



ÓBUDAI EGYETEM
ÓBUDA UNIVERSITY

TUDOMÁNYOS DIÁKKÖRI DOLGOZAT

GALAXISOK ÉSZLELÉSE ÉS OSZTÁLYOZÁSA ÉGBOLTKÉPEKEN NEURÁLIS HÁLÓZATTAL

Szerző(k):

Szabó Alex

mérnökinformatikus BSc. szak, IV. évf.

Konzulens(ek):

Pintér Ádám

tanársegéd

Budapest, 2022.

GALAXISOK ÉSZLELÉSE ÉS OSZTÁLYOZÁSA ÉGBOLTKÉPEKEN NEURÁLIS HÁLÓZATTAL

Szabó Alex

Óbudai Egyetem

Neumann János Informatikai Kar, BSc. IV. évfolyam

Konzulens(ek): Pintér Ádám, tanársegéd

A galaxisok észlelése és osztályozása kulcsszerepet tölt be a csillagászati kutatásoknál. Elsősorban a főosztályok megállapítására van szükség, ami lehet spirális vagy elliptikus. Gyűrűs szerkezetek spirális galaxisoknál figyelhetők meg, ritkán fordulnak elő és még kevésbé ismertek. Felismerésük emberek számára sem mindig egyszerű. Jelenleg mintegy 2 trillió galaxis létezik, ezek közül teleszkópjaink évente több milliót térképeznek fel. A projekt célja az észlelés és osztályozás automatizálása egy erre kifejlesztett szoftverrel, ami jelentősen felgyorsítaná a folyamatot.

A dolgozatomban az elérhető módszerek és hasonló megoldások áttekintése alapján a galaxisok osztályozására két különböző módszer került kialakításra. Elsőként egy Faster R-CNN alapú neurális hálózat lett kifejlesztve, ami különböző méretű és arányú határolókeretekre osztja fel a képet és azokban keresi az objektumot. A RoI pooling a Mask R-CNN-nél bemutatott RoIAlign-ra lett cserélve, mivel a feature mapok kimetszésénél bilineáris interpolációt alkalmaz, így több információt őriz meg. A számítás intenzív függvények, mint pl. a határolókeret átfedések kiszámítása GPU-n kerülnek végrehajtásra. Tanításához egy égboltkép generáló lett létrehozva, amely a Galaxy Zoo projekt katalógusa segítségével letöltött SDSS égboltfelmérés galaxis és háttérképekből készít képet, azokat augmentációs módosítások után véletlenszerűen elhelyezve. A másik módszer észleléshez DBSCAN-t, osztályozáshoz pedig egy-egy CNN-t használ.

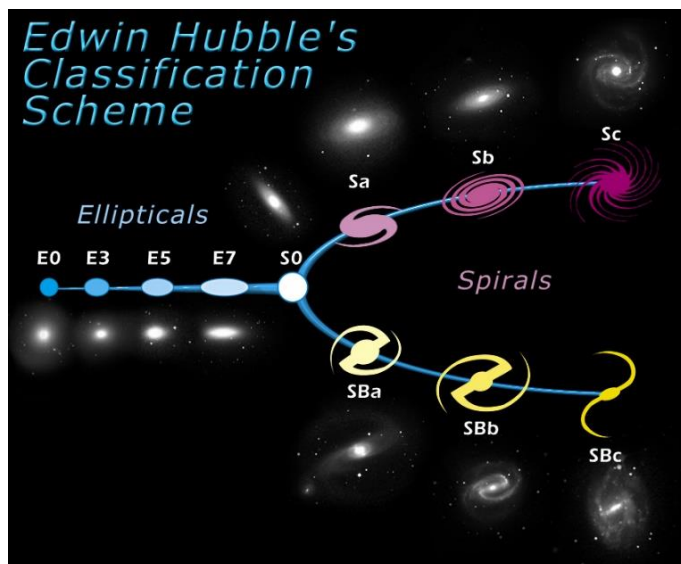
Az eredmények alapján a DBSCAN alapú módszer bizonyult a jobbnak, ami főosztály esetén 99,7%-os, gyűrű esetén pedig 94,9%-os pontosságot és 94,91%-os F1 scoret ért el a Faster R-CNN 93,0%-os főosztályú és 86,0%-os gyűrű pontosságú eredményeihez képest. A betanított hálózathoz egy grafikus felületű program is készült, ahol tetszőleges égboltképen észlelhetők, lementhetők a galaxisok.

TARTALOMJEGYZÉK

1	BEVEZETÉS	2
2	MÓDSZEREK ISMERTETÉSE.....	4
2.1	Adatok	4
2.2	Konvolúciós neurális hálózatok.....	6
3	IRODALOMKUTATÁS.....	11
3.1	Létező megoldások galaxisok felismerésére.	11
3.2	Objektumok detektálásnak lehetőségei képeken	16
3.3	Faster R-CNN	18
3.4	Összefoglalás	21
4	TERVEZÉS	22
4.1	A szoftver célja és felépítése	22
4.2	UML, UI/Wireframe.....	23
5	FEJLESZTÉS	26
5.1	Rendelkezésre álló hardver.....	26
5.2	Adatok előkészítése	26
5.3	RPN	30
5.4	Fast R-CNN	32
5.5	Hálózatok tanítása.....	34
5.6	Faster R-CNN eredmények	37
5.7	DBSCAN és CNN	39
5.8	Eredmények.....	41
6	ÖSSZEFOGLALÁS.....	44
7	ÁBRAJEGYZÉK	45
8	TÁBLÁZATJEGYZÉK	47
9	IRODALOMJEGYZÉK.....	48
10	MELLÉKLETEK	51

1 BEVEZETÉS

Az első és egyben legelterjedtebb galaxis osztályozást Hubble alkotta meg 1926-ban. Eredetileg három típusra osztotta őket: spirális, elliptikus és irreguláris, amit később 1936-ban kiegészítette a lentikulárisal, amely az 1.1 ábrán láthatunk [1]. Univerzumunkban leggyakoribbak az elliptikusok, melyek gömb vagy ellipszoid alakúak, többnyire vörös színűek a benne lévő nagyszámú öreg csillagok miatt. A spirálgalaxisok mindegyikének közepében egy sűrű csillagcsoportból álló dudor alakul ki. Az ezt körülvevő korongban jönnek létre a spirálkarok, melyek rengeteg fiatal csillagból állnak és emiatt kék színben látszódnak. Emellett a spirálgalaxisokat aszerint is megkülönböztetjük, hogy rendelkeznek-e küllővel, ami egy rúd formájú csillagsáv a galaxis központi magja körül. Az 1.1 ábrán látható hangvilla alsó ágában a küllős, felső ágában pedig küllő nélküli spirálgalaxisok szerepelnek. A lentikuláris galaxisoknál megfigyelhető szintén a korong, azonban karokkal nem rendelkeznek. Az elliptikusokhoz hasonlóan ezek is főként öreg csillagokból állnak. Irreguláris galaxisoknak pedig azokat nevezzük, amelyek az előzőek egyikébe sem sorolhatók. Ezek szabálytalan alakúak, spirálkarok vagy központi dudor nélkül.



1.1. ábra. Hubble hangvillája [1].

A csillagászok számára kulcsfontosságú a galaxis típusának ismerte. Adott kutatáshoz csak egy bizonyos fajtájú galaxisok felelnek meg, ezért ha rendelkezésre áll számukra egy osztályozó katalógus, onnan könnyen kiválaszthatják a megfelelő galaxisokat. Jelenleg körülbelül 2 trillió [2] galaxis létezik. Teleszkópjaink milliányi galaxist térképeznek évente [3]. Ezek megismerésére irányuló kezdeményezés a Galaxy Zoo [4] projekt, melynek 2007-es indulása óta eddig több mint egymillió galaxis került osztályzásra. Azonban ez még mindig eltörpül az osztályozatlan galaxisok száma mellett. Már a 70-es évektől elkezdődött a galaxisok azonosítása számítógép segítségével. Ilyen volt például a Faint Object Classification and Analysis System (FOCAS) [5], mely a képen lévő képpontok körüli pontok átlagfényességéből

számolva több mint 90% pontossággal képest volt megkülönböztetni galaxisokat a csillagoktól. Később az SExtractor projekt már neurális hálózattal különítette el a galaxisokat a csillagoktól, mely azonos fényerősségű képterületeket használt bemenetként és a fényesebbek esetén (<20 magnitúdó) közel 95% pontosságot produkált [6]. Az újabb Sloan Digitális Égboltfelmérési Program (Sloan Digital Sky Survey, továbbiakban SDSS) felméréshez készült neurális hálózat [7] és SVM alapú módszer [8] már több, mint 90% pontossággal volt képes meghatározni a Hubble sorozatú galaxisokat. Azóta számos tanulmányban mind galaxisok osztályozásánál, mind másféle képfeldolgozásnál rendkívül jó eredménnyel alkalmazásra kerültek a konvolúciós neurális hálózatok.

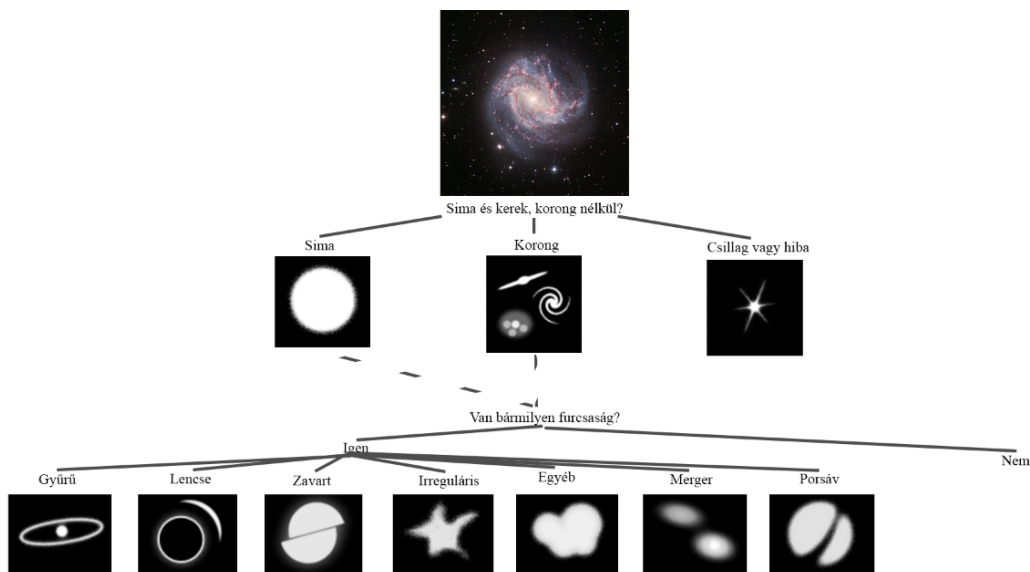
2 MÓDSZEREK ISMERTETÉSE

2.1 Adatok

A képek, teleszkópok által készített égboltfelmérésekből származnak, melyek Világegyetemünk különböző részeit, többféle hullámhosszon pásztázzák. Talán a leghíresebb ezek közül az SDSS, mely 2000-ben kezdte meg működös Új-Mexikóban egy 2,5 méteres tükrös távcsővel. Fotometrikus rendszere kb. 300 és 900 nanométer közötti hullámhosszokon 5 sávban végez mérést, melyből azután színes, RGB képeket készítenek. A létrejött adatokat mindenki számára szabadon elérhetővé teszik. 2009-ig végére égboltunk körülbelül harmada került feltérképezésre. Weboldalukon koordináták megadásával egyszerűen lekérhető az adott terület képe [9].

Ez volt a forrása a Galaxy Zoo projektnek, mely 2007-ben indult azzal a céllal, hogy a galaxisokat morfológiailag azonosítsák. Ebben bárki részt vehet, azok is, akik nem rendelkeznek csillagászati ismeretekkel. A felhasználó véletlenszerűen kap egy galaxisról képet és egy kérdésfán végig haladva kell a válaszokat megadnia. Dolgozatom szempontjából fontos részlet az 2.1. ábrán látható. A teljes, eredeti kérdésfa pedig az irodalom alapján elérhető. Az első kérdés a galaxisok legalapvetőbb morfológiai tulajdonságára fókuszál, melyet Hubble is elkülönített, tehát spirális vagy elliptikus. Míg előbbieknél megfigyelhető a korong, addig az elliptikus galaxisoknál ilyen nem figyelhető meg és simának, homogénnek látszódik. Emellett ha az objektum nem galaxis, a kérdéssorozat véget ér. Ezután az adott típushoz több arra specifikus kérdést kapnak, melyből pontosabb ismeretünk lesz az alakjáról, mint például spirális galaxisok esetén a spirálkarok száma. A végén minden esetben kérdés, hogy látható-e bármilyen különlegesség a képen. Amennyiben igen a 2.1. ábra alján látható hét lehetőség közül választhatnak. Különösen fontos ezek közül a gyűrűk és a mergerek¹ jelenléte.

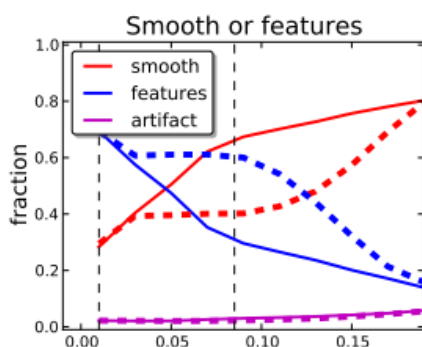
¹ Merger: galaxis egyesülés



2.1. ábra. Galaxy Zoo kérdésfa részlet [10].

Az évek alatt összegyűlt válaszokat katalógusokba gyűjtötték, melyek weboldalukon, szabadon elérhetők. Mivel az adatok torzítással terheltek, további előkészületek szükségesek [4].

Megfigyelhető, hogy minél távolabbi és halványabb egy galaxis, annál kevésbé képesek felismerni az apró jellegzetességeket mind az emberek, mind a gépek. Különösen a spirális és elliptikus galaxisok szétválasztásánál jelentett ez problémát. Mivel a Galaxy Zoo 2-nél az osztályzás egy kérdésfán keresztül zajlik, ezért gyakran a kérdések összefüggenek egymással. Ahhoz, hogy bizonyos alsóbb rétegbeli kérdéseket kapjon az osztályozást végző felhasználó, korábban már annak megfelelő választ kellett adnia, hogy erre az ágra jusson. A szavazatok torzításának megszüntetéséhez ezért egy minimum súlyozott szavazati hányadost és az adott kérdéshez érkezett minimális szavazatot állítottak be.

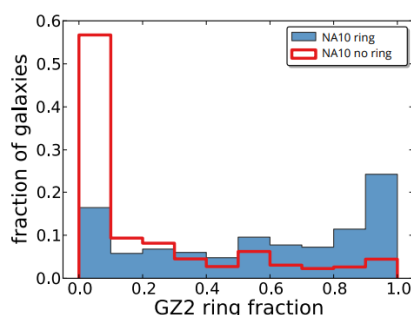


2.2. ábra. Szavazati arányok [11].

A 2.2. ábrán látható folyamatos vonalak az eredeti, míg szaggatott vonalak a korrigált állapotot mutatják. A korrigálást 0,01 vöröseltolódástól kezdték meg. Az SDSS érzékenysége 0,085. Minél távolabbi egy galaxis, általában annál nagyobb a vöröseltolódása és így észlelésük is nehezebb.

Minden döntést megjelöltek egy igaz/hamis jelzővel, amely megmutatja, hogy egy kijelentés megbízható-e az adott galaxisra. Három követelménynek kell teljesülnie, ahhoz hogy igaz legyen. A döntési fában a korábbi válaszok egy minimum szavazási hányaddal rendelkezzenek, ezentúl legalább 20 szavazat az adott kérdésre és végül a torzítás mentesített szavazási hányadnak legalább 0,8-nak kell lennie minden egyes döntési elágazás esetén [11].

Nair és Abraham [12] 2010-ben a gyűrűket 3 féle csoportra bontotta: belső gyűrű, külső gyűrű, nukleáris gyűrű. Galaxy Zoo 2-vel való összehasonlítás során megfigyelhető, hogy a belső gyűrűk észlelése okozta a legnagyobb kihívást a felhasználóknak. A gyűrűket összességükben kb. 75% pontossággal ismerték fel, amennyiben legalább 5 szavazat figyelt meg gyűrűt. Mint ahogy a 2.3. ábrán is látható, a problémát elsősorban a téves gyűrű észlelések jelentik.



2.3. ábra. Gyűrűs galaxisok aránya [11].

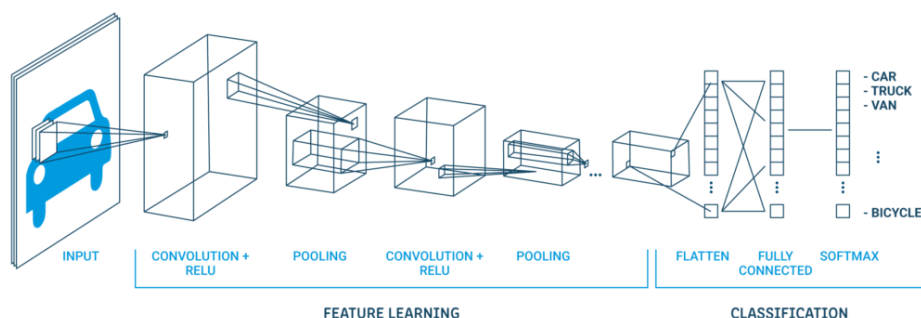
Az így elkészült torzítás mentesített katalógus szintén szabadon elérhető, így az osztályozó szoftver számára szükséges bemeneti adatok könnyedén beszerezhetők a koordináták ismeretében.

Casteels és társai [13] 2013-ban rámutattak arra, hogy a torzítás különösen magas a ritka jelenségeknél és mergereknél. Utóbbinál gyakran úgy tűnhet, mintha két galaxis egymáshoz közel lenne és interakcióba lépnének egymással, azonban ha a vöröseltolódásuk nagymértékben eltér, valójában nem igazi mergerekről van szó. Ennek feloldására a fizikailag közel lévő pároknál kivonták a látszólagos mergerek átlagos valószínűségét.

2.2 Konvolúciós neurális hálózatok

A konvolúciós neurális hálózat alapötlete Hubel és Wiesel megfigyeléseiből származik. Macskák és majmok agytevékenységeit vizsgálták, melynek során két sejtípust azonosítottak: egyszerű és komplex. Az egyszerű sejtek adott irányú élekre reagálnak és lineáris függvényhez hasonló a viselkedésük. Ezek továbbbódnak a komplexebb sejtek felé ahol konvergálnak. A komplex sejtek már érzéketlenek az irányítottságra és képesek összetettebb mintákat is felismerni [14]. Erre alapozva alkotta meg Kunihiro Fukushima az első mesterséges neurális hálózatot, ahol egyszerű és komplex sejtekből álló rétegekből épített fel modellt. Az utolsó rétegben lévő komplex sejtek csak egy bizonyos mintára reagálnak. Minden egyes egyszerű sejtcellán egy súlymátrixot alkalmaz, mely az öntanulási folyamat során képes változni [15].

