Mélytanulás házifeladat

mélytanulóbuvárok csapat

Feladat

“*A transzformátorhálózatok forradalmasították a mélytanulást, különösen a természetes nyelvfeldolgozás területén, és egyre nagyobb figyelmet kapnak a számítógépes látásban is. Nagy előnyöket kínálnak a konvolúciós hálózatokkal (CNN) szemben, például nagyobb rugalmasságot, kisebb érzékenységet a hiperparaméterekre, valamint a bemeneti adatok lokális és globális jellemzőinek hatékony megragadásának képességét. Azonban bizonyos esetekben a CNN-ek még mindig felülmúlják őket, és néhány feladathoz, például szegmentációhoz, még a transzformátoralapú architektúrák is használnak konvolúciós rétegeket.*

A hallgatók feladata: fedezzék fel a transzformátorhálózatokat az orvosi képalkotási szegmentáció területén. Vizsgáljanak meg nyílt forráskódú implementációkat, találjanak és teszteljenek egy CNN-alapú alapmegoldást, majd tanítsanak be 1-2 különböző architektúrájú hálózatot egy szív MRI szegmentációs adathalmazon (más, akár nem orvosi adathalmazok is megengedettek). Tiszta transzformátor, hibrid transzformátor-CNN vagy teljesen konvolúciós architektúrák is megfelelnek, de legalább egy transzformátoralapúnak kell lennie. Hasonlítsák össze a kiválasztott hálózatokat az alábbiak alapján: pontosság, teljesítmény, a hiperparaméterek változásaira való érzékenység, a megvalósítás és betanítás egyszerűsége stb.”

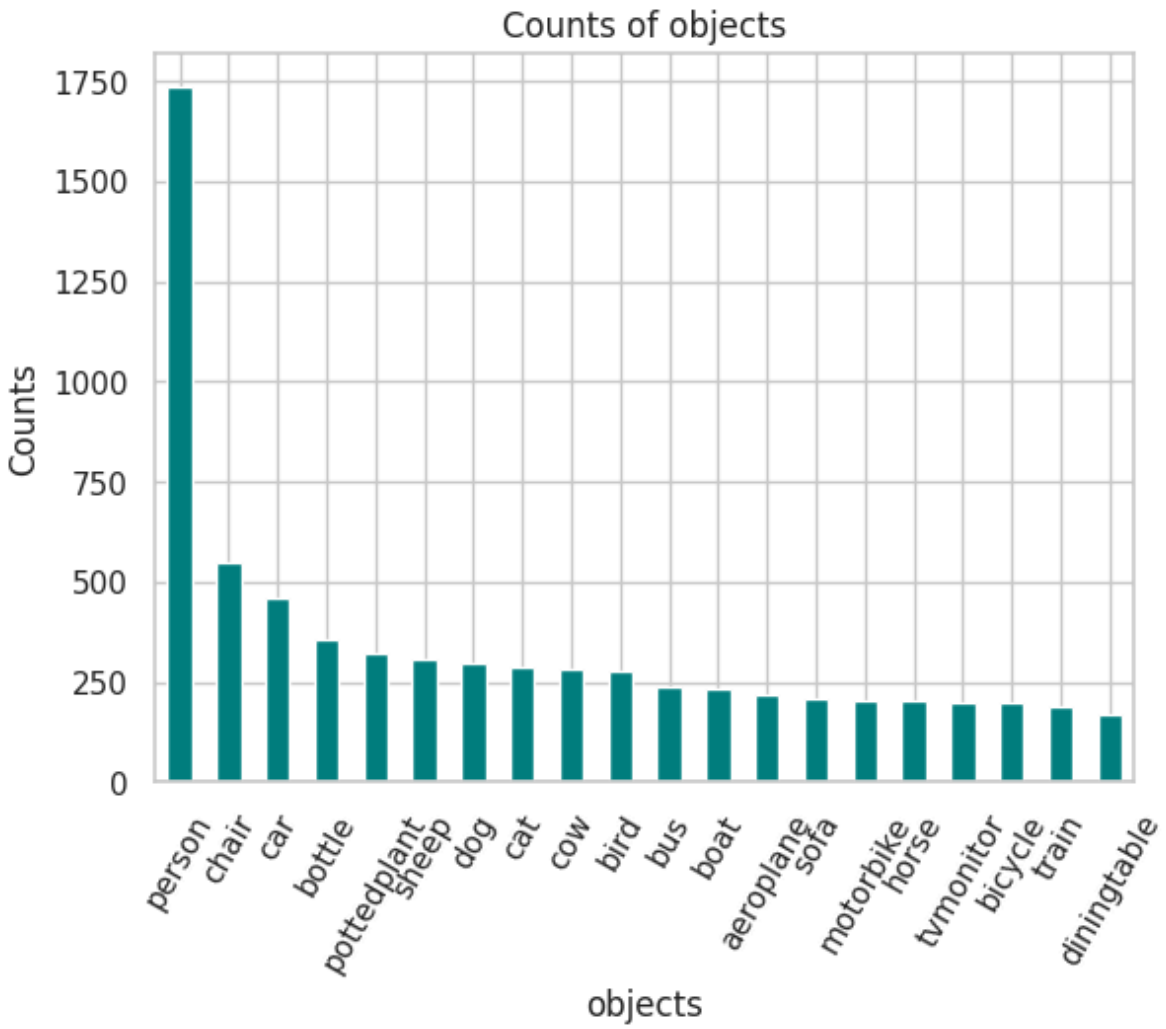
A feladat megvalósításához meg kellett értenünk mit is akarunk csinálni, ehhez nyújtott segítséget az IBM: What is image segmentation? című cikke, amely részletesen körüljárja mit is jelent pontosan a képszegmentálás, például a kép klasszifikációhoz képest.

