



# Mélytanulás Házi Feladat

## Alvengers

*Frecska Hajnalka (C1MTMR)*

*Hugauf Dániel Bálint (F3G8I9)*

*Wágner Bánk (ANKNFJ)*

## Feladat címe: Model Ensemble for Medical Image Segmentation

### Feladat

In this project, you'll dive into the idea of using multiple models together, known as model ensembles, to make our deep learning solutions more accurate. They are a reliable approach to improve accuracy of a deep learning solution for the added cost of running multiple networks. Using ensembles is a trick that's widely used by the winners of AI competitions. Task of the students: explore approaches to model ensemble construction for semantic segmentation, select a dataset (preferentially cardiac MRI segmentation, but others also allowed), find an open-source segmentation solution as a baseline for the selected dataset and test it. Train multiple models and construct an ensemble from them. Analyse the improvements, benefits and added costs of using an ensemble.

2023/24. 1. félév

## A munka előzményei

### 1.1 Bevezető

Az MRI-képek alapján történő szív szegmentálás rendkívül fontos a szívbetegségek diagnosztizálásában és kezelésében. Azért választottuk ezt a témát, mert az adatfeldolgozás és gépi tanulás területén is izgalmas kihívásokat rejtegethet.

A szegmentáló modell létrehozásához a rendelkezésünkre bocsátott forrásokat tanulmányoztuk. Ezek az előkészületek és tanulmányok segítettek abban, hogy hatékonyabb módon tervezzük meg és fejlesszük ki ezt a modellegyüttest, amely a reményeink szerint alkalmas lehet egy szív MRI felvétel szegmentálására.

A dokumentum során bemutatjuk a modell felépítését, a felhasznált adatokat, valamint a tanítás és tesztelés folyamatát is.

### 1.2 Adathalmaz

#### 1.2.1 Adathalmaz elérhetősége

Az adathalmaz [1] a Dijoni egyetem kórháza által, valós orvosi vizsgálatok folyamán készült felvételek. Ezek teljesen anonimizálva lettek a helyi etikai bizottság által állított elvárásoknak megfelelően. Az adatbázisban megtalálható jó minőségű kórképek alkalmasak a gépi tanulási modellek fejlesztéséhez, egyértelmű értékeléséhez. 150 különböző páciens leletei találhatóak meg az adatbázisban, melyek 5 egyenlő méretű csoportra bomlanak (4 kóros, 1 kontrol). Ezen kívül minden lelethez hozzá tartoznak a következő egyéb információk: testsúly, testmagasság, diasztolés (szív elernyedés) és szisztolés (szív összehúzódás) fázispontok. Minden csoport egyértelműen meg volt határozva különböző fiziológiai paraméterek szerint. Ilyenek például a bal vagy jobb diasztolés térfogat vagy ejekciós frakció, az LV helyi összehúzódása, az LV tömege és a szívizom maximális vastagsága.

#### 1.2.2 Adatok típusa

Az adathalmazban minden pácienshez tartozik egy mappa. Ezekben találhatóak meg az .nii kiterjesztésű fájlok, ezek a NIfTI (Neuroimaging Informatics Technology Initiative) formátumához tartoznak, amely gyakran használt fájlformátum az orvosi képalkotás területén, különösen az MRI (Mágneses Rezonancia Képalkotás) és funkcionális MRI (fMRI) adatok tárolására. Ezek a fájlok háromdimenziós térfogat képeket (volumetrikus képeket) tárolnak, valamint hozzájuk kapcsolódó metaadatokat, például a voxel méreteket, képorientációt és egyéb fontos információkat, amelyek szükségesek a neuro képalkotási adatok megfelelő értelmezéséhez és elemzéséhez.

