jelölést, mert ebben a nézetben könnyebb átlátni a szükséges portok számát és típusait. A tápellátást 3,3 és 5V-os feszültségforrások biztosítják. A tesztpontok

nincsenek jelölve az ábrán, de minden kommunikációs vezetékre illetve a különböző tápfeszültségekhez szükséges tenni egyet. A felhasználás és a tesztelés megkönnyítése miatt 2db LED-et helyeztem el a nyomtatott áramkörön. Az áramkör energiamenedzsmentjének tervét szemlélteti az alábbi ábra.



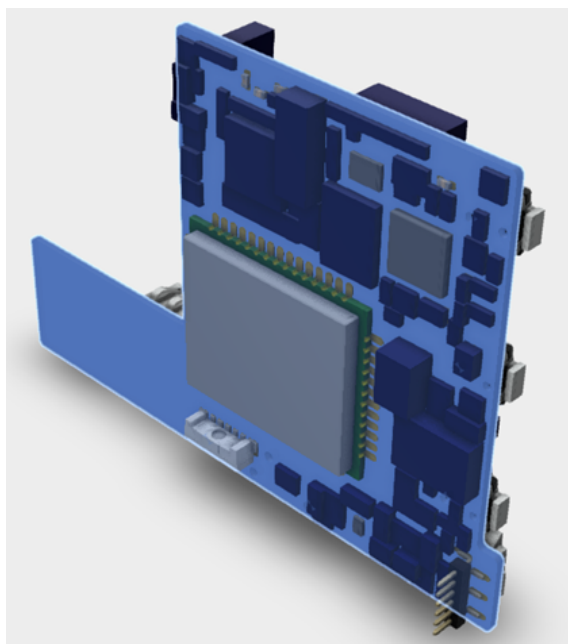
5.3. ábra. Energiamenedzsment tervrajz.

A real-time adatgyűjtés miatt felmerült bennem a Bluetooth kommunikáció alkalmazása. Viszont ennek a technológiának igen magas energiaigénye van a többi alkatrészhez képest, ami miatt kialakult az igény egy szuperkapacitás alkalmazására. Ez az eszköz sokkal több energiát képes tárolni mint egy átlagos kondenzátor, ugyanakkor kisebb helyen. A Cap-XX nevű cég által gyártott szuperkapacitás 5,5V-os feszültségen 1F elektromos kapacitást tud tárolni 21x14 mm-es tokban [46]. A nyomtatott áramkör 3D nézetének robbantott és egyszerű 3D nézetének ábrája látható alább.



5.4. ábra. A PCB robbantott ábrája.

A jelenleg megtervezett nyomtatott áramkör méretei 35mm x 48mm. A PCB alapja FR4-es anyag ami 0,35mm vastagságú. Ebből összesen 3 található. A kész áramkör összesen 4 réteget tartalmaz, melyek közül a középső kettő a tápfeszültséget és a földet foglalja magában, a szélső kettő pedig a jelátviteli réteg. Az IPC által meghatározott minimum vezetősáv szélesség 0,308mm a két vezeték közötti távolság pedig 0,64 mm.



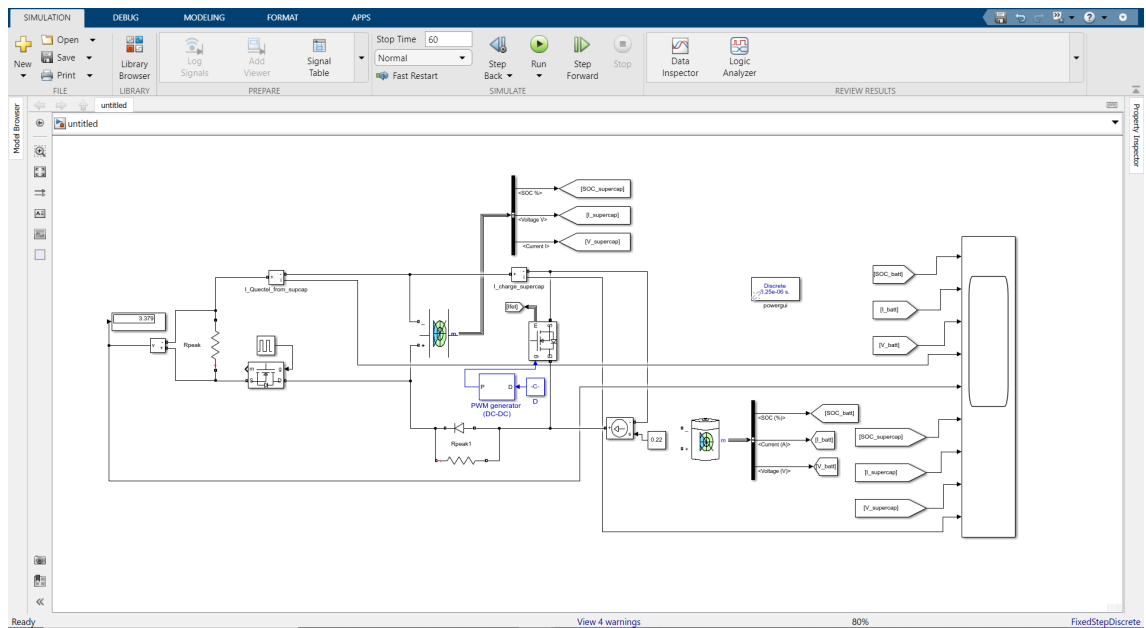
5.5. ábra. A PCB 3D ábrája.

