



ÓBUDAI EGYETEM
ÓBUDA UNIVERSITY

TUDOMÁNYOS DIÁKKÖRI DOLGOZAT

DEEP LEARNING MEGOLDÁS DIGITÁLIS PATOLÓGIAI ROBOTOS MANIPULÁCIÓHOZ

Szerző(k):

Cserfalvi Erik Patrik

mérnökinformatikus BSc. szak, IV. évf.

Konzulens(ek):

Károly István Artúr

doktorjelölt, tudományos
segédmunkatárs

Budapest, 2024.

TARTALOMJEGYZÉK

1	Bevezetés	3
1.1	Motiváció és célkitűzés	4
1.2	Kapcsolódó munkák	7
2	Felhasznált eszközök és módszerek.	16
2.1	Blender	16
2.1.1	Blender főbb funkciói	16
2.1.2	Blender workflow	17
2.1.3	Suzanne	17
2.2	Blender Annotation Tool	18
2.3	MinIO.	21
2.4	Kafka	22
3	Megvalósítás és eredmények	25
3.1	Szintetikus keypoint detekciós adathalmaz	25
3.1.1	Üveglemezek	26
3.1.2	Keypointok.	28
3.1.3	Változások reprezentálása	28
3.1.4	Annotációs maszk	29
3.1.5	Adathalmazok generálása	30
3.2	Tanítási folyamat	32
3.3	Kiértékelés.	33
4	Összefoglalás és jövőbeli feladatok.	36
5	Irodalomjegyzék	38
6	Ábrajegyzék.	43
7	Rövidítések	45

Kivonat

A patológia vagy kórtan az orvostudomány azon ága, amely különböző sejtek és szövetek tanulmányozásával betegségek diagnosztikájával foglalkozik. Digitális patológiáról akkor beszélünk, ha ezek a szövetminták, egy erre a célra kifejlesztett szkennelő berendezés segítségével, digitalizálásra kerülnek. A digitális patológiának számos előnye van a hagyományos megközelítéssel szemben. A minták hatékonyabban tárolhatók, könnyebb kezelhetőséget és egyszerűbb hozzáférést biztosít, mindemellett az így kapott nagy felbontású képek különböző deep learning modellek tanítására is felhasználhatóak. Rengeteg korábban archivált szövettani minta viszont még nincs digitalizálva és a rendelkezésre álló minták számát figyelembe véve ez a folyamat igen hosszadalmas és erőforrás igényes, mivel a mintákat egyesével emberi beavatkozás által szükséges feldolgozni. Ezen folyamat automatizálása nem egyértelmű a rengeteg változó környezeti tényező miatt. Ilyen például a mintákat tároló dobozok típusa, mérete, a benne lévő minták száma és elhelyezkedése, valamint a környezeti megvilágítás. A dolgozat ennek a feladatnak az automatizálására egy deep learning alapú megoldást kínál, amit a nagy mennyiségű változó tényező indokol. A tároló dobozban lévő minta üveglemezeket egy deep learning modell detektálja. Ezt követően a detektált üveglemezeket egy robot a szkennelő berendezésbe helyezi. Megfelelő erőforrások hiányában a deep learning modell tanítására szintetikus adathalmaz készült a Blender 3D modellező alkalmazás segítségével, ami lehetővé teszi fotórealisztikus képek renderelését. Ezzel a megközelítéssel készült képek képesek szimulálni a változatos környezeti faktorokat. A Blender Annotation Tool alkalmazásával a fotórealisztikus képekhez annotációs maszkok készültek a deep learning modell tanításához. Az így létrejövő adathalmazon egy módosított Keypoint R-CNN tanult fel az üveglemezek detektálására. A megközelítés eredményessége valós minták detektálásával került kiértékelésre.