Minden pácienshez 5 NIfTI fájl is tartozik. Egy 4 dimenziós kép, amely 3 dimenziós képek szekvenciája, tehát az időbeli változások megfigyelésére alkalmas. Ezen kívül két, 3 dimenziós kép, amely az MRI során meghatározott időben készült pillanatképek. Illetve az ezekhez tartozó címkék (grund truth), amelyeket a tanítás során felhasználhatunk. A szegmentáláshoz 4 osztályt különít el: **bal kamra endokardiuma**, **bal kamra epikardiuma**, **jobb kamra endokardiuma**, **minden más** (háttér)

### 1.3 Elméleti összefoglaló a modellekről

#### 1.3.1 U-net

Az U-Net [2] egy konvolúciós neurális hálózat (CNN) architektúra, amelyet általában képszegmentálási feladatokhoz használnak. Az "U-Net" név az U alakú architektúrájából ered. Két fő részből áll: egy **kódoló**ból és egy **dekódoló**ból. A kódoló felelős a bemeneti kép kontextuális információinak rögzítéséért, míg a dekódoló a lokalizációért.

**Kódoló**

A kódolót konvolúciós és pooling rétegek sorozata, amelyek fokozatosan csökkentik a bemeneti kép térbeli felbontását, miközben növelik a jellemzők mélységét. A bemeneti információt absztraktabb reprezentációba tömöríti.

**Dekódoló**

A jellemzőtérképet az eredeti bemeneti felbontásig felskálazza. Felcserélt konvolúciókat (dekonvolúciót) használ az információ fokozatos visszanyeréséhez.

**Skip Connection**

A modell fontos része a skip connection. A kódolás során egyre alacsonyabb szinteken egyre több csatornával, de egyre kisebb felbontású képeket kapunk, ami miatt a pontos térbeli pozíciókról információt veszítünk. A skip connection arra való, hogy a dekóder ág konvolúciós blokkjai ne csak az alacsonyabb szintekről érkező, kevésbé részletes térbeli információt tartalmazó adatot kapja meg inputként, hanem a kódoló megfelelő szintjén lévő konvolúciós blokk kimenetét is. Az architektúra egy végső konvolúciós réteggel zárul, amely a szegmentációs maszkot állítja elő. Ez a réteg a szegmentálási feladatban szereplő osztályok számával megegyező számú kimeneti csatornával rendelkezik.

**1.3.2 LinkNet**

Az U-Net-hez hasonlóan a LinkNet-et [3] szemantikus szegmentálásra használják, ahol a cél a bemeneti kép minden egyes pixeléhez címkét rendelni.

**Kódoló-dekódoló architektúra**

A LinkNet az U-Net-hez hasonlóan kódoló-dekódoló struktúrát követ. A kódoló felelős a bemeneti kép jellemzőinek kinyeréséért, a dekódoló pedig a szegmentáló maszk létrehozásáért.

**Skip Connection**

Ezek a kapcsolatok a megfelelő rétegeket kötik össze a kódoló és a dekódoló között, hasonlóan az UNet ugrásos kapcsolataihoz.

**Residual blokkok a kódolóban**

A LinkNet Residual blokkokat használ a kódolóban, amelyeket a ResNet architektúrája ihletett. Ezek a blokkok segítenek megoldani az eltűnő gradiens problémát, lehetővé téve a hálózat hatékonyabb képzését, különösen mély architektúrák esetén.

**Spatial Attention modul**

A LinkNet tartalmaz egy Spatial Attention modult, amely segít a hálózatnak a bemeneti kép relevánsabb részeire összpontosítani a szegmentálás során. Ez a figyelem mechanizmus segíti a képen található hosszú távú függőségek megragadásában.

**Batch normalizálás**

A batch normalizáció a hálózat egészében az aktivációk normalizálására és a képzés felgyorsítására szolgál.

**1.3.3 Feature Pyramid Network (FPN)**

A Feature Pyramid Network (FPN) [4] egy másik népszerű konvolúciós neurális hálózat (CNN) architektúra.

**Több skálájú feature piramis**

Az FPN a több léptékű objektumfelismerés kihívását egy feature-piramis létrehozásával oldja meg. A piramis különböző térbeli felbontású jellemzőtérképekből áll, lehetővé téve a hálózat számára, hogy különböző léptékű információkat rögzítsen. Ez hasznos a különböző méretű objektumok felismeréséhez a képen.

**Bottom-up és Top-down architektúra**

Az FPN kombinálja az alulról felfelé irányuló (a hagyományos konvolúciós hálózathoz

hasznoló) és a felülről lefelé irányuló útvonalat. Az alulról felfelé irányuló útvonal magában foglalja a tipikus konvolúciós rétegeket, amelyek magas szintű jellemzőket vonnak ki a bemeneti képből. A felülről lefelé irányuló útvonal a nagy felbontású jellemzőknek a piramis alacsonyabb szintjeiről a magasabb szintekre történő felfelé mintavételezését és összevonását foglalja magába.