5.3. Alkatrész választás

Az alkatrészek esetében igyekeztem a nagyobb gyártók termékei közül választani, mert tapasztalataim szerint pontosabb gyártástechnológiával dolgoznak. Az IPC szabvány javaslata szerint igyekeztem Class 1 besorolású alkatrészeket választani. Ez azt jelenti, hogy kapacitások esetén a C0G osztályú kondenzátorokat részesítettem előnyben. Ezek szélesebb hőmérsékleti tartományban tudnak működni, nagyobb pontossággal, illetve használatukkal elkerülhető az „énekítő kondenzátor” hatás, ami a nagy frekvenciás jelek esetén a kondenzátor által kibocsátott sípoló hangot jelenti. A mikrofonnal történő mérés miatt ez igen fontos tényező. Az ellenállások tekintetében 0,1% pontosságú alkatrészeket választottam. A korábban használt MPU6050 helyett most a Bosch BHI270 IMU-t választottam, ami jobb pontosságot szolgáltat és az adatlapja szerint alkalmas gesztusdetekciós eszközök fejlesztésére [6].

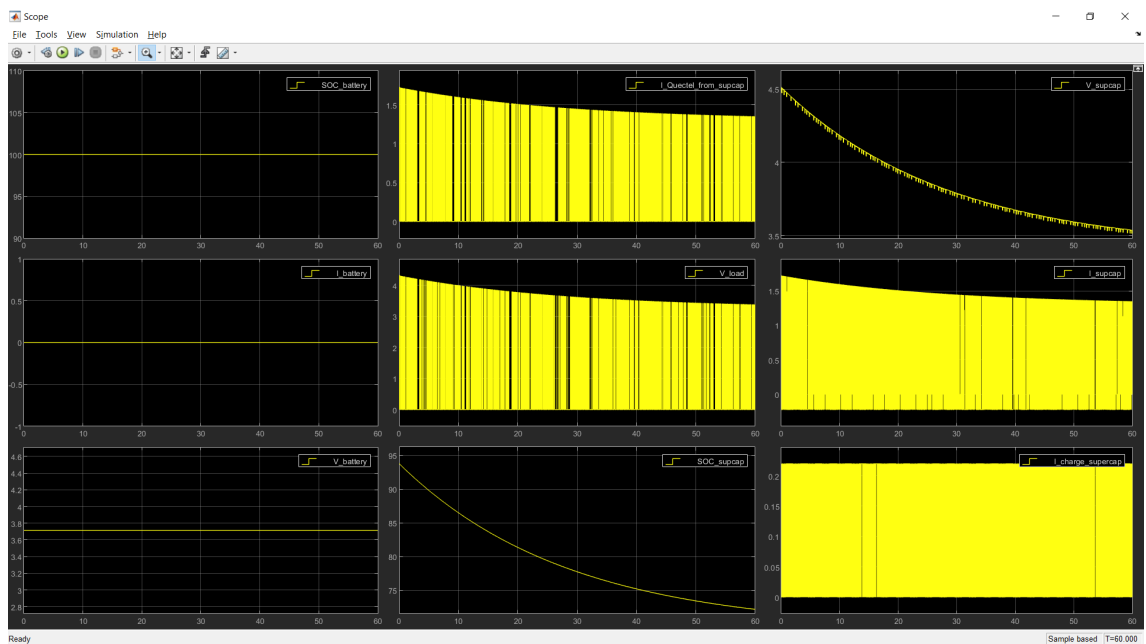
5.4. Tesztelés

Az alkatrészek közül eddig a mikrofont, az IMU-t, egy STM32 mikrovezérlőt, illetve a tápellátás biztosításához szükséges darabokat szereztem be és teszteltem le. Problémával eddig egyedül a szuperkapacitás esetén találkoztam. A legtöbb Bluetooth modul használat közben 350-550 mA-es áramfelvétel csúcsokat produkál körülbelül másodpercenként kétszer. Ez egy egyszerű akkumulátor esetében nem tolerálható áramfelvétel, mert a cella idő előtti tönkremeneteléhez vezet. Emiatt kerülne beépítésre a szuperkapacitás. Viszont egy általam készített szimuláció alapján ez a 0,5 másodperc nem elég arra, hogy a szuperkapacitás feltöltsön és ki tudja szolgálni a Bluetooth modult. Nagyobb akkumulátor esetleg megoldást jelenthet, mert nagyobb áram leadására képes, ugyanakkor a méret is jelentősen megnövekedne. A szimulációt Matlab szoftverben készítettem el.



5.6. ábra. A szimuláció felépítése.

Az alábbi ábrán jól látható, ahogyan a szuperkapacitás feszültsége az idő előrehaladtával egyre csökken, ami miatt 1 perc után már nem tudja ellátni a feladatát.



5.7. ábra. 60 másodperces szimuláció eredménye.

6. fejezet

Jövőbeni tervek és továbbfejlesztés