Abstract

Pathology, or the study of diseases, is the branch of medicine that focuses on diagnosing diseases through the study of various cells and tissues. Digital pathology refers to the digitization of these tissue samples using a specialized scanning device developed for this purpose. Digital pathology offers numerous advantages over traditional methods. Samples can be stored more efficiently, provide easier handling and simpler access, and the resulting high-resolution images can be used to train various deep learning models. However, many previously archived histological samples have not yet been digitized, and considering the number of available samples, this process is quite lengthy and resource-intensive since the samples need to be processed individually through human intervention. Automating this process is not straightforward due to numerous variable environmental factors. These include the type and size of the storage boxes, the number and arrangement of samples within them, and the ambient lighting conditions. This thesis proposes a deep learning-based solution to automate this task, justified by the large number of variable factors. A deep learning model detects the glass slides containing the samples within the storage box. Following this, an industrial robot places the detected glass slides into the scanning device. Due to the lack of adequate resources, a synthetic dataset was created for training the deep learning model using the Blender 3D modeling application, which allows the rendering of photorealistic images. This approach can simulate various environmental factors in the generated images. Annotation masks were created for these photorealistic images using the Blender Annotation Tool to train the deep learning model. A modified Keypoint R-CNN was trained on the resulting dataset to detect the glass slides. The effectiveness of this approach was evaluated by detecting real samples.

1 BEVEZETÉS

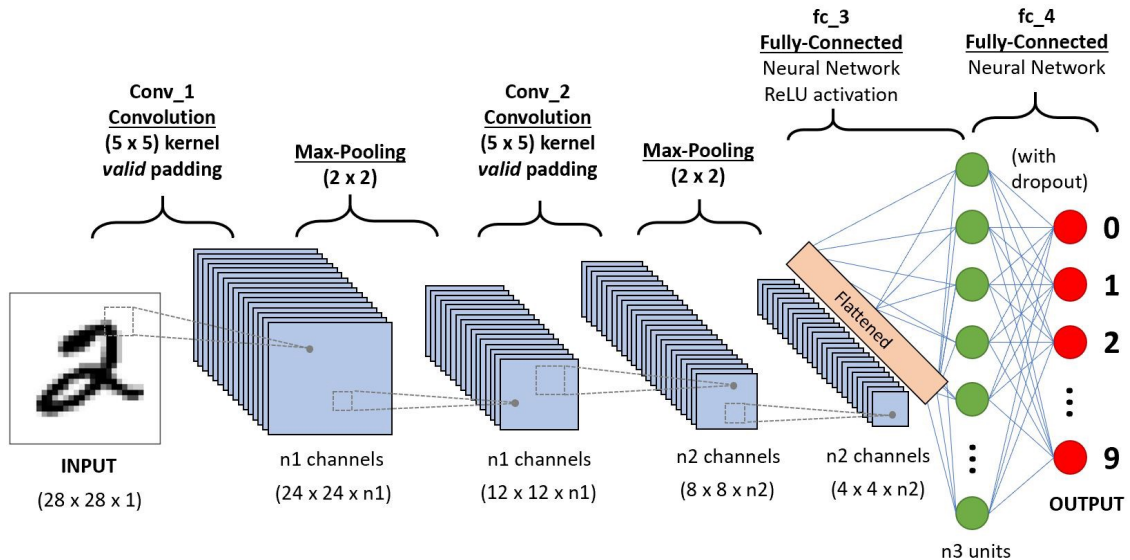
Az utóbbi évek eredményei deep learning (továbbiakban DL) terén alapjaiban változtatták meg az informatikát és formálták át a mindennapjainkat. A gépi tanulás a mesterséges intelligencia azon ága, mely az emberi tanulási folyamatot algoritmusok segítségével imitálja. Ezen terület alá tartozik a DL, mely 3, vagy több réteget tartalmazó neurális hálókat jelent. A neurális hálók a felépítésükben és működésükben az emberi agy működését veszik alapul és a tanításukhoz nagy mennyiségű adatra van szükség. Felépítésük szerint egy bemeneti, egy kimeneti és több rejtett réteggel rendelkeznek. A rétegeket alkotó neuronokat súllyal rendelkező élek kötik össze. A neuron a bemeneteinek súlyozott összegét számítja ki és hozzá adja az eltolást. A több rejtett réteg használata az adatok mélyebb megértésben segít [1]. Egyre több és több területen hasznosítják ezen technológia által nyújtott széleskörű lehetőségeket.