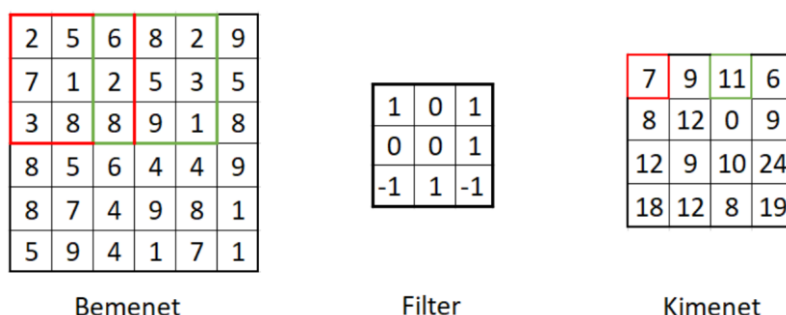
Ehhez hasonló az újabb CNN-ok felépítése is, melyből egy példa az alábbi 2.4. ábrán látható. Minden CNN három fő részre bontható: bemeneti réteg, rejtett réteg és kimeneti réteg. Emellett rendelkeznek különböző számú szűrőkkel, melyek képesek a képből kiszűrni információkat, így például éleket és összetettebb formákat tudnak találni. Ezek mérete jellemzően sokkal kisebb a bemenet méreténél. Utolsó dimenziójuk megegyezik a bemeneti mátrix utolsó dimenziójával. A rejtett rétegben történik a hálózat tanítása, mely során a súlyok értékei folyamatosan változnak, ezáltal képessé válnak később új, ismeretlen alakzatok felismerésére.



2.4. ábra. Konvolúciós neurális hálózat járművek felismerésére [16].

A tanulási folyamat előreterjesztéssel kezdődik. A bemeneti réteg minden esetben egy mátrix, amely kép esetén: szélesség×magasság×csatornák száma. Ezen elvégezve a konvolúciót lépünk a rejtett rétegekbe.

A művelet folyamata az 2.5. ábrán látható. Ami lényegében elemenkénti mátrixszorzások összegzését jelenti és a végén egy skalárt kapunk eredményül. A szűrőt végig csúsztatjuk ablakként a bemeneti mátrixon és az így képzett skalárokból álló mátrixot aktivációs térképnek nevezzük. Első két dimenziója a szűrő méretétől és lépésköztől függ. Utolsó dimenziója pedig az alkalmazott szűrők számával egyezik meg.



2.5. ábra. Konvolúció folyamata.

Az aktivációs térkép első két dimenziója ilyen esetben mindig kisebb lesz a bemeneténél, annak széleknél való adatvesztéséhez vezet. Erre nyújt megoldást az 2.6. ábrán látható zero padding, mely segítségével körülvesszük nullákkal a bemeneti mátrixot, így könnyen elkerülve a dimenziócsökkenést. Látható, hogy az eredeti 6×6-os mátrix, a kimeneten szintén 6×6-os méretű maradt a korábbi 4×4-el szemben.

0	0	0	0	0	0	0	0
0	2	5	6	8	2	9	0
0	7	1	2	5	3	5	0
0	3	8	8	9	1	8	0
0	8	5	6	4	4	9	0
0	8	7	4	9	8	1	0
0	5	9	4	1	7	1	0
0	0	0	0	9	0	0	0

Bemenet

1	0	1
0	0	1
-1	1	-1

Filter

11	-2	4	2	2	2
1	7	9	11	6	9
12	8	12	0	9	8
14	12	9	10	24	-6
8	18	12	8	19	-2
16	16	8	28	2	8

Kimenet

2.6. ábra. Zero padding alkalmazása.

Célszerű lehet ezen egy aktivációs függvényt végrehajtani. Az egyik legelterjedtebb a ReLU². Mint ahogy az egyenletéből (1) látható, minden negatív számból nullát képez, míg a pozitív értékeket változatlanul hagyja.

$$f(x) = x^+ = \max(0, x) \quad (1)$$

Ezután ismét következhet egy konvolúciós réteg. Azonban ahogy a rétegen egyre mélyebbre haladunk, a paraméterek száma jelentősen megnövekszik, ami egyrészt nagyobb számításigénnyel jár, másrészt túlillesztés is előfordulhat. Ennek kiküszöbölésére lehetőség van egy ún. pooling réteggel, amely az előző rétegbeli neuron csoportokat egyetlen neuronba összpontosítja. Két fő elterjedt módszer a max pooling és az average pooling. Előbbi a neuroncsoport maximális értékét adja vissza, amiből egy tulajdonság létezésére következtethetünk, míg átlagszámolás esetén annak képen való átlagos előfordulását kapjuk meg.

A végső, teljesen összekapcsolt rétegnél minden egyes neuron összekapcsolódik az előtte lévő réteg minden neuronjával. Ezen egy aktivációs függvény, például a softmax-ot végrehajtva minden egyes paraméter 0 és 1 közötti értéket kap, a valószínűségének megfelelően. Akkor érdemes alkalmazni, mikor az egyes osztályozandó esetek kölcsönösen kizárják egymást. Ezen hívjuk meg a veszteségfüggvényt, amellyel validáljuk modell eredményének jóságát. Ilyen lehet például az átlagos négyzetes eltérés.

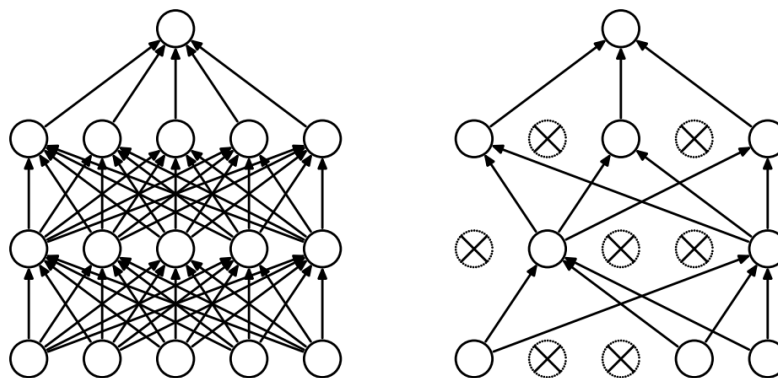
Ennek ismeretében egy visszaterjesztést hajtunk végre, hogy az egyes neuronok megfelelő irányba módosuljanak. Minden neuron függ az előtte lévő réteg neuronjaitól. Azon a neuronok súlyát, amelyek helyesen felismerték a mintát, megerősítjük, míg, amelyek tévedtek gyengítjük. Az utolsó rétegtől visszafelé haladva kiszámoljuk az adott réteg súlyainak gradiensét. Ezt elvégezzük az összes tanulómintán, majd ezután az úgynevezett gradiensereszkedés során módosítjuk az aktuális súlyok értékét a kapott gradiens, tanulási rátával módosított értékével. Egyes szigmoid aktivációs függvények alkalmazása esetén előfordulhat, hogy a gradiens túl kicsivé válik vagy teljesen eltűnik. ReLU aktivációs függvénnyel azonban ez kiküszöbölhető. Az epochok során a veszteség fokozatosan csökken egy adott értékig. Célszerű lehet a teljes

² ReLU: rektifikált lineáris egység (rectified linear unit)

mintát, a rendelkezésre álló memória tekintetében, kisebb kötegekre bontani. Így bár elsőre nem a legoptimálisabb módosításokat kapjuk meg, a folyamat jelentősen felgyorsul [17].

A modell működésében problémát jelenthet a túlillesztés. Ha a szükségesnél komplexebbé válik a rendszerünk, akkor az egyszerűbb mintákban is komplexitást keres. Kis mennyiségű tanítási adat vagy magas epoch szám esetén egyre jellemzőbbé válik. Ekkor bár a tesztekénél pontos lesz, ismeretlen adatokon kevésbé működik jól, mint egy egyszerűbb modell. Ezzel szemben az alulillesztésnél a modell nem lesz képes felismerni komplexebb szerkezeteket túl egyszerű felépítése miatt. A cél mindkettő elkerülése, azáltal hogy a tanulás pontossága mellett a teszt és tanulás közti különbséget is minimalizáljuk.

Különböző regularizációs technikák alkalmasak a túlillesztés csökkentésére. Az egyik ilyen módszer a 2.7. ábrán látható dropout. A tanulási folyamat során adott valószínűséggel kitörölünk neuronokat a hálóból, így a megmaradt neuronokat kényszerítve önálló tanulásra. Ezáltal a háló ritkává válik tanítás alatt. Később teszt során a tanítás alatt ideiglenesen kitörölt neuronokat ismét használjuk, azonban a súlyértékét a törlés valószínűségével szorozzuk, így értéke csökken.



2.7. ábra. Dropout folyamata [18].

A bemeneti rétegnél közel az összes neuront érdemes megtartani, majd a további rétegeknél általában a fele eltávolítható. Teszteredmények alapján ugyanazon a konvolúciós hálózat minden rétegeire alkalmazva a dropoutot, közel 2%-kal csökkent a modell hibaszázaléka. Különböző tanítási méreteken tesztelve az tapasztalható, hogy a dropout leginkább a közepes méretű adatmennyiségeknél előnyös. MNIST³-ből 100 és 50 ezer közti méreteknél, 5000-nél volt a leghatékonyabb. Hátránya ennek a módszernek, hogy a tanítási időt két-háromszorosára növeli. Max-norm módszerrel az adott réteg bementeti súlyai értékéhez egy maximális korlát tartozik. Ha ezt valamelyik súly meghaladja, akkor egy konstans értékkel csökken minden súly. Súlycsökkentés vagy max-norm technikákkal párosítva a dropout és a max-norm kombinációja bizonyult a legpontosabbnak [18].

³ MNIST: neurális hálózatok számára készített képadatbázis

Egy másik népszerű regularizációs módszer a batch normalizáció. A hálózat minden egyes rétegénél normalizálja a kimeneti neuronok átlagát és szórását két paraméter – skalár és eltolás – segítségével. Ezen paramétereket a hálózat menet közben szintén megtanulja a hiba visszaterjesztés által. A módszer előnye, hogy nagyon gyorsá teszi a tanulást, így jóval kevesebb epochra van szükség a megfelelő pontosság eléréséhez. Emellett sikeresen csökkenti a túlillesztést is [19]. Christian és társai 2020-as eredményei alapján a batch normalizáció kb. 2% - al javította egy CNN pontosságát [20]. A tanulási rátát tízszeresére növelve további 2-3%-kal így összesen 5%-kal javult a pontosság. Míg dropout esetén majdnem 6% - al rontotta azt. Érdekes még figyelembe venni a tanulási idő közel 10%-kal való emelkedését batch normalizációnál.

3 IRODALOMKUTATÁS

3.1 Létező megoldások galaxisok felismerésére

3.1.1 Különböző megoldások Galaxy Challengehez

2014-ben egy Galaxy Challenge nevű nemzetközi verseny zajlott le, melynek célja a lehető legjobb galaxis osztályozó modell elkészítése volt. A képek az SDSS DR7-ből származtak, a hozzájuk tartozó címkék pedig a Galaxy Zoo projektből.

Elsőként Dieleman és társai [21] általi nyertes modellt részletezem. A rendelkezésükre álló közel 60 ezer kép viszonylag kevés a tanításhoz, könnyen túlillesztést jelensége fordulhatna elő. Ennek kiküszöbölésére hasonló esetekben is jellemző adatbővítést és paraméter megosztást alkalmaztak. Ezen felül dropoutot és a modellek átlagolását végezték el. Adatbővítés során forgatási, tükrözési, eltolási, átméretezési műveletek, valamint fényesség állítása közül véletlenszerűen választottak ki ötöt minden egyes kép esetén. Ezentúl a képeket kisebb darabokra szeletelték, melyek középen egymást átfedték. A bemeneti képek RGB formátuma pontosabbnak bizonyult a szürkeárnyalatossal szemben, ezért azokat megtartották színesnek.

3.1. táblázat. Dieleman és társai neurális hálózatának felépítése [21].

	Type	No. of features	Filter size	Non-linearity	Initial bias	Initial weights
1	Convolutional	32	6×6	ReLU	0,1	N(0; 0,01)
2	Convolutional	64	5×5	ReLU	0,1	N(0; 0,01)
3	Convolutional	128	3×3	ReLU	0,1	N(0; 0,01)
4	Convolutional	128	3×3	ReLU	0,1	N(0; 0,1)
5	Dense	2048	-	maxout	0,01	N(0; 0,001)
6	Dense	2048	-	maxout	0,01	N(0; 0,001)
7	Dense	37	-	Constraints	0,1	N(0; 0,01)

A 3.1. táblázatban látható neurális hálózat 4 konvolúciós- és 2 vagy 3 dense réteg egyedi torzítás értékeket kapott. A harmadik réteget kivéve 2×2-es max pooling rétegeket is illesztettek be közéjük. Az aktivációs térképeken ReLU-t alkalmaztak. A folyamat végén összefűzték és a sűrű rétegen átvezették őket az utolsó kimeneti rétegig. A valószínűségeket ezen egy ReLU és normalizálás segítségével kapták meg.

Emellett létrehoztak 17 különböző modellt a szűrők méretének és számának változtatásával, valamint a sűrű rétegek aktivációs függvényeinek ReLU-ra cserélésével. Az így kapott modellek eredményeit azután átlagolták. Elliptikus galaxisokat 87,79% míg spirálisokat 82,52% pontossággal azonosította. Gyűrűs galaxisok esetén meglehetősen jó, 90,97%-os precizitást ért el. Mergerek esetén a modell 73,98%-os precizitással pontatlannak bizonyult [21].

Tu és Truyen [22] által készített modell harmadik helyezést ért el, amely szintén CNN – t használt. A versenynél a modellek jóságát a galaxisokra érkezett válaszok és azoktól való eltérés négyzetes átlagában mérték. Dieleman és társainak [21] modelljének $7,492 \cdot 10^{-2}$ eredményétől mindössze $3,77 \cdot 10^{-3}$ -al maradt el. Előkészületként a képek 200×200 pixeles középső területe kivágásra került, majd ezt még 128×128 pixelesre átméretezték. A galaxisok többségénél nem történt információvesztés, mivel azok általában a képek közepén találhatók, emellett a számításigény így jelentősen lecsökkent. Augmentációként a nyertes modellhez hasonló technikákat alkalmaztak. A hálózat felépítése az alábbi 3.2. táblázatban látható.

Aktivációs függvényként minden réteg esetén ReLU-t alkalmaztak. A túlillesztést és még további sebességnövelést segíti a 7-es és 8-as rétegben elvégzett 0,5-ös dropout. Ezenkívül $5 \cdot 10^{-4}$ mértékű súlycsökkentést is alkalmaztak. A képeket véletlenszerűen megkeverték és 98:2 arányban felbontották tanítási és validációs adathalmazra. A kezdeti súlyértékek 0 és 0,01 közötti Gauss eloszlásból kerültek beállításra és a torzítás mindenhol 0 volt. A tanulási ráta 140 epochig 0,01 értékű volt. Ekkor tizedére csökkentették, majd 166 és 174 epochnál ismét tovább csökkentették. Összesen 182 epochig futott a tanítás.

Miután a CNN megfelelően pontosná vált a kimenetét egy másik neurális hálózatba vezették. Utóbbit úgy tanították, hogy a kimenetek átlaga és normális eloszlása nulla legyen. A hiperparamétereket állítva, több neurális hálózatot is tanítottak. A végeredményhez egy CNN–t, egy neurális hálót és két átlagolt értéket használtak, mely egy CNN és egy neurális háló kimeneteiből származott [22].

3.2. táblázat. Tu és Truyen neurális hálózatának felépítése [22].

Layer	1	2	3	4	5	6	7	8	9
Stage	conv +max	conv +max	conv	conv	conv	conv +max	full	full	full
#channels	48	96	192	192	384	384	2048	2048	37
Filter size	5×5	5×5	3×3	3×3	3×3	3×3	-	-	-
Conv. stride	1×1	1×1	1×1	1×1	1×1	1×1	-	-	-
Pooling size	3×3	2×2	-	-	-	3×3	-	-	-
Pooling stride	3×3	2×2	-	-	-	3×3	-	-	-
Zero- padding size	-	-	-	-	-	-	-	-	-
Spatial input size	120×120	39×39	18×18	16×16	14×14	12×12	4×4	1×1	1×1

Érdemes megfigyelni Rohit és Irfan [23] által készített modelleket a versenyhez, ugyanis nekik a hardver szűkössége miatt nem volt lehetőségek CNN alkalmazására. Egyik próbálkozásuk a 70-es évekbeli FOCAS–hoz [5] hasonlóan középső pixelek vizsgálatán alapult. 10×10 pixeles

képrészletet kivágtak a képek közepéből. Más projektekhez hasonlóan itt is a színes képeket használták. Augmentációként az általában használt forgatást, eltolást, tükrözést, skálázást használták. Érdekesség, hogy itt forgatásnál 0 és 360 fok közötti véletlenszerű értéket választottak, míg a legtöbb esetben csak néhány fix értékkel módosítják. Tu és Truyen [22] úgy találta, hogy 90 fok mellett még egy 45 fokos forgatást is alkalmazva nem javított a modell jóságán. A modell RGB színekhez tartozó értékek összegéből számolt egy átlagos intenzitás értéket. Majd az egyes színekhez tartozó átlagértéket elosztva a teljes átlaggal megkapni, hogy melyik szín átlaga mekkora részben járul hozzá a teljes átlaghoz. Így kapva RGB(0), RGB(1) és RGB(2) értékeket. Ebből egy hash-t számolt az alábbi képlettel:

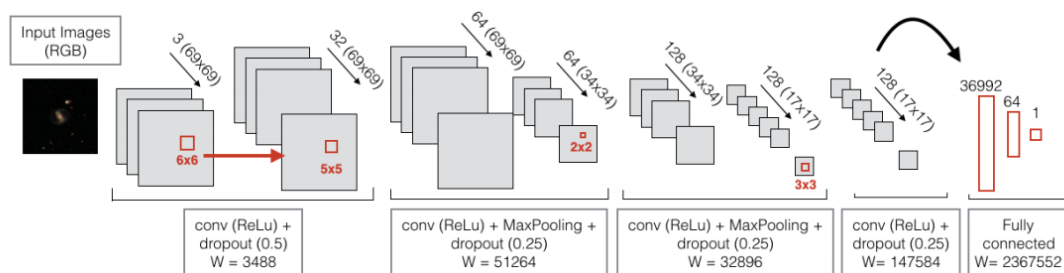
$$\text{hashérték} = \text{RGB}(0) * \text{faktor} * \text{faktor} + \text{RGB}(1) * \text{faktor} + \text{RGB}(2) \quad (2)$$

A szorzótényezőnek 10-et használtak kísérletezések eredményeképpen. Ezt a hash értéket kiszámította minden egyes képhez. Ezek alapján hasonló színekből álló csoportokat hozott létre és ezekre átlagolta a galaxis 37 féle valószínűségi értékeit. Minden tesztben lévő kép esetén a középső 10×10 pixeles képrészlet színei hashelve lettek, így egyetlen számértéket kapva. Ezután megkereste a hozzáillő csoportot és hozzárendelte a csoporthoz tartozó osztályértékeket a tesztképhez. A verseny eredményénél az átlagos négyzetes hiba gyöke (RMSE) 0,16 volt, amely bár elmarad a CNN-ek pontosságától, azonban sokkal gyorsabb azoknál.