A projektet véleményem szerint mindenképp érdemes tovább folytatni. A jövőben további hangadatok gyűjtését fogom végezni, hogy a DS-CNN neurális hálóval jobb eredményeket tudjak elérni, illetve más hangok feldolgozására alkalmas algoritmusokat is ki fogok próbálni. Először egy rekurrens AI lesz a tesztelendő algoritmus, mert a szakirodalomkutatás során több esetben olvastam az által szolgáltatott pozitív eredményekről. Reményeim szerint ha a hangfelismerés megbízhatóbban működik, akkor a kombinált neurális hálók is jobb hatásfokot fognak elérni. Az eszköz tervezése szempontjából a PCB jelenleg gyártás alatt van, de a dolgozat leadásának időpontjáig nem érkezett meg, ezért nem szerepel róla kép a dokumentumban. Az energiaellátás problémáját egy nagyobb energiatároló képességű akkumulátorral fogom orvosolni, különben a Bluetooth kommunikáció nem valósul meg. Amint minden eszközt leteszteltem, a NYÁK beültetése lesz a következő lépés illetve annak tesztelése.

7. fejezet

Összefoglalás

Munkám során adatgyűjtést végeztem és a korábbi 310.000 rekordból álló gyorsulás és szögelfordulási adatokat tartalmazó adathalmazt kiegészítettem további 1.656.733 adatsorral, ami 368 pernyi mérésnek felel meg. Tanítást végeztem módosított VGG16 és AutoML SKlearn 2.0 algoritmusokkal, illetve kipróbáltam a DS-CNN MI algoritmust, amivel az evőeszközök és tányérok egymáshoz ütdődésének hangját valamint fémdobozos üdítőitalok kibontásának hangját ismertettem fel. A VGG16 esetén 98%-os pontosságot értem el, az SKlearn 2.0 esetén ez a paraméter 94% volt. A hangfelismerő algoritmusom 86%-os pontossággal működött. Kipróbáltam a különböző mozgás és hangfelismerő algoritmusok kombinációját, aminek eredménye sajnos elmarad az elvártaktól. Ennek oka a hangfelismerő háló pontatlansága volt. Kutatásom folyamán rengeteg szakirodalmat áttekintettem a mesterséges intelligencia alapú gesztusfelismerésről illetve megismerkedtem behatóbban az IPC-2221A szabvánnyal. Ez alapján PCB-t terveztem és küldtem gyártásba. Bosch BHI270 IMU-t illetve Keyes KY-038 mikrofon modult teszteltem, valamint a Bluetooth alkatrész működtetéséhez szükséges energiafogyasztásról Matlab szimulációt készítettem. Emellett az 5.3 ábrán látható energia menedzsment áramkört leteszteltem. Munkámat a jövőben a továbbfejlesztési lehetőségek részen belül ismertetekkel fogom folytatni.