### Piramis-pooling

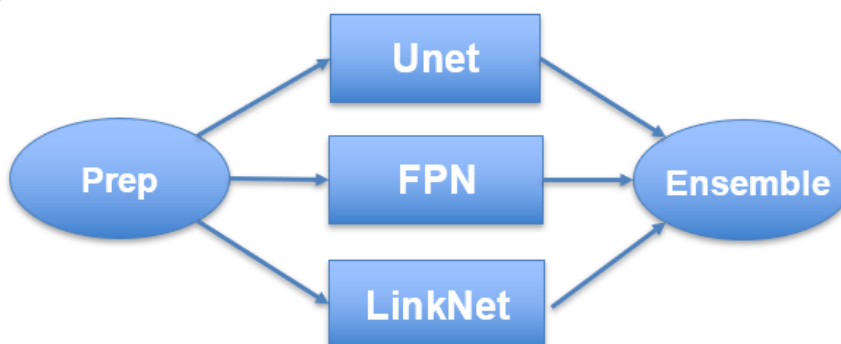
Ennek során a különböző piramisszintekről származó jellemzők összevonása és egymáshoz kapcsolása történik, ami a kép átfogóbb ábrázolását biztosítja.

## Az elvégzett munka ismertetése

### 2.1 Modellezés

#### 2.1.1 Pipeline

A modellezési pipelineban több elképzelésünk is volt a megoldásra a félév folyamán. Az első elképzelésünkönkor 5 konténert futtattunk volna. Az első konténerben történt volna az adatok letöltése és a preprocessálás. Ezt megvárva a következő 3 konténerben futottak volna a szegmentálási modelljeink. Ezek docker volume-on és network-on keresztül kommunikáltak volna. A előfeldolgozás lefutása után mindhárom konténer az inputjára megkapta volna az előfeldolgozott képeket. Ez után a szegmentálás elvégzése után az utolsó, ötödik konténer megkapta volna a modelleknek a kimeneteit és ezeket felhasználva elvégezte volna az ensemble-t. Vizuálisan így nézett volna ki ez a struktúra:



Ennél a pipeline futtatásnál több gondba is ütköztünk. Mivel a modelleket mind python 3.9 alatt teszteltük így az alap image-t is annak választottuk. Ez viszont egy 9GB körüli image ami az összes futtatásakor már túl sok helyet foglalt a gépeinken és nem voltunk képesek ezeket megépíteni.

Ilyenkor próbáltuk a szükséges image-t kisebbre cserélni, de a szűkített méretű imagekben különböző felhasznált könyvtárak nem futottak.

Ezek után próbáltuk az egész hálózatot bele szűkíteni egy darab konténerbe is. Viszont az adathalmaz nagysága (kb. 8GB) és a szükséges könyvtárak és file-okkal együtt már szintén túl kevésnek bizonyult a számítógépeink erőforrása.

Ezt a hálózatot viszont csatoltuk a leadott anyagok között, a **docker-compose.yaml** fájlban kell a service-k alatt a final-t kikommentezni és a train\_test-ről pedig leszedni a kommentet és kiadni a **docker-compose up** parancsot. A beadásban benne lévő hálózatot a colab-ból kiszedett könyvtárakkal hoztuk létre.

A másik (final) service pedig az általunk egy adott képre szóló predikciót visszaadó pipeline-t tartalmazza. Ez 1 konténerben elérhetővé teszi az összes modellt, amit egy gradio-s felületen lehet használni. A **docker-compose up** parancs kiadása után, hogy ha felépült az image a **localhost:7860** címen érhető el a IU.

### 2.1.2 Előfeldolgozás

A gépi tanulási feladatokban, mint például a szegmentálás, az adatokat szükséges előfeldolgozni olyan formátumba, amely kompatibilis a használt könyvtárakkal vagy keretrendszerekkel, lehetővé téve a hatékony adatkezelést, és az előfeldolgozást. A NumPy tömbök numerikus számításokra vannak optimalizálva, ami gyorsabb adatfeldolgozást tesz lehetővé, mint a voxel-információk közvetlen elérése az NIfTI-fájlokból. Ezért az előfeldolgozás elsődleges célja a NumPy tömbökké való konvertálás.