CNN-ek esetén a hálózat a tanítás során megtanul bizonyos mintákat, melyekkel később képes új képet is felismerni. Kísérletet tettek e minták kézzel történő beállítására úgy, mint entrópia, átlag színérték és kontúr mintafelismerők, melyet egy SVM-ben alkalmaztak. A kapott valószínűségeket ezután egy Extra Trees Regresszorral feldolgozták, így 0,115 RMSE értéket értek el. Látható egy CNN-ek pontosságával szemben még ez is jelentősen elmarad. [23]

3.1.2 Egy újabb CNN alapú galaxis osztályozó

Sánchez és társai [24] Galaxy Zoo 2 katalógus alapján SDSS DR7 képeket használtak bemenetként. Az eredetileg 424×424-es képeket leskálázták 69×69 pixel méretűre, hogy felgyorsítsák a tanulási folyamatot. A 3.1. ábrán látható neurális hálózat 4 konvolúciós rétegből és 4 szűrőből állt. A túlillesztés elkerülésére ezek közé egy-egy max-pooling réteget illesztettek be, valamint dropoutot is alkalmaztak. Ez a túlillesztés csökkentése mellett gyorsabbá is tette a hálózatot.



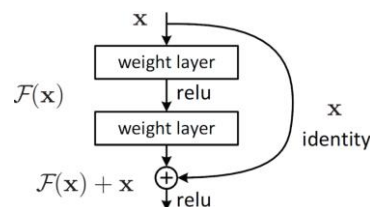
3.1. ábra. Neurális hálózat felépítése [24].

Az első rétegnél nagyobb, bár az általánosan alkalmazott 0,8-1,0-nél kisebb, 0,5 értékű dropoutot használtak, míg a többinél 0,25-öt. Az aktivációs függvény minden réteg esetén a ReLU volt. 50 epochig futtatták 30-as köteggel és 0,001-es tanulási rátával. Adatbővítésként skálázást, forgatást, tükrözést és eltolást alkalmaztak, mely a túlillesztést csökkentette. A veszteségfüggvénynek bináris-keresztentropiát alkalmaztak Adam optimalizációval és szigmoid aktivációval.

Csak azokat az adatokat használták, amelyekre legalább 5 szavazat érkezett. A különböző kérdésekhez saját, egyedi modelleket tanítottak. A nehezebben felismerhető tulajdonságokra is fókuszáltak, mint például a mergerek. Ezek a típusok a teljes katalógus alig 7%-át teszik ki. Pozitív esetnek azt tekintették, ahol a merger valószínűsége nagyobb, mint 0,7 és annak a valószínűsége, hogy semmilyen furcsaság nem szerepel kisebb, mint 0,45. Ellenpéldának azokat vették, ahol a merger valószínűsége kisebb, mint 0,4 és a különlegesség valószínűsége kevesebb, mint 0,1 és legalább 10 szavazat érkezett arra, hogy nincs semmilyen különlegesség. Így összesen 1400 mintára csökkent a mergerek száma. Az elliptikus/spirálisnál 98,2%-os pontosságot értek el, mergerek esetén pedig 97%-ot [24].

3.1.3 Reziduális hálózat alapú megközelítés

CNN-on kívül más módszerrel is próbálkoztak galaxisok osztályozására, ilyen például a reziduális hálózat (továbbiakban ResNet). Hagyományos CNN-ek esetén a modell pontossága először növekszik újabb rétegek hozzáadásával, azonban egy idő után ez megfordul, és egyre pontatlanabbá válik. Itt a hiba oka azonban nem a túlillesztés. He és társai [25] által bemutatott reziduális hálózatok lehetővé teszik a rétegek közti rövidebb átjárást, ahogy az a 3.2. ábrán látható.



3.2. ábra. Reziduális hálózat rétegei [25].

A modell számára könnyűvé válik az identitásfüggvény felismerése. Így a rétegek számának növelésével nem csökken a pontosság, még abban az esetben sem, ha a reziduális blokkon belüli rétegek rontanának rajta, ilyenkor egyszerűen át lesznek ugorva ezek a rétegek. Kedvező esetben azonban új ismereteket gyűjthetnek, melyek tovább javítják a modell pontosságát. Többféle ResNet is létezik, melyeket elsősorban mélységük jellemez. Sokrétegű hálónál már 1×1 -es konvolúciók is alkalmazásra kerülnek, melyek képesek csökkenteni a számításigényt, azáltal, hogy csökkentik a bemenet dimenziószámát. Így például színes kép esetén a modell nem az összes színre, hanem csak a legfontosabbra végzi el a nagyobb méretű és költségesebb konvolúciót.

3.3. táblázat. Különböző ResNet-ek számítási igénye [25].

Rétegek száma	18	34	50	101	152
FLOPS ⁴	$1,8 \times 10^9$	$3,6 \times 10^9$	$3,8 \times 10^9$	$7,6 \times 10^9$	$11,3 \times 10^9$

Zhu és társai [26] galaxis morfológiák osztályozására egy módosított reziduális hálózatot alakítottak ki. Az adatokat az SDSS DR7 Galaxy Zoo Challengehez készített képekből vették. Majd Willet és társai [11] alapján egy szűkebb, megbízható mintát készítettek belőle. Mint ahogy a 3.3. táblázatban lévő FLOPS értékekből is látható, a nagyon mély ResNet-ek rendkívül számításiigényesek lehetnek, emiatt csökkentették a mélység számát és növelték a dimenziókat. A 3.4. táblázatban is látható hálózat a $64 \times 64 \times 3$ bemeneti képen először $64, 6 \times 6 \times 3$ méretű szűrőt alkalmaz, 1-es lépésközzel, mely korábban Dieleman és társainál [21] is megfelelőnek bizonyult. Az első konvolúciós réteg után egy 2×2 -es max-pooling réteg következik, 2-es lépésközzel, ami által a kép $32 \times 32 \times 64$ méretűvé válik. Ez a réteg lesz a bemenete 4 konvolúciós csoportnak, amelyek mind 2 reziduális blokkot tartalmaznak.

Az első csoport esetén, az egyik blokk $1 \times 1, 64 \times k, 3 \times 3, 64 \times k, 1 \times 1, 256 \times k$, míg a másik $1 \times 1, 64 \times k, 3 \times 3, 64 \times k, 1 \times 1, 256 \times k$ (2-es lépésközzel) típusú konvolúciókból épülnek fel, ahol k dimenzió növelő tényező. A következő csoport az előző csoport utolsó rétegét kapja bementként, tehát a második csoport első rétege $1 \times 1, 128 \times k$ méretű konvolúció és így tovább. Egyedül a legutolsó csoport, végső rétegében van eltérés, ahol kihagyásra kerül az alulmintavételezés, így a lépésköz itt marad 1. Az 1×1 -es rétegek a dimenzió csökkentése után növelik azokat, így a paraméterek számának csökkenésével felgyorsul a tanulási folyamat. A legutolsó réteg egy 4×4 -es ablakú average-pooling, aminek kimenete egy $1 \times 1 \times 4096$ méretű vektor. Ebből softmax alkalmazásával egy teljesen összekapcsolt réteget kaptak. Több száz esetet megvizsgálva arra jutottak, hogy a legoptimálisabb k érték a 2. Az aktivációs térképek a csoportokon belül megosztásra kerülnek egymással, majd az adott csoport utolsó rétegénél mérete megfelelődik és a szűrők száma duplájára nő.

3.4. táblázat. Zhu és társai neurális hálózatának felépítése [26].

Layer name	Output size	Depth
Conv 1	64×64	$6 \times 6, 64$
Max-pooling	32×32	$2 \times 2, \text{stride } 2$
Conv 2	16×16	$\begin{bmatrix} 1 \times 1 & 64 \times k \\ 3 \times 3 & 64 \times k \\ 1 \times 1 & 256 \times k \end{bmatrix} \times N$
Conv 3	8×8	$\begin{bmatrix} 1 \times 1 & 128 \times k \\ 3 \times 3 & 128 \times k \\ 1 \times 1 & 512 \times k \end{bmatrix} \times N$

⁴ FLOPS: Lebegőpontos műveletek másodpercenként. (Floating Point Operations Per Second)

Conv 4	4×4	$\begin{bmatrix} 1 \times 1 & 256 \times k \\ 3 \times 3 & 256 \times k \\ 1 \times 1 & 1024 \times k \end{bmatrix} \times N$
Conv 5	4×4	$\begin{bmatrix} 1 \times 1 & 512 \times k \\ 3 \times 3 & 512 \times k \\ 1 \times 1 & 2048 \times k \end{bmatrix} \times N$
Avg-pooling	1×1	$4 \times 4, 5 - d, \text{softmax}$

Több, korábbi elterjedt és sikeres CNN modellel összevették eredményeiket melyek a 3.5. táblázatban láthatók. Ez a modell bizonyult a legpontosabbnak 95,21%-kal, amelyet a ResNet-50 követ 94,69%-el. A többi modell is alig maradt el ezektől. Végző eredménynek 10 futtatás közül a legjobbat választották. Annak ellenére, hogy ezek a modellek nem galaxis felismerésére lettek tervezve, de mégis közel hasonló pontosságot értek el, mint az erre kifejlesztett ResNet [26].

3.5. táblázat. Különböző neurális hálózatok eredményei [26].

Model	Accuracy (%)	Precision	Recall	F1	AUC
Dieleman	94,6528	0,9455	0,9465	0,9456	0,9793
AlexNet	92,2569	0,9207	0,9226	0,9215	0,9809
VGG	93,6458	0,9348	0,9365	0,9353	0,9846
Inception	94,5139	0,9447	0,9451	0,9448	0,9852
ResNet-50	94,6875	0,9458	0,9469	0,9461	0,9823
Ours	95,2083	0,9512	0,9521	0,9515	0,9823

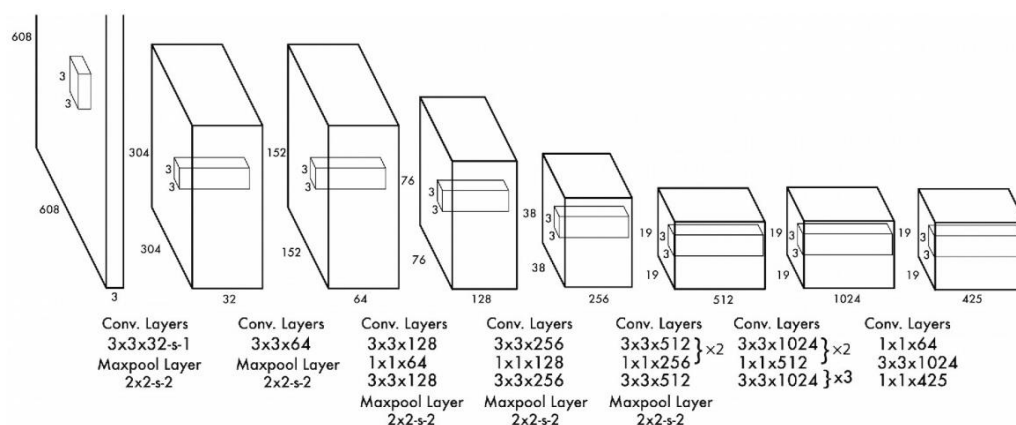
3.2 Objektumok detektálásnak lehetőségei képeken

Objektumok detektálásánál a neurális hálózat három részből épül fel. Két gyakori elterjedt megoldás is létezik, a Region Based Convolutional Neural Networks (továbbiakban R-CNN) és a You May Only Look Once (továbbiakban YOLO). A hálózat gerincét egy osztályozó háló alkotja, mely lehet például CNN vagy ResNet. Ezt követi egy nyak például Feature Pyramid Network (továbbiakban FPN), mely összeköti a fejvel, amely lehet például két teljesen összekapcsolt réteg. Minden objektumot egy határolókeretbe tesznek, melyeket 5 paraméter határoz meg: x és y a keret középpontjának helye, w és h a keret szélessége és magassága valamint az objektum kereten belüli létének valószínűsége. Eredményességüknek fontos mérőszáma a metszet az unió felett (IoU), mely az elvárt és becsült határolókeretek közti eltéréseket fejezi ki. A két módszer folyamatában lényegesek eltérések vannak.

Az R-CNN a teljes képet kis részekre bontja és mindegyikre lefuttatja az osztályozó algoritmust. Hátrány, hogy adott esetben a háttérzajt is objektumként érzékeli, valamint egyúttal rendkívül számításgényes és ezért lassú is mind a tanulás, mind a detektálás. Ez azonban galaxisok észlelésénél kevésbé jelenet problémát, mint valós idejű videók esetében.

A YOLO ezzel szemben a teljes képet figyelembe veszi, így elkerülve az előbb említett háttérzaj problémáját. Először a képen megkeresi az objektumot és azután végez csak klasszifikációt, ezáltal jóval gyorsabb is. A teljes képet egy $S \times S$ méretű mátrixra bontja fel. Állítható paraméterként a képrácsok mérete, valamint a rácsonkénti határolókeretek száma [27].

Galaxisok detektálása és osztályozása egyben is lehetséges például YOLO technikával, amire González és társai [28] tettek kísérletet. Adatok forrásának SDSS és Galaxy Zoo képeit használták. A 23 konvolúciós és 5 max-pooling rétegből álló hálózat felépítése 3.3. ábrán látható.

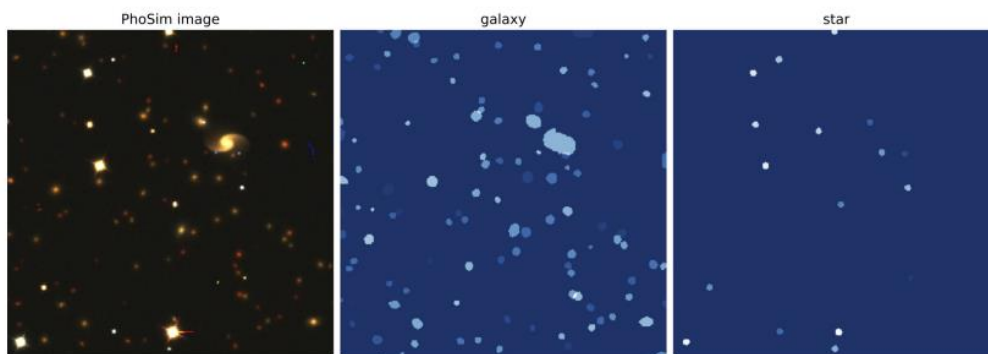


3.3. ábra. YOLO technikájú neurális hálózat, galaxis detektáláshoz [28].

Az egész projekt a Darknet keretrendszert használta fel. Elliptikusoknál több, mint 90%-os, míg spirálisoknál körülbelül 70%-os recallt ért el. Az IOU minden esetben 70% körül mozgott [28].

R-CNN alkalmazásával is léteznek kiváló megoldások. Burke és társai [29] galaxisok és csillagok detektálására tettek kísérletet ennek egy bővített változatával, a Mask R-CNN-el. Az első fázis szintén régiókat választ ki, azonban ezután, az R-CNN-től eltérően, párhuzamosan előállít egy bináris maszkot a határolókereteken és címkéken kívül. FPN sémára épül, ami a rétegek közötti feature továbbítást teszi lehetővé. Miután a megfelelő feature map kiválasztásra kerül a hálózat gerinceként szolgáló CNN-el egy Region Proposal Network (RPN) végig pásztázza a képet olyan területek után kutatva, ahol objektum jelenléte lehetséges. A határolókeretek különböző méretüknél fogva ronthatnak az osztályozó gerinc pontosságán. Ennek elkerülése végett egy RoIAlign réteg használható, amivel azonos méretű régiókat állít elő az osztályozó számára.

A hálózat gerinceként egy ResNet-101-et alkalmaztak. A tanítást szimulált képeken végezték el. Ehhez a PhoSim [31] szoftvert használták fel, mely képes valósághű spirális, elliptikus, irreguláris galaxisokat, valamint csillagokat is előállítani.

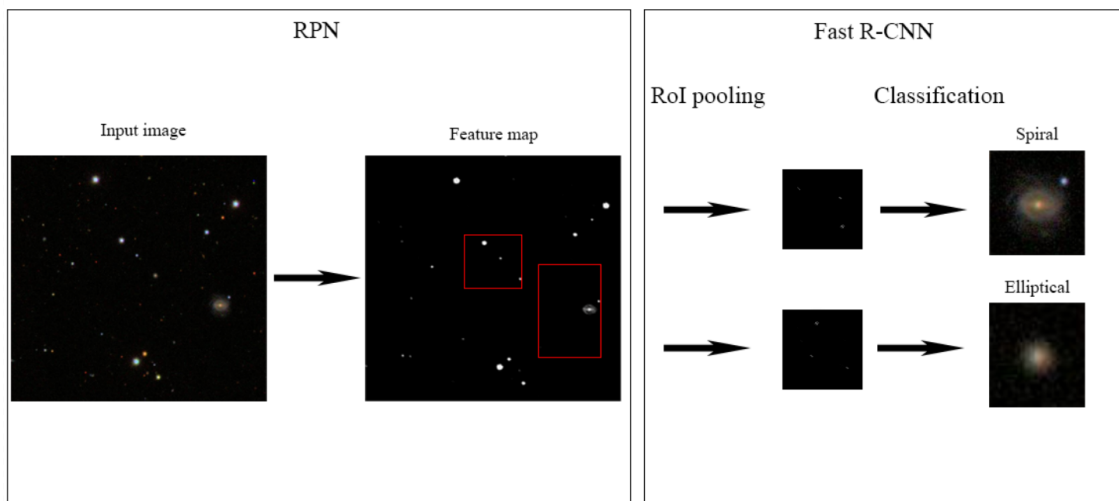


3.4. ábra. PhoSim által generált képek [31].

Ezer darab 512×512 pixeles képet állítottak így elő, melyen aztán több, szokásos adatbővítési technikát is alkalmaztak, mint forgatás, tükrözés. Emellett a galaxisokhoz és csillagokhoz generáltak egy a 3.4. ábrán is látható 512×512 pixeles g-hullámhosszú (464 nm körüli) képet, melyek ugyanazok, mint a színes kép esetén. Ezzel a galaxisok és csillagokhoz tartozó maszkot kapták meg. Validáláshoz még további 250 valóságú kép generálódott le. Egy már korábban Abdulla [32] által MS COCO képein betanított hálózat szűrőit rögzítették és tanították át. A folyamat során először a fej 0,001-es, majd a többi réteg 0,0001-es tanulási rátával tanult újra. Ezt később folyamatosan csökkentették 10^{-6} -ig. 50 epochig zajlott a tanítási folyamat. Veszteségfüggvényként bináris keresztentropiát alkalmaztak, mely 0,209 osztályozó és 0,208 határolókeret veszteséghez konvergált. Galaxisoknál az átlagos precízió 83,9%-ot ért el 0,5-ös IoU esetén. Összesen 6341 objektumot észlelt galaxisnak, ennek 99,38%-a valóban galaxis volt [29] [30].