Irodalom

- [1] Fiv Asim. *AndroSensor*. URL: <https://play.google.com/store/apps/details?id=com.fivasim.androsensor&hl%5C%5C=en&gl=US>. (accessed: 17.12.2021).
- [2] AutoML. “BOHB: Robust and Efficient Hyperparameter Optimization at Scale”. (2020). URL: https://www.automl.org/blog_bohb/ (elérés dátuma 2022).
- [3] Microsoft Azure. “Azure Cosmos DB - NoSQL and Relational Database”. (2021). URL: <https://azure.microsoft.com/en-us/products/cosmos-db/> (elérés dátuma 2022).
- [4] Ekaba Bisong. “Google Colaboratory”. (2019). DOI: 10.1007/978-1-4842-4470-8_7. URL: https://www.researchgate.net/publication/336108351_Google_Colaboratory.
- [5] Bernadette M; RandlesIrene V; PasquettoMilena GolshanMilena GolshanChristine L; Borgman. “Using the Jupyter Notebook as a Tool for Open Science: An Empirical Study”. (2017). DOI: 10.1109/JCDL.2017.7991618. URL: https://www.researchgate.net/publication/318751639_Using_the_Jupyter_Notebook_as_a_Tool_for_Open_Science_An_Empirical_Study.
- [6] Bosch. “BMI270 IMU combining accelerometer and gyroscope”. (October. 2021). URL: <https://www.bosch-sensortec.com/media/boschsensortec/downloads/datasheets/bst-bmi270-ds000.pdf>.
- [7] J. Brownlee. “A Gentle Introduction to the Gradient Boosting Algorithm for Machine Learning”. (2016). URL: <https://machinelearningmastery.com/gentle-introduction-gradient-boosting-algorithm-machine-learning> (elérés dátuma 2022).
- [8] Nabeel Siddiqui; Rosa H. M. Chan. “Multimodal hand gesture recognition using single IMU and acoustic measurements at wrist”. (2020). URL: <https://journals.plos.org/plosone/article?id=10.1371/journal.pone.0227039>.
- [9] Electronic Clinic. “KY-038 Sound Sensor with Arduino Circuit Code, Clap project”. (2021). URL: <https://www.electronicclinic.com/ky-038-sound-sensor-with-arduino-circuit-code-clap-project/> (elérés dátuma 2022).
- [10] Francisco Javier Ordóñez ; és Daniel Roggen. “Deep Convolutional and LSTM Recurrent Neural Networks for Multimodal Wearable Activity Recognition”. (2015). DOI: 10.3390/s16010115. URL: <https://www.mdpi.com/1424-8220/16/1/115/htm>.
- [11] és Kardos P. Devosa I. Grósz T. “A Mesterséges Intelligencia (AI) az egészség szolgálatában”. (). URL: http://real.mtak.hu/122585/1/2020_3_CSC_001_Devosa.pdf (elérés dátuma 2022).