Néhány példa:

- Autonóm vezetés: kulcsfontosságú szerepet játszanak a DL modellek a jármű navigációjában, tájékozódásában és a forgalmi helyzetek értelmezésében [2]
- Robotika: DL modellek használatával képes sok robot objektumok felismerésére, navigációra, intelligens interakciókra a környezetükkel [3]
- Orvosi diagnosztika: nagy segítséggel vannak az orvoslásban dolgozó szakemberek számára, megkönnyítik a különböző vizsgálatok folyamatát [4]
- Nyelvi feldolgozás: DL modellek lehetővé teszik az emberi nyelv értelmezését és ez alapján a kérdések, válaszok generálását [5]
- Tudományos kutatások: a kutatók ezen modellek használatával képesek nagy adathalmazok elemzésére, és ezek alapján előrejelzések készítésére [6]

Az egyik legjelentősebb terület a gépi látás, ahol az utóbbi időkben jelentős fejlődést figyelhattunk meg. A gépi látás több problémára bontható, ilyen például a képek osztályozása, objektumok, keypointok detektálása, szemantikus és instance szegmentálás [7]. Ezen feladatok megoldására az egyik legnépszerűbb megközelítés a konvolúciós neurális hálók használata [8]. Ezen hálók az egyik legnépszerűbb és legtöbbször használt modellek közé sorolhatók. Bemenetükként több dimenziós mátrixokban érkező adatokat dolgoznak fel, mint például képek, amik általában 3 darab 2D mátrixból állnak, melyek elemei pixel intenzitásokat jelölnek. Az ilyen adatstruktúrákat DL viszonylatban tenzoroknak is szokás nevezni. A konvolúciós hálók (röviden CovNet, vagy CNN) általánosan több rétegből tevődnek össze. Ilyen például a konvolúciós réteg, ami a kritikus képi jellemzők kinyeréséért felelős, vagy például a pooling réteg, ami pedig a paraméterek számának csökkentését végzi. A konvolúció során különböző méretű filterek használatával (3x3, 5x5) csúszóablakszerűen haladnak végig a bemeneti tenzoron és hoznak létre feature map-eket. A modellekbe aktivációs függvények (például ReLU) segítségével vezetik

be a nem lineáris viselkedést. Több rétegnyi konvolúció, pooling és aktivációs függvény követi egymást egy ilyen modellben. Végül egy flatten réteg, ami a bemeneti többdimenziós adatokat egy 1D vektorra alakítja. Az így létrejövő vektort egy teljesen összekötött réteg dolgozza fel, ami a kimenetein a különböző osztályokhoz tartozás valószínűségét adja eredményül. Ezen modelleket széles körben alkalmazzák és több eltérő feladatban is kiemelkedő sikereket érnek el [8].



1.1. ábra: Átlagos konvolúciós háló felépítése¹

1.1 Motiváció és célkitűzés

A patológia az orvostudomány azon ága, amely a betegségek tanulmányozásával és diagnosztikájával foglalkozik. Segítségével a szakemberek képesek a betegségek mélyrehatóbb megértésére, ami megkönnyíti a diagnózis elkészítését. A patológia egyaránt foglalkozik a sebészeti beavatkozások előtti, utáni szövettani vizsgálatokkal. Mindemellett kulcsfontosságú szerepet játszik az epidemiológiai kutatásokban, a betegségek terjedésének feltérképezésében és megértésében. Ez a tudományág rendkívül fontos szerepet játszik a modern orvoslás terén.