3.3 Faster R-CNN

A következőkben, a korábban ismertetett R-CNN-re alapuló Faster R-CNN-t mutatom be részletesebben. Ez a modell két fő modulból épül fel. Egy RPN-ből és egy Fast R-CNN-ből, amint az a 3.5. ábrán látható. Mivel a kettőt együtt akarjuk futtatni, szükségünk van közös, osztott konvolúciós rétegekre is. Az RPN egy adott képen kijelöli azokat a régiókat, ahol az általunk keresett objektum valamelyike megtalálható és egy valószínűségi pontszámot rendel hozzájuk. Majd ezután a Fast R-CNN végzi el a konkrét objektum osztályozását.

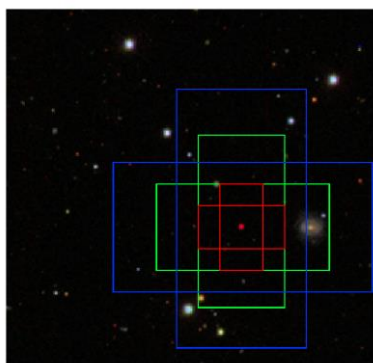


3.5. ábra. A Faster R-CNN felépítése [33].

A régiók megtalálásához egy kis csúszóablakot mozgatunk az utolsó, osztott konvolúciós réteg által készített feature map-en keresztül. A kijelölt régiók ezután két teljesen összekapcsolt rétegbe továbbítódnak, egy box-regressziós- és egy box-osztályozó rétegbe.

Minden csúszóablak régiónál több propozíciót is felkínálhat a modell, legyen ezek száma k . Ekkor a box-regressziós réteg $4k$ kimenetű lesz, amely k darab határolókeret koordinátáit tárolja. Az osztályozó réteg pedig $2k$ pontokból áll majd, melyek azt jelzik, hogy az adott propozíciónál objektum található vagy háttér. A box-regressziót feature mapeken hajtjuk végre, melyek tetszőleges méretű RoI poolingből származnak. Minden regresszor egy skáláért és méretarányért felelős és egymás súlyait nem osztják meg.

Egy anchor a csúszóablak középpontjában áll. Skála és méretarány hiperparaméterekkel rendelkezik, melyek alapesetben 3 értékűek. Így minden egyes pozíciónál 9 anchorot kapunk, ahogy az a 3.6. ábrán is látható. Az anchorok előnye, hogy transláció invariánsak, azaz ha egy képen egy objektumot valamilyen irányban mozgatunk, ugyanúgy képes felismerni a modell. Emellett más módszerekkel szemben kevésbé számításigényes is, mivel itt csak egyetlen bemeneti skálára van szükség az anchorok kiszámításához.



3.6. ábra. Egy adott ponthoz tartozó anchorok [33].

Az RPN tanítása során minden egyes anchorhoz egy bináris (1-es értékű igaz) osztálycímkét rendelünk, ami igaz értékével jelzi az objektumok jelenlétét. Két esetben rendelünk igaz értéket hozzá. Egyik a ground-truth box-al való legnagyobb IoU fedettség, a másik pedig azon anchorok, melyek IoU-ja legalább 0,7 bármelyik ground-truth box-al. Egyetlen ilyen határolókeret több anchorhoz is igaz címkét rendelhet. Hamis értéket kapnak azok az anchorok, amelyek még nem kaptak igaz címkét és IoU értékük kisebb, mint 0,3 az összes ground-truth box esetén. Azok az anchorok, melyek nem kaptak címkét, nem számítanak bele a tanításba.

RPN tanítása visszaterjesztéssel, mint pl. gradiensereszkedéssel lehetséges. Annak elkerülése végett, hogy negatív minták irányába torzítson, véletlenszerűen kiválasztunk adott számú (ez esetben 256) olyan anchorot, melyeknél a pozitív-negatív címék aránya 1:1. Amennyiben nincs 128 pozitív minta a képen, negatívokkal egészítjük ki.

Minden új réteget 0 átlagú Gauss eloszlással inicializálunk, melynek standard eloszlása 0.01. A rétegek (pl. osztott konvolúciós rétegek) ImageNet osztályzáshoz való betanítással jön létre. Tanulási rátának 0.001-et használtak 60k mini-batchnél és 0.0001-et a következő 20k mini-batchnél. 0.9-es momentumot és 0.0005-ös súlycsökkentést használtak.

Az RPN ImageNet-el való inicializálása és finomhangolása után az általa generált proposíciókat kapja meg a Fast R-CNN modell, amely szintén ImageNet-el kerül inicializálásra. Ezután a detektáló hálózatot használjuk az RPN tanítás inicializálásához, azonban rögzítjük az osztott konvolúciós rétegeket és csak az RPN réteghez egyedi rétegeket finomhangoljuk. Az osztott rétegeket rögzítve finomhangoljuk végül a Fast R-CNN egyedi rétegeit is.

Anchoroknál 3 skálát 128, 256, 512 és 3 méretarányt, 1:1, 1:2, 2:1 használtak [33]. Tanítás során azok az anchorok amelyek kilógnak a képből kihagyásra kerülnek, így nem befolyásolják a veszteségfüggvényt. Ha nem hagyjuk ki őket, az jelentős hibákhoz vezet, és a tanítás divergál. Tesztelés során meghagyjuk őket és a kép széleihez illesztjük a kilógó részeket. Az anchoroknál különböző számú skálát és méretarányt teszteltek, melynek eredménye a 3.6. táblázatban látható. Ebből az következik, hogy a pontossághoz leginkább a különböző skálák járulnak hozzá.

3.6. táblázat. Különböző anchorbeállítások [33].

settings	anchor scales	aspect ratios	mAP (%)
1 scale, 1 ratio	128 ²	1:1	65,8
	256 ²	1:1	66,7
1 scale, 3 ratios	128 ²	{2:1, 1:1, 1:2}	68,8
	256 ²	{2:1, 1:1, 1:2}	67,9
3 scales, 1 ratio	{128 ² , 256 ² , 512 ² }	1:1	69,8
3 scales, 3 ratios	{128 ² , 256 ² , 512 ² }	{2:1, 1:1, 1:2}	69,9

Az egymást átfedő RPN proposíciók csökkentése végett non-maximum suppressiont alkalmaztak. Ezzel a művelettel az átfedésben lévő proposíciók közül a legmagasabb értékű

kerül csak megtartása. Nem befolyásolta a pontosságot, azonban jelentősen csökkentette a régiók számát.

3.4 Összefoglalás

Már a konvolúciós neurális hálózatok forradalmának hajnalán lezajlott Galaxy Challenge verseny első tíz helyezettje is ilyen módszert használt. Az akkori győztesnél az újabb ResNet-50 is kicsivel jobb eredményt ért el. Ezenkívül Sánchez és társai [24] féle CNN a spirális/elliptikus, merger és gyűrűs galaxisok esetén is sokkal jobb precizitást ért el Dieleman és társainak hálózatahoz [21] képest.

Összevetve céloznak tehát leginkább egy Sánchez és társainak [24] megoldásához hálózatahoz hasonló CNN vagy ResNet felelni meg osztályozóként. Előbbi esetén többféle módosítás is érdekes lehet, úgymint a regularizációs módszer átalakítása vagy konvolúciós rétegek számának növelése.

Emellett ResNet is sikeres megoldás lehet. Sánchez és társainak [24] kiváló eredménye részben a jobb képadatbázisnak, részben pedig annak köszönhető, hogy az egyes feladatokhoz külön hálózatot tanítottak. Ezt a technikát alkalmazva ResNet esetén, valamint kihasználva, hogy általában a mélyebb hálózatok (például ResNet-101) pontosabb eredményekre képesek, megfelelő pontosságú hálózat építése lehetséges.

Detektáló hálózatnak R-CNN alapú megoldás tűnik jobbnak. Mivel itt álló képekről van szó, tehát a YOLO által nyújtott sebességnövekedés nem játszik fontos szerepet. Burke és társai által bemutatott [29] ResNet-101 osztályozót használva egy Mask R-CNN gerinceként, sikeresen alkalmazta azt galaxisok és csillagok elkülönítésére. Így akár céloknak megvalósítására egy hasonló modell megfelelő lehet.

4 TERVEZÉS

4.1 A szoftver célja és felépítése

A program fő feladata osztályozás, mely galaxis képeket kapva mindegyiket osztályozza és elmenti egy fájlba koordinátával és morfológiai információkkal együtt. A program elsősorban a spirális és elliptikus galaxisokat lesz képes elkülöníteni, emellett a gyűrűs szerkezetet is azonosítani tudja majd. Egy másik lehetősége a programnak, hogy egy adott csillagászati képet megadva bemenetként, azon megtalálja a galaxisokat, megjelölje és osztályozza is az előzőek alapján.

Szükséges eszközök:

- képletöltő
- égboltkép generáló
- osztályozó hálózatok
- detektáló hálózat
- eredmények mentése

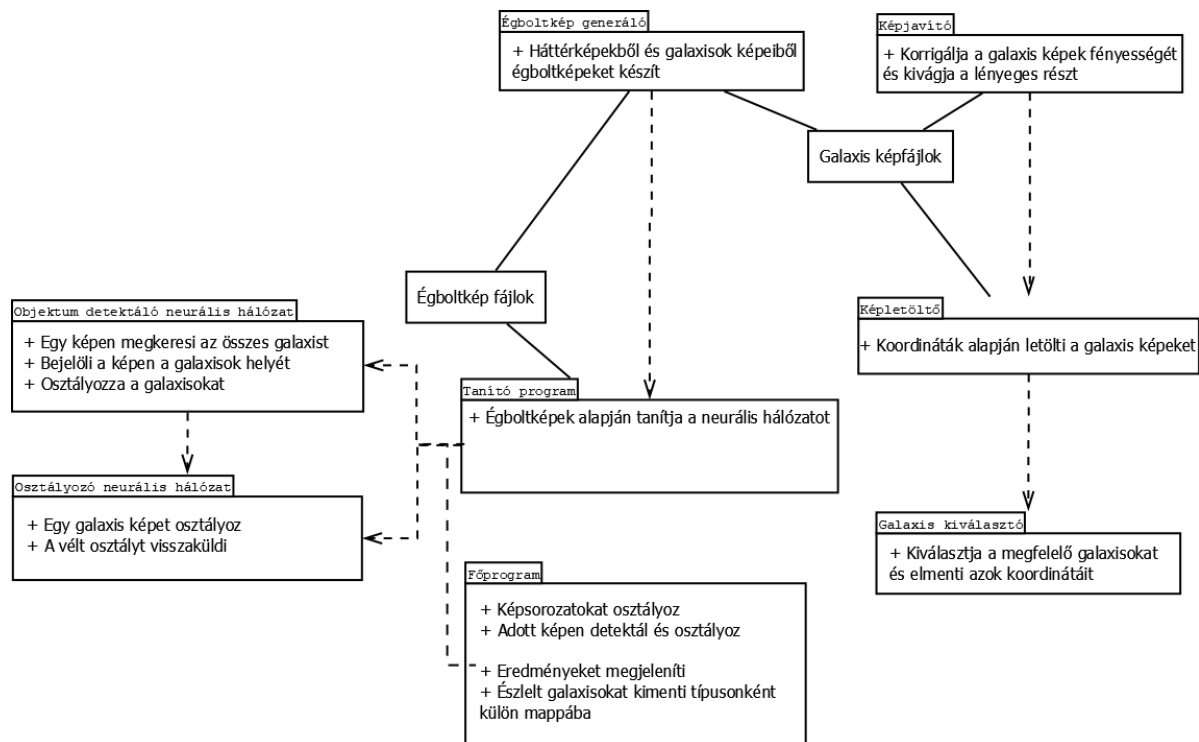
A képletöltő az SDSS szerveréről gyűjti majd be a tanításhoz szükséges képeket. A galaxisok szűrése is itt, a letöltés előtt történik. A teljes katalógusból a számomra érdekes, spirális, elliptikus és gyűrűs galaxisok kerülnek kiválasztásra.

A begyűjtött képeket az égboltkép generáló feldolgozza, majd egy mesterséges égboltképet készít belőlük. Ez amiatt fontos, mivel nehéz olyan SDSS képeket találni, amelyeken sok galaxis szerepel, emellett ezek osztályozása időigényes és a saját forrásból származó hibalehetőség is nagyobb. A folyamat felgyorsítása végett saját égboltképeket célszerű készíteni. Mivel én helyezem el a galaxisokat a képeken, így azok típusa és helye a generálás során egyből automatikusan mentésre kerül majd egy szöveges fájlba.

Az észleléshez és osztályozáshoz a korábbiak alapján R-CNN alapú hálózat kialakítását érdemes megpróbálni, ilyen pl. a Faster R-CNN. Ennek tanítása és éles alkalmazása CPU-val rendkívül időigényes, így GPU-t érdemes mindenképp használni. Jelenleg AMD típusú kártyával rendelkezem, így a lehetőségek kissé korlátozottak. A rendszer az 4.1. ábrán látható összetevőkből áll majd.

Egyik lehetőség az AMD által készített ROCm [34] használata. Ez elérhető Tensorflow és PyTorch esetén is, melyek rendelkeznek neurális hálózatokhoz szükséges könyvtárakkal. Azonban csak Linux rendszerre elérhető. Másik lehetőség Tensorflow és DirectML segítségével Windows alapon futtatás, azonban ez jelentősen elmarad teljesítményben az előzőhöz képest. Így elsősorban az előbbit érdemes megpróbálni alkalmazni. Minden esetben Python programozási nyelvre lesz szükség.

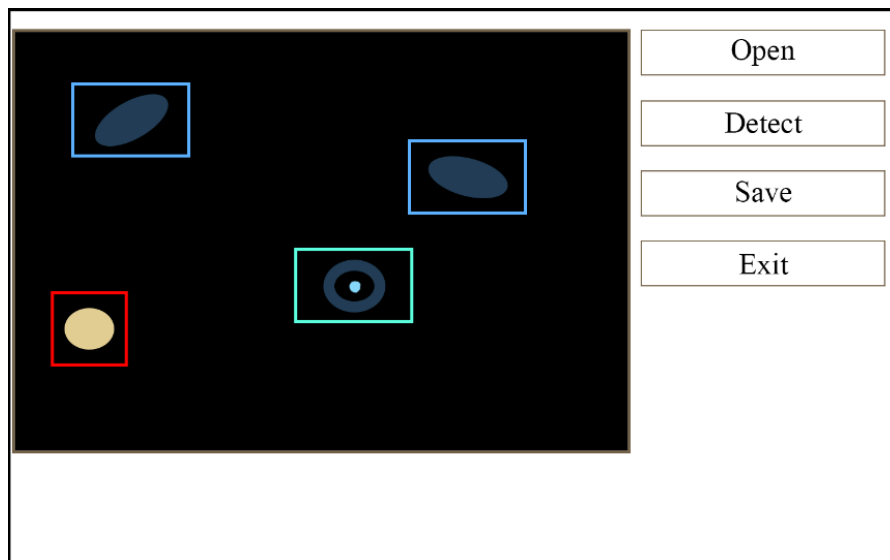
A kész betanított hálózat használatával egy adott képet megnyitva, azon észlelhetők lesznek a galaxis típusok. Amellett, hogy megjelenítődik egy ablakban, lehetőség lesz a képen szereplő galaxisokat külön – külön fájlokba menteni.



4.1. ábra. Rendszer komponensei

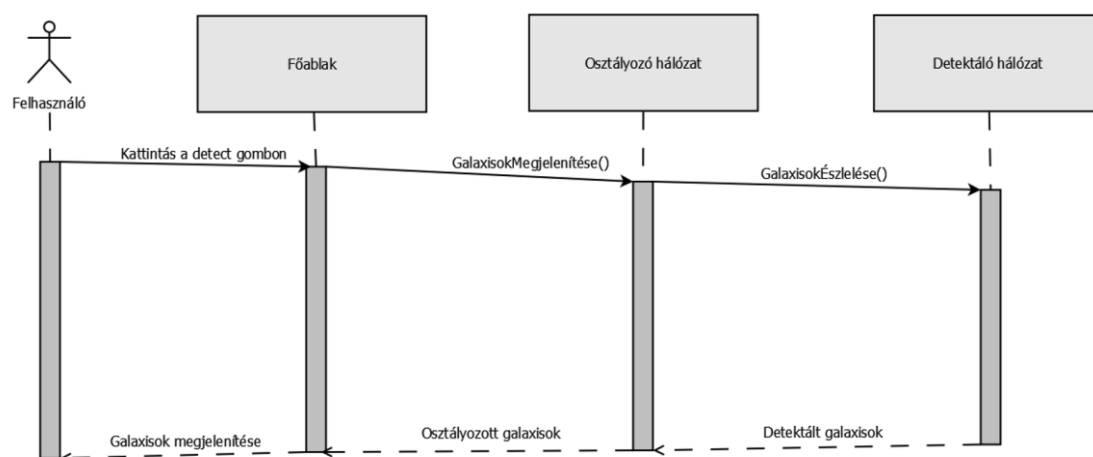
4.2 UML, UI/Wireframe

A program wireframe tervét mutatja a 4.2. ábra. Egy keretben látható lesz az általunk kiválasztott kép és rajta a program által detektált és osztályozott galaxisok típusuknak megfelelően bekeretezve. Ezután lehetőség lesz lementeni a megtalált galaxisokat, típusonként külön mappába.

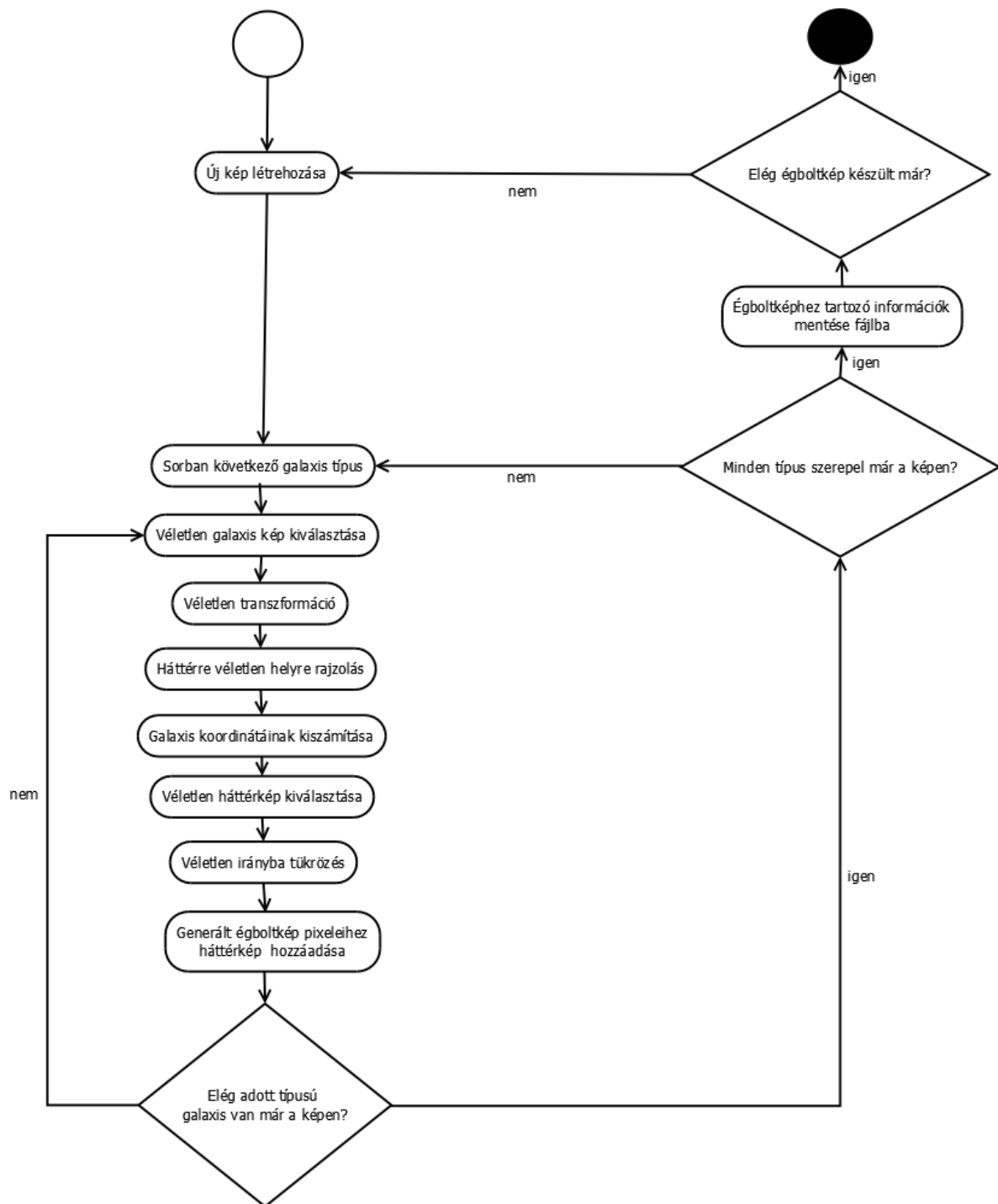


4.2. ábra. A tervezett program főablakának kinézete.