Adatbázis

A feladat megoldásához első lépésként adatokra volt szükségünk, ezért elkezdtünk általánosan ismert és gyakran hivatkozott adathalmazokat keresni. Egyik ilyen hivatkozás volt a SuperAnnotate: Image segmentation detailed overview című cikke, mely ajánlotta a Pascal Voc adatbázist. Így esett a választásunk a Pascal 2012 nevű adatbázisra
A Pascal 2012 tanító/validációs adatai tartalmazták a szegmentálás előtti és utáni képeket egyaránt, ahogy az alábbi ábrákon látszik.

A szegmentált képekhez minden esetben tartozik xml, amely leírja a képen látható objektumok osztályát és azt a bounding boxot, melybe az objektum található a képen. Az alábbi ábrán látható ennek egy példája, amely a korábbi ábrákhoz tartozik.

Azt lehet mondani, hogy az adatbázisunk általánosságban egyenletes eloszlású volt az objektumok előfordulása szempontjából. Azonban ez az egyenletes eloszlás ha az ábrára tekintünk általánosságban egy relatíve alacsony 256-os előfordulást jelent. Van két kicsit gyakrabban előforduló objektumtípus, a szék és az autó, de természetesen van egy mindenki számára szembetűnő osztály, a személyek.



Arra a döntésre jutottunk, hogy megpróbáljuk kiegyenlíteni az eloszlást és a személyek és az egyébb 256-nál frekventáltabban előforduló objektumok képeit kiszedjük az adathalmazból, természetesen úgy, hogy a többi a 256-nál kevesebbszer előforduló objektumok képeit ne töröljük és ezzel ne rontsuk le a megjelenésüket az adatbázisban. Ezen felül a 256-nál kevesebbszer előforduló objektumok reprezentáltságát mesterséges módon javítottuk, különböző kép augmentáló módszerekkel elkészítettük az eredeti képek módosított másolatait, amik kellően mások voltak ahhoz, hogy ne kétszer tanulják a modellek ugyanazt az adatot.
Ilyen módszerek voltak az alábbiak:

- Zaj generálás
- Véletlenszerű fényerő és kontraszt generálás
- Blur generálás
- Kép élesítés

Az alábbi ábrán látható egy egyszerű példa, a véletlenszerű fényerő és kontraszt generálásra.

Az adatok felosztásánál törekedtünk arra, hogy a következőkben bemutatandó modellek, ugyanolyan train-validation-teszt adatokkal rendelkezzenek.

Modellek

Alapvetően 3 modellel igyekeztünk megoldani a házi feladat problémáját. 2 konvolúciós és egy vision transformer modellel.
A konvolúciós modellek amelyekkel dolgoztunk a ResNet-50 és a ResNet-101 voltak:

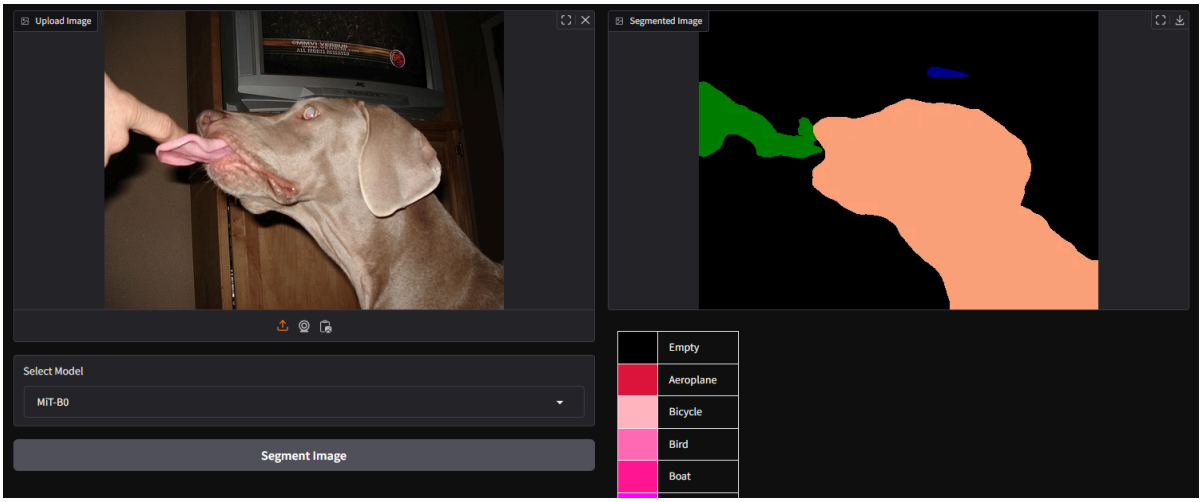
- A ResNet-50 egy 50 szakaszból álló konvolúciós háló, ez az 50 szakasz 50 konvolúciós réteget jelöl. A rétegeken belül bottleneck blokkokkal igyekszik lecsökkenteni a paraméterek számát, ezzel is gyorsítva a modell teljesítményét. A bottleneck csökkenti a bemeneti dimenziót (1x1 konvolúcióval), majd elvégez egy 3x3 konvolúciót, és végül visszaállítja az eredeti dimenziót egy másik 1x1 konvolúcióval.
- A ResNet-101 az 50-esnek egy erősített változata, amely 101 konvolúciós rétegből épül fel, ezért természetesen magasabb erőforrás igénye van mint kisebb változatának.

A vision transformer modell esetében a MiT-B0 modellre esett a választásunk. A modell alapvetően a hierarchikus jellemző tanulásra épít mint ma számos más modell. Ezt úgy lehet elképzelni, hogy szétbontja a képet többszörös patch-ekre és ezeknek a patcheknek a méretét a hálózat különböző szintjein csökkenti. Ez egy piramis-szerű reprezentációt hoz létre, amely eltérő felbontású jellemzőket biztosít. Ezt alapvetően egy hibrid rétegezéssel támogatja meg, ezalatt a hibrid konvolúciós és transzformátor rétegek keveredését értjük. A modell a hagyományos transzformátorok self-attention mechanizmusát optimalizálja, hogy a nagy felbontású képeknél jobban skálázható legyen. Ez azt jelenti, hogy a képek elemeit egymáshoz képest összehasonlítja és ezzel egy súlyozott “hálózatot” hoz létre az adatokból, ezzel megjelenik az elemek egymáshoz képesti relevanciája, ami egy kifinomultabb következtetési formát eredményez.

Igyekeztünk korábbi ötleteket magunkhoz ragadni és azokat felhasználva sikeresebb képszegmentálót létrehozni. A Hugging Face egy remek dokumentációt biztosított a szegmentáció megvalósításához.

Front end UI

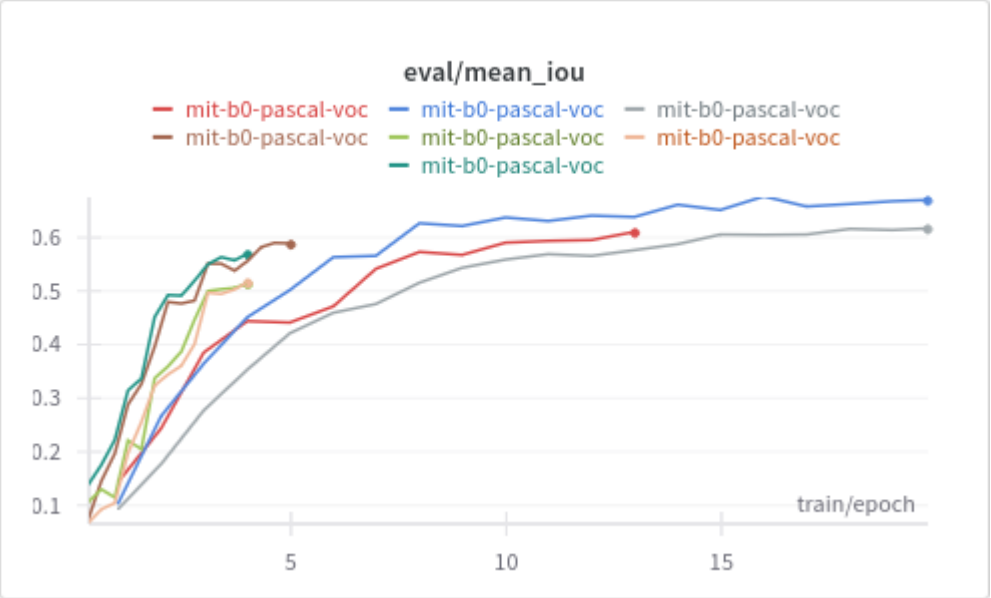
A UI megvalósítására a Gradio csomagot választottuk. A Gradio egy gyors mód a tanuló modell bemutatására. Webes felületet nyit, ahol könnyedén használhatjuk a modelleket és megtekinthetünk számos vizualizációt. A megjelenítéshez kellett egy kép feltöltő és megjelenítő komponens, ennek a komponensnek a működésének a kialakítására találtunk számos dokumentációt a gradio hivatalos oldalán. Mivel 3 modellünk is van, ezért modell kiválasztást támogató UI-t szerettünk volna készíteni, így felvettünk egy Dropdown komponenst is. A



