



M Ű E G Y E T E M 1 7 8 2

Budapesti Műszaki és Gazdaságtudományi Egyetem
Villamosmérnöki és Informatikai Kar
Méréstechnika és Információs Rendszerek Tanszék

Gerinc szegmentálása 3D konvolúciós neurális hálózat felhasználásával

TDK dolgozat

Készítette:

Bálint Marcell

Konzulens:

dr. Hullám Gábor István
Révy Gábor

2023

Tartalomjegyzék

Kivonat	i
Abstract	ii
1. Bevezetés	1
1.1. A kutatás célja és jelentősége	1
1.2. A dolgozat felépítése	2
2. Feldolgozott szakirodalom	3
2.1. Komputertomográfia	3
2.1.1. 3D szeletorientációk	4
2.1.2. Képszegmentáció számítógépek nélkül	5
2.2. Képfeldolgozás és szegmentáció	6
2.2.1. Szakértői módszerek	6
2.2.1.1. Régió alapú módszerek	6
2.2.1.2. Él alapú módszerek	7
2.2.1.3. Modell alapú megközelítések	7
2.2.2. Neurális hálózatok alkalmazása	8
2.2.2.1. U-Net architektúra	8
2.2.2.2. V-Net architektúra	8
2.2.2.3. Hibrid modellek	9
2.3. Létező implementációk vizsgálata	9
2.3.1. Webes gerincszegmentáció U-Net alapon	9
2.3.2. MRI szegmentáció a SAU-Net felhasználásával	9
2.3.3. Gerincszegmentáció az nnU-Net modell felhasználásával	10
3. Implementáció elméleti háttere	11
3.1. Konvolúciós neurális hálózatok (CNN)	11
3.1.1. Konvolúciós rétegek	12
3.1.2. Pooling rétegek	12
3.1.3. Teljesen összekapcsolt rétegek	13
3.1.4. Konvolúciós neurális hálózatok napjainkban	13
3.2. A 2D U-Net modell	14
3.2.1. Architektúra és rétegek	14
3.2.2. Rétegsorrend és forma	15
3.2.3. Matematikai műveletek	15
3.2.3.1. Konvolúciós rétegek	15
3.2.3.2. ReLU aktivációs függvény	16
3.2.3.3. Max-pooling rétegek	16
3.2.3.4. Összehasonlítás egy általános CNN architektúrával	16
3.2.4. Összefoglalás	17

3.3.	A 3D U-Net modell	17
3.3.1.	Architektúra és rétegek	17
3.3.2.	Rétegsorrend és forma	17
3.3.3.	Matematikai műveletek	18
3.3.4.	Teljesítmény és komplexitás	18
3.4.	Hibrid modellek előnyei és kihívásai	18
4.	Adathalmazok bemutatása és elemzése	20
4.1.	Tanító és validáló adathalmazok	20
4.1.1.	CT adatok tárolása NIFTI formátumban	21
4.1.2.	TotalSegmentator adathalmaz	21
4.1.3.	Az adathalmaz felhasználásának előnyei és hátrányai	21
5.	Implementáció	23
5.1.	Adatok előkészítése	23
5.1.1.	Kétdimenziós adathalmaz előkészítése	23
5.1.1.1.	Nifti formátumú felvételek beolvasása	24
5.1.1.2.	Szeletek újraméretezése	25
5.1.1.3.	Padding megadása a szeletekhez	26
5.1.1.4.	Szeletek ablakozása	26
5.1.1.5.	Szeletek mentése numpy formátumban	27
5.1.2.	Háromdimenziós CT felvételek előkészítése	27
5.1.2.1.	Előfeldolgozás a kétdimenziós modell segítségével	28
5.1.2.2.	Háromdimenziós felvételek feldarabolása a teljesítmény növeléséhez	28
5.2.	Neurális hálózatok implementációja	29
5.2.1.	Az ELEKTRONN keretrendszerből átvett U-Net modell	29
5.2.2.	2D U-Net implementációja	30
5.2.2.1.	2D U-Net tanítása	31
5.2.2.2.	A tanítás hardverigénye és futási ideje	33
5.2.3.	3D U-Net implementációja	33
5.2.3.1.	3D U-Net tanítása	34
5.2.3.2.	A tanítás hardverigénye és futási ideje	35
5.3.	Hibrid pipeline felépítése	35
5.3.1.	Bemeneti képek feldolgozása	35
5.3.2.	Bemeneti képek feldarabolása 2D szeletekre	35
5.3.3.	Szeletek szegmentálása a kétdimenziós modell által	36
5.3.4.	A szegmentált szeletek összefűzése háromdimenziós felvételekké	36
5.3.5.	A gerinc kivágása az eredeti CT felvételtől	36
5.3.6.	A felvétel szegmentálása a háromdimenziós modell segítségével	36
5.3.7.	A szegmentált felvétel mentése	36
6.	Teljesítményértékelés	37
6.1.	A dolgozatban tárgyalt modellek eredményei	37
6.2.	Metrikák és eredmények	37
6.3.	Szegmentációs maszkok összehasonlítása	38
7.	Továbbfejlesztési lehetőségek	39
7.1.	Univerzális neurális hálózatok	39
7.2.	Csigolya-specifikus szegmentálás	39
8.	Összegzés	40

Köszönetnyilvánítás	41
Irodalomjegyzék	42

Kivonat

A gerincproblémák diagnosztizálásának egyik leggyakrabban használt módszere a gerinc CT felvételek elemzése. A diagnózis felállítása során az orvosok több kulcsfontosságú tényezőt és jellemzőt vizsgálnak meg, hogy átfogó képet kapjanak a páciens állapotáról. Ilyen például a csontszerkezet integritásának vizsgálata. Ez a folyamat egyszerű diagnózisoknál is sok időt vehet igénybe, komplex esetekben (többszörös sérülés, korábbi műtétek ellenőrzése) pedig jelentősen megnőhet az elemzésével töltött idő. Trauma esetén gyakran sürgős beavatkozás szükséges, így a folyamat felgyorsítása életmentő lehet. A csigolyák mérete és alakja a gerinc mentén folyamatosan változik. A felvételek minősége és a páciensek egyedi genetikai jegyei miatt szakértői megoldások használata nehéz, jellemzően csak a végeredmény minőségének romlása mellett lehetséges. Emiatt a probléma megoldására egy gépi tanulós módszert javaslok, amihez megfelelően annotált adathalmazok szükségesek. Több kétdimenziós szegmentáló módszer is elérhető elfogadható teljesítménnyel, ezek viszont nem tudják kihasználni a háromdimenziós CT felvételek térbeli információit. Az elérhető háromdimenziós modellek számítási igénye hatalmas, így az ilyen neurális hálózatok tanítása és használata magas költségekhez és hosszú futásidőhöz vezet.

Kutatásom során a gerinc szegmentálására egy hibrid U-Net modellt vizsgálok meg, ami kétdimenziós előfeldolgozás segítségével redukálja az információt a háromdimenziós modell gyorsabb és pontosabb tanítása érdekében. Az implementálni kívánt eljárást a kutatási célokra elérhető VerSe 2019 adathalmazon értékelem ki, majd hasonlítom össze egy "state of the art" konvolúciós neurális hálózat eredményeivel.

Abstract

One of the predominant approaches for diagnosing spinal problems is the analysis of spinal CT scans. When making a diagnosis, doctors examine several key factors and features to get a comprehensive picture of the patient's condition. For example, the integrity of the bone structure. This process can be time-consuming even for simple diagnoses, and in complex cases (multiple injuries, checking previous surgeries) the time spent on analysis can increase significantly. In the case of trauma, the patient may also need urgent intervention, so speeding up the process can be life-saving. The size and shape of the vertebrae along the spine are constantly changing. Due to the quality of the images and the unique genetic traits of the patients, expert solutions are difficult to use, typically only with a deterioration in the quality of the final result. For this reason, I propose a machine learning approach to solve this problem, which requires appropriately annotated datasets. Several two-dimensional segmentation methods are available with acceptable performance, but they cannot exploit the spatial information of three-dimensional CT images. The computational demand of the available three-dimensional models is huge, leading to high costs and long runtimes for training and using such neural networks.

In my thesis, I explore a specialized U-Net model designed for spine segmentation. This model employs a two-step process: first, it simplifies the data using two-dimensional pre-processing, and then it uses this simplified data to more quickly and accurately train a three-dimensional model. I evaluate the method on the VerSe 2019 dataset available for research purposes and compare it to the results of a state-of-the-art convolutional neural network used for biomedical image segmentation.

1. fejezet

Bevezetés

A komputertomográfia (CT) az orvostudomány egyik legfontosabb képalkotó diagnosztikai eszköze, amely forradalmasította a betegellátást és a klinikai kutatásokat egyaránt. A CT képes részletes, háromdimenziós képeket készíteni az emberi test belső struktúráiról, így lehetővé téve az orvosok számára, hogy pontos diagnózist állítsanak fel és hatékony terápiás stratégiákat dolgozzanak ki. A CT használata különösen előnyös olyan komplex anatómiai területeken, mint például a gerinc, ahol a pontos diagnózis kritikus jelentőségű. A CT felvételeknek köszönhetően az orvosok képesek azonosítani a csigolyák közötti eltéréseket, a gerincvelő sérüléseit, daganatokat, gyulladásokat és egyéb patológiás állapotokat. A felvételek segítségével nem csak a diagnózis, de a sebészeti tervezés és a kezelési eredmények nyomon követése is sokkal pontosabbá válik. A feladat komplexitása miatt magas szakértelmet igénylő folyamat, amely gyakran órákig vagy akár napokig is eltarthat, attól függően, hogy az adott esetben milyen részletességgel szükséges az adatok feldolgozása. Mindezek mellett felvételek manuális elemzése nem mentes az emberi hibáktól sem, amelyek a diagnózis pontosságát is befolyásolhatják.

A CT technológia folyamatos fejlődése és a nagyobb felbontású felvételek megjelenése biztosítja a technológia megbízható működését, így új lehetőségeket is nyit a kutatási területen. A kutatók egyre inkább felismerik a háromdimenziós képelemzés jelentőségét, amely mélyebb betekintést nyújt a test pillanatnyi anatómiai állapotába. Az automatizált rendszerek több szektorban bizonyították, hogy jól alkalmazhatók a terhelés csökkentésére, így az egészségügy szempontjából is megmutatkozik az automatizált, intelligens rendszerek iránti igény.

1.1. A kutatás célja és jelentősége

Jelen kutatás célja egy olyan automatizált rendszer kifejlesztése, amely képes a gerinc háromdimenziós szegmentálására konvolúciós neurális hálózatok (CNN) segítségével. A kutatás jelentősége több aspektusban is megmutatkozik. Először is, a 3D modellek használata lehetővé teszi az orvosok számára, hogy sokkal részletesebb képet kapjanak a gerinc anatómiai és patológiai állapotáról. A 3D modellek nem csak a csigolyák közötti térbeli viszonyokat mutatják be, hanem a gerincet körülvevő szövetekkel és struktúrákkal való kapcsolatot is jobban megértik, ami kritikus lehet például daganatos megbetegedések, illetve gerincsérülések diagnosztizálásánál.

Másodszor, a gerinc háromdimenziós szegmentációja és megjelenítése jelentős időmegtakarítást eredményezhet. A manuális képelemzés órákig, vagy akár napokig is eltarthat és magas szintű szakértelmet igényel. Ezt a folyamatot felgyorsíthatjuk, ha leredukáljuk azt a területet, amit az orvosoknak elemeznie kell. Adott esetben a gerinc modelljének gyors vizsgálata után a szakemberek foglalkozhatnak a problémásnak ítélt területek alapos vizs-

gálatával, amely jelentősen csökkentheti az orvosok és radiológusok munkaterhét, és lehetővé teszi számukra, hogy több időt fordítsanak a betegellátásra és a terápiás stratégiák kidolgozására. A módszerrel készített szegmentáció más modellek bemeneti állapotaként is megállhatja a helyét tovább növelve a területen végzett folyamatok automatizálását.

Végül, de nem utolsósorban, a kutatás hozzájárulhat az orvosi képzés és az orvostudományi kutatások területén alkalmazott mesterséges intelligencia fejlődéséhez. A 3D szegmentációs technikák további fejlesztése és optimalizálása nem csak a gerinc, de más komplex anatómiai struktúrák vizsgálatát is forradalmasíthatja, így széles körben hozzájárulva az orvostudományi diagnosztika és terápia hatékonyságának növeléséhez [12]

Összegezve a jelen kutatás célja egy olyan, pontos és hatékony rendszer kifejlesztése, amely képes a gerinc automatikus szegmentálására.

1.2. A dolgozat felépítése

Jelen dolgozat 7 fejezeten át taglalja a kutatás eredményeit, a teljesség igénye nélkül röviden itt olvasható ezen fejezetek tartalma:

1. **Bevezetés:** Ebben a fejezetben bemutatom a kutatás hátterét, célját és jelentőségét. Kitérek a CT képzés fontosságára, különösen a gerinc diagnosztikájában, és bemutatom az automatizált képelemzés szükségességét.
2. **Feldolgozott szakirodalom:** Ebben a részben áttekintem azokat a kulcsfontosságú szakirodalmi forrásokat és kutatásokat, amelyek relevánsak a témához. Bemutatom a képfeldolgozás és szegmentáció alapjait, a neurális hálózatok és konvolúciós neurális hálózatok (CNN) alkalmazását, valamint a U-Net és 3D U-Net architektúrákat.
3. **Implementáció elméleti hátttere:** Ebben a fejezetben mélyebben bemutatom a kutatásban használt modell architektúrákat, az adatok előfeldolgozását és az adathalmazok előfeldolgozását, valamint a tanítási és validációs adathalmazokat. Kitérek a hibrid 2D-3D U-Net architektúra elméleti alapjaira is.
4. **Az adathalmaz bemutatása és elemzése:** A fejezetben bemutatom a TotalSegmentator adathalmazt, kitérek a felhasználás előnyeire és arra is, hogy milyen hátrányokkal jár a kiértékelés szempontjából az adathalmaz felhasználásának.
5. **Implementáció:** Itt részletezem a modell implementációjának lépéseit, az adatok előkészítésétől kezdve a tanítási folyamaton át a teljesítményértékelésig. Az eredményeket összehasonlítom a TotalSegmentator modellel, hogy bemutassam a hibrid modell előnyeit és korlátait.
6. **Teljesítményértékelés:** Bemutatom a kutatás során vizsgált modellek teljesítményét és összehasonlítom a legjobban teljesítő háromdimenziós modellel a TotalSegmentator modellel.
7. **Következtetések és jövőbeli munka:** Ebben a záró fejezetben összegzem a kutatás fő eredményeit, és kitérek a jövőbeli kutatási irányokra. Bemutatom, hogy a kutatás milyen mértékben járult hozzá az orvosi képzés és diagnosztika területéhez, és milyen további fejlesztések lehetségesek.

2. fejezet

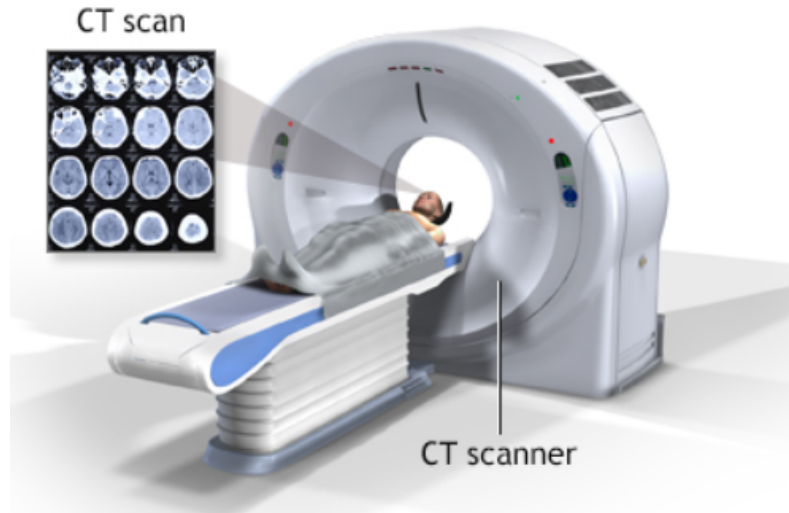
Feldolgozott szakirodalom

A fejezet célja, hogy átfogó képet adjon azokról a kutatási területekről és módszerekről, amelyek közvetlenül kapcsolódnak a gerinc CT képek szegmentálásához. A témában elérhető szakirodalom széleskörű és multidiszciplináris, magába foglalja az orvostudományt, a képfeldolgozást, a gépi tanulást és a neurális hálózatokat. A szakértői módszerek, például a manuális szegmentáció és a hagyományos képfeldolgozási technikák hosszú ideje adnak általános megoldást a probléma megoldására, de az utóbbi években egyre inkább teret hódítanak a gépi tanulási algoritmusok, különösen a konvolúciós neurális hálózatok (CNN). A CT képek szegmentálása egy aktívan kutatott terület, amelynek jelentősége az orvosi diagnosztika és terápia szempontjából vitathatatlan. A manuális szegmentáció egy magas szakértelmet igénylő feladat, amely rendkívül időigényes és jelentős az emberi hiba lehetősége. Ezért egyre nagyobb figyelmet kapnak az automatizált módszerek, amelyek gyorsak és költséghatékonyak. A mesterséges intelligencia alapú megoldások, különösen a deep learning alapú algoritmusok az utóbbi években jelentős fejlődést mutattak a kutatási területen, és egyre több klinikai alkalmazásban válnak elérhetővé. [12] [22]

Ebben a fejezetben részletezem a képfeldolgozás és szegmentáció alapvető koncepcióit, a szakértői módszerek előnyeit és korlátait, valamint a gépi tanulási algoritmusok, különösen a konvolúciós neurális hálózatok alkalmazását és hatékonyságát a gerinc CT felvételek szegmentálásában. Mindenekelőtt viszont had mutassam be a CT felvételek készítésének tudományát és azt is, hogy milyen módon tárolhatjuk és dolgozhatjuk fel ezeket az adatokat.