A felhasználói felület észlelés gombja (*Detect*) során lefutó lépéseket szemléletli a 4.3. ábra. Égboltképek generálásának folyamata pedig az alábbi 4.4. ábrán látható folyamat alapján zajlik le.



4.3. ábra: Egy képen szereplő galaxisok észlelése és megjelenítése



4.4. ábra. Égboltképek generálásának folyamata

5 FEJLESZTÉS

5.1 Rendelkezésre álló hardver

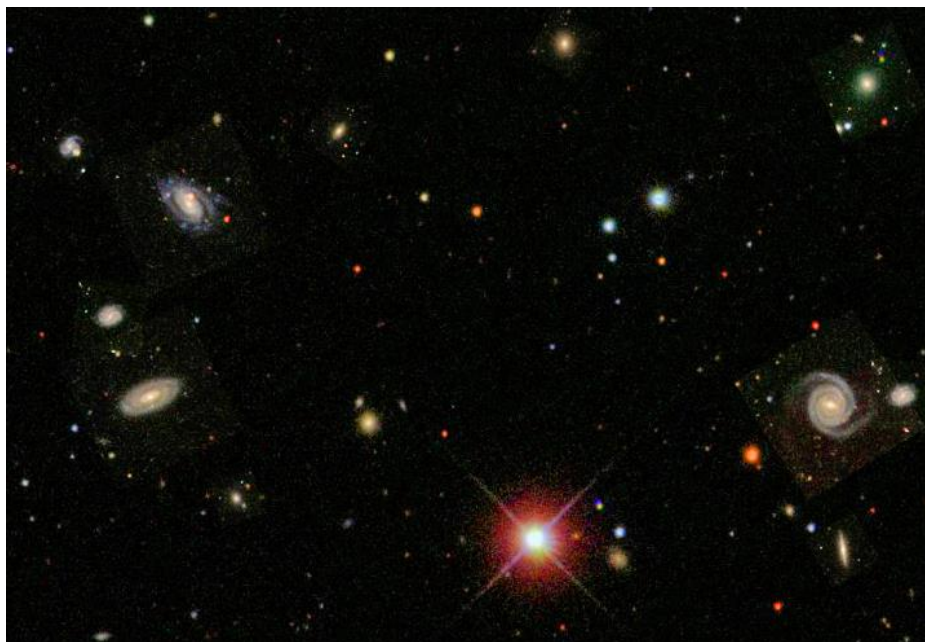
A szoftver fejlesztéséhez és működtetéséhez a következő komponenseket használtam fel:

- processzor: AMD Ryzen 5 3600X
- GPU: Nvidia GeForce RTX 3060 Ti
- memória: 16 GB

5.2 Adatok előkészítése

Az égboltképek készítéséhez először képeket gyűjtöttem azokról a galaxisfajtákról, amelyeket később osztályozni szeretnék. A galaxisokat a Galaxy Zoo 2 projektben használt SDSS képekből választottam ki. Ehhez Willet és társai [11] és Hart és társai [35] által torzításmentessé tett galaxis katalógust használtam fel. Olyan spirális, elliptikus és gyűrűs galaxisokat választottam ki, amelyek a flag jelző szerint egyértelműen az adott típushoz tartoznak, valamint amely a legtöbbek által került osztályzásra. Azonban átvizsgálva a képeket kiderült, hogy egy-egy így is téves kategóriába lett sorolva. Ezen tévesen osztályozott galaxisok száma bár elenyésző, eltávolításukkal kb. 5%-ot javult a modell pontossága.

Ezen a képeken azonban a galaxis mellett egyéb felesleges objektumok is láthatók, valamint az űr sem homogén fekete színű. Ennek akkor van jelentősége, ha egy adott háttérképre ezeket a képeket beillesztem, akkor láthatóvá válnak a galaxisról készült kép élei, mely alább az 5.1. ábrán látható.



5.1. ábra. Eredeti, korrigálatlan galaxisok képeiből generált égboltkép.

Ahhoz, hogy megszüntessem a problémát, először egy fényesség korrekciót alkalmaztam minden galaxis képre. Olyan függvényt kerestem, ami a magasabb értékeket kevésbé, míg a nullához közeli értéket jelentősen csökkenti. A logaritmust végül megfelelőnek találtam. A galaxis képen minden pixel esetén összeadtam a 3 szín értékét és megkerestem, mekkora a legnagyobb érték. Ez adta a logaritmus alapját. Majd minden pixel esetén az adott 3 szín értékének vettem a logaritmusát és az így kapott szín értékekkel írtam felül a pixeleket. Több kísérlet után arra jutottam, hogy ezt kétszer végrehajtva kapom a legjobb eredményt. Alább az 5.2. ábrán látható egy példa.



5.2. ábra. Egy galaxis képe korrigálás előtt (bal) és után (jobb).

A körülötte lévő objektumok eltávolításához készítettem egy segítő programot. Ez a képet rácsokra bontja, és minden négyzetrácshoz kiszámolja az átlagos pixelmennyiség értékét. Az így kapott mátrix közepétől indulva nyolc irányban egy adott küszöbértékig halad a keresés, mely a galaxis szélét jelzi. A galaxis ezen szélek mentén kivágásra kerül. A legtöbb kép esetén tökéletes megoldást ad, mindössze néhány kép esetén volt egy kicsit pontatlan, melyek könnyen észrevehetők és módosíthatók voltak. A végeredmény az 5.3.-as ábrán látható. Ezeket használja a program égboltképek generálásához.



5.3. ábra. Egy tanításhoz előkészített galaxis képe.

Jelenleg típusonként ~ 3500 kép készült, de későbbi szükség esetén, könnyen növelhető. Ezek a képek véletlenszerűen elforgatásra kerülnek 0 és 359 fok között majd átméretezésre $1/2$ és $1/6$ érték között. A generált égboltkép háttéréhez olyan képet kerestem az SDSS oldalán, melyen galaxis vagy nem, vagy elenyésző mennyiségben szerepel.

Jelenleg 22 háttérkép van, de ezek száma is növelhető igény esetén. A háttérképek is véletlenszerűen vízszintes vagy függőleges tükrözésre kerülnek. Ezekre helyezek el spirális és elliptikus galaxisokat, egyenlő arányban, jelenleg $12 - 12$ -t. Ezek mindegyike fele arányban tartalmaz gyűrűs változatot. Alább az 5.5. látható egy kész, generált égboltkép, mely az 5.4. ábrán lévő háttérrel használta alapul.

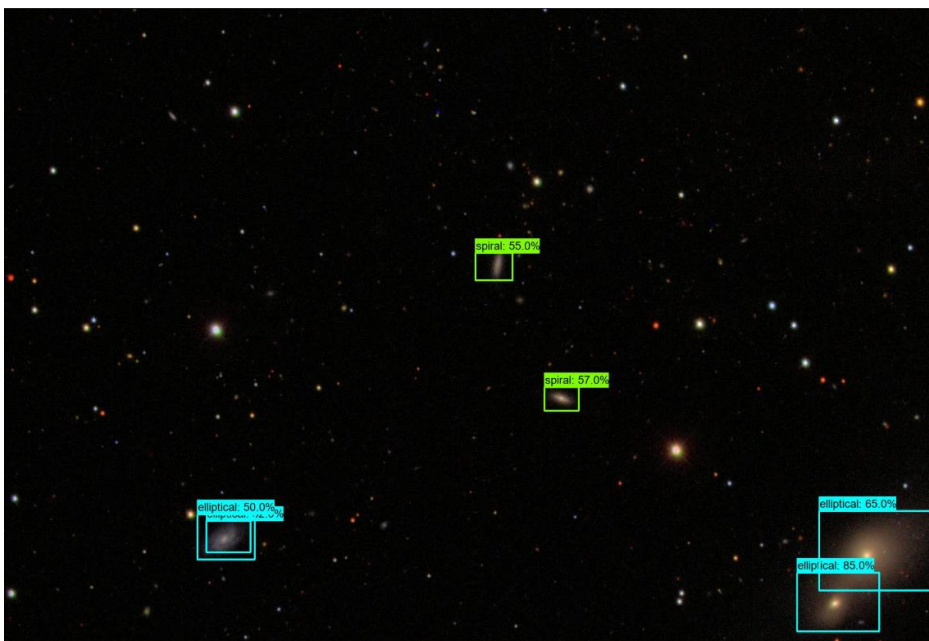


5.4. ábra. Égboltkép háttér, galaxisok nélkül.

Az így készült képeket és a rajta szereplő objektumok listáját kapja meg a detektáló hálózat. Ahhoz, hogy kiderüljön a generált képek megfelelők-e a tanításhoz, a Tensorflow Object Detection API-jával végeztem tanítást. Ennek hatalmas erőforrás igénye volt, így a képeket szélesség és magasság esetén is negyedére kellett csökkenteni, hogy lefusson. Kis mintaszámú adatokon és mindössze kb. fél óra tanítás után az alábbi 5.6. ábrán látható eredményt produkálta egy valódi, nem általam generált képen.



5.5. ábra. Generált égboltkép, galaxisokkal együtt.



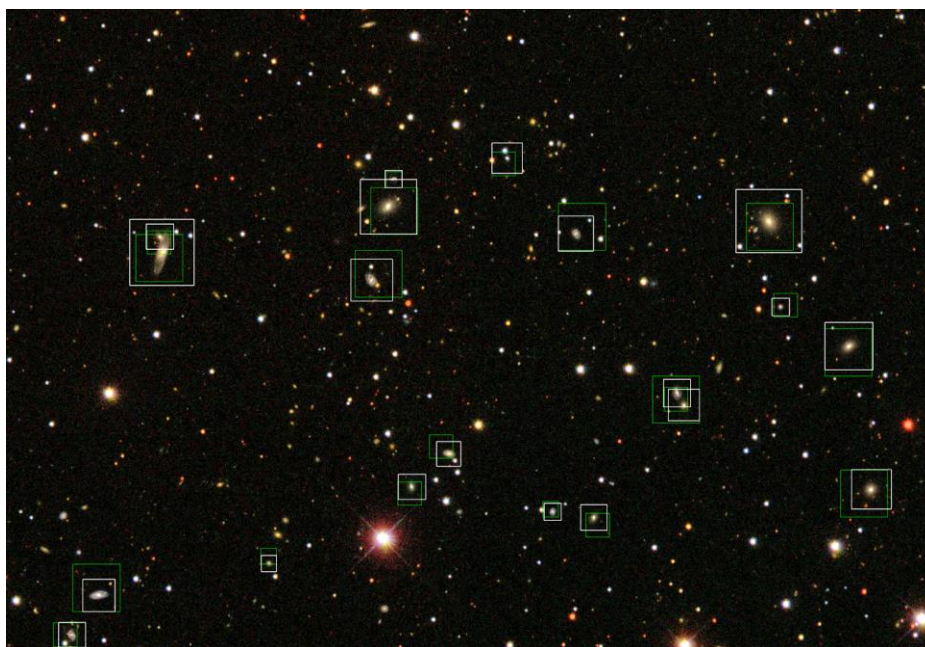
5.6. ábra. A Tensorflow Object Detection API Faster R-CNN hálózata által észlelt galaxisok.

Bár az egyik galaxis esetén tévedett, összességében biztatónak tűnik a technikai körülményeket tekintve. Így a detektáló modellem ilyen módon kapja a tanításhoz az adatokat.

5.3 RPN

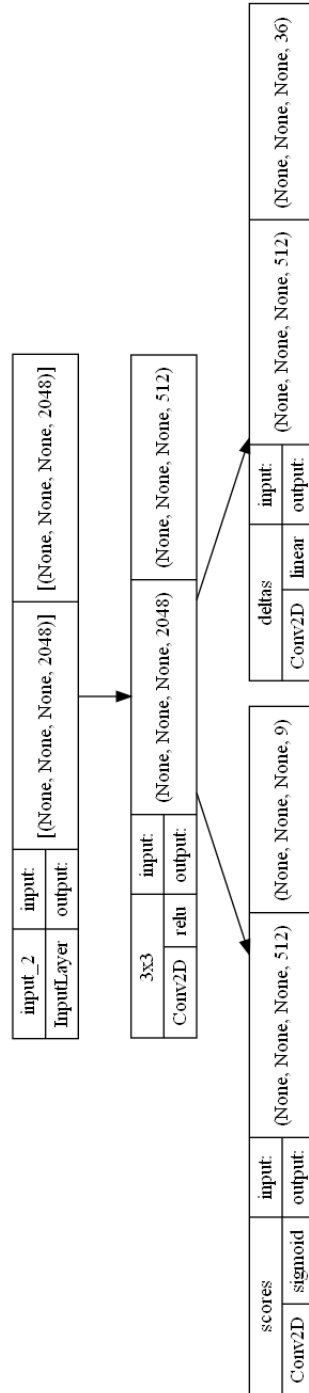
A kép először egy előre betanított CNN-en halad át. Az eredeti képen az anchorok akkora távolságra kerülnek elhelyezésre és eredetileg akkora méretűek, mint amennyivel a CNN által generált feature map kisebb az eredeti képnél. Minden anchorhoz 9 különböző anchor-box tartozik, 3 különböző méret (1,2,3) és 3 különböző méretarány (1:1, 1:2, 2:1).

A korábban ismertetett módszer szerint kerülnek az előtéri objektumok listájába való kiválasztásra azok az anchor-boxok, melyek leginkább hasonlítanak az eredeti ground-truth-boxra. Az alábbi 5.7. ábrán látható fehér színnel a ground-truth-boxok, zölddel pedig a pozitív anchor-boxok. Ahány pozitív előtéri anchor-box szerepel egy képen, ugyanannyi kerül kiválasztásra a tanításhoz a hátterek közül is.



5.7. ábra. Egy generált égboltképen szereplő galaxisok ground-truth-box-ai (fehér szín) és a hozzájuk előtéri objektumnak kiválasztott anchor-box-ok (zöld szín).

Az RPN hálózat 3 elemből épül fel. A CNN által készített feature map szolgál bemenetként, amit egy konvolúciós réteg követ. Ehhez kapcsolódik azután a box-regressziós és box-osztályozó réteg. Az elemek az 5.8. ábrán láthatók. A modell tanításához a mintaképekből 256 méretű batchet kap.



5.8. ábra. RPN hálózat felépítése

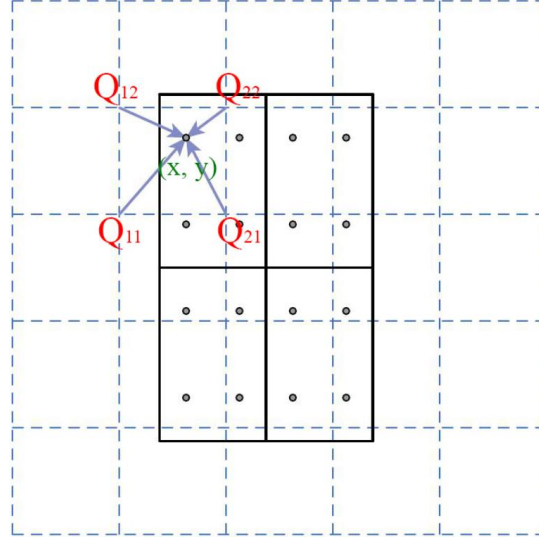
Annak meghatározása, hogy egy adott régióban szerepel-e objektum, bináris jellegű, ezért szigmoid aktiváció kerül itt alkalmazásra. A határolókeretekhez tartozó regresszió bármilyen értéket felvehet, ezért ezt lineáris aktiváció követi. Osztályozásnál 9 szűrő szerepel, minden határolókerethez egy, míg regresszió esetén pedig határolókeretenként 4, tehát összesen 36.

5.4 Fast R-CNN

Ahhoz, hogy az osztályozó hálózathoz ne kelljen minden egyes anchor-boxot megvizsgálnia, az RPN segítségével először egy szűrés fut le. Az RPN az összes anchor-boxhoz meghatároz egy számot, hogy mekkora eséllyel található ott objektum, valamint az anchor pozícióját és méretét milyen mértékben szükséges módosítani a pontosabb illeszkedés érdekében. Minden anchor-box koordinátái frissítésre kerülnek ezekkel az adatokkal. Ezután az objektumosságot jelző szám alapján növekvő sorba lesznek rendezve és az a felső réteg kerül megtartásra, amelyek nagyobb eséllyel tartalmazznak objektumot (top 3000). Azonban így még mindig több anchor-box is tartozhat egy objektumhoz, amelyek átfedik egymást. Ennek csökkentésére használható a non-maximum suppression algoritmus. Lényege, hogy minden anchor-boxnál megvizsgálja, hogy létezik-e egy másik anchor-box, mellyel a kettőnek átfedése elér egy adott határértéket. Amennyiben eléri és annak az objektumossága nagyobb, amivel összehasonlítjuk az aktuális elemet, akkor a jelenlegi anchor-box elvetésre kerül. Ha egyik, másik elemnél se fordul ez elő, akkor megtartjuk. Az így megmaradt anchor-boxhoz kiszámítjuk az átfedéseket a ground-truth-boxokkal és azokat választom ki az osztályozó réteg pozitív bemenetéhez, amelyek egy bizonyos IOU határértéket elérnek (pl. 0,7).