A laboratóriumok többsége már minden újonnan készített szövetmintát digitalizál, viszont rendkívül sok minta áll rendelkezésre mely a technológia megjelenése előtt készült és melyre ezen folyamat még vár. A szövettani minták digitalizálása egy hosszadalmas és manuális beavatkozást igénylő feladat. A minták feldolgozásához emberi beavatkozás

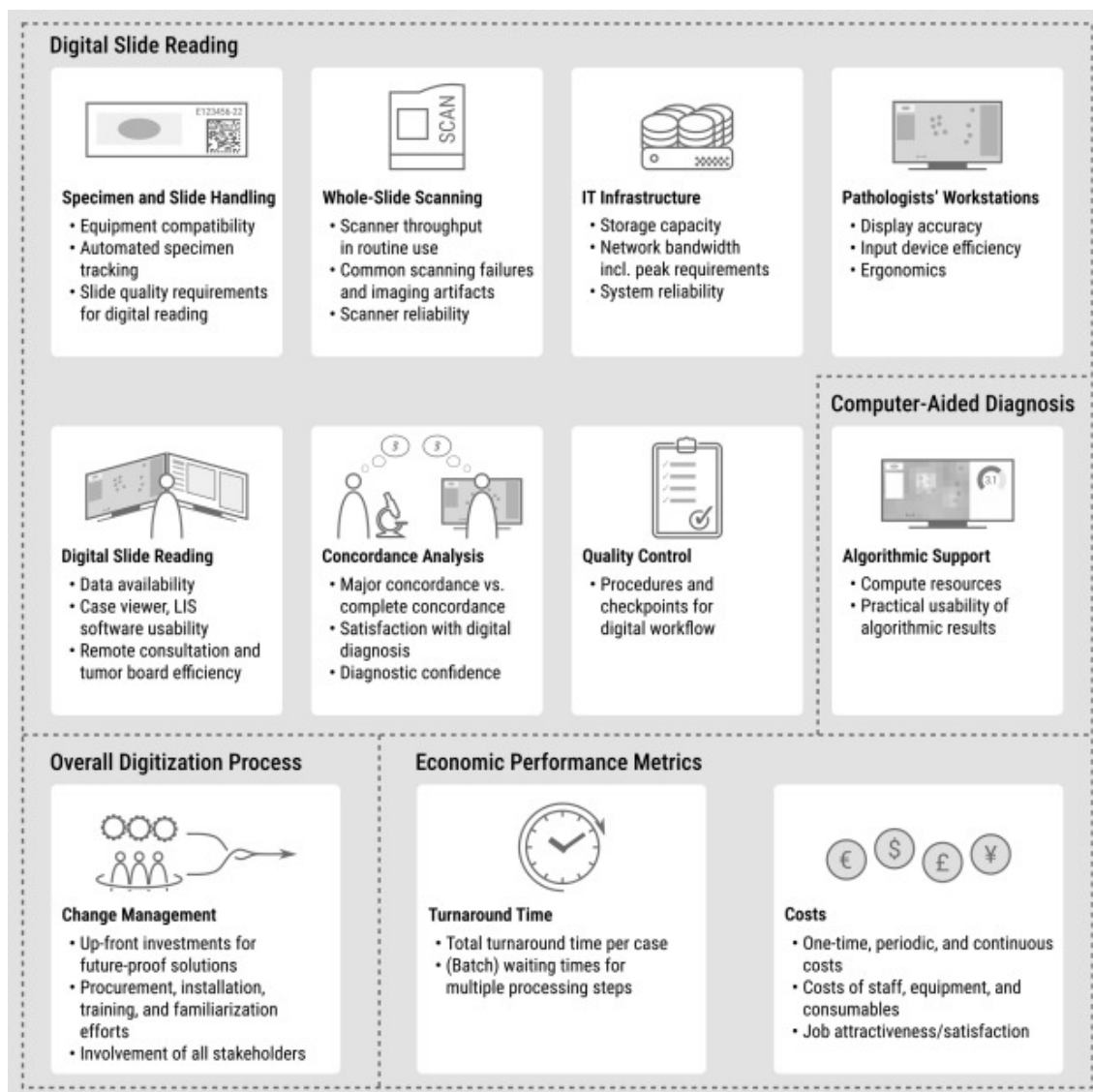
¹<https://medium.com/@mitchhuang777/image-recognition-with-cnns-improving-accuracy-and-efficiency-dd347b636e0c>

szükséges. Ezen felül kompatibilitási problémák, szkennerek és képfeldolgozási hibák is előfordulhatnak. A folyamat automatizálása sem egyértelmű, mivel rengeteg változó tényező van jelen, ilyen például a használt tároló doboz típusa, mérete, minták száma, elhelyezkedése, a környezet és a megvilágítás [9].