2.1. Komputertomográfia

A Komputertomográfia, ismertebb nevén CT egy rendkívül fontos és hatékony képalkotó technika az orvosi diagnosztikában és terápiás tervezésben egyaránt. Fizikai alapelve szorosan kapcsolódik a hagyományos 2D röntgenképalkotáshoz. A különbség az, hogy a CT felvételek transzaxiális szeletekből állnak össze, azaz keresztmetszetek sorozatát fűzzük egymás után egy háromdimenziós felvétel előállításához.



2.1. ábra. Az ábrán egy CT gép látható. [27]

A hagyományos CT eszközök a detektor gyűrű egy teljes forgásával egy-egy kétdimenziós képszeletet hoznak létre. Ezt követően az asztalt a szeletek közötti távolságnak megfelelően továbbmozgatják. Ezen módszer hátránya, hogy a képalkotás viszonylag lassú; egyetlen szelet elkészítése akár 3 másodpercig is eltarthat, ami lágy szövetek elmozdulásához vezethet a légzési mozgások miatt. Az 1989-ben bemutatott spirál CT eszközök ezt a problémát orvosolják azáltal, hogy a detektor forgatása és az asztal elmozdulása párhuzamosan történik, ami gyorsabb képalkotást tesz lehetővé.

Az intenzitásértékek normalizálása révén a CT kiválóan alkalmas különböző szövetsűrűségek, így a csontok és lágy szövetek megkülönböztetésére. A Hounsfield-értékek (HU) segítségével pedig pontosan jellemezhető az anyagok röntgenabszorpciós képessége. A Hounsfield-érték kiszámításához az alábbi képletet használhatjuk:

$$HU = \frac{\mu_X - \mu_v}{\mu_v - \mu_l} \times 1000 \quad (2.1)$$

ahol μ_X az adott anyag elnyelődési tényezője, míg μ_v és μ_l a víz és a levegő elnyelődési tényezőit jelölik.

A CT képalkotás térbeli felbontása kiváló, alkalmazása az 1970-es évek óta vált széles körben elfogadottá. A képek általában 512×512 képpont méretűek, ami elegendő részletességet biztosít az orvosi alkalmazásokhoz. A technika azonban nem mentes a röntgensugárzás okozta potenciális kockázatoktól, ezért a használata során az alacsony dózis és a szükségesség elvei érvényesülnek.

Összességében a CT egy olyan eljárás, amely lehetőséget ad a belső anatómiai struktúrák pontos és részletes vizualizációjára, ezáltal jelentősen hozzájárul a diagnosztikai pontosság növeléséhez és a számítógéppel segített műtéttervezés fejlődéséhez.

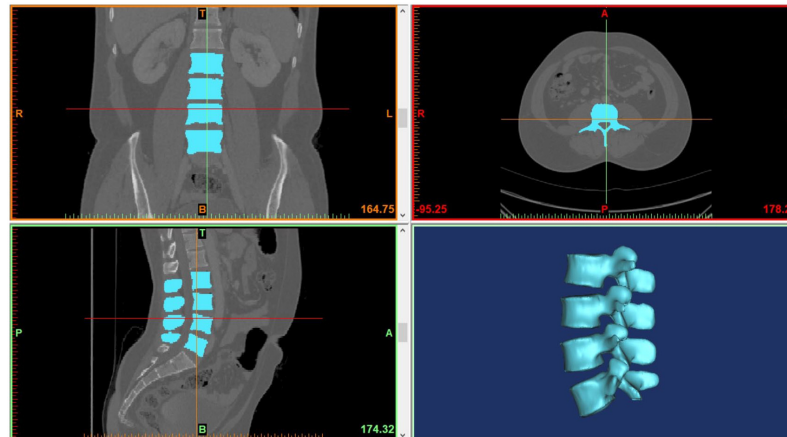
2.1.1. 3D szeletorientációk

A háromdimenziós képalkotás során a szeletorientáció meghatározása kulcsfontosságú. Ebben segítségünkre van az úgynevezett **RAS (Right-Anterior-Superior)** koordináta-rendszer, amely egy orvosi diagnosztikai vizsgálatokban széles körben alkalmazott standard. Az RAS koordináta-rendszerben az 'R' az 'Right' (jobb) oldalt, az 'A' az 'Anterior' (elülső) részt, míg az 'S' a 'Superior' (felső) részt jelöli.

A RAS koordináta-rendszer alapján három fő szeletorientációt különböztetünk meg:

1. **Transzaxiális:** A beteg hosszanti tengelyére merőleges képsíkokat jelöli.
2. **Sagittális:** A bal-jobb tengelyre merőleges képsíkokat képviseli.
3. **Koronális:** A hátulról-előre tengelyre merőleges képsíkokat foglalja magában.

Amennyiben a szeletek nem esnek egybe ezekkel a fő irányokkal, **ferde szeleteknek** nevezzük őket. A RAS koordináta-rendszer és a szeletorientációk együttes ismerete elengedhetetlen a pontos és informatív vizuális reprezentációk előállításához, valamint a diagnosztikai pontosság növelése érdekében. A különböző orientációkat a 2.2 ábra szemlélteti.



2.2. ábra. Gerinc CT felvétel rendre koronális, transzaxiális és sagittális nézetben. [4]

2.1.2. Képszegmentáció számítógépek nélkül

A manuális képszegmentáció egy széles körben alkalmazott módszer, amelyben egy szakértő, például egy orvos vagy képfeldolgozási szakember grafikus felhasználói felületen kijelöli az érdeklődésre számot tartó anatómiai vagy strukturális területeket. Ez a módszer az emberi szakértő által nyújtott intuitív és komplex mintafelismerési képességeknek köszönhetően rendkívül pontos lehet. A manuális szegmentáció különösen hasznosnak bizonyul egyedi vagy ritka esetekben, ahol az automatizált módszerek nem nyújtanak megfelelő pontosságot vagy nem elég érzékenyek.

Noha megvannak az előnyei, a manuális szegmentáció rengeteg nehézséggel is jár. Az egyik legnagyobb kihívás, hogy a módszer rendkívül időigényes. Ez különösen problémás lehet nagy adathalmazok esetében, hiszen a feladatok párhuzamosítása még több szakember számtalan munkaóráját emészti fel. Továbbá, mivel az emberi beavatkozás szükséges, a módszer hajlamos lehet a szubjektivitásra és az emberi hibára, amelyek befolyásolhatják a további elemzések és diagnosztikai folyamatok minőségét.

A manuális szegmentáció rossz skálázhatósága miatt az informatikai eszközök fejlődésével felmerült az igény a folyamat automatizálására. Hosszú időn keresztül szakértői módszerekkel próbálták a feladat bizonyos aspektusait kiváltani, de ezek a megoldások még mindig nem nevezhetők automatikusnak, ugyanis gyakran szakértői beavatkozást igényelnek. Természetesen szakértői megoldásokból is találunk teljesen automatizált módszert, illetve olyan megközelítést, ami nem igényli szakértők ellenőrzését, az ilyen módszereket viszont általában nehéz kifejleszteni. Az elmúlt években egyre elterjedtebbek a gépi tanuláson alapuló módszerek, amik a jövőben jó eséllyel automatizálhatják majd a folyamatot.

2.2. Képfeldolgozás és szegmentáció

A képfeldolgozás és szegmentáció alapvető fogalmak az orvosi képalkotás és analízis területén. A képfeldolgozás egy olyan technika, amelynek célja a digitális képek minőségének javítása, információtartalmuk kiemelése vagy éppen azok analízise. A folyamat magában foglalhat számos algoritmust és módszert, például zajszűrést, éldetektálást vagy kontraszt-növelést, amelyek segítenek a kép tulajdonságainak kiemelésében vagy módosításában. [6]

A szegmentáció egy specifikus képfeldolgozási technika, amelynek célja a kép különböző részeinek, vagy "szegmenseinek" az elkülönítése. Ebben a kontextusban a szegmensek olyan pixelekből állnak, amelyek hasonló tulajdonságokkal rendelkeznek, például azonos intenzitással vagy textúrával. A szegmentáció lehet egyszerű küszöbérték-alapú módszer, de lehet komplex gépi tanulási algoritmus is. Az orvosi képalkotásban a szegmentáció gyakran használt módszer a különböző anatómiai struktúrák, például csigolyák, izmok vagy agyi régiók elkülönítésére a felvételeken. [19]

A képfeldolgozás és szegmentáció tehát kulcsfontosságú eszközök az orvosi képalkotásban, melyek lehetővé teszik pontos diagnosztik felállítását és a hatékony terápiás stratégiák kialakítását. Ebben a szekcióban részletesen bemutatom ezeknek a technikáknak az alapjait, valamint azokat az előnyöket és korlátokat, amelyek az orvosi alkalmazásokban való használatukkal járnak.

2.2.1. Szakértői módszerek

A "szakértői módszerek" alatt olyan technikákat és eljárásokat értünk, amelyeket az orvosi képalkotásban és analízisben használnak és amelyek nem tartalmazzak gépi tanulási algoritmusokat. Ezek a módszerek gyakran félig automatizált eljárások, amik bizonyos határfokig képesek ellátni egy-egy feladatot, de ennek ellenére is szakértői felülvizsgálatot igényelnek. A szakirodalom sok különböző módszerrel csoportosítja a szakértői módszereket. A kutatásaim alatt nem találtam egyértelmű kategóriákat a módszerek felsorolására, így Tanács Attila publikációja alapján [28] csoportosítottam a módszereket. Az alábbiakban leírt módszereket is az ő munkája alapján fejtettem ki.

2.2.1.1. Régió alapú módszerek

A régió alapú módszerek általában két vagy több különböző területre osztják a felvételt az egyes képpontok intenzitásértékei alapján. A legerjedtebb technika ebben a kategóriában a globális küszöbölés, amely egy előre meghatározott intenzitásszint (T) alapján osztályozza a képpontokat előtérbe és háttérbe. Ez a módszer különösen gyors és alkalmazható specifikus esetekben, például csontok vagy más, nagy kontraszttal rendelkező struktúrák szegmentálására. Kiemelten fontos, hogy a küszöbértéknek a megválasztásakor figyelembe kell venni a különböző régiók intenzitását, hogy a szegmentációs folyamat alacsony hibával menjen végbe.

Hasonlóan elterjedt módszerként beszélhetünk a régiónövelő algoritmusokról, amelyek egy, vagy több kiindulópontból (magból) indulnak és a hasonló intenzitású szomszédokat adják hozzá az éppen szegmentált területhez. A módszer rugalmas, de a kiválasztott kiindulási pontok és a homogenitási kritériumok jelentősen befolyásolják az eredményt.

Említésre méltóak még a fuzzy szegmentációs módszerek. Az algoritmus lényege, hogy "fuzzy" kapcsolatot definiál a képpontok között. Ez a kapcsolat azt jellemzi, hogy a két adott képpont "mennyire tartozik egy régióba". A módszer előnye, hogy finomabb, árnyaltabb szegmentációt tesz lehetővé, ám a sikeres alkalmazás kulcsa a megfelelő fuzzy affinitás és küszöbértékek kiválasztása, ami sok esetben komplex feladatnak bizonyul.

2.2.1.2. Él alapú módszerek

Az él-alapú módszerek a képpontok közötti intenzitáskülönbségek alapján azonosítanak határokat vagy éleket a képen. Az éldetekció tipikusan a kép gradiensének, azaz az elsőrendű deriváltjának vizsgálatával történik. A gradiens iránya, amely a "legnagyobb csökkenés" irányát jelöli, a gradiens komponenseinek arányából számítható. A másodrendű derivált, ismertebb nevén a Laplace operátor, szintén hasznos lehet az éldetekcióban, különösen a szélsőértékek azonosításában.

Diszkrét digitális képek esetében konvolúciós maszkokat, például Roberts, Prewitt, Sobel és Frei-Chen maszkokat használnak a gradiens és a Laplace operátor közelítésére. Az éldetekció után sok esetben szükség van utófeldolgozásra, mivel az élek általában nem folytonosak.

A kategóriában elterjedt megoldás a Canny éldetektor algoritmus használata, amely élelnyomást, hiszterézist és jellemző szintézist is magában foglal, és széles körben használt az éldetekció területén. Az él hiszterézis egy olyan technika, amely az élek lokális környezetét vizsgálja, és az egyező gradiensméretek és -irányok alapján dönt az élek hozzáadásáról vagy elnyomásáról. A zajérzékenység csökkentése érdekében előzetes simító lépések, például Gauss simítás is alkalmazhatók.

Összességében az él-alapú módszerek gyors és hatékony szegmentációs technikákat kínálnak, de kihívást jelent a zajérzékenység és az éldarabok összekötésének szükségessége.

Összességében a régió alapú szegmentációs módszerek előnye, hogy egyszerűek, gyorsak és hatékonyak lehetnek bizonyos jól elkülönülő intenzitású régiók azonosításában. Ugyanitt, ezek a módszerek gyakran nem teljesítenek jól bonyolultabb struktúrák vagy összetett textúrák esetén, és a választott küszöbértékek vagy kiindulási pontok jelentősen befolyásolhatják az eredmény minőségét. Továbbá, a globális küszöbértékek nem mindig alkalmazhatók változó intenzitású régiók esetén, és az automatizációs szint gyakran korlátozott. Ugyan a fuzzy alapú megközelítések lehetővé teszik a finomabb szegmentációt, de a megfelelő határfokhoz az algoritmus bonyolult paraméterezési eljárásokat követel meg, illetve a kimenet gyakran igényel összetett utófeldolgozást.

2.2.1.3. Modell alapú megközelítések

A modell-alapú megközelítések a szegmentációban kihasználják az anatómiai struktúrák globális és lokális hasonlóságait és különbségeit. Egy gyakran alkalmazott technika ebben a kategóriában a Cootes féle Pont-Eloszlási Modell (PDM - Point Distribution Model). A PDM létrehozásának alapvető lépései a következők:

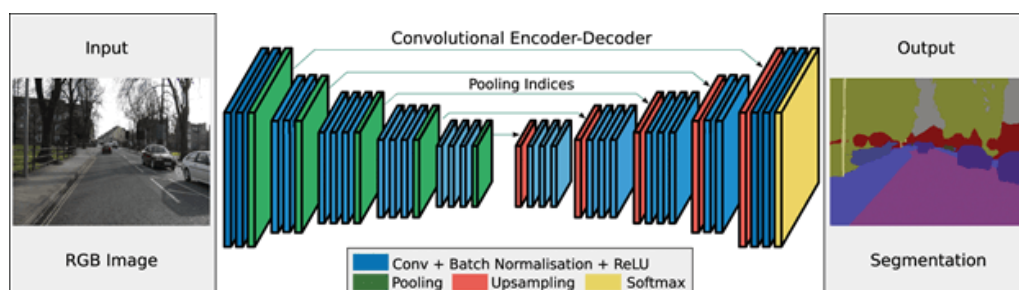
1. Gyűjtsünk össze nagy számú képet ugyanarról az anatómiai szervről (M jelölje a képek számát). Fontos, hogy ezek a képek reprezentálják a szerv alakjának és méreteinek változatosságát a populációban.
2. Minden képen jelöljük ki N darab egymásnak megfeleltethető pontot, manuálisan vagy automatikusan. Ezek a pontok közelítsék a szerv körvonalát, és lehetnek összekötve vonalakkal 2D-ben vagy felszínhálókkal 3D-ben.
3. Transzformáljuk a pontthalmazokat egy közös referencia térbe, például pontpárillesztő módszerrel, figyelembe véve az eltolást, elforgatást és skálázást. Ezzel a lépéssel a megfeleltethető pontok egymás „közelébe” kerülnek, és ezek „szóródása” alapul szolgál az alak statisztikai analíziséhez.

A továbbiakban kovariancia-mátrixot számítunk a pontok átlagából és szórásából, majd a kovariancia-mátrix sajátértékeit és sajátvektorait használjuk az alakok variabilitásának modellezésére.