Tekintettel arra, hogy az adathalmaz összesen 150 páciens adatait tartalmazza, amelyből 150-et használunk fel tanításra, szükségesnek véltük az adatok augmentálását.

Ez segít bővíteni az adathalmaz méretét oly módon, hogy az eredetileg rendelkezésre álló adatokból újakat generálunk. A nagyobb adathalmazzal javíthatjuk a modellünk teljesítményét, mivel sokkal több és változatosabb adattal találkozunk a tanítás során.

**Adatelőkészítés lépései:** Adatok beolvasása → Adatok átkonvertálása numpy formátumba → Átméretezés elvégzése → Normalizálás

A képek mérete ezután (256,256,10) méretet vett fel. Mivel egy pácienshez egy nifti kép több szeletet jelent (10-13 darab), így az augmentációhoz nifti képként továbbítottunk, hogy ne keveredjenek össze az egyes páciensek képszeletei. Az augmentációs lépéseket belül numpy tömbként dolgoztunk velük. Az **augmentálás** során véletlenszerűen választottunk ki 25 páciens, amelyek képein különböző átalakításokat végeztünk el.

- Forgatás 90°-al a z tengely körül
- Forgatás 270°-al a z tengely körül
- Tükrözés az y tengelyre
- Tükrözés az x tengelyre
- Random zaj hozzáadása

Ugyanezeket megtettük a címkék esetén is, illetve a teszt adathalmazon is.

A végső training és test adathalmazt numpy tömbbő alakjuk át, majd elmentjünk .npz fájlokba. A tanítás során ezen adathalmazzal dolgozunk tovább.

### 2.1.3 Tanítás bemutatása

Az említett modellek mindegyikére elvégeztük a tanítást az előfeldolgozott adatokkal. A használt veszteségfüggvények a Dice Loss és Focal Loss

#### Dice Loss

A Dice Loss egy olyan veszteségfüggvény, amely segít az alakzatok pontosságának és az átfedés mértékének értékelésében, különösen a bináris vagy többszörös osztályozási problémák esetén.

**Dice Loss** =  $1 - (2 * \text{intersection}) / (\text{estimated} + \text{ground truth})$

- **Intersection:** A modell által becsült és a valóságos értékek közötti átfedés aránya.
- **Estimated:** becsült érték
- **Grund Thruth:** valós érték

Minél alacsonyabb a Dice Loss, annál jobb a modell teljesítménye az alakzatok megfelelő lefedésében.

```
dice_loss = sm.losses.DiceLoss(class_weights=np.array([0.25, 0.25, 0.25, 0.25]))
```