- [12] San Francisco Diabetes Teaching Center at the University of California. “Types of Insulin :: Diabetes Education Online”. (). URL: <https://dtc.ucsf.edu/types-of-diabetes/type2/treatment-of-type-2-diabetes/medications-and-therapies/type-2-insulin-rx/types-of-insulin/> (elérés dátuma 2022).
- [13] Dr. Diag. “Gyógyszer és egyéb kémiai anyagok okozta diabetes mellitus”. (). URL: <http://www.drdiag.hu/kereso/diagnosztika.adatlap.php?id=90393> (elérés dátuma 2022).
- [14] Docker Documentation. “Use containers for development”. (2021). URL: <https://docs.docker.com/language/java/develop/> (elérés dátuma 2022).
- [15] Microsoft Azure Documentation. “Introduction to Blob (object) Storage - Azure Storage”. (2021). URL: <https://learn.microsoft.com/en-us/azure/storage/blobs/storage-blobs-introduction> (elérés dátuma 2022).
- [16] Katharina Eggenberger; Matthias Feurer. “Auto-Sklearn 2.0: The Next Generation”. (2020). URL: <https://www.automl.org/auto-sklearn-2-0-the-next-generation/>.
- [17] Matthias Feurer és tsai. “Auto-Sklearn 2.0: Hands-free AutoML via Meta-Learning”. (2020).
- [18] Ilka Gdanietz. “Postprandial Blood Glucose - Why it’s important after eating”. (2018). URL: <https://www.mysugr.com/en/blog/postprandial-glucose-why-your-blood-sugar-is-so-important-after-eating/>.
- [19] GeeksforGeeks. “VGG-16 | CNN model”. (2020). URL: <https://www.geeksforgeeks.org/vgg-16-cnn-model/>.
- [20] Diabetologia Hungarica. “Cukorbetegség”. (). URL: http://www.diabet.hu/upload/diabetes/magazine/02050103.HTM?web_id= (elérés dátuma 2022).
- [21] ASSOCIATION CONNECTING ELECTRONICS INDUSTRIES. “Generic Standard on Printed Board Design”. (2021). URL: <https://www.ipc.org/TOC/IPC-2221A.pdf> (elérés dátuma 2022).
- [22] J. Johnsona. “4 Types of Artificial Intelligence”. (). URL: <https://www.bmc.com/blogs/artificial-intelligence-types/> (elérés dátuma 2022).
- [23] Severin D. Dobrea; és M.-C. Dobrea. “Head Gesture Recognition using a 6DOF Inertial IMU”. (2020). DOI: 10.15837/ijccc.2020.3.3856. URL: <http://univagora.ro/jour/index.php/ijccc/article/view/3856>.
- [24] MaBisz. “Így javíthatja a mesterséges intelligencia az egészségügyi ellátást”. (). URL: <https://mabisz.hu/szemle/?p=40334> (elérés dátuma 2022).
- [25] Mbientlab. “MetamotionS description”. (2021). URL: <https://mbientlab.com/metamotions/>.
- [26] Jorge Piazzentin Ono és mtsai. “PipelineProfiler: A Visual Analytics Tool for the Exploration of AutoML Pipelines”. (2020). URL: <https://deepai.org/publication/pipelineprofiler-a-visual-analytics-tool-for-the-exploration-of-automl-pipelines>.
- [27] és T. May P. M. Sørensen B. Epp. “A depthwise separable convolutional neural network for keyword spotting on an embedded system”. (2021). URL: <https://asmp-urasipjournals.springeropen.com/articles/10.1186/s13636-020-00176-2> (elérés dátuma 2022).
- [28] és T. May P. M. Sørensen B. Epp. “Combining Local Appearance and Holistic View: Dual-Source Deep Neural Networks for Human Pose Estimation”. (2021). (Elérés dátuma 2022).