Ennek a megközelítésnek az előnye, hogy a statisztikai modellek segítségével olyan alakzatok is vizsgálhatók, amelyek nem szerepelnek a tanító halmazban, így nagyobb rugalmasságot és pontosságot biztosít a szegmentációs folyamatban. [28]

2.2.2. Neurális hálózatok alkalmazása

A neurális hálózatok az utóbbi években forradalmasították a képfeldolgozás és képszegmentáció területét. A konvolúciós neurális hálózatok (CNN), például, különösen hatékonyak képi adatok hierarchikus jellemzőinek tanulásában, ami lehetővé teszi a komplex szegmentációs feladatok megoldását is. A mély tanulási algoritmusok, mint a *U-Net* vagy a *V-Net*, nem csak magas szintű pontosságot érnek el, de képesek adaptálódni a képi adatok változatosságához és komplexitásához is. Ezen kívül, a neurális hálózatok gyakran kevesebb manuális előfeldolgozást és jellemzőkinyerést igényelnek, ami jelentősen csökkenti az emberi beavatkozás szükségességét. A neurális hálózatok alkalmazása a képszegmentáció területén nem csak a pontosság és az hatékonyság növelését teszi lehetővé, hanem új lehetőségeket is nyit a komplex és kihívást jelentő orvosi és ipari alkalmazásokban. Természetesen a módszernek megvannak a saját limitációi. A neurális hálózatok tanítása gyakran rengeteg adatot vesz igénybe, aminek a megszerzése sok időbe és pénzbe kerülhet (már ha egyáltalán rendelkezésre áll az adat). Az adatbeszerzés folyamatát gyakran a GDPR szabályok is megnehezítik, így külön szakértőkre van szükség, akik a jog ezen aspektusát jól ismerik. Szintén komoly nehézség, hogy a neurális hálózatok nem rendelkeznek minden esetben jó általánosítási képességgel. Ez azt jelenti, hogy annak ellenére, hogy egy modell képes a tanítási és validációs adathalmazokon jól teljesíteni, valódi tesztelés közben gyakran alulteljesíthet a rendszer. A módszer további nehézségeit hosszan sorolhatnánk még, összességében azt kell szem előtt tartanunk, hogy a mesterséges intelligencia alapú megoldások sem alkalmazhatók minden területre. Adott esetben egy szakértői módszer alapú algoritmus jobb eredményt képes elérni, mint egy-egy neurális hálózat.



2.3. ábra. Életszerű példa a képszegmentációra neurális hálózatok felhasználásával. [2]

2.2.2.1. U-Net architektúra

A *U-Net* architektúra az egyik legelterjedtebb megoldás a gerincszegmentációban. A *U-Net* kifejezetten orvosi képfeldolgozási feladatokra lett tervezve, és képes nagy pontossággal szegmentálni a gerinc különböző struktúráit. Az architektúra előnye, hogy kevesebb adatot igényel a tanításhoz, és mégis képes magas szintű pontosságot elérni. A modell továbbá jól skálázható, így alkalmazható különböző felbontású és méretű képek esetében is [22].

2.2.2.2. V-Net architektúra

A *V-Net* egy 3D-s változata a *U-Net* architektúrának, és kifejezetten 3D-s képfeldolgozási feladatokra lett tervezve. A *V-Net* kiválóan alkalmas a gerinc komplex 3D-s struktúrájának modellezésére és szegmentálására. A *V-Net* architektúra különösen hatékony a csigolyák

közötti tér és a csigolyák geometriai formájának pontos meghatározásában [15]. Mivel a kutatás célja a jelenleg elérhető modellek hatásfokának növelése, valamint a hardverigények csökkentése, így nem egy eleve optimalizált modellel, hanem a *U-Net* háromdimenziós változatával foglalkoztam.

2.2.2.3. Hibrid modellek

A hibrid modellek, például a 2D és 3D *U-Net* kombinációja, egyre népszerűbbek a gerincszegmentáció területén. Ezek a modellek a 2D és 3D információk előnyeit egyesítik, lehetővé téve a gerinc struktúrájának részletes és pontos modellezését. A hibrid modellek különösen hasznosak lehetnek olyan esetekben, ahol a 3D információ elengedhetetlen, de a számítási erőforrások korlátozottak [5].

2.3. Létező implementációk vizsgálata

A konvolúciós neurális hálózatok, különösen a *U-Net* architektúrák terén elért fejlődés jelentős mértékben hozzájárul az orvosi képfeldolgozás, így a gerinc szegmentáció területén is tapasztalható innovációkhoz. A technológiai fejlődés gyors üteme és a kutatási eredmények bőséges mennyisége miatt fontosnak tartom az aktuális megoldások rövid ismertetését. Ebben a fejezetben bemutatok pár modern modellt, illetve azokat a kihívásokat, amelyek aktuálisan jelen vannak ezen a kutatási területen. A vizsgálattal lehetőséget biztosítok az olvasó számára, hogy betekintést kapjon a témában folyó kutatásokról, és megértse, hogy milyen kontextusban helyezkedik el a saját munkám a meglévő ismeretek tükrében.

2.3.1. Webes gerincszegmentáció U-Net alapon

Kim és munkatársai [10] egy deep learning alapú, webes automatikus gerincszegmentációs módszert fejlesztett ki, amely különösen a hátfájás diagnosztizálásában lehet hasznos. A kutatásukban 344 CT képet használtak fel. Az adathalmazból 330 felvételt a hálózat tanítására használtak fel, a maradék 14 CT-t pedig tesztelési célokra különítették el. A módszerük U-Net architektúrára alapul, és a tesztelés során átlagosan 90,4%-os Dice-score értéket, és 96,81%-os pontosságot értek el. Az algoritmusuk előnye, hogy a webes felületen keresztül bármilyen operációs rendszeren futtatható, és nincs szükség különleges hardverkövetelményekre. A rendszer képes a felhasználó által feltöltött képek gyors és pontos szegmentációjára. A kutatók Flask webes keretrendszert használtak a fejlesztés során, amely egy Python alapú mikro webes keretrendszer.

2.3.2. MRI szegmentáció a SAU-Net felhasználásával

A SAU-Net egy új módszer gerinc MRI felvételek szegmentálására, amely az inter-slice attention mechanizmusát alkalmazza. A SAU-Net architektúrája két fő komponensből áll: egy Dense U-Net struktúrából, amely az inter-slice jellemzők kinyerésére szolgál, és egy inter-slice attention modulból (ISA), amely az egymás melletti szeletek információját használja a szegmentációs eredmények finomítására. [31] A Dense U-Net struktúra az encoder-decoder U-Net architektúrára épül, viszont dense kapcsolatokat ad hozzá a konvolúciós rétegekhez, hogy növelje a jellemzőkinyerés hatékonyságát. Az ISA modul attention maszkokat generál, amelyek kiemelik a legfontosabb régiókat és elnyomják az irreleváns területeket, így javítva a 3D szegmentációs teljesítményt. A módszer kiemelkedő teljesítményt mutatott a SpineSegT2W adathalmazon, ahol jobb eredményeket ért el, mint a hagyományos 2D és 3D U-Net modellek. A SAU-Net Dice score-ja 89,86%, a precizitása 89,75% volt. [31] A SAU-Net előnye, hogy képes a 3D információk hatékony felhasználására anélkül, hogy jelentős számítási erőforrásokat igényelne, ami különösen

releváns lehet a jelenlegi kutatásban, ahol a számítási költségek minimalizálása is egy fontos cél. A módszer tehát nem csak magas szegmentációs pontosságot kínál, hanem az erőforrás-hatékonyság szempontjából is előnyös lehet.

2.3.3. Gerincszegmentáció az nnU-Net modell felhasználásával

A tanulmány célja egy olyan szegmentációs modell kifejlesztése, amely könnyen használható, nyilvánosan elérhető, és képes az anatómiailag releváns struktúrák széles körének szegmentálására különböző klinikai beállításokban. A modell fejlesztéséhez egy sokszínű, 1204 CT-sorozatból álló adathalmazt használtak. A modell teljesítményét egy 4004 betegből álló, egész testet ábrázoló CT-adathalmazon is tesztelték, és az életkorfüggő változásokat is vizsgálták. A modell magas pontosságot mutatott, a Dice hasonlósági együtthatója 0,943 volt. Ez a tanulmány is alátámasztja, hogy a U-Net modell megfelelően felhasználható anatómia struktúrák széles körének szegmentálására, ami jelen kutatás alapjait is tovább erősíti. [29]

3. fejezet

Implementáció elméleti hátttere

Az előző fejezetekben részletesen bemutatam a képfeldolgozás és szegmentáció széleskörű szakirodalmát, különös tekintettel a konvolúciós neurális hálózatokra (CNN). A jelen fejezet kritikus jelentőségű az egész kutatási projekt szempontjából, mivel itt fogom alaposan körbejárni az általam választott implementációs módszer elméleti alapjait. E fejezet célja, hogy világossá tegye az olvasó számára, milyen alapelvek és technikai részletek mentén épül fel a 2D és 3D U-Net modell, illetve az ezen komponensekből előállított hibrid 2D-3D U-Net architektúra, amelyet kutatásaim során használtam. Az elméleti keretek tisztázása nemcsak a kutatás alapjainak pontos megértése miatt elengedhetetlen, hanem azért is, hogy a kutatás reprodukálhatósága biztosítva legyen. A rendelkezésre álló szakirodalom alapján bemutatom az említett architektúrákat, működésüket és matematikai háttérüket.

3.1. Konvolúciós neurális hálózatok (CNN)

Az emberi idegrendszer működéséből inspirálódva, a Mesterséges Neurális Hálózatok (ANN) komplex rendszerek, melyek magas számú, összekapcsolt számítási csomópontból (neuronokból) állnak. Ezek a csomópontok együttműködve komplex feladatokat oldanak meg, valamint optimalizálják a rendszer kimenetét különböző bemenetek alapján. Az ANN architektúra számos változatban létezik. Az architektúra alapján kialakíthatunk például előrecsatolt neurális hálózatokat (FNN), korlátozott Boltzmann gépeket (RBM) és rekurrens neurális hálózatokat (RNN). Az ANN-ben alkalmazott két alapvető tanulási paradigma a felügyelt és nem felügyelt tanulás. Az felügyelt tanulás során előre címkézett bemeneteket használnak, míg az nem felügyelt tanulásnál a sikeresség egy költségfüggvény csökkentésén vagy növelésén alapul. Képfeldolgozási feladatokban általában az felügyelt tanulás a domináns.

A Konvolúciós Neurális Hálózatok (CNN) az ANN egy speciális típusát képezik, amelyeket elsősorban képminták felismerésére használnak. A CNN felépítése hasonló az ANN-hez, de a rétegek neuronjai három dimenzióba vannak szervezve: magasság, szélesség és mélység. A hálózat általános esetben három fő rétegtípusból épül fel: konvolúciós rétegek, pooling rétegek és teljesen összekapcsolt rétegek. Az ANN egyik legnagyobb korlátozása a nagy számítási igény, amely különösen képfeldolgozási feladatok esetében jelent problémát. A konvolúciós neurális hálózatok erre adnak megoldást azzal, hogy csökkentik a szükséges paraméterek számát és ezzel egyidejűleg a hálózat általánosító-képességét is növelik, csökkentve az overfitting (túltanulás) esélyét. [18]

3.1.1. Konvolúciós rétegek

A konvolúciós réteg célja a bemeneti adatok lokális részének elemzése. A réteg egy vagy több konvolúciós szűrőt (kernelt) alkalmaz a bemeneti adatok egy kis részére. Matematikailag egy konvolúciós művelet két függvény (f és g) között a következőképpen definiálható:

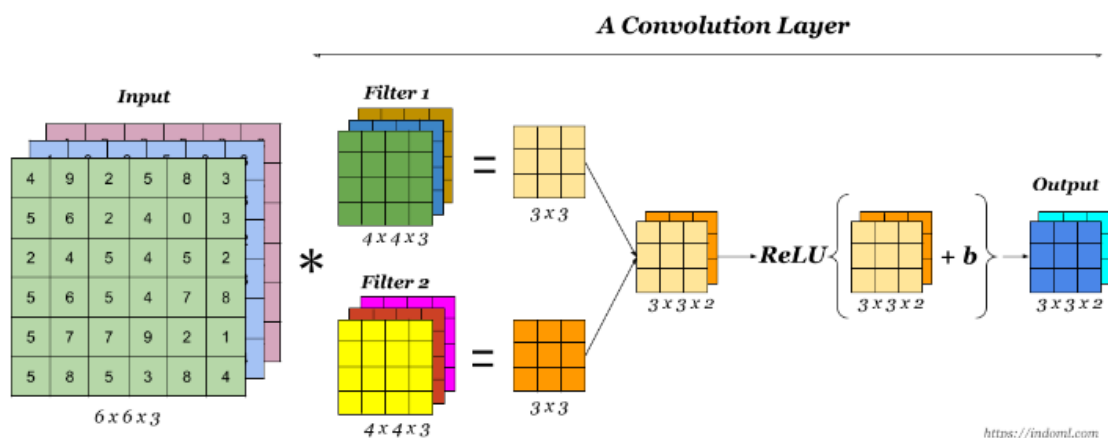
$$(f * g)(t) = \int_{-\infty}^{\infty} f(\tau)g(t - \tau)d\tau \quad (3.1)$$

Ahol f és g a konvolválts függvények, τ egy változó, és t az idő vagy a pozíció változója, attól függően, hogy melyik területen alkalmazzuk a konvolúciót. A képfeldolgozásban a konvolúciós művelet diszkrét és a kép pixelin alkalmazzuk.

A konvolúciós réteg számítása

$$o_{i,j} = \sum_{m,n} I_{i+m,j+n} \cdot W_{m,n} \quad (3.2)$$

Ahol $o_{i,j}$ az output pixel értéke, I az input kép, és W a konvolúciós kernel. Az m és n változók végigiterálnak a kernel dimenzióin. [18] Az alábbi ábrán egy konvolúciós neurális hálózat konvolúciós rétegének működése látható.



3.1. ábra. A konvolúciós réteg kimenetének előállításának alapja, Forrás: [20]

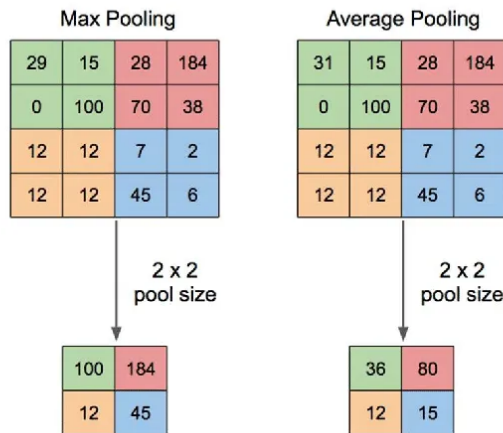
Ez a konvolúció számtalan CNN architektúrában megtalálható. A kutatás során én is ezt a koncepciót használtam a 2D-s U-Net modellnél, illetve hasonlóképpen ennek a konvolúciós módszernek egy 3D-s változatát a 3D U-Net modell implementálásakor.

3.1.2. Pooling rétegek

A pooling réteg célja a bemeneti adatok térbeli redukciója, ami segít csökkenteni az összesített paraméterek számát és növeli a modell toleranciáját az eltolásokhoz és transzformációkhoz. Általánosan elterjedt a max pooling, amely a bemeneti ablakon belüli pixelértékek közül a legnagyobbat választja ki. A pooling réteg max-pooling alkalmazásával az alábbi módon számolja az outputot:

$$o_{i,j} = \max_{m,n} I_{i+m,j+n} \quad (3.3)$$

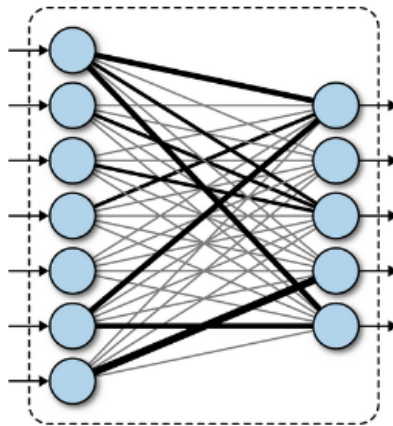
Ebben az esetben $o_{i,j}$ az output pixel értéke, és I az input kép. Az m és n változók itt is végigiterálnak a pooling ablak dimenzióin. [18] Az alábbi ábra segít megérteni a max-pooling és az average pooling módszereket.



3.2. ábra. A pooling réteg kimenetének előállítása adott bemenet alapján, Forrás: [21]

3.1.3. Teljesen összekapcsolt rétegek

Ezekben a rétegekben nincsen semmi különlegesség, implementációjuk megegyezik a standard neurális hálózatokban megszokott teljesen összekapcsolt rétegekkel, azaz minden neuron kapcsolódik az előző és a következő réteg összes neuronjához. Az utolsó teljesen összekapcsolt réteg gyakran egy softmax aktivációs funkciót használ, ami kiszámítja az osztályok valószínűségek eloszlását. [18] Az alábbi ábrán a jobb megértés érdekében egy neurális hálózat két rétege látható, a rétegek közötti kapcsolatokkal.



3.3. ábra. Teljesen összekapcsolt rétegek Forrás: [21]

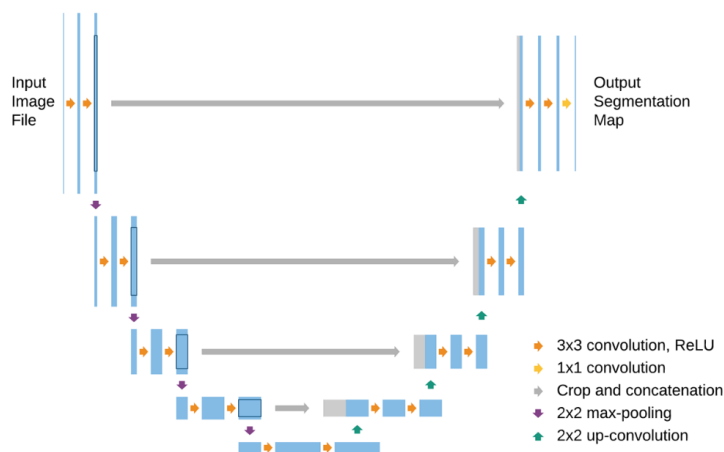
3.1.4. Konvolúciós neurális hálózatok napjainkban

A konvolúciós neurális hálózatok (CNN) az utóbbi években váltak dominánssá a gerincszegmentáció területén, elsősorban kiemelkedő teljesítményük és adaptivitásuk miatt. A gerincszegmentáció különösen összetett feladat, mivel a gerinc anatómiája rendkívül változatos és komplex. A CNN alapú megoldások ebben a kontextusban is bizonyították hatékonyságukat, és számos alkalmazási területen, például a csigolyák és az intervertebrális lemezek automatikus azonosításában, mutatnak kiemelkedő eredményeket.