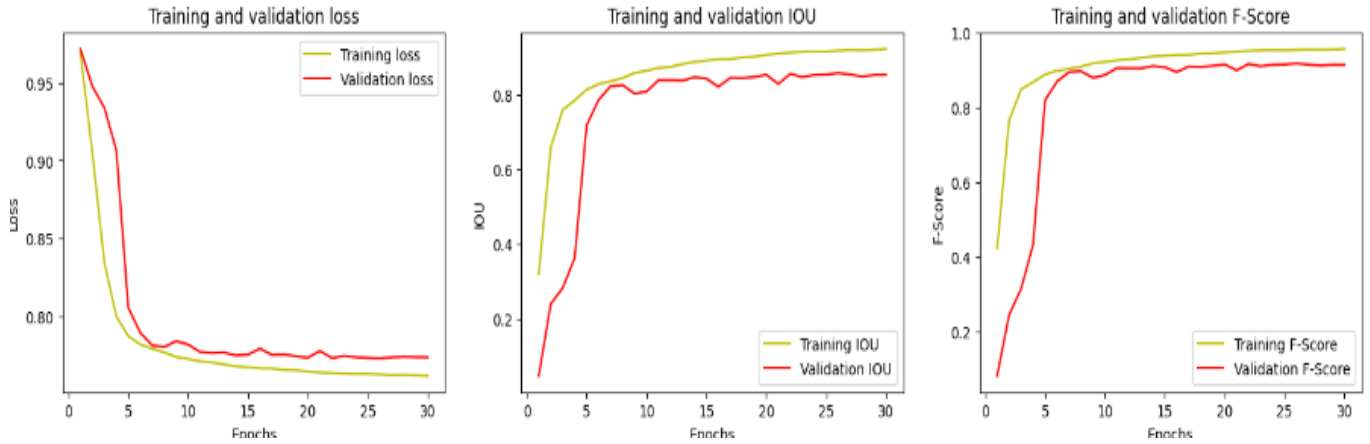
#### Focal Loss

A Focal Loss egy másik veszteségfüggvény, amely a gyengén osztályozott vagy túlterhelt adathalmazok esetén használatos, azzal a céllal, hogy kezelje a "háttérzajt", vagyis az osztályok közötti jelentős egyensúlyhiányt, tehát az osztályok egyike sokkal gyakoribb, mint a többi. A nehezebben osztályozható példákra fókuszál, csökkentve a könnyen osztályozható példák

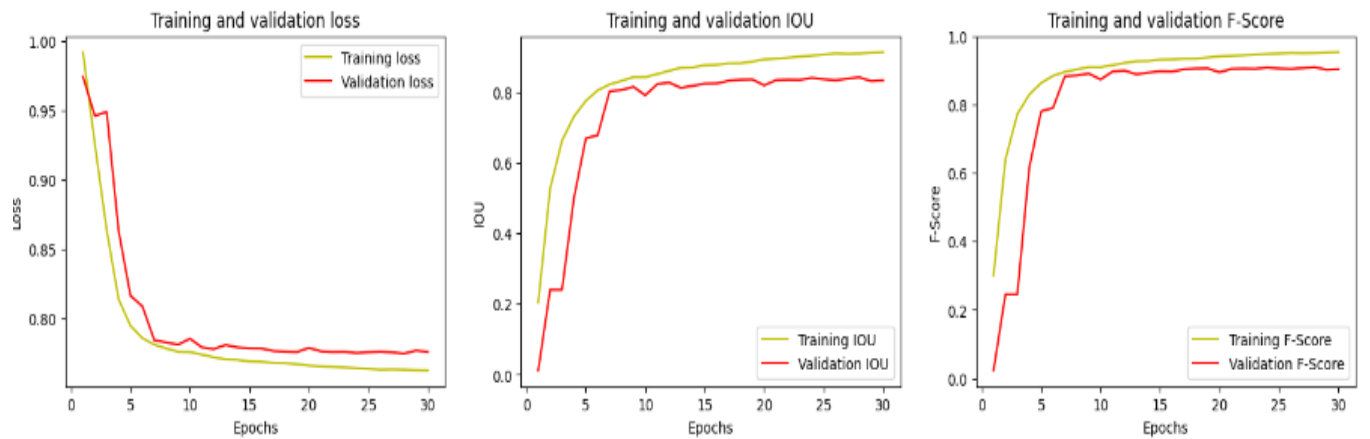
súlyosságát. Tehát amíg a dice lossnál egyenlő súlyokkal szerepelnek az osztályok, addig focal loss-nál a class imbalance-t próbálja kiegyenlíteni.

A tanítás kiértékeléséhez az IoU (Intersection over Union) és F1 értéket használtuk.