A pozitív elemek mellé kb. ugyanannyi negatívra is szükség van, ahhoz hogy a tanítás kiegyenlített legyen. Ezeket a korábban említett objektumosságot kifejező szám alapján sorba rendezett lista alsó rétegeiből veszem ki. Majd a pozitívokhoz hasonlóan a ground-truth-boxokkal való átfedéseket kiszámolom. Amennyiben egy bizonyos határérték alatt marad az IOU, akkor megtartása kerül. Itt fontos megemlíteni, hogy először az eredeti Faster R-CNN-nél [33] is használt 0,3-as határértékkel tanítottam a modellt. Azonban a pontossági értékek mindössze 75%-ot értek el. A probléma forrását vizsgálva kiderült, hogy a 0,3-as érték túl nagy, mivel előfordult, hogy egy-egy galaxisnál bár az IOU 0,3 – 0,4 között volt, ez egy galaxis közepét fedte le, amelyet a hálózat negatív, azaz háttérként kapott meg. Miután ezt a határértéket lecsökkentettem 0,01-re, a téves kijelölések megszűntek és a hálózat 90%-os pontosságot ért el.

Az osztályozó hálózat bemeneteinek 4 féle adatra van szüksége: feature map részlet, galaxis főtípusa, gyűrűvel rendelkezés állapota, anchor-boxnak ground-truth-boxtól való eltérése. A főtípus és a gyűrű adott, míg az utolsó az RPN-nél már megismert módon számítható. A feature map részlet számítása viszont nem triviális. A Fast R-CNN hálózat végén, két teljesen összekapcsolt réteg található. Ezen rétegeknek azonos méretű bemeneti adatokra van szüksége. Azonban az általunk kiválasztott anchor-boxokhoz tartozó feature map részletek mérete eltérő lehet. Ezért a kiválasztott részleteket át kell alakítani egységes alakra. Erre több módszer is létezik, mint pl. a RoI Pooling vagy RoI Align. Előbbi hátránya, hogy kétszer is történik adatvesztés. Először, amikor a koordinátákat elosztjuk a feature map stride-jával, hogy megkapjuk a régióhoz tartozó feature map indexeket. Ezután pedig amikor ezt egy adott méretű tömbbe akarjuk leképezni ismét az osztás egészrészével számolunk. Erre nyújt megoldást a RoI Align, melynek elvét szemléletesen az 5.9. ábra, mely a Mask R-CNN-nél [36] került bemutatásra.



5.9. ábra. A bilineáris interpoláció elve. He és társai [36] alapján.

Az adott régió annyi cellára bontódik fel, amekkora méretű feature map részleteket készítek (a képen 2×2 , a programban 3×3). Minden egyes cellából 4 ponton készül mintavételezés, melyet a 4.10. ábrán az (x, y) pont jelöl. Ezek értékét bilineáris interpolációval kapom meg a szomszédos feature map pontok értékeiből, melyek az 5.9. ábrán Q -val vannak jelölve és a számítás lépéseit az alábbi egyenletek írják le, ahol x_i és y_j az adott Q_{ij} ponthoz tartozó koordináták. Minden cellából a maximális érték kerül a végső feature map részletbe.

$$f(x, y_1) = \frac{x_2 - x}{x_2 - x_1} * f(Q_{11}) + \frac{x - x_1}{x_2 - x_1} * f(Q_{21}) \quad (3)$$

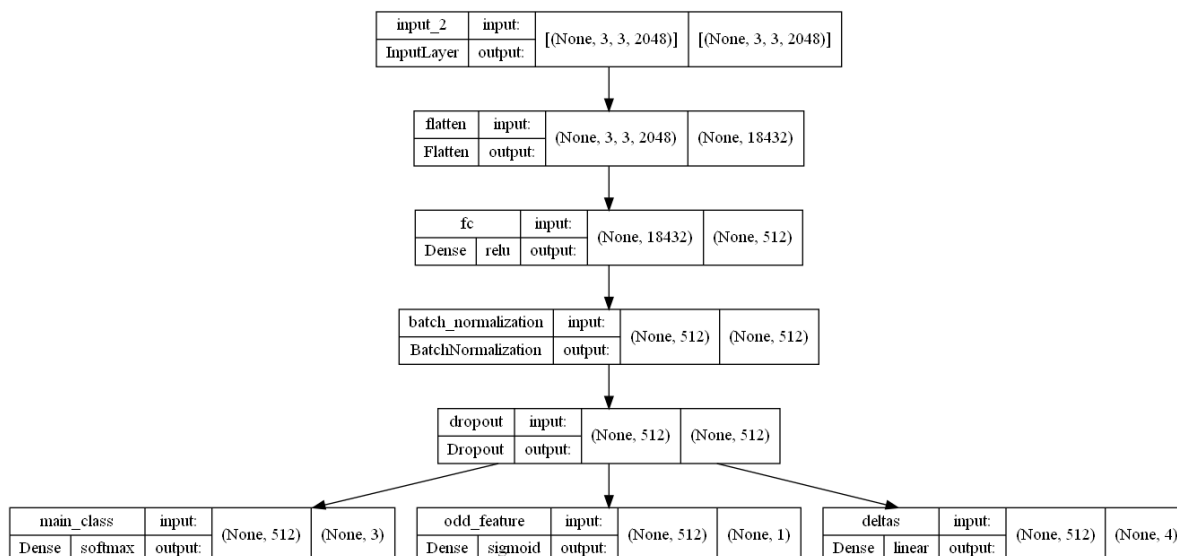
$$f(x, y_2) = \frac{x_2 - x}{x_2 - x_1} * f(Q_{12}) + \frac{x - x_1}{x_2 - x_1} * f(Q_{22}) \quad (4)$$

$$f(x, y) = \frac{y_2 - y}{y_2 - y_1} * f(x, y_1) + \frac{y - y_1}{y_2 - y_1} * f(x, y_2) \quad (5)$$

Ez az eljárás szintén elég számításigényes. 128 db 2048 mélységű batch $3 \times 3 \times 2048$ tömbbe alakításához közel 10 millió pontban kell az interpolációt elvégezni. Először egy függvénnyel kiszámoltam azokat a pontokat ahol interpolációt kell végezni, amiket a tényleges interpolálás során felhasználtam. Processzorral azonban ~20 másodpercbe telt ennek elvégzése, míg GPU segítségével ez 50 ms körüli időre volt csökkenthető, ami körülbelül 400-szoros gyorsítást jelent.

Ezeket az elkészített feature map részleteket kapja meg az osztályozó hálózat, melynek felépítése az alábbi 5.10. ábrán látható. Az első rétegek az eredeti Faster R-CNN szerinti felépítést követik. Ezután egy dropout réteg került hozzáadásra 0,25-ös értékkel, hogy kisebb legyen a túlillesztés. Ezt kapja meg végül a főkategória- és gyűrűsséget osztályozó, valamint a határolókeret regressziós rétege. Softmax aktivációt alkalmazok a főkategóriák megállapításához, mivel háttér, spirális vagy elliptikus lehet egy adott részlet, amik egymást kizárják. A gyűrűsség észlelése bináris feladat, ezért szigmoid aktiváció kerül itt alkalmazásra.

A határolókerethez tartozó számok pedig bármilyen értéket felvehetnek, akár negatívok is lehetnek, ezért itt egyszerű lineáris aktiváció szerepel.



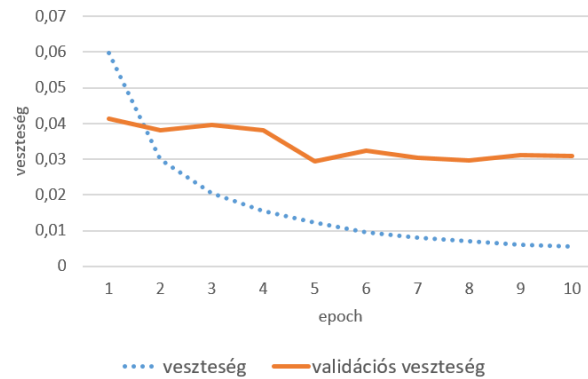
5.10. ábra. Fast R-CNN hálózat felépítése.

5.5 Hálózatok tanítása

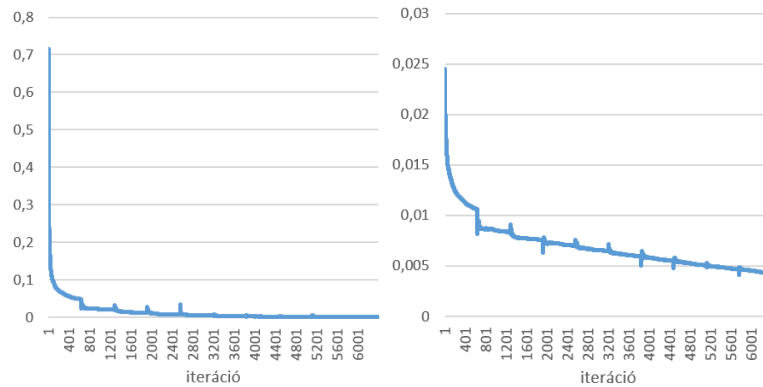
A tanításhoz 3 különböző adatkészletet hoztam létre: tanítás, validáció és teszt. A tanítási folyamat két részben történt, először az RPN került betanításra, majd annak kész súlyait használva a Fast R-CNN is. Learning ratenak a kis értéket választottam, 10^{-4} -ről indul és minden epoch alatt csökken, a teljes epoch szám végére 10^{-5} -t ér el. Ennél nagyobb értékeknél divergált a veszteségfüggvény. A tanítás során folyamatosan mérésre került a tanítási és validálási veszteség és pontossági értékek. Annyi epochon át futott a tanítás, amíg az utolsó 3 epochnál csökkenés volt tapasztalható a validálási veszteségnél.

A tanítás és inference gyorsítása érdekében bizonyos függvényeket GPU-ra írtam át. A határolókeretek átfedésének és non-maximum suppressionnek a számítása is jól párhuzamosítható. Mivel a GPU-k jobban kezelik a párhuzamos feladatokat, így jóval hamarabb végzi el a feladatot. Határolókeret esetén kb. 500x-os gyorsítás volt tapasztalható. Bár a végső inference esetén ez elegendő, azonban a tanításnál az RPN esetén a ResNet50-el, Fast R-CNN esetén pedig az RPN-vel való inference visszafogta a folyamatot. A teljes folyamat idejének szinte egészét ezek töltötték ki. Ezek elvégzése azonban már nem gyorsítható. Ahhoz, hogy a tanítás még gyorsabb legyen a hálózatok bemenetéhez szükséges tömbök elkészítése után azokat kötegekben lemezre is lementettem. Ezután minden epoch esetén ezekből a fájlokból kerülnek a tömbök beolvasásra. A hatékonyabb tanítás érdekében a tömbökben szereplő egyes tanítóminták sorait véletlenszerűen összekevertem.

Az RPN hálózat tanítása látható az 5.11., 5.12. ábrákon. A validációs veszteség az 5. epoch után már nem javul tovább, ezért az akkori állapot súlyai kerülnek mentésre, a túlillesztés elkerülése érdekében. Ezeket a súlyokat használja fel ezután az osztályozó hálózat.

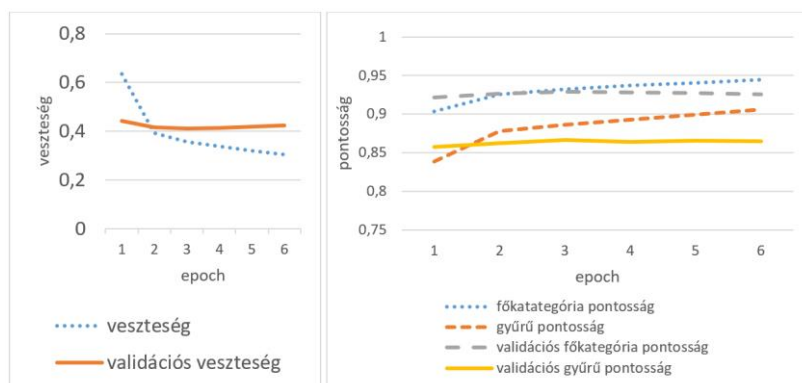


5.11. ábra. RPN hálózat tanítása során a teljes tanítási és validációs veszteségek.



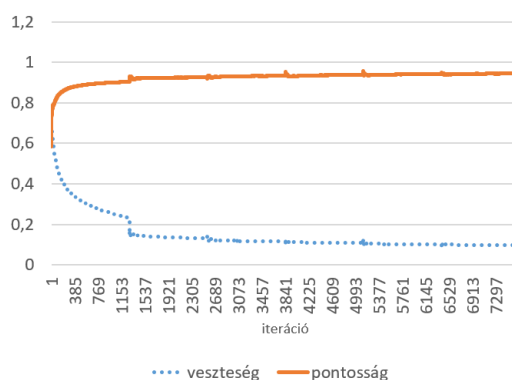
5.12. ábra. Az osztályozó (bal kép) és regressziós (jobb kép) réteg veszteség értékei a tanítás során.

Az R-CNN tanítása során a veszteség és pontosság értékeket monitoroztam. Ahogy az 5.13. ábrán látható, a harmadik epoch után a validációs veszteség növekedni kezd, ami túlillesztésre jellemző, ezért a súlyok értékei itt kerültek lementésre.



5.13. ábra. Fast R-CNN hálózat tanítása során a tanítási és validációs veszteségek (bal kép), valamint a fő kategória és gyűrűsség pontossági értékei (jobb kép).

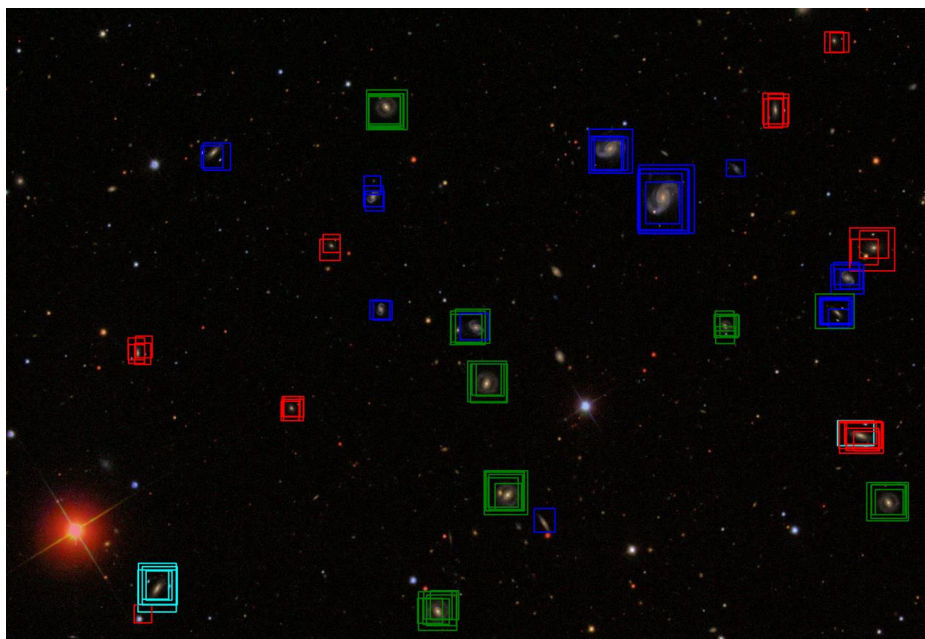
Az 5.14. ábrán látható a tanítási adatokon való tanításnál az egyes iterációkhoz tartozó értékek. Egy epoch 1280 iterációból áll. Jól látható tehát, hogy a hálózat már az első epoch végére megtanulja egész pontosan felismerni a galaxisokat, ezután már csak kis mértékben javul.



5.14. ábra. Fő kategóriás osztályok tanítási folyamata.

5.5.1 Utófeldolgozás

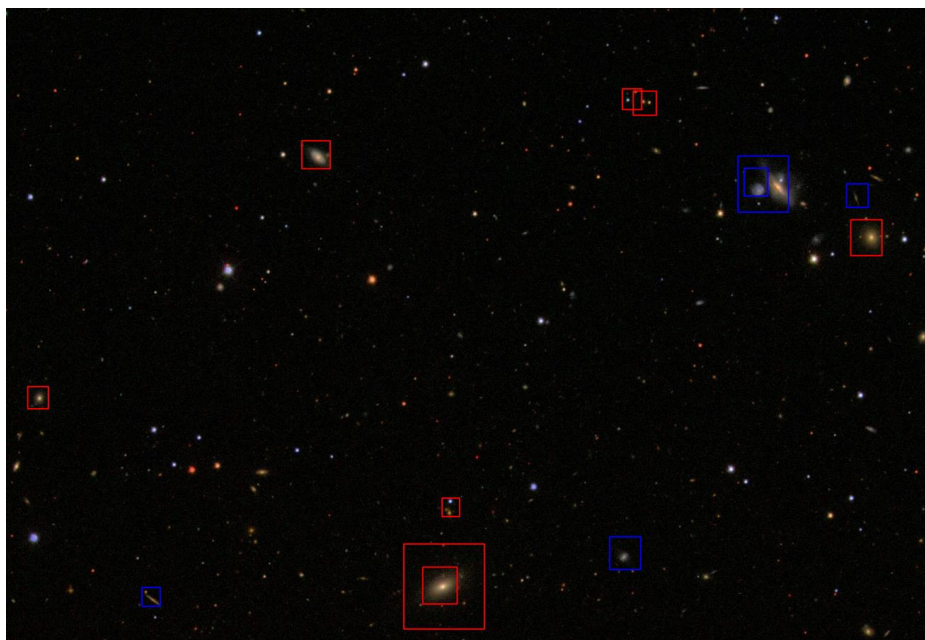
Az osztályozó hálózat kimenete tartalmazza az anchor-boxokhoz tartozó galaxis fő típusát, gyűrűsségét és az anchor-box alakjának módosításához szükséges számokat. Utóbbival frissítésre kerülnek az alakot leíró számok. Galaxisok megjelenítésénél és mentésénél azok lesznek figyelembe véve, amelyeknél a valószínűség egy bizonyos határértéket elér. Jelenleg fő típusnál 90%, gyűrűsségnél 75% ez az érték. Egy-egy átfedés még így is előfordult a kapott eredményeknél, ahogy az az 5.15-ös ábrán is látható, ezért erre ismét egy non-maximum suppressiont végeztem el, 0,5-ös határértékkel. Így már minden galaxisra csak 1 határolókeret tartozik, ahogy az az eredményeknél az 5.18-as is látható.



5.15. ábra. Generált égboltképen észlelt galaxisok. Az elliptikusok piros, a spirálisok kék, a gyűrűs elliptikus cián, a gyűrűs spirálisok pedig zöld színnel.