Összességében a CNN alapú architektúrák és modellek kiválóan alkalmazhatók a gerincszegmentáció területén, így kutatás során én is egy CNN alapú hálózatot implementáltam.

3.2. A 2D U-Net modell

A U-Net architektúra egy rendkívül hatékony és népszerű konvolúciós neurális hálózat (CNN), melyet eredetileg egészségügyi felvételek szegmentálására fejlesztettek [22]. A modell alakja "U" alakú, amit a konvolúciós (encoder) és dekonvolúciós (decoder) részek együttese alkot. Ebben a szakaszban részletesen bemutatom a 2D U-Net modell elméleti háttérét, annak rétegeit és a matematikai műveleteket, amelyek a háttérben történnek.



3.4. ábra. Kétdimenziós *U-Net* architektúra Forrás: Siddique et al. [26]

3.2.1. Architektúra és rétegek

A U-Net modell encoder-decoder architektúrájú. Az encoder rész konvolúciós rétegekből, aktivációs függvényekből és max-pooling rétegekből áll. Minden konvolúciós lépés tartalmaz egy vagy több 3×3 konvolúciós réteget. A konvolúciós réteget minden esetben egy ReLU (Rectified Linear Unit) aktivációs függvény követi. A rétegek sorát végül egy 2×2 max-pooling réteg zárja, amely felezi az adatok térbeli dimenzióit.

$$\text{Konvolúció: } Y = X * K$$

$$\text{ReLU: } f(x) = \max(0, x)$$

$$\text{Max-Pooling: } \max(\text{Local Region})$$

Az encoder után egy bottleneck réteg következik, amelynek célja a jellemzők kiemelése a bemeneti adatokból. Ezt követően a decoder rész indul, amely konvolúciós rétegekből, aktivációs függvényekből és upsampling rétegekből áll össze. Az upsampling rétegek során a kép méretét növeljük, és összefűzzük az encoder oldalról kapott jellemzőtérképeket a dekódolt jellemzőtérképekkel. Ez egy nagyon fontos gondolat a *U-Net* architektúrájában, hiszen ezzel az átcsatolással az upsampling réteg átveheti a downsampling réteg képeinek eredeti térbeli információját torzítás nélkül a szegmentációs maszkok pontosabb meghatározásának érdekében.

Upsampling: Nearest Neighbor vagy Transposed Convolution

Átcsatolás: $C = [A; B]$

3.2.2. Rétegsorrend és forma

Az eredeti publikációban Ronneberger et al. [22] az alábbi formát javasolják a modellhez:

- Encoder: $Conv1 \rightarrow ReLU \rightarrow Conv2 \rightarrow ReLU \rightarrow MaxPooling$
- Bottleneck: $Conv \rightarrow ReLU$
- Decoder: $Upsampling \rightarrow Concatenation \rightarrow Conv \rightarrow ReLU$

A konvolúciós rétegek szűrőinek és csatornáinak száma a modell mélységével exponenciálisan nő, míg a max-pooling és upsampling rétegek a kép felbontását csökkentik vagy növelik.

3.2.3. Matematikai műveletek

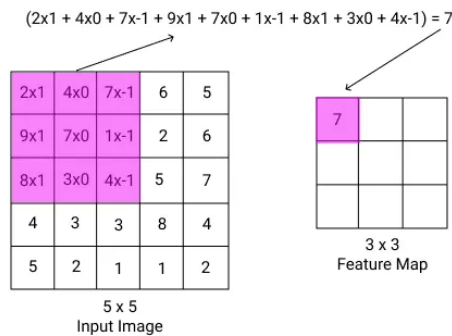
A 2D *U-Net* architektúrában lineáris és nemlineáris matematikai műveletek sora található meg. A dolgozat szempontjából kiemelten fontosnak tartom, hogy a modell elméleti háttere ne csak informatikusok számára (programkód formájában) legyen olvasható, hanem bármilyen természettudományokkal foglalkozó kolléga megértse azt. Ennek érdekében az alábbi szekcióban részletesen kifejtem a modell rétegeinek matematikai leírását, hogy általánosan érthető legyen a tárgyalt architektúra működése.

3.2.3.1. Konvolúciós rétegek

A konvolúciós (Conv) rétegek lineáris műveletek, amelyek a bemeneti jellemzőtérkép és egy konvolúciós kernel (szűrő) konvolúcióját hajtják végre. A konvolúciós művelet matematikai leírása a következő:

$$Y_{i,j} = \sum_m \sum_n X_{i+m,j+n} \cdot K_{m,n} \quad (3.4)$$

ahol Y a kimeneti jellemzőtérkép, X a bemeneti jellemzőtérkép és K a konvolúciós kernel. Egy konvolúciós lépés folyamata a 3.5 ábrán tekinthető meg.



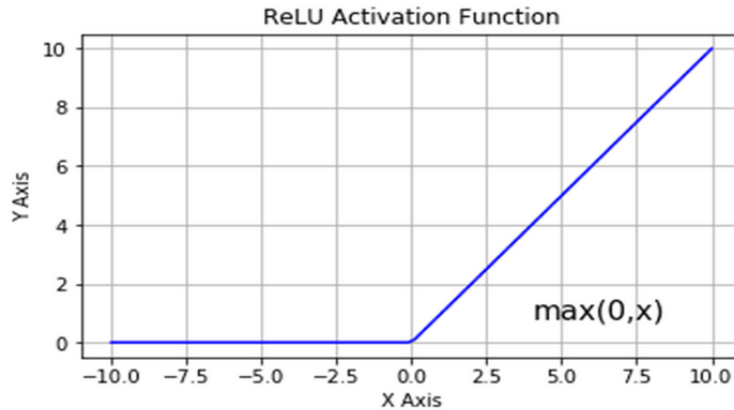
3.5. ábra. Egy konvolúciós lépés [23]

3.2.3.2. ReLU aktivációs függvény

A ReLU (Rectified Linear Unit) egy nemlineáris aktivációs függvény, mely a következőképpen definiálható:

$$f(x) = \max(0, x) \quad (3.5)$$

A ReLU célja, hogy nemlineáris tulajdonságokat adjon a hálózathoz, lehetővé téve számára, hogy komplexebb függvényeket approximáljon. A ReLU függvény a 3.6 ábrán tekinthető meg.



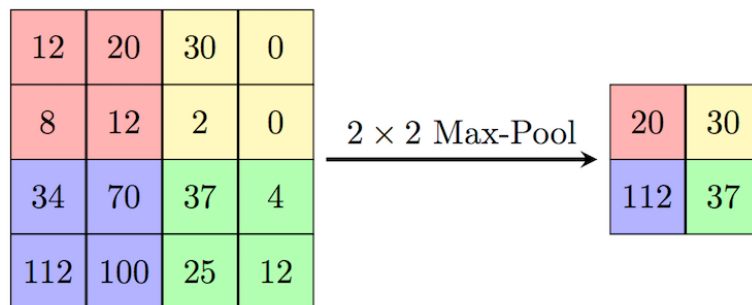
3.6. ábra. A Rectified Linear Unit függvény [17]

3.2.3.3. Max-pooling rétegek

A Max-Pooling egy másik nemlineáris művelet, amely a jellemző térkép méretének csökkentésére szolgál.

$$\text{Max}(\text{Local Region of } X) \quad (3.6)$$

A Max-Pooling alkalmazásával a modell kisebb méretű jellemző térképeket generál, így csökkentve a számítási követelményeket, miközben megtartja az adatok lényeges jellemzőit. A Max-Pooling folyamata a 3.7 ábrán tekinthető meg.



3.7. ábra. Max-pooling folyamata []

3.2.3.4. Összehasonlítás egy általános CNN architektúrával

Ha összehasonlítjuk az irodalomkutatás konvolúciós neurális hálózatokat ismertető részét 2.2.2 a *U-Net* modell leírásával arra lehetünk figyelmesek, hogy az architektúra szinte

megegyezik. A modell a CNN architektúrák mintapéldája, ennek ellenére néhány kiemelkedő elméleti változtatással (ReLU, concatenation) bizonyos adathalmazokon jobban teljesít a U-Net, mint egy átlagos CNN.

3.2.4. Összefoglalás

Összességében a 2D U-Net modell a komplexitás és a hatékonyság ideális egyensúlyát kínálja, lehetővé téve a gyors és pontos képszegmentációt minimális hardverkövetelmények mellett. Az encoder és decoder közötti kapcsolatok lehetővé teszik az alacsony szintű jellemzők és magas szintű kontextus egyesítését, ami pontosabb szegmentációt eredményez.

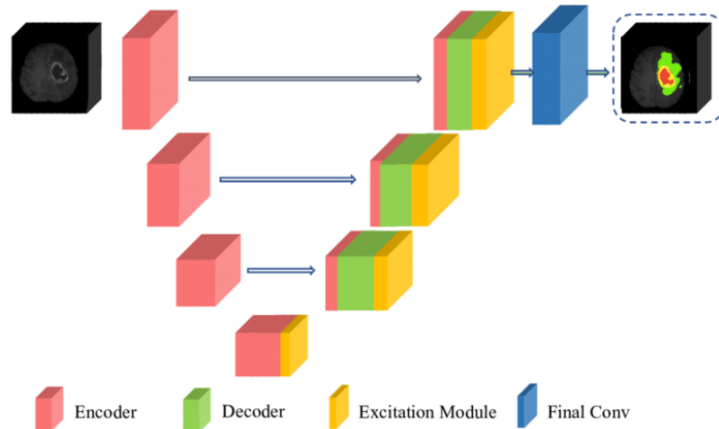
3.3. A 3D U-Net modell

A 3D *U-Net* az eredeti 2D *U-Net* architektúra továbbfejlesztett változata, amely kifejezetten a volumetrikus adatok, például a CT és MRI felvételek, automatikus szegmentálására lett tervezve.

Míg a 2D *U-Net* alkalmazása már elterjedt és sikeresen megold számos képszegmentációs problémát, a 3D verzió lehetőséget ad arra, hogy komplexebb, térbeli struktúrákban rejlő információkat is figyelembe vegyünk.

A kétdimenziós variáns alacsony számításigényének és kiemelkedő teljesítményének hála sok területet meghódított már.

Ennek köszönhetően, a 3D *U-Net* egyre növekvő jelentőséggel bír az orvosi képfeldolgozásban, valamint olyan területeken, ahol a térbeli összefüggések kulcsszerepet játszanak. [32]. A 3.8-as ábrán a modell architektúráját tekinthetjük meg.



3.8. ábra. Háromdimenziós U-Net architektúra. Forrás: Çiçek et al. [9]

3.3.1. Architektúra és rétegek

A 3D U-Net encoder-decoder architektúrája azonos az kétdimenziós *U-Net* architektúrával, azonban az összes alapegység — beleértve a konvolúciós, max-pooling és upsampling rétegeket — háromdimenziós változatban vannak implementálva. A 3D konvolúciós rétegek lehetővé teszik a volumetrikus kontextus figyelembevételét, amely általánosságban javítja a szegmentációs teljesítményt.

3.3.2. Rétegsorrend és forma

Az architektúra az alábbi rétegekből épül fel:

- Encoder: $3DConv1 \rightarrow ReLU \rightarrow 3DConv2 \rightarrow ReLU \rightarrow 3DMaxPooling$
- Bottleneck: $3DConv \rightarrow ReLU$
- Decoder: $3DUpsampling \rightarrow 3DConcatenation \rightarrow 3DConv \rightarrow ReLU$

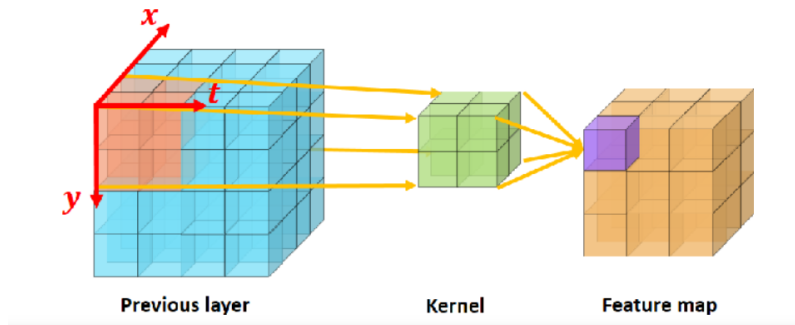
A konvolúciós rétegek mérete általában $3 \times 3 \times 3$, a max-pooling és upsampling rétegek pedig $2 \times 2 \times 2$ méretűek.

3.3.3. Matematikai műveletek

A 3D U-Net alapvető matematikai műveletei hasonlóak a 2D változathoz. Az alapvető különbség a konvolúciós műveletek dimenziószámában van. A 3D konvolúció a következőképpen definiálható:

$$Y_{i,j,k} = \sum_m \sum_n \sum_o X_{i+m,j+n,k+o} \times K_{m,n,o} \quad (3.7)$$

Itt Y a kimeneti jellemzőtérkép, X a bemeneti jellemzőtérkép és K a konvolúciós kernel. A 3D konvolúció lehetővé teszi a térbeli kapcsolatok mélyebb tanulmányozását, amelyek nem jelennek meg a 2D konvolúcióban. A 3.9-es ábra a 3D konvolúció egy lépését vizualizálja a jobb megértés érdekében.



3.9. ábra. Háromdimenziós konvolúció. [8]

3.3.4. Teljesítmény és komplexitás

Bár a 3D U-Net modell komplexitása magasabb, a modell eredményének potenciális javulása - különösen a 3D képi adatoknál - gyakran indokoltá teszi az ezzel járó számítási költségeket. Az architektúra képes felismerni a volumetrikus adatokban rejlő komplex összefüggéseket és ezáltal nagy pontossággal szegmentál [32].

3.4. Hibrid modellek előnyei és kihívásai

Az elmúlt években az orvosi képfeldolgozás területén a mély tanulás alkalmazása forradalmasította a képszegmentálási módszereket. Ebben a kontextusban a 2D és 3D U-Net architektúrák kiemelkedő teljesítményt mutatnak [22, 32]. Mindazonáltal, a 3D modellek magas számítási követelményekkel rendelkeznek, amely korlátozza azok használatát erőforrás-korlátozott környezetekben. Ezzel szemben a 2D modellek gyorsabbak és kevesebb erőforrást igényelnek, de nem képesek kihasználni a térbeli összefüggéseket, amelyek gyakran kritikusak az orvosi képszegmentálásnál [1].

A hibrid 2D-3D modellek célja, hogy kihasználják mindkét modell előnyeit. Az ilyen modellek általában egy 2D szegmentáló modellt használnak az alapvető jellemzők kiemelésére, amelyeket aztán egy 3D modell vesz át a további feldolgozásra. Ezzel a megközelítéssel a rendszer képes minimalizálni a számítási költségeket, miközben megtartja a 3D modellek által kínált magas szegmentálási pontosságot.

Természetesen egyszerre több modell implementálása rengeteg komplikációval járhat, a 2D és 3D modellek közötti információátvitel optimalizálása pedig igen komplex feladat lehet.

4. fejezet

Adathalmazok bemutatása és elemzése

A neurális hálózatok tanítása során elengedhetetlen a megfelelő adathalmaz kiválasztása a modell tanításához és teszteléséhez. A kutatás alatt versenytársként használt modell adathalmazának felhasználása a saját modellünk tanítására számos előnyt és kihívást jelent, amelyeket alaposan meg kell vizsgálnunk, hogy megértsük a módszer hatásait.

4.1. Tanító és validáló adathalmazok

A téma átfogó bemutatásához elengedhetetlen a felhasznált adathalmazok bemutatása és elemzése, hiszen a kutatás során prezentált folyamat egyik legjelentősebb lépése a tanítás. A modell teljes tanítási folyamatához három részre osztjuk az adathalmazunk: tanító halmaz, teszt halmaz és validáló halmaz. Ebben az alfejezetben ezeket a fogalmakat szeretném tisztázni.

A tanító adathalmaz a neurális hálózatok alakításának alapvető eszköze. Ezen az adathalmazon dolgozik a modell, a paraméterek iteratív finomítása során, hogy az optimális súlyokat találja meg. A tanítási folyamat során az adathalmaz minden egyes eleme áthalad a neurális hálózaton, és a kapott kimenetet összehasonlítja a várt eredménnyel. A hálózat paramétereit a hibafüggvény gradiensének segítségével aktualizáljuk. Az adathalmazt tehát az iteratív tanító folyamatban használjuk arra a célra, hogy a hálózatnak megtanítsuk az elvégezni kívánt feladatot.

A validáló adathalmaz funkcióját tekintve egyfajta hidat képez a tanító és a tesztelő adathalmaz között. Miközben a tanító adathalmaz segítségével a modell paramétereit finomítjuk, a validáló adathalmazt azért használjuk, hogy a modell általánosítási-képességét értékeljük, ezzel elkerülve a túltanítást (overfitting) vagy alultanítást (underfitting). Ezen adathalmaz segítségével választjuk ki a legjobban teljesítő modellarchitektúrát és hiperparamétereket, valamint monitorozzuk a tanítási folyamat során a modell teljesítményét.