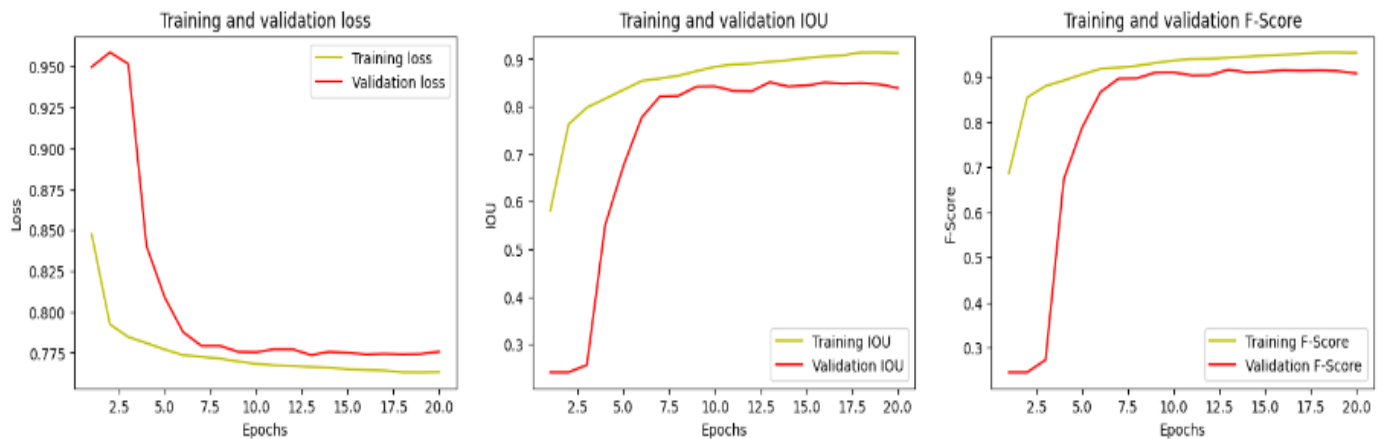
A következő ábrák a veszteség és a metrikák alakulását mutatják.



3. Kép: Loss, IoU és F1 Score alakulása a U-Net esetén



2. Kép: Loss, IoU és F1 Score alakulása a LinkNet esetén



1. Kép: Loss, IoU és F1 Score alakulása az FPN esetén

Model	size	batch size	l_rate	epochs	time	IoU	F1
UNet	290 MB	8	0.0001	30	15 min	0.8295	0.9459
LinkNet	250 MB	8	0.0001	30	15 min	0.817	0.9405
FPN	280 MB	8	0.0001	20	20 min	0.8225	0.9396

1 Táblázat: Az egyes modellek esetén használt paraméterek és a kiértékelés eredménye

### 2.1.4 Ensemble modell és kiértékelése

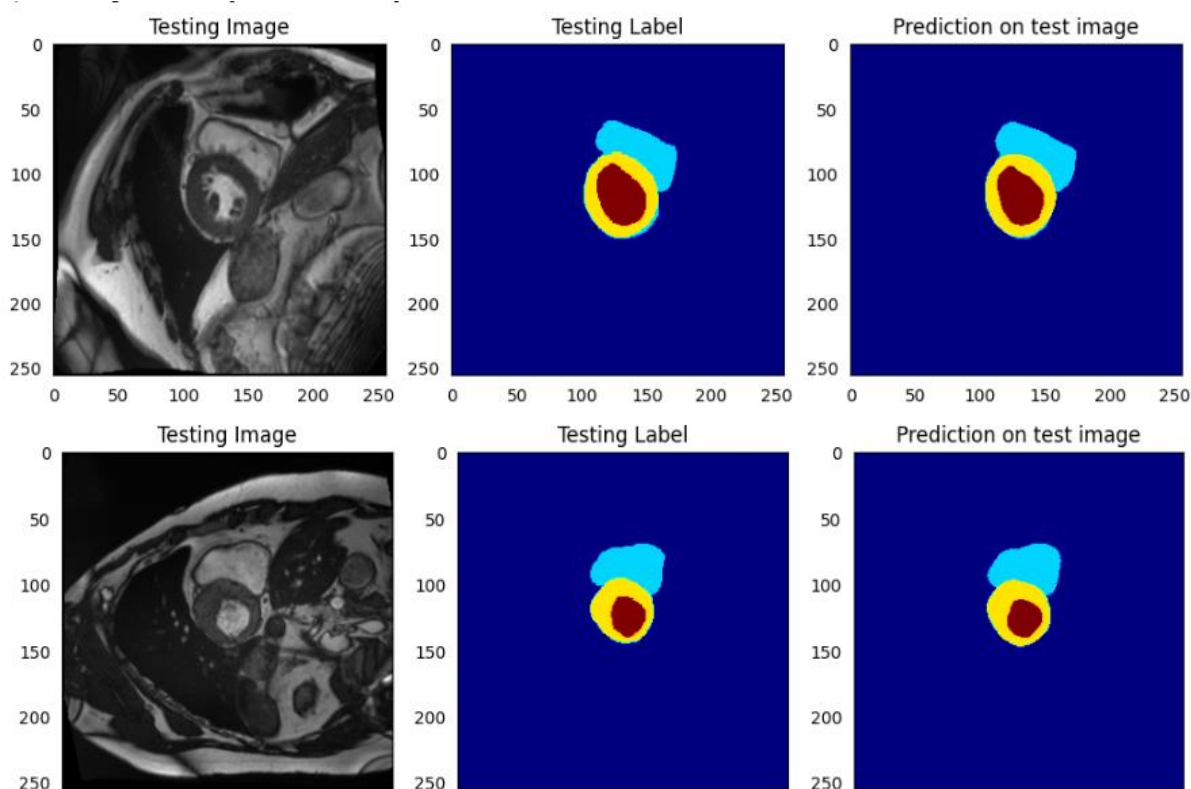
Az egyes modellek használata mellett lehetőség van egy modellegyüttes futtatására is. Ehhez a modellek predikciói különböző súllyal kerülnek be a végső predikcióba. Az optimálisához legközelebb eső súlyokat a 3 modellre (az iou alapján) random gridsearch-el keressük meg.