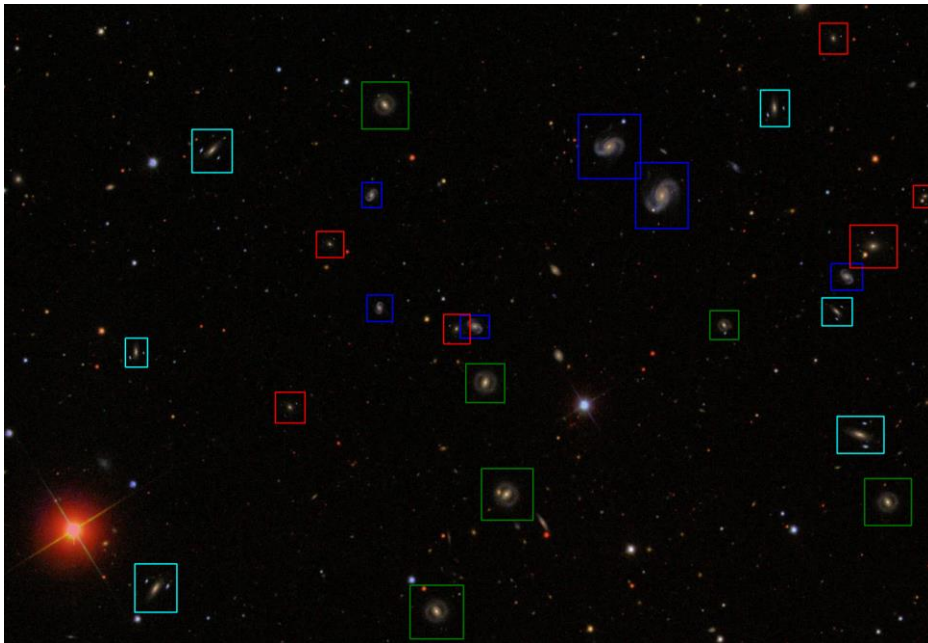
5.6 Faster R-CNN eredmények

Az alábbi 5.16. ábrán láthatók egy valódi, SDSS égboltképen felismert galaxisok. A tényleges galaxisok mindegyikét helyesen osztályozta, azonban 3 csillagnak tűnő objektumot elliptikus galaxisnak vélt tévesen.

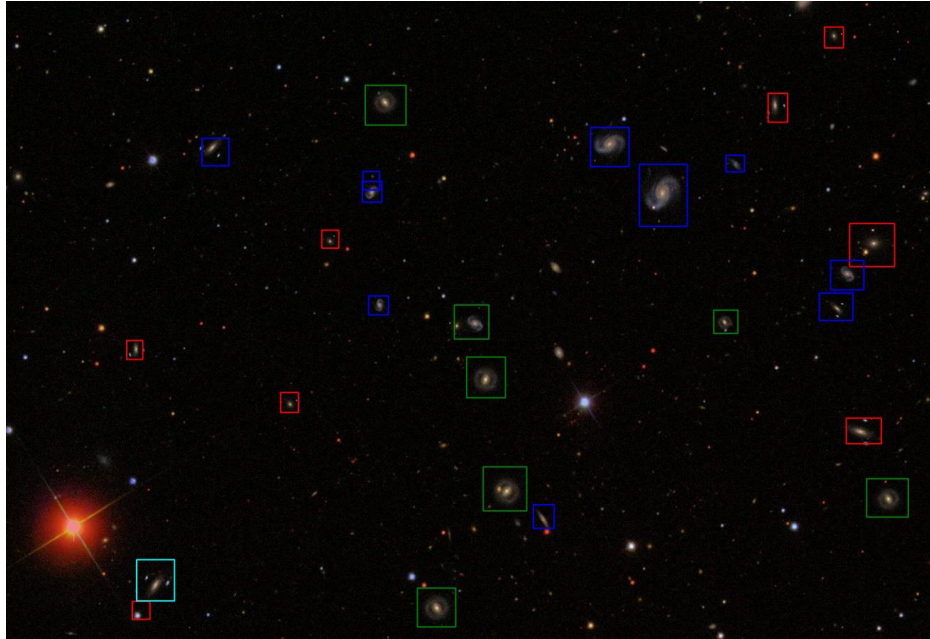


5.16. ábra. Egy SDSS égboltképen észlelt galaxisok. Az elliptikusok piros, a spirálisok kék színnel.

Az 5.18. ábrán egy általam generált teszt égboltképen felismert galaxisok láthatók. Mindegyik galaxist megtalálta és három csillagot észlelt galaxisnak tévesen. A főkategóriába is helyesen besorolta őket egy galaxist leszámítva. Gyűrűsség esetén (zölddel és ciánnal jelölt galaxisok) az tapasztalható, hogy ezeknél a spirálisokat helyesen megtalálja, azonban elliptikusok közül mindössze csak egyet ismer fel. Ennek oka az lehet, hogy egyrészt ezekből a galaxisokból igen kevés létezik. A teljes közel 240 ezer galaxist tartalmazó katalógusból, mindössze 100 bizonyult megfelelőnek, míg spirálisok esetén több ezer rendelkezésre állt, amikből végül 3246-ot használtam fel a tanításhoz. Másrészt nehézséget jelenthet a modell számára, hogy itt az eltérések nagyon csekélyek, emberek számára is sokszor nehezen megállapítható a gyűrű jelenléte. Érdekes megfigyelni továbbá, hogy az eredeti 5.16-os ábrán látható határolókeretekhez képest az észlelt 5.18-as ábrán szereplő határolókeretek egyes galaxisoknál pontosabban illeszkednek. Az eredeti képen az automatikus képalkotás miatt előfordul, hogy néha egy galaxis az égboltkép generálás során kissé pontatlan határolókeretet kap.



5.17. ábra. Generált égboltképen szereplő eredeti galaxisok. Az elliptikusok piros, a spirálisok kék, a gyűrűs elliptikus cián, a gyűrűs spirálisok pedig zöld színnel.



5.18. ábra. Generált égboltképen észlelt galaxisok. Az elliptikusok piros, a spirálisok kék, a gyűrűs elliptikus cián, a gyűrűs spirálisok pedig zöld színnel.

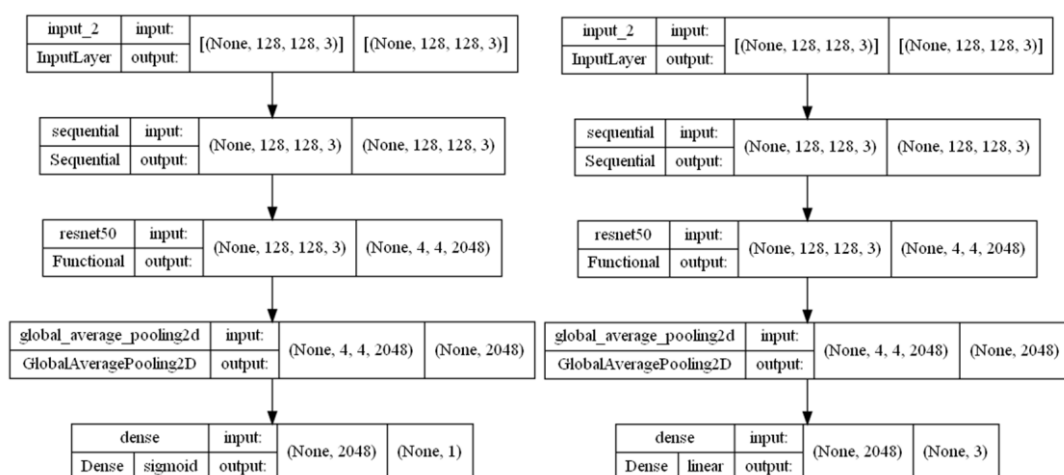
5.7 DBSCAN és CNN

A Faster R-CNN már egészen jó eredményeket mutatott, érdemes lenne megvizsgálni más módszerrel is a feladatot. A galaxisok égboltképen való helyének megállapítása akár DBSCAN [37][38] algoritmussal is lehetséges. Ezután az így kimetszett képszeletet felhasználva azonosítja a főkategóriát és a gyűrűs szerkezetet egy-egy CNN.

Az adatokon történt változtatás. Több kép került begyűjtésre. Főkategóriánként 10 – 10 ezer galaxis, míg gyűrűs és gyűrű nélküli esetén 5 – 5 ezer galaxist választottam ki. Mindegyik galaxishoz elérhető az SDSS Data Release 7 és Data Release 9-beli kép is. A kettő között expozíciós és színbeli különbségek figyelhetők meg. Mindkét fajtát letöltve, így a teljes adatállomány duplájára nőtt. A tanítás, validálás és teszteléshez az adatokat 70%, 30%, 10% arányban bontottam fel.

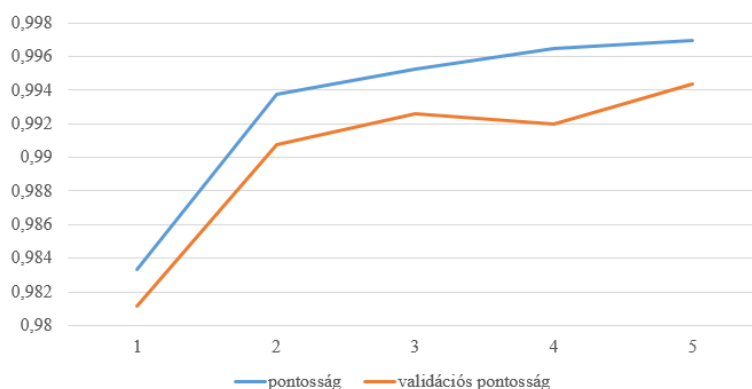
A DBSCAN egy klaszterezési módszer, ami az egymással összetartozó pontokat keresi meg. Egyik hátránya, hogy jelentős a számításgigéje, azonban futási sebessége a neurális hálózathoz mérhető. Másik hátránya pedig, hogy tisztán intenzitáson alapul, így a kellően nagy és fényes csillagokat is a galaxisok közé veszi. Ez azonban csak annyit jelent, hogy a főkategória tanításánál az elliptikus és spirális mellett háttér észlelésére is tanítani kell a modellt. A DBSCAN-t meghatározó paraméterek: intenzitási küszöbérték, körülvevő pontok száma, belső ponttól vett sugár. Többféle variáció kipróbálása után a következő számok bizonyultak a legmegfelelőbbnek: 50, 8, 2. Azok a clusterek, amelyek szélessége vagy magassága kevesebb, mint 32 pixel, elvetésre kerültek.

A főkategóriát és gyűrűt osztályozó hálózat egy-egy CNN, melyek felépítése az 5.19. ábrán látható. Minden kép 128×128 pixel méretűre lett átméretezve, ezek a hálózat bemenetei. Tanítás során a következő augmentációk lettek végrehajtva: tükrözés, forgatás, nagyítás/kicsinyítés, négyirányú eltolás és kontraszt változtatás. Ezek közül az eltolás eredményezte a legnagyobb mértékű javulást a teljes szoftver tesztelése során. Ennek oka az lehet, hogy a CNN-ekhez letöltött képeken szereplő galaxisok mindig teljesen középen találhatók, azonban a DBSCAN által észlelt és kivágott galaxisok néha egy kissé el vannak attól csúszva. Az így módosított képeket kapja meg ResNet50 hálózat. Főkategóriák esetén 3 különböző osztály van, ezért veszteségfüggvénynek a kategorikus keresztentrópiát használtam. Gyűrűknél bináris jellegű a feladat, ezért itt szigmoid aktivációs függvényt és bináris keresztentrópiát alkalmaztam.



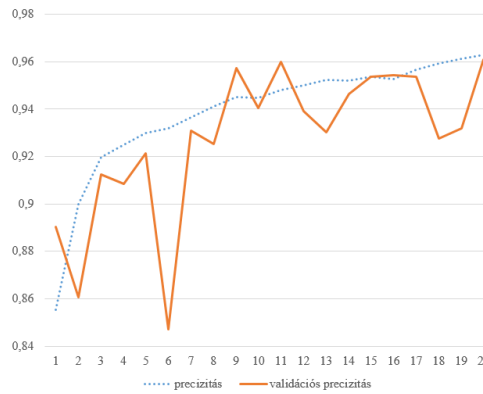
5.19. ábra. A két CNN hálózat felépítése.

A főkategória osztályozó hálózat tanítási folyamata az 5.20. ábrán látható. 5 epochig futott és a legjobb pontosságú súlyok kerültek megtartásra.



5.20. ábra. A főkategória osztályozó hálózat tanítása.

A gyűrűt osztályozó hálózat 20 epochon át került tanításra, ami közben kapott értékek az 5.21. ábrán láthatók. Ez esetben a legjobb precizitású súlyok kerültek mentésre.



5.21. ábra. A gyűrű osztályozó hálózat tanítása.

5.8 Eredmények

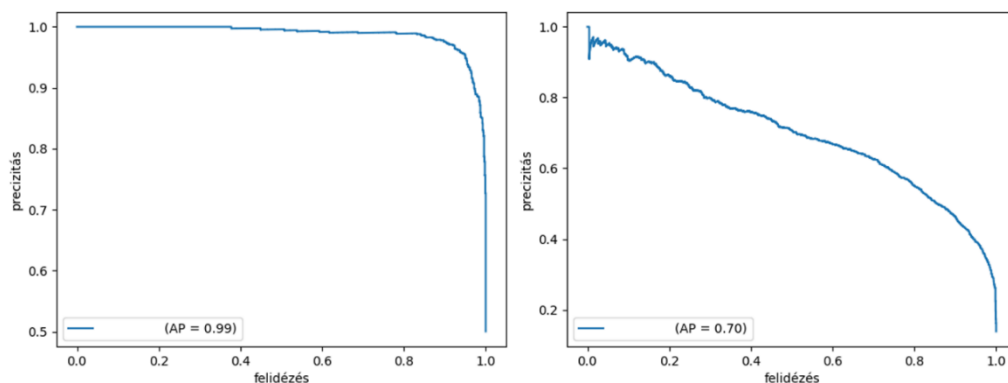
A kész, betanított hálózatok eredményei az alábbi 5.1. táblázatban láthatók. Az értékek alapján az új módszer minden téren sokkal jobban teljesít.

5.1. táblázat. A hálózatok összehasonlítása.

Modell	főosztály pontosság	gyűrű pontosság	gyűrű precizitás	gyűrű felidézés	gyűrű F1 score
CNN	99,70%	94,90%	95,72%	94,00%	94,91%
Faster R-CNN	92,81%	89,28%	58,13%	74,33%	65,24%

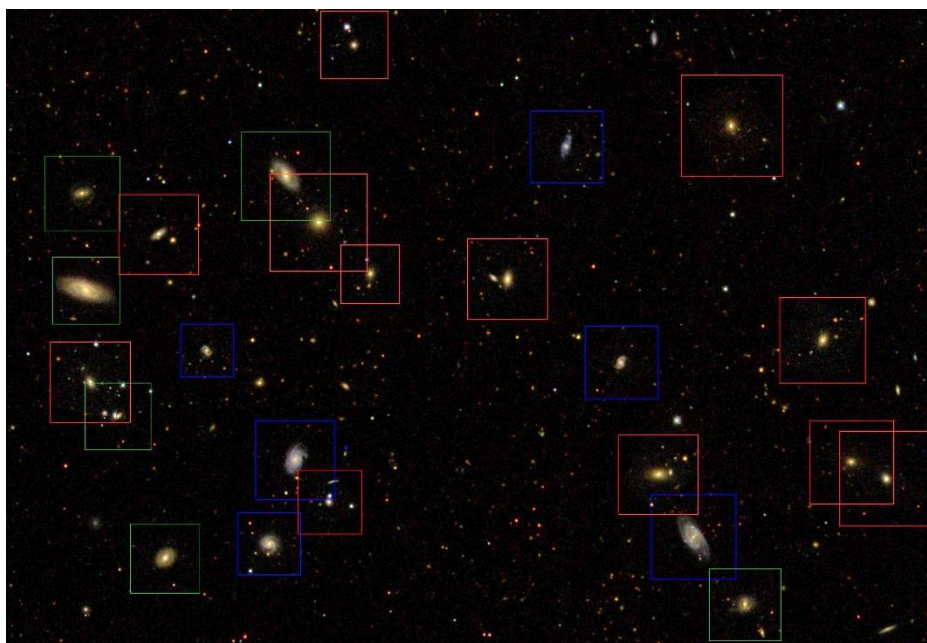
A gyűrűk esetén a nagy különbség oka az is, hogy a tanítás során a gyűrű és gyűrű nélküli galaxisok száma egyenlőtlen. Ez abból fakad, hogy elliptikus galaxisok esetén csak gyűrű nélküliek állnak rendelkezésre. Így ha főosztályonként egyenlítjük ki a mintákat, gyűrűk esetén egyenlőtlenség keletkezik. Kipróbálásra került ennek fordított esete is, mikor gyűrűsség alapján történt a kiegyenlítés, ekkor bár nőtt a precizitás és felidézés, azonban a főosztályok azonosítása túlságosan leromlott. Ezért Faster R-CNN esetén szükség lenne egy újabb hálózat hozzáadására, ami a gyűrűsséget állapítaná meg.

A két módszer eredményei közti különbség az 5.22. ábra precizitás – felidézés görbéjén is megfigyelhető.

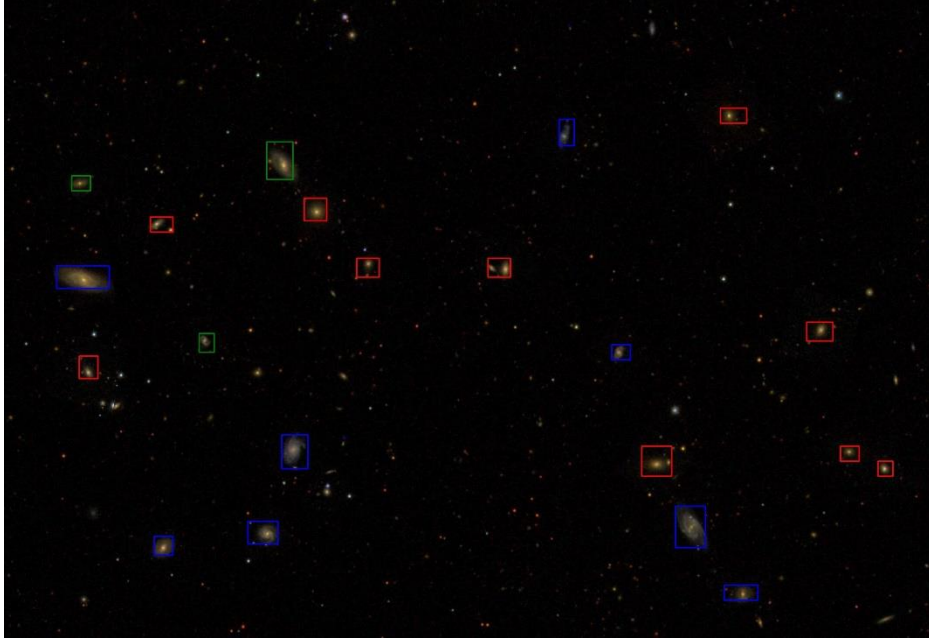


5.22. ábra. A két osztályozó hálózat precizitás – felidézés görbéje. A bal képen az új CNN, a jobb képen pedig a Faster R-CNN.