A tesztelő adathalmaz a modell végső értékelésére szolgál. Ez az adathalmaz teljesen független a tanító és validáló adathalmaztól, és csak a teljesítmény értékelésére használjuk, hogy objektív képet kapjunk a modell általánosítási-képességéről és teljesítményéről. Egy jól teljesítő modell a tesztelő adathalmazon is magas pontosságot, szenzitivitást és specificitást mutat, amelyek indikátorai a modell alkalmazhatóságának.

A tanító, validáló és tesztelő adathalmazok összetett, de szükséges eszközök a neurális hálózatok fejlesztésében és értékelésében. Ezen adathalmazok megfelelő alkalmazása és menedzsmentje kritikus a modell sikeres implementációjához és a robosztus, megbízható teljesítményéhez.

4.1.1. CT adatok tárolása NIFTI formátumban

Mivel az elérhető adathalmazok első sorban *nifti* fájlformátumot használnak, a teljesség igénye nélkül fontosnak tartottam ennek a bemutatását is. A NIFTI (Neuroimaging Informatics Technology Initiative) fájlformátum széles körben használt az agyi képalakítás és más orvosi képalakítási alkalmazások területén. A formátum célja, hogy egyetlen, jól strukturált fájlban tárolja az orvosi képek adatait és metaadatait. A NIFTI fájlformátum lehet egy- vagy kétrészes, ahol az első rész (a fejléc) tartalmazza az általános információkat, mint például a kép dimenziói, a voxel méretek, és az orientáció. A második rész a tényleges képadatokat tárolja. A NIFTI fájlok általában `‘.nii’` vagy `‘.nii.gz’` kiterjesztéssel rendelkeznek, az utóbbi pedig egy tömörített változat. A NIFTI fájlformátum támogatja a többdimenziós adattömbök tárolását, így nem csak 3D-s térbeli adatok, de idősoros (4D) adatok tárolására is alkalmas. A formátum rugalmassága és átfogó támogatottsága miatt vált iparági sztenderddé számos képalakító modalitásban.

4.1.2. TotalSegmentator adathalmaz

A kutatás céljait szem előtt tartva a TotalSegmentator modell által használt adathalmazt választottam a modellem tanítására és validálására. Az adathalmaz és a modell Wasserthal és munkatársai [29] kutatásának eredményeként született. Az adathalmaz kifejezetten képszegmentációra lett tervezve, ezáltal egy sokszínű és átfogó gyűjtemény, amelyet a Bázeli Egyetemi Kórház képarchiválási és kommunikációs rendszeréből válogattak össze. Összesen 1368 CT páciens felvételeit vették alapul három különböző évből (2012, 2016 és 2020), így biztosítva az adathalmaz változatosságát. Az adathalmazból kizárták azokat a CT sorozatokat, amelyek hiányosak voltak, vagy amelyekben az emberi annotátor bizonyos struktúrákat nem tudott egyértelműen szegmentálni.

Érdekes megvizsgálni a folyamatot, ami alapján Wasserthal és munkatársai elkészíteték az adathalmazt, ugyanis rengeteg párhuzamot találhatunk a szakirodalom kutatásnál felsorolt módszerekkel. A felvételek annotálásának folyamata szinte minden képszegmentációs módszert felhasznál a felvételek előkészítésére. A kiinduló adathalmazban 5 páciens felvételeit eleinte kézzel szegmentálták szakemberek, majd a későbbiekben (modernebb megközelítéssel szakértői módszerek helyett neurális hálózatokkal) 2.2.2 emelték ki a CT felvételeken látható anatómiai struktúrákat. Végezetül a hibás részeket szintén szakértők segítségével manuálisan javították.

Az adathalmaz végső formájában 1204 CT sorozatot tartalmaz. Ezt a gyűjteményt tovább bontották egy 1082 páciensből álló tanító adathalmazra (90%), egy 57 páciensből álló validáló adathalmazra (5%), és egy 65 páciensből álló teszt adathalmazra (5%). Jelen kutatás más arányokban bontja fel az adathalmazt tanító, validáló és tesztelő halmazokra, mivel egyszerre két adathalmazzal dolgozunk.

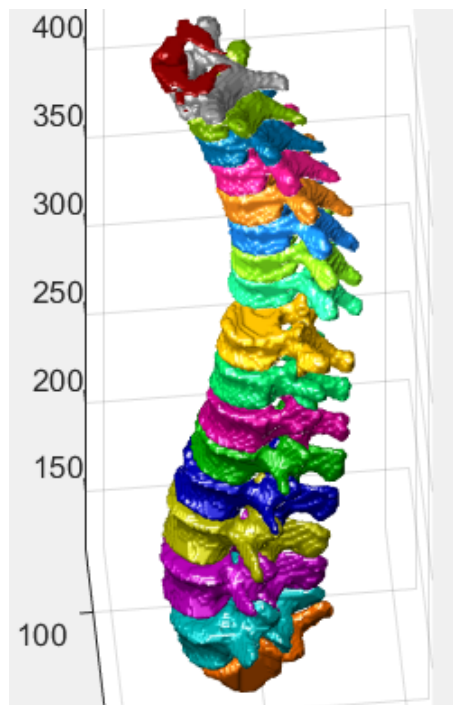
4.1.3. Az adathalmaz felhasználásának előnyei és hátrányai

A kutatás céljából a TotalSegmentator adathalmazának kiválasztása tűneht a megfelelő megoldásnak. Számtalan előnnyel jár, hogy a két modell ugyan azokon az adatokon tanult. Mivel az adathalmaz megegyezik, mélyebb bepillantást kapunk a modellek nyers teljesítményébe, összehasonlíthatjuk, hogy egy-egy bemenetként szolgáló felvételen hogyan teljesítettek különböző módon. Mindezek mellett a modellreől tudjuk, hogy **512x512** pixeles felvételeket használ bemenetként, így a pipeline is alkalmazhatja ezt az egyszerűbb összehasonlítás jegyében.

A sok pozitívummal ellentétben persze hátránya is van annak, hogy az adathalmazok megegyeznek. Mivel nem férünk hozzá a TotalSegmentator tanító, validáló és teszt rész-adathalmazokra osztó algoritmusához, így nem tudjuk felvételenként vizsgálni a modellek

pontosságát, hiszen nem tudhatjuk, hogy egy adott felvétel benne volt-e a TotalSegmentator tanító halmazában. (Amennyiben igen, a modellünk hátrányba kerül, hiszen lehet, hogy a mi verziónk épp először látja a felvételt. Éppen ezért alaposan át kell gondolnunk, hogy hogyan fogjuk összehasonlítani a modellek teljesítményét, hogy a leginkább "fair" megoldást hatjsuk végre.

Összefoglalva tehát, az adathalmaz felhasználásának több előnye és hátránya is van. A lényeg az, hogy amíg szem előtt tartjuk a potenciális problémákat és kihasználjuk az adathalmaz nyújtotta előnyeket érdemes lehet felhasználni az adatokat a kutatáshoz.



4.1. ábra. Az adathalmaz egyik szegmentálási maszkja látható a felvételen. A különböző színe különböző osztályokat, azaz egy-egy csigolyát jelölnek.

5. fejezet

Implementáció

Ezen fejezet célja, hogy bemutassa és részletezze a hibrid 2D-3D *U-Net* pipeline implementációját, különös tekintettel az adatok előkészítésére és feldolgozására, valamint a 2D és 3D modellek tanítására. A fejezet olvasásával az olvasó megismerheti a teljes pipeline működését, a leírt információk alapján pedig reprodukálhatja a kutatás eredményeit.

5.1. Adatok előkészítése

Az adatok előkészítése egy olyan kritikus szakasz, amely meghatározza egy gépi tanulási projekt sikerességét. Az adatok minősége és felhasználási módja közvetlenül befolyásolja a modell tanításának hatékonyságát, valamint a modell általánosító-képességét az új, nem látott adatokon. Ebben a szekcióban az általam javasolt modellhez implementált képfeldolgozási és adataugmentációs lépéseket mutatom be, kitérve arra, hogy ezek a lépések hogyan járulnak hozzá a modell tanítási folyamatának optimalizálásához. Az adatok megfelelő előkészítése nemcsak az eredmények pontosságát javíthatja, hanem időt és erőforrást is megtakaríthat, valamint befolyásolhatja a neurális hálózat hardverigényét.

A képfeldolgozás alapvető eszköze annak, hogy a modell megértse és kiértékelje az adathalmazt. Egy jól megtervezett képfeldolgozási pipeline lehetőséget ad arra, hogy a modell első sorban a lényeges jellemzőkre fókuszáljon, miközben minimalizálja az irreleváns információk hatását. Az adataugmentáció egy további, elengedhetetlen technika, melynek segítségével a modell sokkal robosztusabb lesz. Az adataugmentáció során kisebb transzformációkat hajtunk végre az eredeti képeken, így növelve az adathalmaz méretét és diverzitását. Ez elősegíti a jobb általánosítást és csökkenti a túltanulás kockázatát.

A képfeldolgozás és az adataugmentáció nem csak a modell teljesítményét befolyásolja, hanem a szükséges hardverigényeket és a tanításra fordított időt is. Nagy felbontású, komplex képek feldolgozása gyakran igényel nagy mennyiségű memóriát és számítási kapacitást. Az előfeldolgozás során lehetőségünk van a képek méretének csökkentésére, így csökkentve a hardverigényeket anélkül, hogy jelentős információvesztés történne. Az előfeldolgozást végrehajthatjuk a teljes adathalmazon, majd elmenthetjük a változásokat, ami rövidebb futásidőt eredményezhet az adatok beolvasásánál.

5.1.1. Kétdimenziós adathalmaz előkészítése

A kétdimenziós adathalmaz előkészítése során több alapvető lépés is végrehajtásra kerül a modell teljesítményének optimalizálása érdekében. Elsőként a SimpleITK könyvtár segítségével beolvassuk a CT képeket, és egy dedikált Python objektumban tároljuk azok szeletek közötti távolságát és a CT felvétel Hounsfield értékeit. A képeket a szeletek közötti távolság alapján újramintavételezzük, majd padding-et alkalmazunk az újraméretezés közbeni torzulás elkerülése végett. Ezt követően az adatokat újraméretezzük, hogy meg-

feleljenek a modell bemeneti követelményeinek. A képeken a Hounsfield értékek alapján ablakozást végzünk, hogy kiemeljük a releváns, csontokra vonatkozó információkat. Végül, egy függvény segítségével a szeleteket különböző nézetekre transzformáljuk. Ezen lépések részletes kifejtése és magyarázata a következő alfejezetekben kerül bemutatásra.

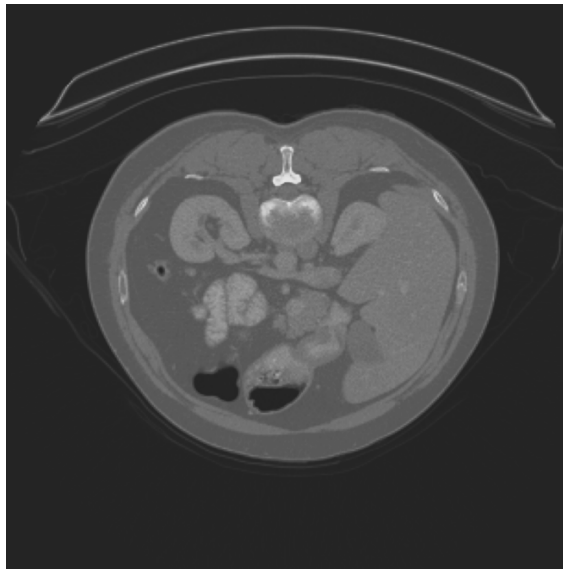
5.1.1.1. Nifti formátumú felvételek beolvasása

A Nifti formátumú CT felvételeket a python-hoz modulként elérhető SimpleITK könyvtár segítségével dolgoztam fel. A SimpleITK egy széles körben használt könyvtár az orvosi képek feldolgozásában, amely lehetővé teszi a Nifti formátumú adatok könnyű és hatékony beolvasását [3][30][13]. A Nifti formátum az egyik legelterjedtebb adatformátum a CT és MRI képek körében, amely a képi információk mellett metaadatok tárolására is lehetőséget biztosít.

Az egyik legkritikusabb paraméter, amit a képek beolvasása során tárolunk, a pixel spacing, azaz a szeletek közötti távolság. Ez az információ kritikus jelentőségű a képek későbbi feldolgozása és elemzése során. A pixel spacing segít az összes kép térbeli felbontásának egységesítésében, ami nélkülözhetetlen az adathalmaz konzisztenciájának fenntartásához.

A képek beolvasása után a Housfield értékeket tároló háromdimenziós mátrix dimenzióit felcseréljük, így lehetővé téve a transzaxiális, sagittális és koronális nézetek generálását. Minden egyes nézet különböző információkat hordoz, és ezek kombinációja növelheti a modell teljesítményét és általánosító-képességét. A lépés különösen fontos annak érdekében, hogy a kétdimenziós *U-Net* modellt több különböző nézetben is tesztelhesük.

A beolvasott adatok és metaadatok tárolásához egy Python objektumot hozunk létre. Az objektum struktúrája úgy van megtervezve, hogy képes legyen tárolni a Hounsfield értékek háromdimenziós mátrixát és a kapcsolódó pixel spacing információt. Az adatokat a numpy könyvtár segítségével kezeljük, amely az adatok gyors és hatékony manipulációját teszi lehetővé [7].



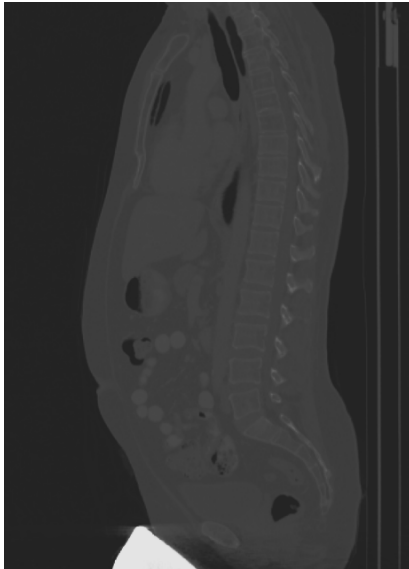
5.1. ábra. Ezen az ábrán a változtatás nélküli, eredeti felvétel látható, az eredeti indexelés miatt transzaxiális nézetben.

5.1.1.2. Szeletek újraméretezése

A szakirodalomban széles körben alkalmazott TotalSegmentator modell 512x512 pixel méretű képeket használ [29]. A hibrid 2D-3D *U-Net* architektúra is hasonlóan van konfigurálva, így ennek megfelelően, az előfeldolgozási folyamat során az összes képet át kell méretezni.

Az erőforrás-hatékonyság szempontjából is előnyös az átméretezés: kisebb képek kevesebb VRAM-ot igényelnek, ami lehetővé teszi a gyorsabb és gazdaságosabb tanítást. Ez különösen fontos, ha komoly GPU kapacitás csak korlátozottan áll rendelkezésre, vagy a modell tanítása és a gradiens számítása meghaladja rendelkezésre álló hardver képességét.

Érdemes megjegyezni azonban, hogy az átméretezésnek lehetnek hatásai a modell pontosságára is. A túl nagy méretcsökkenés például információvesztést eredményezhet, amely ronthatja a szegmentáció minőségét. A méretválasztásnál tehát kompromisszumot kell kötni a számítási igények és a pontosság között.



(a) Ezen az ábrán látható a pixel spacing alapján újramintavételezett, immár sagittális nézetbe transzformált szelet.



(b) Ez az ábra pedig a kép átméretezését mutatja be.

5.2. ábra. A felvételek normalizálásának két lépése: újramintavételezés és átméretezés

5.1.1.3. Padding megadása a szeletekhez

A CT-szeletek előfeldolgozása során nem elhanyagolható szempont a képek méretének standardizálása, amely gyakran megköveteli a képek újraméretezését 5.1.1.2. Az adathalmazban általában nem minden kép rendelkezik négyzet alakú mérettel, sőt, a különböző nézetekben 2.1.1 - például transzaxiális, sagittális, koronális - eltárolt képek formája sok esetben hosszúkás lehet. Ezen méretbeli eltérések kompenzálása érdekében a képekhez ún. "paddinget," vagyis kitöltést adunk.

A padding nélküli képek újraméretezése torz képet eredményezne, ami jelentős mértékben befolyásolná a modell teljesítményét és a szegmentálás pontosságát. Az ilyen torzítások, például a csontok vagy egyéb anatómiai struktúrák elnyúlása, félrevezető információt szolgáltathatnak a modell számára, és ennek következtében hibás szegmentálást eredményezhetnek.

Általában két típusú paddinget alkalmazunk: az "edge paddinget," amely a kép széléinél ismétli az utolsó pixelértékeket, és a "zero paddinget," amely egyszerűen nullával tölti ki a kép széléit. Jelen esetben a "zero padding"-et választottam, hiszen a véletlenszerű értékek torzítanák a felvételek térbeli információját.

Érdemes megjegyezni, hogy a padding hozzáadása növeli a kép méretét, és ezzel együtt a szükséges hardverterhelést is, azonban ez általában elfogadható kompromisszum, mivel a modell teljesítményének és pontosságának javulása jellemzően meghaladja a növekvő hardverigényeket.