**IoU értékek, ha a modellek egyenlő súllyal szerepelnek:**

- IOU unet\_pre-trained = 0.8197997212409973
- IOU linknet\_pre-trained = 0.8025847673416138
- IOU fpn\_pre-trained = 0.809705376625061
- IOU az egyenlő arányban súlyozott ensemble esetén = **0.8218475**

**Random\_GridSearch**

- Max IOU az optimális súlyokkal: **0.824257** {'wt1': **0.97**, 'wt2': **0.28**, 'wt3': **0.64**}
- Best single model's IOU: **0.8197997**
- Ensemble teljesítménye 0.544% jobb, mint a legjobb modellnek egyedül (IOU értékben számolva: 0.00445 javulás)



4 Kép: A tesztelés során kapott ensemble modell eredménye vizuálisan szemlélítve 2 képen

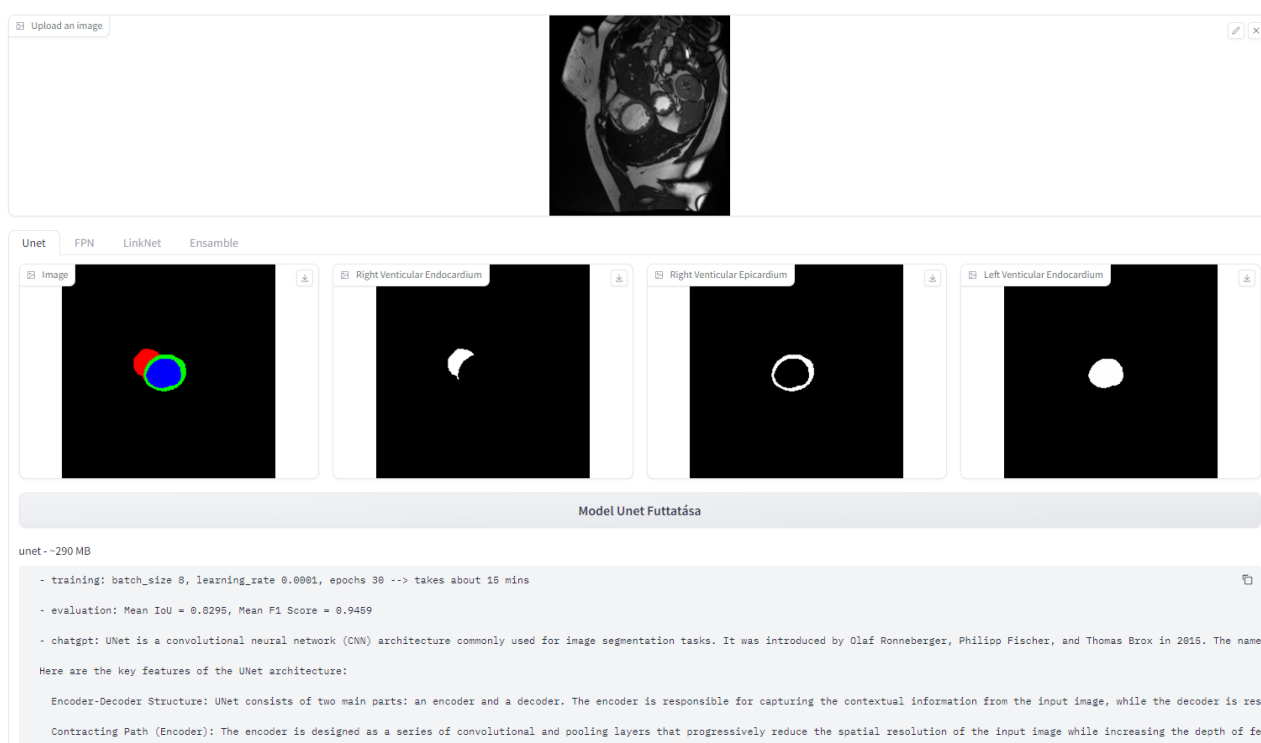
### 2.1.5 User Interface Gradioval

A **docker-compose up** parancs kiadása után, hogy ha felépült az image a **localhost:7860** címen érhető el a IU. Az oldalon meg fog jelenni egy felület, ahova a szegmentálásra kiválasztott képet kell feltölteni. Az alatta lévő részen a fülök közül választhatunk, hogy melyik modell alapján kapott predikciót szeretnénk megtekinteni. Majd a gomb megnyomásával visszakapjuk a prediktált képet, amit aztán akár le is tölthetünk. Az oldal legalján pedig a feladatunk leírása található, angol nyelven.

A szegmentálás során 4 osztályt különböztetünk meg:

- bal kamra endokardiuma
- bal kamra epikardiuma
- jobb kamra endokardiuma
- minden más (háttér)

Model Ensemble project for the Deep Learning course at BME by the team Alvengers



5 Kép: UI kinézete predikció során