- [29] R. Péter. “Az IPC szabványrendszere”. (2021). URL: <https://www.elektro-net.hu/gyartoso/5062-az-ipc-szabvanyrendszere> (elérés dátuma 2022).
- [30] Arnold Pető. *Fan noise reduction*. Budapest: Budapest University of Technology et al., 2012.
- [31] Scientific Reports. “Artificial Pancreas, Injection Sites, and Fast-Acting Insulin: Diabetes Trends and Challenges”. (). URL: <https://www.mddionline.com/components/artificial-pancreas-injection-sites-and-fast-acting-insulin-diabetes-trends-and> (elérés dátuma 2022).
- [32] Scientific Reports. “Global, regional, and national burden and trend of diabetes in 195 countries and territories: an analysis from 1990 to 2025”. (). URL: <https://www.nature.com/articles/s41598-020-71908-9/figures/1> (elérés dátuma 2022).
- [33] L. Roberts. “Understanding the Mel Spectrogram”, Analytics Vidhya”. (2021). URL: <https://medium.com/analytics-vidhya/understanding-the-mel-spectrogram-fca2afa2ce53> (elérés dátuma 2022).
- [34] Strasser Gergő Szántó Marcell Szász László. “A diabétesz kezelésének optimalizálása gesztusdetekciós alapon”. (). (Elérés dátuma 2022).
- [35] Georgi Marcus; Christoph Amma; és Tanja Schultz. “Recognizing Hand and Finger Gestures with IMU based Motion and EMG based Muscle Activity Sensing”. (2015). DOI: 10.5220/0005276900990108. URL: <https://www.scitepress.org/papers/2015/52769/52769.pdf>.
- [36] Magyar Diabetes Társaság. “A diabéteszesek számának alakulása magyarországon”. (). URL: http://www.diabet.hu/info.aspx?web_id=&sp=179 (elérés dátuma 2022).
- [37] Magyar Egészségügyi Menedzsment Társaság. “Biztonságos-e a mesterséges intelligencia alkalmazása az egészségügyben?”. (). URL: <https://memt.hu/biztonsagos-e-a-mesterseges-intelligencia-alkalmazasa-az-egeszsegugyben/> (elérés dátuma 2022).
- [38] TDK. “MPU-6050 | TDK”. (2021). URL: <https://invensense.tdk.com/products/motion-tracking/6-axis/mpu-6050/>.
- [39] Heather Thompson. “Artificial Pancreas, Injection Sites, and Fast-Acting Insulin: Diabetes Trends and Challenges”. (febr. 28, 2012). URL: <https://www.mddionline.com/components/artificial-pancreas-injection-sites-and-fast-acting-insulin-diabetes-trends-and%20Challenges>.
- [40] WHO. “Diabetes”. (). URL: <https://www.who.int/news-room/fact-sheets/detail/diabetes> (elérés dátuma 2021).
- [41] Wikipedia. “Mel scale”. (2021). URL: https://en.wikipedia.org/w/index.php?title=Mel_scale&oldid=1117532023 (elérés dátuma 2022).
- [42] T. Yiu. “Understanding Random Forest”. (2021). URL: <https://towardsdatascience.com/understanding-random-forest-58381e0602d2> (elérés dátuma 2022).
- [43] L. M. Zhang. “Pairwise Neural Networks (PairNets) with Low Memory for Fast On-Device Applications”. (2020). DOI: 10.48550/arXiv.2002.04458. URL: <https://www.mdpi.com/1424-8220/16/1/115/htm>.
- [44] IoT Zóna. “Mesterséges intelligencia az egészségügyben”. (). URL: <https://iotzona.hu/egeszseg/mesterseges-intelligencia-az-egeszsegugyben> (elérés dátuma 2022).

Ábrák jegyzéke

1.1. A diabéteszesek számának változása az idő függvényében [32].	6
1.2. Az inzulin és vércukorszint változása a nap folyamán [18].	7
1.3. A CGMS és az inzulinpumpa elhelyezkedése [39].	7
1.4. Az AI típusok felosztása [22].	8
1.5. Az AI medicinában betöltött szerepköre [37].	10
2.1. IMU és mikrofontömb segítségével megvalósított adatgyűjtés [8].	12
2.2. MetaMotionS karpánt nyomtatott áramköre [25].	13
2.3. Az AutoML algoritmus felépítése [16].	15
2.4. BOHB optimalizálás versus Bayes-optimalizálás sebessége [2].	15
2.5. A Jupyter Notebook architektúrája [5].	16
2.6. A Jupyter Notebook architektúrája [4].	17
3.1. A mel skála és a frekvencia skála közti függvénykapcsolat [33].	20
3.2. Leves kanalizásának mel spektrogrammja.	20
3.3. Keyes KY-038 mikrofon modul	21
3.4. Egy DS-CNN felépítése [28].	23
3.5. A program logikai rendszerterve.	24
4.1. Pipeline Profiler eredménymegjelenítő [26].	28
5.1. IPC szabványfa [21].	32
5.2. Logikai rendszerterv.	33
5.3. Energiamedzsmment tervrajz.	34

5.4. A PCB robbantott ábrája.	34
5.5. A PCB 3D ábrája.	35
5.6. A szimuláció felépítése.	36
5.7. 60 másodperces szimuláció eredménye.	36

Táblázatok jegyzéke

2.1. A VGG16 háló legjobb eredménye	14
2.2. A saját készítésű háló legjobb eredménye	14
2.3. Az SKlearn 1.0 eredménye	15
2.4. Az SKlearn 2.0 eredménye	16
4.1. VGG16 CNN eredménye.	28
4.2. AutoML SKlearn 2.0 eredménye.	29
4.3. DS-CNN eredménye.	29
4.4. VGG16 és DS-CNN kombinációjának eredménye.	30
4.5. SKlearn 2.0 és DS-CNN kombinációjának eredménye.	30