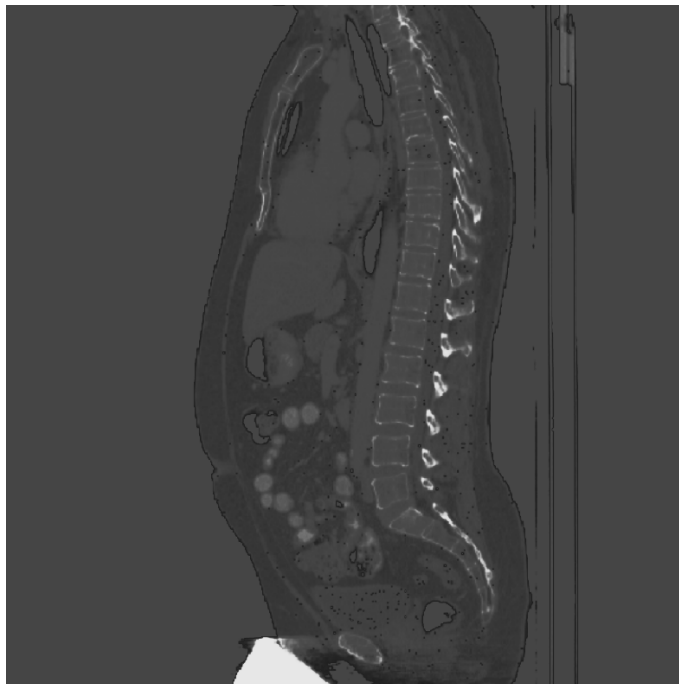


5.3. ábra. Ez az ábra a padding hozzáadását mutatja be az adott felvételhez

5.1.1.4. Szeletek ablakozása

A CT-szeletek Hounsfield-értékeket tárolnak. Ez lehetővé teszi a képfeldolgozás folyamatában az ablakozás használatát. Általánosságban az emberi csigolyák Hounsfield-értéke 400 és 1800 között van [16]. Ezen értékintervallumot használjuk a szeletek ablakozáshoz, azaz minden pixel, amelynek Hounsfield-értéke ebben az intervallumban található, kiemelésre kerül. Fontos megjegyezni, hogy az ablakozási eljárás csupán kiemeli a csontokat, de a további értékeket az adott szelet egyéb információit nem törli, így megmarad a képek tér-

beli információja is. A javasolt hibrid pipeline, különösen a háromdimenziós *U-Net* modell előnyt kovácsol ezen információk felhasználásából.



5.4. ábra. Ezen az ábrán az ablakozás eredményeként kiemelt gerincet láthatjuk.

5.1.1.5. Szeletek mentése numpy formátumban

Az előfeldolgozási folyamat utolsó lépéseként a CT-szeletek információtartalmát numpy formátumban mentjük el. Ennek több indoka és előnye is van, amelyek a modell tanításának hatékonyságát és a reprodukálhatóságát egyaránt szolgálják. A numpy formátum előnye, hogy könnyen olvasható és írható, továbbá jól integrálódik a Python tudományos számításokat támogató ökoszisztémájába, ideértve a SimpleITK-t, és a PyTorch-ot is [7].

Az adatok tárolásakor gyakran használják a float16 (fél-lebegőpontos) formátumot, amely kevesebb tárhelyet igényel a float32 vagy float64 formátumokhoz képest, miközben megőrzi az adatok pontosságát a számunkra releváns tartományban. Ez a kutatás szempontjából kiemelkedően fontos, mivel az irodalomkutatásban bemutatott adatok alapján a *U-Net* kifejezetten hardverigényes, főleg háromdimenziós kontextus esetén.

Az előfeldolgozott adatok elmentése jelentős időmegtakarítást eredményezhet a modell tanítási fázisában, mivel az előfeldolgozási lépések nem kell, hogy minden egyes tanítási iteráció során megismétlődjenek. Ez az egységesség és a reprodukálhatóság szempontjából is előnyös: biztosak lehetünk benne, hogy minden kísérlet ugyanazon előfeldolgozott adathalmazon fog futni.

Az előfeldolgozott adatok tárolása tárhelyet igényel, de az olcsó tárhely és a számítási kapacitások növekedése miatt ez egyre kevésbé jelent problémát. Az időmegtakarítás, amit az előfeldolgozott adatok használata eredményez, gyakran felülmúlja a tárhelyköltségeket.

5.1.2. Háromdimenziós CT felvételek előkészítése

A háromdimenziós CT felvételek előkészítésének alfejezete két kulcsfontosságú részre bomlik. Az első részben bemutatom, hogy hogyan használom fel a kétdimenziós modell által szegmentált maszkokat a háromdimenziós adathalmaz előkészítésére. A második részben

pedig bemutatok egy módszert, aminek a segítségével redukálhatjuk a hálózat hardver igényét.

5.1.2.1. Előfeldolgozás a kétdimenziós modell segítségével

Az előfeldolgozás fázisában a kétdimenziós *U-Net* modell által generált maszkokat használjuk fel, hogy összeállítsuk a háromdimenziós *U-Net* modell adathalmazát. A felvételek beolvasása ugyan úgy történik, mint a kétdimenziós adathalmaz előfeldolgozása esetén 5.1.1.1. A különbség mindössze annyi, hogy mostmár nem kell figyelni az egyes szeletek orientációjára, hiszen mostantól a teljes felvételt egyben kezeljük. Mivel a háromdimenziós felvételek előfeldolgozása nagyon hasonló a kétdimenziós adathalmaz előkészítéséhez 5.1.1, az alfejezet csak azokat a folyamatok tárgyalja külön, amik a két folyamat során különböznek.

A felvételeken elvégezzük az ablakozást 5.1.1.4, illetve átméretezzük a felvételeket 5.1.1.2. Az alapvető transzformáció után beolvassuk a memóriába az adott felvétel kétdimenziós szegmentációját. A szegmentált kétdimenziós felvételeket újra összefűzzük egy háromdimenziós felvétellé. Ezek után egy algoritmust futtatunk a szegmentált képen, ami egy olyan téglatest koordinátáit adja meg, ami magába foglalja a teljes szegmentált gerincet. Mivel a két felvétel dimenziói megegyeznek, ezért a kapott téglatest koordinátái alapján kivághatjuk a számunkra érdekes téglatestet az eredeti felvételtől. Ez azért fontos, mert így redukálhatjuk a felvétel dimenzióit megtartva kellő mennyiségű térbeli információt amellet, hogy csökkentjük a felvétel méretét és ezzel együtt a hálózat tanításához szükséges hardverigényt. Gyakorlatilag egy attention mechanizmust építünk a pipeline-ba, amivel növelni kívánjuk a modell szegmentálási pontosságát 2.3.2.

Az így összeállított felvételt végezetül elmentjük numpy formátumban, hogy a folyamatot ne kelljen minden tanítás esetén lefuttatni. A pipeline szempontjából fontos kiemelni, hogy a háromdimenziós létrehozása függ a kétdimenziós neurális hálózat adathalmazától és magától a modelltől is, így ezt érdemes előbb implementálnunk.

5.1.2.2. Háromdimenziós felvételek feldarabolása a teljesítmény növeléséhez

A dolgozat több fejezetében is taglalja a *U-Net* modell kiemelkedően magas hardverigényét. Ez főleg háromdimenziós kontextusban korlátozhatja a végtermék futtatásának és a hálózat tanításának körülményeit. Éppen ezért az implementációs fázisban javasolok egy megoldást a bemeneti adatok méretének csökkentésére, hogy kisebb számítási igényt érzünk el hardver oldalon az eredmények minimális romlása mellett.

Tegyük fel, hogy a háromdimenziós felvétel rendre (x, y, z) koordinátarendszerben került eltárolásra. Ez azt jelenti, hogy bármely (x, y) definiál egy transzaxiális szeletet, azaz egy kétdimenziós felvételt. A z tengely értékein végigiterálva megkapjuk az adott felvétel minden transzaxiális szeletét. Ebből az információból következik, hogy a z tengely mentén feldarabolva a felvételt kisebb, térbeli darabokat kapunk a CT felvételtől. Ezeket a darabokat felhasználhatjuk a háromdimenziós hálózat tanításához, hiszen megfelelően nagy z érték mellett a felvételekben használható térbeli információt adunk a hálózatnak. Ez jelentős többletinformációt biztosít a kétdimenziós képekhez képest, így javíthatja a modell kimenetét. Mindezek mellett a

Fontos megjegyezni, hogy a feldaraboláshoz használt n érték nem feltétlenül osztja a z tengelyt. Ebben az esetben a kutatás során tárgyalt zéró padding módszer 5.1.1.3 helyett azt javaslom, hogy vegyük a kép utolsó teljes z méretű darabját. Ez a zéró padding-ellentétben több térbeli információt ad a modellnek a pontosabb szegmentációhoz.

5.2. Neurális hálózatok implementációja

A neurális hálózatok implementációjának mélyebb megértése nem csak a kódolási technikák és algoritmusok ismeretét igényli, hanem azt is, hogy tisztában legyünk a mély tanulásban rejlő elméleti alapokkal és azok gyakorlati alkalmazásával. Ebben a fejezetben nem elégszem meg azzal, hogy csak a felhasznált modellek technikai részleteit ismertessem; bemutatom azt is, hogy az implementáció során, milyen alapelvek, tudományos meglátások és kutatási eredmények vezettek a választott modellek kialakításához és alkalmazásához.

Ezen túlmenően, hangsúlyozom azoknak a kutatóknak és fejlesztőknek a fontosságát, akik a neurális hálózatok ezen területén eredeti munkát végeztek, és akiknek kódja vagy elméleti hozzájárulása nélkül a jelenlegi projekt nem jöhetett volna létre. Az általam eszközölt módosítások, fejlesztések és integrációs lépések bemutatása mellett külön kitérek arra is, hogy ezek a változtatások hogyan járulnak hozzá a projekt sikeréhez.

Az olvasó a fejezet végére egy átfogó képet kap arról, milyen technikai és elméleti kihívásokkal jár az ilyen jellegű modellek implementációja, és milyen megoldási stratégiákat alkalmaztam ezen kihívások leküzdésére.

5.2.1. Az ELEKTRONN keretrendszerből átvett U-Net modell

A mély tanulás és képfeldolgozás területén számos kész keretrendszer áll rendelkezésre, amelyek különböző modellek implementálását és alkalmazását teszik lehetővé. Ezen keretrendszerek közül az ELEKTRONN egy kiemelkedően hasznos eszköz, amely kifejezetten az orvosbiológiai képszegmentálásra specializálódott. Az általam végzett munkában azonban nem az ELEKTRONN keretrendszer egészét használtam fel, hanem csupán egy specifikus fájlját, amely magában foglalja a U-Net modell alapjait mind a 2D, mind a 3D változatban. Az ELEKTRONN keretrendszert Martin Drawitsch készítette a Max Planck Neurobiológiai Intézetben Münchenben. A projekthez kutatásaim során hivatalos publikációt nem találtam, a teljes projekt az alábbi linken érhető el: <https://github.com/ELEKTRONN/elektronn3/tree/master>.

A U-Net modell eredetileg Ronneberger és munkatársai által került bemutatásra az [22] publikációban. Bár Çiçek és kollégái [32] javasoltak egy tisztán 3D változatot az architektúrából, a jelenlegi implementációt Martin Drawitsch az eredeti implementáció alapján készítette, több javítással és módosítással kiegészítve. A kód alapjául Jackson Huang munkája szolgált, akitől Martin Drawitsch a <https://github.com/jaxony/unet-pytorch> linken található (2D) U-Net implementációt vette át. Jackson Huang publikációja lehetővé teszi a kétdimenziós modell egyszerűbb paraméterezését a felhasználók számára. Az általam használt változat főbb eltérései Jackson Huang kódjától Martin Drawitsch leírása alapján:

- Kétdimenziós adatok helyett háromdimenziós felvételekkel is működik, 5D tenzorok felhasználásával.
- Alapértelmezésben 3D konvolúciót, 3D poolingot stb. használ.
- Lehetőséget nyújt *planar_blocks* architektúra használatára 2D/3D konvolúciós hálózatokhoz.
- Frissítve lett a PyTorch 1.3 és Python 3.6 verziókhöz.

A fenti információk mellett szeretném megjegyezni, hogy az eredeti U-Net modellt és annak változatait számos kutató és fejlesztő munkája alapján adaptáltam a saját projektjeimhez. Ezen forrásokat a dolgozatban több helyen megjelöltem, publikációik nélkül a jelenleg tárgyalt pipeline sem jöhetett volna létre.

5.2.2. 2D U-Net implementációja

A 2D U-Net modell paraméterezését empirikus módon végeztem, a különböző beállításokat a későbbiekben összehasonlítom majd. A pipeline-ban integrált modell paraméterei a reprodukálhatóság kedvéért az alábbiakban olvashatók némi magyarázattal:

- **in_channels=1, out_channels=1:** Mivel szürkeárnyaltos képekkel dolgozom, egy bemeneti és egy kimeneti csatornát használtam. Ez egyszerűsíti a modell architektúráját és csökkenti a szükséges számítási erőforrásokat. A kimeneti csatornák számából az is jól látható, hogy a modell nem jelzi külön kategóriákkal a csigolyákat, csak csigolya és "egyéb" közt tesz különbséget. Ez csökkenti a tanításhoz szükséges hardverigényt, illetve jobb tanulást eredményez a kétdimenziós modell esetén, hiszen a hasonló csigolyákról komoly térbeli információ hiányában nehéz megmondani, hogy pontosan melyik csigolya látható a képen.
- **n_blocks=5:** A konvolúciós blokkok számát négynek választottam. Ahogy azt már a modell architektúrájának bemutatásánál ismertettem, minden encoder lépés során felezzük a kép méretét és egy információ mátrixot állítunk elő a kép tartalmáról. Az 5 felezés 512 pixeles oldalméret esetén az encoder végére 32x32 pixeles képeket állít elő, így ezzel a beállítással a hálózat képes megragadni a képek lokális és globális jellemzőit is.
- **start_filts=32:** A konvolúciós rétegek szűrőinek kezdeti számát 32-re állítottam, és ezt az értéket exponenciálisan növeltem a hálózat mélyebb rétegeiben.
- **up_mode='transpose', merge_mode='concat':** Az encoder szakaszban transzponált konvolúciót használtam, és a jellemzőtérképeket konkatenációval egyesítettem.
- **activation='relu':** Az aktivációs függvényként a Rectified Linear Unitot, vagy röviden ReLU-t használtam. Ez az aktivációs függvény jól skálázódik mély neurális hálózatokon. A ReLU nemcsak a tanulási sebességet gyorsítja fel azáltal, hogy segít kezelni a vanishing gradient¹ problémát, hanem a modell általánosító képességét is javítja.
- **normalization='batch':** A gyorsabb tanítás és jobb általánosító-képesség elérése érdekében batch normalizációt alkalmaztam.
- **attention=False:** Az attention mechanizmus aktiválásával lehetőséget biztosít a modell számára, hogy a módszer által fontosnak tartott területekre fókuszáljon. Sajnos a tanítások során ez a feature nem bizonyult hasznosnak, viszont lassította a modellt. A felesleges hardveres terhelés elkerülése érdekében kikapcsoltam ezt a feature-t.
- **dim=2, conv_mode='same':** Két dimenziós konvolúciókat használtam, és a 'same' padding módot alkalmaztam, hogy megőrizzem a jellemzőtérképek méretét a konvolúciós folyamat során.

A **BATCH méretet** 16-nak választottam, annak érdekében, hogy a modell képes legyen jobban általánosítani, és hogy elegendő variancia legyen az egyes BATCH-ek között. Továbbá, a **learning rate** paramétert 10^{-5} -re állítottam, mivel a kétdimenziós adathalmaz

¹A "vanishing gradient" probléma egy olyan jelenség neurális hálózatok mély rétegeiben, amely során a gradiensek a backpropagation folyamatában olyan kicsivé válnak, hogy az alsó rétegek súlyai gyakorlatilag nem frissülnek, ami jelentősen lassítja a tanulási folyamatot vagy akár meg is állíthatja azt.

különösen nagy, több mint 500 000 képet tartalmaz. Kisebb értékek mellett tapasztalataim szerint a modell már az első epoch végére túltanult (overfitting).

A modellt **512x512** pixeles képeken tanítottam. Ebben a felbontásban a képek informatívak és élesek, miközben a modern hardverek nyújtotta teljesítmény mellett a tanítás nem válik túlzottan hardverigényessé. Az ADAM optimalizálót² használtam, amely kiválóan alkalmas nagy adathalmazokon történő tanításhoz.

Veszteségfüggvényként a Dice Loss-t alkalmaztam, amely matematikailag a következő képlettel definiálható:

$$\text{Dice Loss} = 1 - \frac{2 \times |Y \cap P|}{|Y| + |P|}$$

ahol Y az eredeti címkék halmaza (maszk), és P a predikciók halmaza (modell output). A Dice Loss értéke 0 és 1 között változik, ahol az alacsonyabb érték jobb teljesítményt jelent. Ez a veszteségfüggvény elterjedten alkalmazott az objektum-detekció és szegmentáció problémáira, mivel figyelembe veszi az igaz pozitív predikciók és a valós címkék közötti átfedést.

5.2.2.1. 2D U-Net tanítása

A modell tanítási folyamatának lebonyolítását empirikus módon végeztem, hipotéziseket állítottam fel és ezeket próbáltam tesztelni. Az alábbi kérdések megvizsgálását tartottam lényegesnek a feladat szempontjából:

- A modell kisebb felbontású felvételeken jobban teljesít.
- A modellbe épített attention mechanizmus segít a pontosabb szegmentálásban.
- Ha a modell több epochon keresztül tanulna anélkül, hogy overfitting-et érzelnénk, jobb lesz az általánosító-képessége, mintha már az első epochban túltanulna.