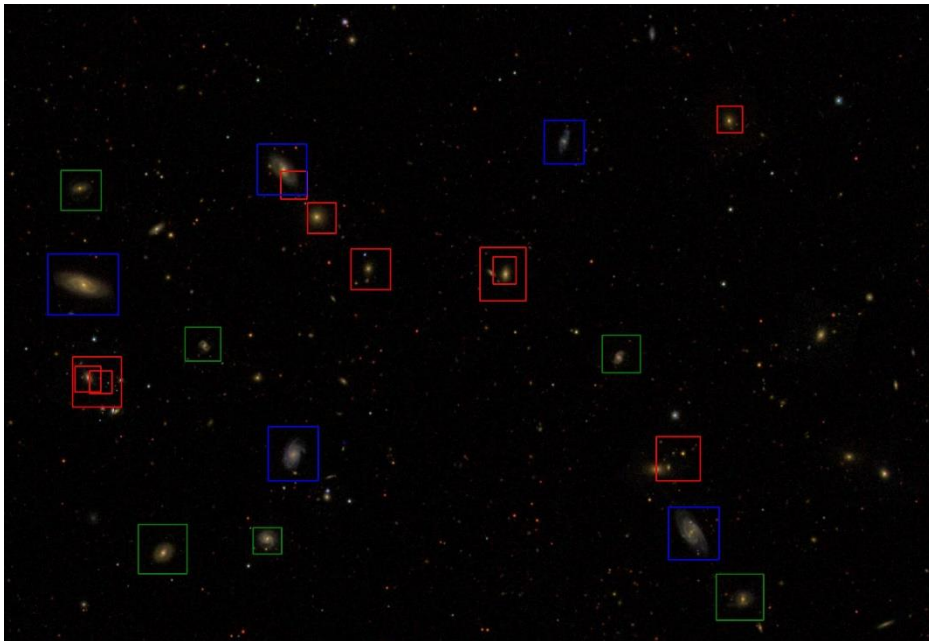
A következő 5.23., 5.24. és 5.25. ábrákon láthatók a hálózatok eredményei egy tesztképen. Megfigyelhető, hogy a DBSCAN féle módszer több galaxist ismert fel. Gyűrűk esetén a Faster R-CNN hármat talált meg a hatból, azonban három gyűrű nélkülit is annak azonosított. A DSBCAN módszer csak kettő gyűrűs galaxist észlelt, azonban mindössze egy spirális esetén jelzett gyűrűt tévesen.



5.23. ábra. Generált égboltképen szereplő eredeti galaxisok. Az elliptikusok piros, a spirálisok kék, a gyűrűs elliptikus cían, a gyűrűs spirálisok pedig zöld színnel.



5.24. ábra. Generált égboltképen DSBCAN és CNN által észlelt galaxisok. Az elliptikusok piros, a spirálisok kék, a gyűrűs elliptikus cián, a gyűrűs spirálisok pedig zöld színnel.



5.25. ábra. Generált égboltképen Faster R-CNN által észlelt galaxisok. Az elliptikusok piros, a spirálisok kék, a gyűrűs elliptikus cián, a gyűrűs spirálisok pedig zöld színnel.

6 ÖSSZEFOGLALÁS

Dolgozatom célja galaxisok észlelése és osztályozása volt égboltképeken. A főkategória (spirális és elliptikus) meghatározása mellett a galaxis gyűrű felismerése is cél volt. A szakirodalom áttekintése alapján neurális hálózatokkal legcélszerűbb ezt megvalósítani. Ezek közül is a Faster-RCNN tűnt a legjobb megoldásnak, ezért először egy ilyen jellegű hálózatot építettem fel. Tanításához készítettem egy programot, ami megfelelő galaxis képeket gyűjt be és azokból égboltképeket generál. Az észlelt és osztályozott galaxisok megjelenítésére és mentésére egy grafikus felületű program is készült. A Faster R-CNN az első teszteredmények alapján főkategóriáknál ~93%, míg gyűrű felismerésénél ~86% pontosságot ér el. Ezután kialakítottam egy másik módszert, aminél az észlelést DBSCAN-nel, az osztályzásokat pedig egy-egy CNN-nel végeztem el. Az adatok is bővítésre kerültek, amivel újratanítottam a Faster R-CNN-t is. A két módszert összehasonlítva az újabb DBSCAN és CNN alapú bizonyult a jobbnak, amely 99,7%-os főosztály, 94,9%-os gyűrű pontosságot, valamint gyűrű esetén 94,91%-os F1 score-t ért el.

7 ÁBRAJEGYZÉK

1.1. ábra. Hubble hangvillája [1].	2
2.1. ábra. Galaxy Zoo kérdésfa részlet [10].	5
2.2. ábra. Szavazati arányok [11].	5
2.3. ábra. Gyűrűs galaxisok aránya [11].	6
2.4. ábra. Konvolúciós neurális hálózat járművek felismerésére [16].	7
2.5. ábra. Konvolúció folyamata.	7
2.6. ábra. Zero padding alkalmazása.	8
2.7. ábra. Dropout folyamata [18].	9
3.1. ábra. Neurális hálózat felépítése [24].	13
3.2. ábra. Reziduális hálózat rétegei [25].	14
3.3. ábra. YOLO technikájú neurális hálózat, galaxis detektáláshoz [28].	17
3.4. ábra. PhoSim által generált képek [31].	18
3.5. ábra. A Faster R-CNN felépítése [33].	19
3.6. ábra. Egy adott ponthoz tartozó anchorok [33].	19
4.1. ábra. Rendszer komponensei	23
4.2. ábra. A tervezett program főablakának kinézete.	24
4.3. ábra: Egy képen szereplő galaxisok észlelése és megjelenítése.	24
4.4. ábra. Égboltképek generálásának folyamata	25
5.1. ábra. Eredeti, korrigálatlan galaxisok képeiből generált égboltkép.	26
5.2. ábra. Egy galaxis képe korrigálás előtt (bal) és után (jobb).	27
5.3. ábra. Egy tanításhoz előkészített galaxis képe.	28
5.4. ábra. Égboltkép háttér, galaxisok nélkül.	28
5.5. ábra. Generált égboltkép, galaxisokkal együtt.	29
5.6. ábra. A Tensorflow Object Detection API Faster R-CNN hálózata által észlelt galaxisok.	29
5.7. ábra. Egy generált égboltképen szereplő galaxisok ground-truth-box-ai (fehér szín) és a hozzájuk előtérri objektumnak kiválasztott anchor-box-ok (zöld szín).	30
5.8. ábra. RPN hálózat felépítése	31
5.9. ábra. A bilineáris interpoláció elve. He és társai [36] alapján.	33
5.10. ábra. Fast R-CNN hálózat felépítése.	34
5.11. ábra. RPN hálózat tanítása során a teljes tanítási és validációs veszteségek.	35
5.12. ábra. Az osztályozó (bal kép) és regressziós (jobb kép) réteg veszteség értékei a tanítás során.	35
5.13. ábra. Fast R-CNN hálózat tanítása során a tanítási és validációs veszteségek (bal kép), valamint a főkategória és gyűrűsség pontossági értékei (jobb kép).	36
5.14. ábra. Főkategóriás osztályok tanítási folyamata.	36
5.15. ábra. Generált égboltképen észlelt galaxisok. Az elliptikusok piros, a spirálisok kék, a gyűrűs elliptikus cían, a gyűrűs spirálisok pedig zöld színnel.	37

5.16. ábra. Egy SDSS égboltképen észlelt galaxisok. Az elliptikusok piros, a spirálisok kék színnel.....	37
5.17. ábra. Generált égboltképen szereplő eredeti galaxisok. Az elliptikusok piros, a spirálisok kék, a gyűrűs elliptikus cián, a gyűrűs spirálisok pedig zöld színnel.....	38
5.18. ábra. Generált égboltképen észlelt galaxisok. Az elliptikusok piros, a spirálisok kék, a gyűrűs elliptikus cián, a gyűrűs spirálisok pedig zöld színnel.	39
5.19. ábra. A két CNN hálózat felépítése.	40
5.20. ábra. A főkategória osztályozó hálózat tanítása.	40
5.21. ábra. A gyűrű osztályozó hálózat tanítása.	41
5.22. ábra. A két osztályozó hálózat precizitás – felidézés görbéje. A bal képen az új CNN, a jobb képen pedig a Faster R-CNN.	42
5.23. ábra. Generált égboltképen szereplő eredeti galaxisok. Az elliptikusok piros, a spirálisok kék, a gyűrűs elliptikus cián, a gyűrűs spirálisok pedig zöld színnel.....	42
5.24. ábra. Generált égboltképen DSBCAN és CNN által észlelt galaxisok. Az elliptikusok piros, a spirálisok kék, a gyűrűs elliptikus cián, a gyűrűs spirálisok pedig zöld színnel.	43
5.25. ábra. Generált égboltképen Faster R-CNN által észlelt galaxisok. Az elliptikusok piros, a spirálisok kék, a gyűrűs elliptikus cián, a gyűrűs spirálisok pedig zöld színnel.	43

8 TÁBLÁZATJEGYZÉK

3.1. táblázat. Dieleman és társai neurális hálózatának felépítése [21].	11
3.2. táblázat. Tu és Truyen neurális hálózatának felépítése [22].	12
3.3. táblázat. Különböző ResNet-ek számítási igénye [25].	15
3.4. táblázat. Zhu és társai neurális hálózatának felépítése [26].	15
3.5. táblázat. Különböző neurális hálózatok eredményei [26].	16
3.6. táblázat. Különböző anchorbeállítások [33].	20
5.1. táblázat. A hálózatok összehasonlítása.	41

9 IRODALOMJEGYZÉK

- [1] „A Hubble hangvilla”, <http://cas.sdss.org/dr3/hu/proj/basic/galaxies/tuningfork.asp>, utoljára megtekintve 2022.10.22.
- [2] Christopher J. Conselice, Aaron Wilkinson, Kenneth Duncan, Alice Mortlock, „The Evolution of Galaxy Number Density at $z < 8$ and its Implications”, The Astrophysical Journal, vol. 830, no. 2, 2016, p. 83
- [3] „Sloan Digital Sky Survey galaxies”, <http://skyserver.sdss.org/dr16/en/proj/basic/galaxies/galaxieshome.aspx>, utoljára megtekintve 2022.10.22.
- [4] „Galaxy Zoo”, <https://www.zooniverse.org/projects/zookeeper/galaxy-zoo/>, utoljára megtekintve 2022.10.22.
- [5] Jarvis, J. F., Tyson, J. A., „Focas: faint object classification and analysis system”, Astronomical Journal, vol. 86, 1981, pp. 476-495
- [6] Bertin E., Arnouts S., „SExtractor: Software for source extraction”, Astronomy and Astrophysics Supplement, vol. 117, 1996, pp. 393-404
- [7] N. M. Ball, J. Loveday, M. Fukugita, O. Nakamura, S. Okamura, J. Brinkmann, R. J. Brunner, „Galaxy types in the Sloan Digital Sky Survey using supervised artificial neural networks”, Monthly Notices of the Royal Astronomical Society, vol. 348, no. 3, 2004, pp. 1038–1046
- [8] M. Huertas-Company, J. A. L. Aguerri, M. Bernardi, S. Mei and J. Sánchez Almeida, „Revisiting the Hubble sequence in the SDSS DR7 spectroscopic sample: a publicly available Bayesian automated classification”, A&A, vol. 525, 2011, pp. 1-13
- [9] „SDSS DR7 Finding Chart Tool”, <http://skyserver.sdss.org/dr7/en/tools/chart/chart.asp>, utoljára megtekintve 2022.10.22.
- [10] „Galaxy Zoo Decision Trees”, https://data.galaxyzoo.org/gz_trees/gz_trees.html, utoljára megtekintve 2022.10.22.
- [11] Kyle W. Willett , Chris J. Lintott , Steven P. Bamford, Karen L. Masters, Brooke D. Simmons , Kevin R.V. Casteels, Edward M. Edmondson, Lucy F. Fortson, Sugata Kaviraj, William C. Keel , Thomas Melvin , Robert C. Nichol , M. Jordan Raddick, Kevin Schawinski, Robert J. Simpson, Ramin A. Skibba, Arfon M. Smith, Daniel Thomas, „Galaxy Zoo 2: detailed morphological classifications for 304,122 galaxies from the Sloan Digital Sky Survey”, Monthly Notices of the Royal Astronomical Society, vol. 435, no. 4, 2013, pp. 2835-2860
- [12] Nair P. B., Abraham R. G., „A Catalog of Detailed Visual Morphological Classifications for 14,034 Galaxies in the Sloan Digital Sky Survey”, The Astrophysical Journal Supplement Series, vol. 186, no. 2, 2010, pp. 427-456
- [13] Kevin. R. V. Casteels, Steven P. Bamford, Ramin A. Skibba, Karen L. Masters,4,5 Chris J. Lintott, William C. Keel, Kevin Schawinski, Robert C. Nichol and Arfon M. Smith, „Galaxy Zoo: quantifying morphological indicators of galaxy interaction”, Monthly Notices of the Royal Astronomical Society, vol. 429, no. 2, 2013, pp. 1051-1065

- [14] D. H. Hubel, T. N. Wiesel, „Receptive fields and functional architecture of monkey striate cortex”, *The Journal of Physiology*, vol. 195, no. 1, 1968, pp. 215–243
- [15] Kunihiko Fukushima, „Neocognitron: A self-organizing neural network model for a mechanism of pattern recognition unaffected by shift in position”, *Biological Cybernetics*, vol. 36, 1980, pp. 193–202
- [16] „What Is a Convolutional Neural Network?”, <https://uk.mathworks.com/discovery/convolutional-neural-network-matlab.html>, utoljára megtekintve 2022.10.22.
- [17] Ian Goodfellow, Yoshua Bengio, Aaron Courville, „Deep Learning”, MIT Press, 2016
- [18] Nitish Srivastava, Geoffrey Hinton, Alex Krizhevsky, Ilya Sutskever, Ruslan Salakhutdinov, „Dropout: A Simple Way to Prevent Neural Networks from Overfitting”, *Journal of Machine Learning Research*, vol. 15, no. 1, 2014 pp. 1929-1958
- [19] Sergey Ioffe, Christian Szegedy, „Batch Normalization: Accelerating Deep Network Training by Reducing Internal Covariate Shift”, *Proceedings of the 32nd International Conference on Machine Learning*, vol. 37, 2015, pp. 448-456
- [20] Christian Garbin, Xingquan Zhu, Oge Marques, „Dropout vs. batch normalization: an empirical study of their impact to deep learning”, *Multimedia Tools and Applications*, vol. 79, 2020, pp. 12777–12815
- [21] Sander Dieleman, Kyle W. Willett, Joni Dambre, „Rotation-invariant convolutional neural networks for galaxy morphology prediction”, *Monthly Notices of the Royal Astronomical Society*, vol. 450, no. 2, 2015, pp. 1441–1459
- [22] Tu Dinh Nguyen, Truyen Tran, Galaxy Zoo Challenge with Convolutional Neural Networks, 2014, https://github.com/tund/kaggle-galaxy-zoo/blob/master/report/gz_report.pdf
- [23] Rohit Kumar Jha, Irfan Hudda, Galaxy Zoo Challenge Project Report, 2014, <https://cse.iitk.ac.in/users/cs365/2014/submissions/rkjha/project/report.pdf>
- [24] H. Domínguez Sánchez, M. Huertas-Company, M. Bernardi, D. Tuccillo and J. L. Fischer, „Improving galaxy morphologies for SDSS with Deep Learning”, *Monthly Notices of the Royal Astronomical Society*, vol. 476, no. 3, 2018, pp. 3661-3676
- [25] Kaiming He, Xiangyu Zhang, Shaoqing Ren, Jian Sun, „Deep Residual Learning for Image Recognition”, *IEEE Conference on Computer Vision and Pattern Recognition*, 2016, pp. 770-778
- [26] Xiao-Pan Zhu, Jia-Ming Dai, Chun-Jiang Bian, Yu Chen, Shi Chen, Chen Hu, „Galaxy morphology classification with deep convolutional neural networks”, *Astrophysics and Space Science*, vol. 364, no. 4, 2019, pp. 1-12
- [27] Joseph Redmon, Santosh Divvala, Ross Girshick, Ali Farhadi, „You Only Look Once: Unified, Real-Time Object Detection”, *Proceedings of the IEEE conference on computer vision and pattern recognition*. 2016, pp. 779–788
- [28] Roberto E. González, Roberto P. Munoz, Cristian A. Hernández, „Galaxy detection and identification using deep learning and data augmentation”, *Astronomy and Computing*, vol. 25, 2018, pp. 103-109

- [29] Colin J. Burke, Patrick D. Aleo, Yu-Ching Chen, Xin Liu, John R. Peterson, Glenn H. Sembroski, Joshua Yao-Yu Lin, „Deblending and Classifying Astronomical Sources with Mask R-CNN Deep Learning”, *Monthly Notices of the Royal Astronomical Society*, vol. 490, no. 3, 2019, pp. 3952-3965
- [30] Kaiming He, Georgia Gkioxari, Piotr Dollár, Ross Girshick, „Mask R-CNN”, 2017 IEEE International Conference on Computer Vision (ICCV), 2017, pp. 2980-2988
- [31] J. R. Peterson, J. G. Jernigan, S. M. Kahn, A. P. Rasmussen, E. Peng, Z. Ahmad, J. Bankert, C. Chang, C. Claver, D. K. Gilmore, E. Grace, M. Hannel, M. Hodge, S. Lorenz, A. Lupu, A. Meert, S. Nagarajan, N. Todd, A. Winans, and M. Young, „Simulation of Astronomical Images from Optical Survey Telescopes using a Comprehensive Photon Monte Carlo Approach”, *The Astrophysical Journal Supplement Series*, vol. 218, no. 1, 2015, pp. 1-14
- [32] Abdulla W., „Mask R-CNN for object detection and instance segmentation on Keras and TensorFlow”, 2017, https://github.com/matterport/Mask_RCNN
- [33] Shaoqing Ren, Kaiming He, Ross Girshick, and Jian Sun, Faster R-CNN: Towards Real-Time Object Detection with Region Proposal Networks, *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 39, no. 6, 2015, pp. 1137-1149
- [34] „AMD ROCm Documentation”, <https://rocm.docs.amd.com/en/latest/>, utoljára megtekintve 2022.10.22.
- [35] Ross E. Hart, Steven P. Bamford, Kyle W. Willett, Karen L. Masters, Carolin Cardamone, Chris J. Lintott, Robert J. Mackay, Robert C. Nichol, Christopher K. Rosslowe, Brooke D. Simmons, Rebecca J. Smethurst, *Galaxy Zoo: Comparing the demographics of spiral arm number and a new method for correcting redshift bias*, *Monthly Notices of the Royal Astronomical Society*, vol. 461, no. 4, 2016, pp. 3363-3682
- [36] Kaiming He, Georgia Gkioxari, Piotr Dollár, Ross Girshick, Mask R-CNN, IEEE International Conference on Computer Vision (ICCV), 2017, pp. 2980-2988
- [37] „Haladó Algoritmusok”, <https://users.nik.uni-obuda.hu/halal/main.pdf>, utoljára megtekintve 2022.10.22.
- [38] Martin Ester, Hans-Peter Kriegel, Jiirg Sander, Xiaowei Xu, A Density-Based Algorithm for Discovering Clusters in Large Spatial Databases with Noise, *Proceedings of the Second International Conference on Knowledge Discovery and Data Mining*, 1996, pp. 226-231.

10 MELLÉKLETEK

A forráskódok ezen az oldalon érhetők el: https://github.com/alexszb/galaxy_detection.