Mivel szerettük volna megjeleníteni az objektum típusokat is a szegmentált képeken, ezért minden típushoz egy saját színt rendeltünk. Ezekhez a szín-típus kapcsolatokhoz egy táblázatot készítettünk.

Hiperparaméter optimalizáció

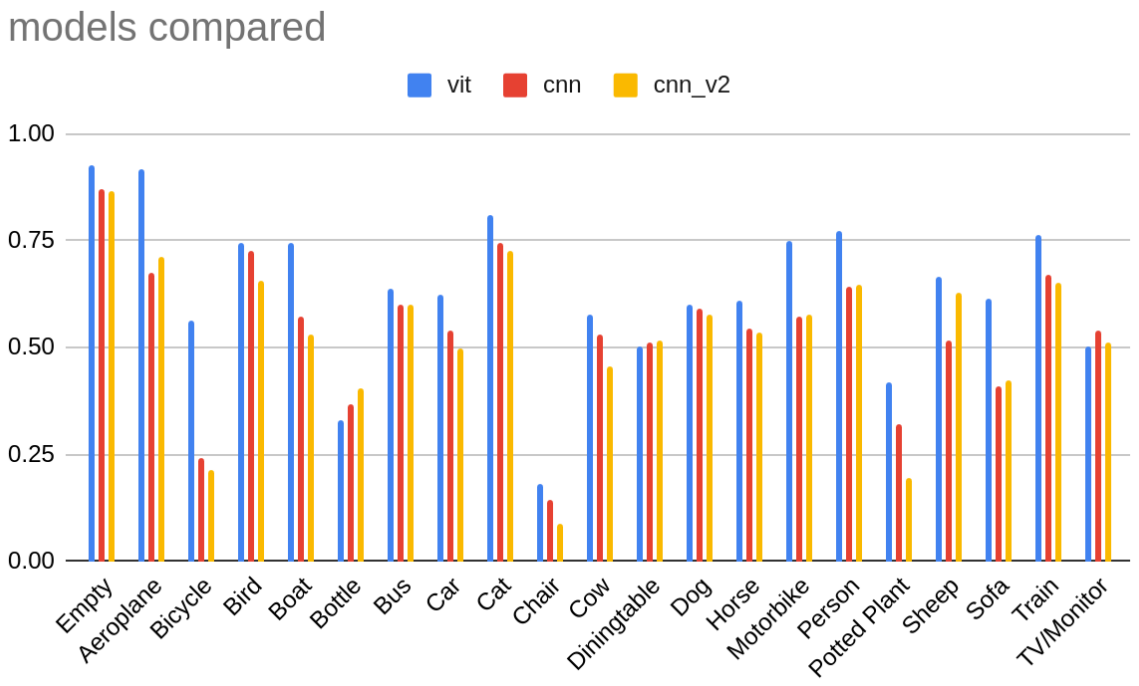
A hiperparaméter optimalizálást manuálisan végeztük elsősorban a batch size és a learning rate megfelelő beállításával, amíg a kívánt IoU értéket el nem értük. Továbbá még warmup ratio-t és weight decay-t alkalmaztunk a Vision Transformer finomhangolása során. A futások logolásához és kielemzéséhez segítségként a Weights and Biases platformját vettük igénybe.



Vision Transformer	
learning rate	3e-4
warmup ratio	0.05
weight decay	0.05
batch size	16

Teszt eredmények

	cnn	cnn_v2	vit
mean IoU	0.540	0.525	**0.632**



Konklúzió

A modellek futtatása során szignifikáns különbséget vettünk észre a két modell típus között. A konvolúciós modellek sokkal nagyobbak, lassabbak és az IoU értékük is látványosan alacsonyabb, a Vision Transformer modellhez képest. Ha osztályszinten tekintjuk az IoU értékeket, akkor is megfigyelhető a VIT előnye, kis kivétellel minden osztály esetén.