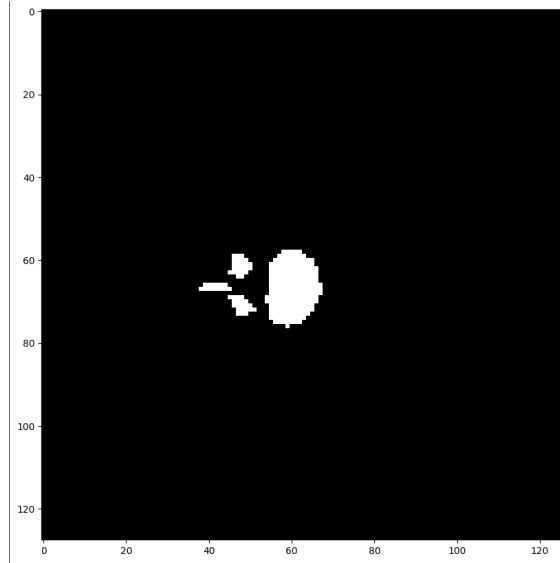
Ezen hipotézisek mindegyikét külön tanításokkal próbáltam bizonyítani. Összesen 5 paraméterezési kombinációt teszteltem, az eredmények az alábbiakban olvashatók.

5.1. táblázat. Különböző paraméterezésű 2D U-Net modellek teljesítménye

Modell	Felbontás	Attention mechanizmus	Learning rate	Dice score
2D U-Net	512x512	off	5×10^{-5}	0.85
2D U-Net	512x512	on	5×10^{-5}	0.85
2D U-Net	512x512	off	1×10^{-5}	0.62
2D U-Net	256x256	off	1×10^{-4}	0.90
2D U-Net	128x128	off	1×10^{-4}	0.93

Az eredményekből egyértelmű választ kapunk a hipotézisekre, noha ezek nem egyeznek minden esetben az eredeti feltevésekkel. A várakozásoknak megfelelően a modell jobban teljesít alacsony felbontású felvételeken. Ez azért lehet, mert minél alacsonyabb a felbontás, annál kevesebb részlet látszik a felvételekből. A **128x128**-as felvételeken a maszk gyakorlatilag csak egy folt, a csigolya belső üregei és külső görbületei eltűnnek. A modellnek így sokkal egyszerűbb dolga van a szegmentálás elvégzése közben. Sajnos ez a fajta pontosság nem mozdítja előre a kutatást, mert ha 128 vagy 256 pixel oldalméretű képekből próbáljuk kivágni a gerincet még rosszabb felbontást kapunk, így az eredmény gyakorlatilag használhatatlan lesz.

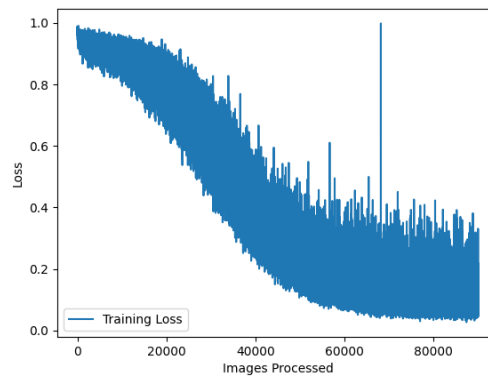
²Az ADAM (Adaptive Moment Estimation) egy olyan optimalizáló algoritmus, amely adaptív tanulási rátákat és momentumszámítást alkalmaz.



5.5. ábra. Egy 128x128 pixel felbontású maszk

Az attention mechanizmus hatásainak megjósolása sajnos nem volt olyan eredményes, mint bemenet felbontására vonatkozó hipotézis. Az ELEKTRONN keretrendszer a Jo Schneider és kollégái által javasolt [24] grid attention mechanizmust implementálja. A kutatás során felhasznált adathalmaz esetén a funkció bekapcsolása nem mutatott semmilyen javulást. Erre egy magyarázat lehet az, hogy a felvételeken a csigolyák mindig egy elég szűk területen belül találhatóak a kép közepe körül, ezt pedig a modell attention mechanizmus nélkül is képes volt megtanulni.

Végül, de nem utolsó sorban érdemes említést tenni az adathalmazon folytatott tanításról. A rendelkezésre álló adathalmaz, amely több, mint 500,000 kétdimenziós szeletet tartalmaz túl nagyra bizonyult a U-Net komplexitásához képest. A modell az első epoch közepére eljutott egy stagnálási pontra, ahonnan az új felvételek már nem tudták segíteni a modell pontosabb predikcióit. Az utolsó hipotézis az volt, hogy ha több felvételt mutatunk a modellnek, mielőtt beégetjük a súlyait, pontosabb szegmentációt fogunk kapni. Sajnos a learning rate állításával csak részlegesen tudtuk elérni a kívánt eredményt. A paraméter 5×10^{-5} -re való csökkentésével kapjuk a legjobb teljesítményt, de a batch loss értékek javulása így is leállt az első epoch 75%-ánál. A paraméter további csökkentése 1×10^{-5} -re pedig teljesen megakasztotta a tanulási folyamatot, hiába nézte meg a modell a felvételeket több epoch-ban, az általánosító-képességét teljességgel elvesztette.



5.6. ábra. Tanítás 10^{-5} -re állított learning rate-el.

5.2.2.2. A tanítás hardverigénye és futási ideje

A hálózat tanítását egy RTX 3090 grafikus kártyán végeztem. A tanításhoz így 24 GB VRAM állt rendelkezésre, amire a modellnek nem volt teljes egészében szüksége. Az adatok betöltése után a gradiensekkel együtt a modell 11.4 GB VRAM-ot használt. Ez modernebb hálózatokhoz képest igen jelentős hardverigényt jelent egy kétdimenziós modell tanításához 16-os batch méret mellett. A nyers adatok alátámasztják az irdoalomkutatást, miszerint az eredeti *U-Net* modell komoly hardverigényekkel rendelkezik. A modellek tanítása között az idő függvényében nehéz lett volna következtetéseket levonni, mivel a paraméterezés nem változtatta olyan nagy mértékben a hálózat számítási igényeit 512×512 pixeles kontextusban. Az attention mechanizmus bekapcsolás nagyjából 40%-al lassította a modell tanítását a megnövekedett számítási igény miatt.

Fontos kiemelni, hogy a hardveren a tanításon kívül más folyamatok is futottak a tanítás mellett, így a fejezetben leírt adatok nem olyan pontosak, mintha egy elkülönített környezetben készítettünk volna benchmarkot.

5.2.3. 3D U-Net implementációja

A 3D modell tanítási folyamata jelentősen összetettebb volt a 2D modellnél. A tanítás megkezdése előtt fontos, hogy tudatosan átgondoljuk miben különbözik a két modell, hogy ki tudjuk használni az új dimenzió adta lehetőséget. Ahhoz, hogy valóban három dimenzióban elemezze az adatokat a modell, el kell végeznünk néhány változtatást az architektúrán, az adathalmazon és a veszteségfüggvényen. A legkézenfekvőbb változtatás ezek közül nyilván az adathalmaz, hiszen mostmár a modellünk 4D tenzorok (batch méret, csatornák száma, szélesség, magasság) helyett 5D tenzorokat vár, hiszen bevezetésre kerül a "mélység" a bemeneti paraméterek listájába.

A modell architektúráját is meg kell változtatnunk annak érdekében, hogy kezelni tudja a 3D-s kontextust. Ugyan a kétdimenziós függvényeink (konvolúció, max-pooling, stb.) is működne 3D-ben, a modell gyakorlatilag újra szeletekre bontaná a bemenetet és egyben értékelné ki sok 2D-s felvételt. Ahhoz, hogy ezt elkerüljük, 3D konvolúciót, 3D max-poolingot stb. kell használnunk. A lényeg az, hogy sehol ne veszítsük el a háromdimenzió nyújtotta extra térbeli információkat.

Ehhez természetesen a veszteségfüggvényt is meg kell változtatnunk, hiszen a jelenleg implementált dice loss 4D tenzorokat vár 5D tenzorok helyett. A függvény implementálásánál érdemes lehet odafigyelni arra is, hogy a háromdimenziós kontextus megtartásával számoljunk, mivel az így kapott eredmény eltérhet 2D-s szeletek veszteségének átlagától.

A modell paramétereit a már jól ismert kétdimenziós modell paramétereit 5.2.2 alapján választottam meg. Az attention mechanizmust a 2D-s modellnél említett érvek miatt továbbra sem használtam, a pipeline-nak hála az attention mechanizmus egyébként is fellelhető az előállított adathalmazban. Azokat a paramétereket, amik nem változtak nem sorolom fel még egyszer, viszont van paraméter, amit meg kell változtatni:

- **batch_size:** A batch méreteket a 3D-s hálózat elrettentő hardverigénye miatt 16-ról 8-ra kellett redukálni.
- **dim:** Az architektúrális változásokat az ELEKTRONN keretrendszerből származó modellnek hála nem nekünk kell implementálni, csupán a dim kapcsolót kell 3-ra állítani, hogy a modell 3D-s konvolúciókat, max-poolingot stb. használjon.
- **learning rate:** Az adathalmaz mérete jelentősen csökkent, hiszen mostmár nem számít minden felvétel külön adatnak. A felvételek kisebb szeletekre darabolásával sikerült nagyjából 10x-es adatmennyiséget előállítanunk, így 10,000 3D-s felvétellel

dolgozhatunk. Ez a méret viszont nem indokolja a learning rate további csökkentését, így ezt visszaállítottam a default 1×10^{-4} -re.

A 3D-s kontextus miatt a veszteségfüggvény is lecsrélttem részben. A tanítás során sima dice veszteségfüggvény mellett a dice veszteségfüggvény és a Bináris kereszt-entrópia (binary cross-entropy) 1:1 arányban súlyozott verzióját is leteszteltem a hálózat szegmen-tációs pontosságának javítása érdekében.

A Binary Cross-Entropy (BCE) veszteségfüggvény matematikailag a következőképpen definiálható:

$$\text{BCE Loss} = -\frac{1}{N} \sum_{i=1}^N [y_i \cdot \log(p_i) + (1 - y_i) \cdot \log(1 - p_i)]$$

ahol N a minták száma, y_i az i -edik minta valós címkéje (0 vagy 1), és p_i az i -edik minta predikciója (a modell kimenete, 0 és 1 közötti érték). A BCE értéke 0 és ∞ között változik, ahol az alacsonyabb érték jobb teljesítményt jelent. Mivel a dice veszteségfüggvény értéke 0 és 1 közé esik, ezt az értéket kétkategóriás osztályozási probléma esetén sigmoid kimeneti függvény mellett használják. A BCE kiválóan alkalmazható bináris osztályozási feladatokra, mivel a logaritmikus büntetés segít a fals pozitív és fals negatív aktivációk szűrésében.

A Dice Loss megfelelően kezeli az osztályok egyenlőtlen eloszlását, ami gyakori az orvosbiológiai képszegmentációban, mivel az érdeklődési terület gyakran csak a kép kis részét foglalja el. Ezzel szemben a BCE minden egyes pixelt függetlenül kezel, viszont ezáltal érzékeny lehet az osztályok kiegyensúlyozatlanságára. A két függvény kombinálása segít enyhíteni az osztályegyensúly hiánya okozta problémákat, miközben redukálja a fals pozitív és fals negatív predikciókat is.

5.2.3.1. 3D U-Net tanítása

A modell tanítása előtt kétdimenziós modell tanításából levont következtetések alapján hipotéziseket állítottam:

- A modell alacsonyabb felbontású felvételen jobban teljesít.
- A pipeline attention mechanizmushoz hasonló előfeldolgozása segíti a pontosabb szegmentációt
- A Dice veszteségfüggvény és a BCE veszteségfüggvény 1:1 arányú alkalmazása jobb pontosabb szegmentációhoz vezet, mint a Dice veszteségfüggvény magában

Ezen hipotézisek mindegyikét külön tanításokkal próbáltam bizonyítani. Összesen 6 paraméterezési kombinációt tetsteltem, az eredmények az alábbiakban olvashatók.

5.2. táblázat. Különböző paraméterezésű 3D U-Net modellek teljesítménye

Modell	Felbontás	Pipeline	Veszteségfüggvény	Dice score
3D U-Net	50x256x256	igen	Dice + BCE	0.947
3D U-Net	50x256x256	igen	Dice	0.918
3D U-Net	50x256x256	nem	Dice	0.891
3D U-Net	50x128x128	igen	Dice	0.952
3D U-Net	50x128x128	nem	Dice	0.901
3D U-Net	100x128x128	igen	Dice	0.948

A tanítási folyamat újra alátámasztotta, hogy az alacsonyabb felbontású képeket a modell jobban pontosabban tudja szegmentálni. Ennek az oka az alacsony felbontásban elveszett részletekben rejlik, sok esetben a modellnek csak nagyobb területeket kell kijelölnie különösebb komplexitás nélkül. A pipeline által az adathalmazba integrált attention mechanizmus az elvártaknak megfelelően működött, a modell minden alkalommal jobban teljesített, amikor olyan felvételeken futtatuk, amiken általánosságban csak a gerinc látszódt a 2D U-Net által segített előfeldolgozásnak hála. Végeredményben pedig a dice és a BCE veszteségfüggvények közös használata is kifizetődött, hiszen összességében ez a modell teljesített a legjobban.

5.2.3.2. A tanítás hardverigénye és futási ideje

A modell tanítását 4 db RTX A5000 típusú GPU-n végeztem, így a felhasznált hardver összesen 96 GB VRAM-ot szolgáltatott. Ez sajnos kevés volt a 100 szeletszámú **256x256** pixeles felvételeken való tanításra, így ezt a modellt nem tudtam előállítani. A kisebb bemenettel rendelkező **50x256x256** is majdnem teljes egészében felhasználta a rendelkezésre álló VRAM-ot. Fontos megjegyezni, hogy nem használtam gradiens checkpointokat (gradient checkpointing), de ezzel a módszerrel - a tanítási sebesség csökkenése mellett - körülbelül a felére lehet csökkenteni a hálózat hardverigényét. Az alábbi táblázatban összehasonlítottam a modellek tanításához szükséges erőforrásokat és a felvételek feldolgozásának sebességét. Az utolsó oszlopban átlagosan 500 pixelnek tekintetem a felvételek mélységét, így kaptam az eredményeket.

5.3. táblázat. A 3D U-Net különböző verzióinak tanításához szükséges erőforrás a bemenetek méretének függvényében

Felbontás	Szeletek száma	Veszteségfüggvény	VRAM	Felvétel/min
256x256	50	Dice + BCE	88.32 GB	34
256x256	50	Dice	88.32 GB	37
128x128	50	Dice	-	57
128x128	100	Dice	-	57

5.3. Hibrid pipeline felépítése

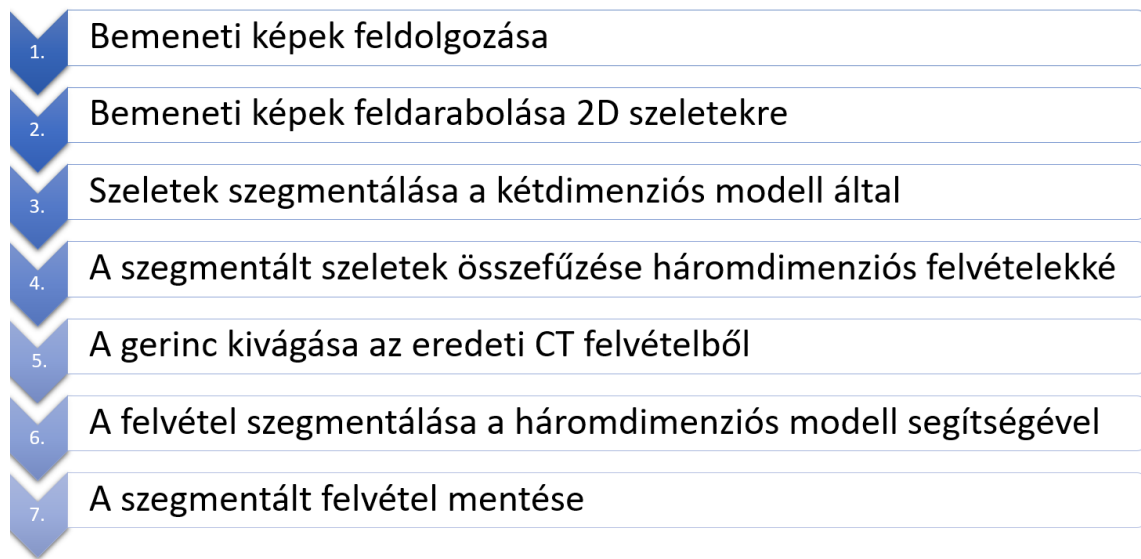
A fejezet, illetve a sokadik fejezet végére eljutottunk a kutatás eredményéhez, egy hibrid 2D-3D U-Net modellhez. A korábbi fejezetekben már bemutattam az összes komponens működését, most pedig ebben az alfejezetben részletesen bemutatom a pipeline főbb lépéseit, amelyek a következők:

5.3.1. Bemeneti képek feldolgozása

Az algoritmus először a bemeneti CT felvételeket normalizálja és kontraszt-szabályozást végez, hogy optimalizálja a neurális hálózatok teljesítményét.

5.3.2. Bemeneti képek feldarabolása 2D szeletekre

A bemeneti felvételek feldolgozása után a teljes 3D CT felvételek feldarabolása következik, így a felvételből kétdimenziós transzaxiális szeletek jönnek létre. Ez lehetővé teszi a kétdimenziós U-Net modell alkalmazását.



5.7. ábra. A kutatás alatt kifejlesztett pipeline lépései.