## 2.2 Összefoglalás

A félév során a feladat célja szív MRI képek pontos és hatékony szegmentálása volt, ami az orvosi képfeldolgozásban egy fontos terület. A konkrét kihívás az volt, hogy olyan meglévő módszereket alkalmazzunk együttesen, amelyek képesek az eltérő szívstruktúrák, elkülönítésére a képeken. A munka során különböző mély tanuló architektúrákat használtunk, többek között a U-Net, LinkNet és az FPN-t. Nagy hangsúlyt fektettünk az adatok előfeldolgozására is, a modell teljesítményének a javításához pedig augmentációt használtunk. Az eredmények azt mutatják, hogy a modell jól teljesített a szegmentálási feladatban, ugyanakkor további finomhangolásra és az adatbázis méretének a növelésére is szükség van, hogy a jövőben további teljesítményjavulást érjünk el.



## Irodalom, és csatlakozó dokumentumok jegyzéke

### A tanulmányozott irodalom jegyzéke:

- [1] bernard. “ACDC Challenge.” *Www.creatis.insa-Lyon.fr*, [www.creatis.insa-lyon.fr/Challenge/acdc/index.html](http://www.creatis.insa-lyon.fr/Challenge/acdc/index.html). Accessed 10 Dec. 2023.
- [2] Ronneberger, Olaf, et al. “U-Net: Convolutional Networks for Biomedical Image Segmentation.” *ArXiv.org*, 18 May 2015, [arxiv.org/abs/1505.04597](https://arxiv.org/abs/1505.04597).
- [3] Chaurasia, Abhishek, and Eugenio Culurciello. *LinkNet: Exploiting Encoder Representations for Efficient Semantic Segmentation*.
- [4] Lin, Tsung-Yi, et al. “Feature Pyramid Networks for Object Detection.” *ArXiv:1612.03144 [Cs]*, 19 Apr. 2017, [arxiv.org/abs/1612.03144](https://arxiv.org/abs/1612.03144).

### Megjegyzés:

*A dokumentum írása során az elméleti bevezetőt, valamint a kódok kommentelését LLM (ChatGPT) segítségével végeztük, majd ellenőrzés után a hibákat kijavítva, némileg átfogalmazva tettük bele a végső kódba/dokumentációba.*