5.3.3. Szeletek szegmentálása a kétdimenziós modell által

Az egyes 2D szeletek szegmentálásra kerülnek a kétdimenziós U-Net modell segítségével. Ebben a lépésben az a kétdimenziós U-Net modell minden felvétel minden pixeléről eldönti, hogy a *háttér* vagy a *csigolya* osztályokba tartozik.

5.3.4. A szegmentált szeletek összefűzése háromdimenziós felvételekké

A szegmentált 2D szeletek összefűzésre kerülnek egy 3D felvétellé. Ebben a lépésben fontos a szeletek közötti sorrend és a szeletek térbeli információjának megőrzése, mivel ezek az adatok a háromdimenziós szegmentáció alapjául szolgálnak majd.

5.3.5. A gerinc kivágása az eredeti CT felvételből

Az eredeti 3D CT felvételből a gerinc kivágásra kerül a kétdimenziós modell kimenete alapján, hogy kizárólag a háromdimenziós modell bemenete kizárólag a szegmentálni kívánt anatómia struktúrákat tartalmazza. Ez javítja a háromdimenziós modell szegmentációs pontosságát az attention mechanizmus elveihez hasonlóan.

5.3.6. A felvétel szegmentálása a háromdimenziós modell segítségével

Az előkészített 3D CT felvétel további szegmentálásra kerül a háromdimenziós U-Net modell segítségével. Ebben a lépésben a modell elvégzi a korábban kiemelt anatómiai struktúrákat, azaz elkülöníti azokat a háttértől.

5.3.7. A szegmentált felvétel mentése

Végül a szegmentált 3D felvétel mentésre kerül, így más alkalmazásokban is elemezhetjük a felvételeinket.

6. fejezet

Teljesítményértékelés

6.1. A dolgozatban tárgyalt modellek eredményei

A javasolt módszer egy 2D és egy 3D CNN betanítását követeli meg. A pipeline eredményeinek jobb értékelése érdekében a kutatás során úgy döntöttem, hogy egy mezei *U-Net* modellt választok mind a 2D-s, mind a 3D-s szegmentációs feladatokra. Noha az architektúra már számtalanszor bizonyított orvosbiológia szegmentációban, a modernebb változatai köztudottan jobb teljesítményt nyújtanak és kisebb hardverigénnyel rendelkeznek. Pontosan ezért a pipelineba olyan modelleket választottam, amik nem érik el a modern hálózatok teljesítményét, így ha mégis jól teljesítenek, az bizonyítja a kutatás eredményességét.

A kutatás során a 2D modell teljesített a leggyengébben. További finomhangolással a modell pontosab szegmentációra is képes lehetne, de a pipeline szempontjából a csigolyák pozíciójának körülbelüli eltalálása kielégítő volt. A modell segítségével sikeresen előállítottam egy adathalmazt, amelybe a pipeline segítségével az attention mechanizmushoz hasonló elveket vezettem be.

A kutatás célja a 3D modell szegmentálási pontosságának maximalizálása volt. A legjobban teljesítő modell **50x256x256**-os inputot vár a pipeline-tól (ami már egy kivágott felvétel a gerincről). A modell 0.947-es dice score-al teljesített a tesztelési adathalmazon,

6.2. Metrikák és eredmények

A javasolt technológia nemcsak a szegmentálás pontosságát optimalizálja, hanem a modell teljesítményét és hardverigényét is. A két modell összehasonlítása ilyen téren az alábbiakban olvasható: A modelleket ugyanazon az RTX 3090 grafikus kártyán teszteltem, az

6.1. táblázat. A dolgozatban vizsgált modellek teljesítményének összehasonlítása

Modell	Felbontás	VRAM	Futásidő	Dice score
2D-3D Pipeline	256x256	12.3 GB	19.1s	0.947
TotalSegmentator	512x512	12.9 GB	26.2s	ismeretlen csigolyákra

alább látható kimenetek elkészítése közben.

6.3. Szegmentációs maszkok összehasonlítása

A kutatás során a pipeline és a 3D modell sok tesztelésen, fejlesztésen esett át. Végeredményben a modell a TotalSegmentator-hoz hasonló szegmentációs képességek szintjére fejlődött, annak ellenére, hogy egy jelentősen egyszerűbb U-Net architektúrát használ. A modell szegmentációs pontossága hiperparaméter optimalizálással, illetve más veszteségfüggvények implementálásával fokozható lehet. Az alábbi ábrán látható az eredeti maszk, a TotalSegmentator által készített szegmentáció, és a dolgozatban javasolt modell. Észrevehető, hogy az anatómiai struktúrákat rendre eltalálta a modellünk, de az élek szegmentálásán még lehetne javítani, mert a kimenet néhol zajos.



(a) A javasolt hibrid pipeline-nal előállított címkézés (b) A TotalSegmentator által előállított címkézés (c) Kézi címkézés

6.1. ábra. A címkézések összehasonlítása 3D-ban

7. fejezet

Továbbfejlesztési lehetőségek

A kutatás közben számos elképzelés fogalmazódott meg bennem a technológia továbbfejlesztésre. Ezen lehetőségek közül szeretném bemutatni azt a kettőt, amelyek a kutatás során szerzett tapasztalataim alapján a legnagyobb potenciállal rendelkeznek.

7.1. Univerzális neurális hálózatok

Először is, a jelenlegi pipeline-ból egy teljes neurális hálózat építése nemcsak a számítási hatékonyság növelését eredményezheti, de a modell összeteljesítményét is javíthatja. A pipeline egyes elemei közötti szorosabb integráció lehetővé teheti a hálózat számára, hogy jobban megértse a komplex interakciókat és jellemző függőségeket. Egy ilyen integráció lehetőséget biztosíthat arra is, hogy az end-to-end tanulás során a modell finomítsa a jellemzők kinyerését és a döntési logikát, anélkül, hogy emberi beavatkozásra lenne szükség. A modell legnagyobb erőssége a pipeline megoldással szemben az lehet, hogy a kétdimenziós és a háromdimenziós U-Net modellek optimalizálhatják egymás kimenetét, ezzel összességében jobb eredményt teremthetnek a teljes hálózat számára.

7.2. Csigolya-specifikus szegmentálás

Másodszor, a csigolyák külön-külön történő szegmentálása újabb dimenziót nyithat meg az analízisben. A csigolyák, mint egyedi osztályok kezelése lehetővé teheti a modell számára, hogy jobban megértse a csigolya-specifikus jellemzőket és patológiákat. Ugyanakkor, a csigolyák gerinc szerinti csoportosítása (pl., lumbalis (L), sacralis (S), stb.) egy másik érdekes alternatíva lehet, amely az anatómiai és funkcionális összefüggések felderítését célozza. A külön csoportokba történő osztályozás, például a lumbalis és a sacralis csigolyák csoportjai, potenciálisan segíthet a modellnek abban, hogy jobban általánosítson új, ismeretlen adatokra, miközben a kimenet továbbra is megőrzi a gerinc szerkezeti és funkcionális integritását.

Összességében, ezek a továbbfejlesztési irányok nem csak a modell pontosságát és hatékonyságát növelhetik, de új, eddig fel nem tett kérdésekre is választ adhatnak, amelyek további kutatásokhoz vezethetnek.

8. fejezet

Összegzés

A dolgozatban taglalt kutatás célja egy olyan módszer kidolgozása, melynek segítségével egy egyszerű *U-Net* modell betanítható az igen összetett TotalSegmentator szintjére, amely a *U-Net* modell egy továbbfejlesztett változatát, az *nn-UNet* architektúrát alkalmazza. A kutatás során egy hibrid 2D-3D pipeline előállítását javasoltam a cél elérésére, amely a kétdimenziós U-Net szegmentációit felhasználva állítja elő a 3D U-Net számára a bemeneti adatokat, az adathalmazba építve az attention mechanizmust. Az architektúra jelentősen jobb fejlődést mutat a sima *U-Net* modell által előállított szegmentációkhoz képest, és a tesztelési adathalmazon 0.947-es dice hasonlósági együtthatóval teljesített. Ez magasabb, mint az összehasonlításra választott TotalSegmentator általános dice hasonlósági együtthatója. Fontos megjegyezni, hogy a TotalSegmentator dice hasonlósági értéke sokkal több anatómiai struktúra szegmentálása során keletkezik, mint a csigolyák száma. A tesztelés során megfigyelhettük, hogy a TotalSegmentator modell pontosabban szegmentál az élek környékén, így kevesebb zaj keletkezik a végső szegmentáción, mint a hibrid 2D-3D pipeline kimenetén.

Összeségében a javasolt módszer az elvárásoknak megfelelően teljesített. A pipeline egy modernebb, komplexebb CNN-el párosítva pedig akár a TotalSegmentator modellnél is pontosabb teljesítményt nyújthat.

Köszönetnyilvánítás

Ezúton szeretném megköszönni konzulenseim, Révy Gábor és dr. Hullám Gábor segítségét a kutatás elvégzésében. Köszönöm nekik a folyamatos konzultálási lehetőséget és rendelkezésreállást. Továbbá köszönet a TotalSegmentator [29] fejlesztőinek az adathalmaz és a modell publikálásáért, és a Verse2019 [25][11][14] adathalmaz létrehozóinak, amely sokat segített a kutatást megelőző, de az adott feladathoz kapcsolódó fejlesztéseimben.

Irodalomjegyzék

- [1] Junaid Ahmad–Muhammad Sajjad–Irfan Mehmood: Volumetric medical image segmentation: A 3d deep learning approach. 2019.
- [2] Vijay Badrinarayanan–Alex Kendall–Roberto Cipolla: Segnet: A deep convolutional encoder-decoder architecture for image segmentation, 2016.
- [3] R. Beare–B.C. Lowekamp–Z. Yaniv: Image segmentation, registration and characterization in r with simpleitk. *Journal of Statistical Software*, 86. évf. (2018) 8. sz.
- [4] Jayanta Kumar Biswas–Masud Rana–Anindya Malas–Sandipan Roy–Subhomoy Chatterjee–Sandeep Choudhury: Effect of single and multilevel artificial intervertebral disc replacement in lumbar spine: A finite element study. *The International Journal of Artificial Organs*, 45. évf. (2022) 2. sz., 193–199. p.
URL <https://doi.org/10.1177/03913988211001875>. PMID: 33706581.
- [5] Q. Dou–L. Yu–H. Chen–Y. Jin–X. Yang–J. Qin–P. A. Heng: 3d deeply supervised network for automated segmentation of volumetric medical images. *Medical Image Analysis*, 2017.
- [6] Rafael C. Gonzalez–Richard E. Woods–Steven L. Eddins: *Digital Image Processing Using MATLAB*. 2004, Pearson Education, Inc.
- [7] Charles R. Harris–K. Jarrod Millman–Stéfan J. van der Walt–Ralf Gommers–Pauli Virtanen–David Cournapeau–Eric Wieser–Julian Taylor–Sebastian Berg–Nathaniel J. Smith–Robert Kern–Matti Picus–Stephan Hoyer–Marten H. van Kerkwijk–Matthew Brett–Allan Haldane–Jaime Fernández del Río–Mark Wiebe–Pearu Peterson–Pierre Gérard-Marchant–Kevin Sheppard–Tyler Reddy–Warren Weckesser–Hameer Abbasi–Christoph Gohlke–Travis E. Oliphant: Array programming with NumPy. *Nature*, 585. évf. (2020. szeptember) 7825. sz., 357–362. p. URL <https://doi.org/10.1038/s41586-020-2649-2>.
- [8] Jie Huang–Wengang Zhou–Houqiang Li–Weiping Li: Sign language recognition using 3d convolutional neural networks. In *2015 IEEE International Conference on Multimedia and Expo (ICME)* (konferenciaanyag). 2015, 1–6. p.
- [9] Mobarakol Islam–V. Vibashan–Jeya Maria Jose Valanarasu–Navodini Wijetilake–Uppal Utkarsh: *Brain Tumor Segmentation and Survival Prediction Using 3D Attention UNet*. 2020. 05, 262–272. p. ISBN 978-3-030-46639-8.
- [10] Young Jae Kim–Bilegt Ganbold–Kwang Gi Kim: Web-based spine segmentation using deep learning in computed tomography images. *Healthcare informatics research*, 26. évf. (2020) 1. sz., 61–67. p.
- [11] H Liebl–D Schinz–A Sekuboyina–JS ..., Kirschke: A computed tomography vertebral segmentation dataset with anatomical variations and multi-vendor scanner data. *Sci Data*, 8. évf. (2021) 1. sz., 284. p.

- [12] Geert Litjens – Thijs Kooi – Babak Ehteshami Bejnordi – Arnaud Arindra Adiyoso Setio – Francesco Ciompi – Mohsen Ghafoorian és mások: A survey on deep learning in medical image analysis. *Medical image analysis*, 42. évf. (2017), 60–88. p.
- [13] B.C. Lowekamp – D.T. Chen – L. Ibáñez – D. Blezek: The design of simpleitk. *Frontiers in Neuroinformatics*, 7. évf. (2013) 45. sz.
- [14] M Löffler – A Sekuboyina – A Jakob – AL Grau – A Scharr – ME Hussein – M Herbell – C Zimmer – T Baum – JS Kirschke: A vertebral segmentation dataset with fracture grading. *Radiology: Artificial Intelligence*, 2020.
- [15] F. Milletari – N. Navab – S. A. Ahmadi: V-net: Fully convolutional neural networks for volumetric medical image segmentation. *2016 Fourth International Conference on 3D Vision (3DV)*, 2016.
- [16] A. Murphy – Y. Baba – J. Kube és mások: Windowing (ct). *Radiopaedia.org*, 2017. URL <https://radiopaedia.org/articles/52108>. Accessed on 25 Oct 2023.
- [17] Naveen: What is relu and sigmoid activation function? 2022. URL <https://www.nomidl.com/deep-learning/what-is-relu-and-sigmoid-activation-function/>. Accessed: 2023.11.01.
- [18] Keiron O’Shea – Ryan Nash: An introduction to convolutional neural networks, 2015.
- [19] Dzung L. Pham – Chenyang Xu – Jerry L. Prince: Current methods in medical image segmentation. *Annual review of biomedical engineering*, 2. évf. (2000) 1. sz., 315–337. p.
- [20] Sabina Pokhrel: Beginner’s guide to understanding convolutional neural networks. 2019. 9. URL <https://towardsdatascience.com/beginners-guide-to-understanding-convolutional-neural-networks-ae9ed58bb17d>. Accessed: 09.10.2023.
- [21] Vishal Rajput: Pooling layers in neural nets and their variants. 2022. 1. URL <https://medium.com/aiguys/pooling-layers-in-neural-nets-and-their-variants-f6129fc4628b>. Accessed: 09.10.2023.
- [22] Olaf Ronneberger – Philipp Fischer – Thomas Brox: U-net: Convolutional networks for biomedical image segmentation. In *MICCAI* (konferenciaanyag). 2015, 234–241. p.
- [23] Rathna S: Convolution neural networks (cnn). 2019. URL <https://medium.com/@rathna211994/convolution-neural-networks-cnn-b6fe90214b1e>. Accessed: 2023.11.01.
- [24] Jo Schlemper – Ozan Oktay – Liang Chen – Jacqueline Matthew – Caroline Knight – Bernhard Kainz – Ben Glocker – Daniel Rueckert: Attention-gated networks for improving ultrasound scan plane detection. 2020. URL <https://github.com/ozan-oktay/Attention-Gated-Networks>.
- [25] A Sekuboyina – AH Bayat – ME Hussein – M Löffler – BM Menze – JS ..., Kirschke: Verse: A vertebrae labelling and segmentation benchmark for multi-detector ct images. *Med Image Anal*, 73. évf. (2021), 102166. p. URL <https://arxiv.org/abs/2001.09193>. Epub 2021 Jul 22.

- [26] Nahian Siddique–Sidike Paheding–Colin P. Elkin–Vijay Devabhaktuni: U-net and its variants for medical image segmentation: A review of theory and applications. *IEEE Access*, 9. évf. (2021), 82031–82057. p.
- [27] Mount Sinai: Head ct scan, 2023.
URL <https://www.mountsinai.org/health-library/tests/head-ct-scan>. Accessed: 2023.10.23.
- [28] Attila Tanács: Orvosi képfeldolgozás/orvosi algoritmusok/távérzékelés, mikroszkópia. 2011., 87–96. p.
- [29] Jakob Wasserthal–Hanns-Christian Breit–Manfred T Meyer–Maurice Pradel-la–Daniel Hinck–Alexander W Sauter–Tobias Heye–Daniel T Boll–Joshy Cyriac–Shan Yang és mások: Totalsegmentator: Robust segmentation of 104 anatomic structures in ct images. *Radiology: Artificial Intelligence*, 5. évf. (2023) 5. sz.
- [30] Z. Yaniv–B.C. Lowekamp–H.J. Johnson–R. Beare: Simpleitk image-analysis notebooks: a collaborative environment for education and reproducible research. *Journal of Digital Imaging*, 31. évf. (2018) 3. sz., 290–303. p.
- [31] Yichi Zhang–Lin Yuan–Yujia Wang–Jicong Zhang: Sau-net: Efficient 3d spine mri segmentation using inter-slice attention. In *Medical Imaging with Deep Learning* (konferenciaanyag). 2020, PMLR, 903–913. p.
- [32] Özgün Çiçek–Ahmed Abdulkadir–Soeren S Lienkamp–Thomas Brox–Olaf Ronneberger: 3d u-net: Learning dense volumetric segmentation from sparse annotation. *MICCAI*, 9901. évf. (2016), 424–432. p.