A digitális patológiának számos előnye van a hagyományossal szemben [9].

- Lehetőség nyílik szakemberek közötti távoli konzultációkra, másodvélemények kikérésére, például online. Mindemellett a diagnózis előállításának idejét is lerövidíti.
- Könnyebb hozzáférést biztosít a szövettani mintákhoz.
- Digitális szövettani minták könnyebben és tartósabban archiválhatók és tárolásuk is egyszerűbb.
- Lehetővé teszi az egyszerűbb keresést a minták közt.
- Az adatok könnyebben és pontosabban elemezhetőek és használhatók fel például DL modelleknél.

A patológiai laborok különböző problémákat tapasztaltak a digitális munkafolyamatokra való áttérés során. Lars Ole Schwen et al. [9] ezeket a nehézségeket térképezte fel kutatása során. Az analóg mikroszkópok használatáról a digitális minták olvasására történő áttérés sok kihívást tartogat. A minták szkennelése megváltoztathatja az előkészítés követelményeit is. Ehhez más hardverekre is szükség lehet a használt eszközök közötti inkompatibilitás elkerülése érdekében. Ezenkívül előfordulhat, hogy a berendezéseket át kell helyezni, hogy a munkafolyamatokhoz szükséges idő ne nőjön meg ugrásszerűen. Ezen változások növelhetik a költségeket, az elvégzendő munka mennyiségét és az előkészítéssel töltött időt is. Ahhoz, hogy a laboratóriumok effektíven tudjanak feldolgozni szövetszövetmintákat megfelelő mennyiségű szkennerekre van szükségük, esetleges meghibásodás esetén pedig tartalékra. Emellett más problémák is akadhatnak a minták előkészítése során. A szkennelési hibák rátáját 2%-ra becslik, ami sokszor a 7%-ot is elérheti [10]. Például, a külső laboratóriumokban használt vonalkódok zavaróak lehetnek a szakemberek számára, mivel sokszor releváns információkat takarnak ki így megnehezítve a szkennelést. Mindemellett a minták beakadhatnak a szkennerekbe ezzel összetörve azokat. Ha a minták nincsenek megfelelően szárítva, akkor a szkennelésre használt állványhoz tapadhatnak. Ezen kívül előfordulhat a szövettani minták leejtése is [11]. Ezt az is okozhatja, ha a címkék nem megfelelően vannak felragasztva így azok túllógnak. Mindezek ellenére összességében a digitális munkafolyamatra történő áttérés több előnnyel rendelkezik, mint hátránnyal [12], [13]. A digitális patológiát alkalmazó laboratóriumok és szakemberek kezdetben sokszor szkeptikusak, de végső soron elégedettek a technológiának köszönhető eredményekkel. A váltás nagy erőforrás befektetésnek tűnik, viszont az ezzel nyert hatékonyságot figyelembe véve kifizetődő. Ezen teljesítmény növekedés szükséges is az egyre csak növekvő diagnosztikai igények miatt és mivel lehetővé teszi különböző gépi tanulási módszerek alkalmazását.



1.2. ábra: Digitális patológiai munkafolyamat [9]

A dolgozat, a korábban készített, még be nem szkennelt szövetszövetminták digitalizálásának az automatizálásra egy deep learning alapú megoldást nyújt, mellyel a folyamat hossza nagyságrendekkel lerövidíthető és az emberi beavatkozás minimalizálható. A javasolt megoldás során a tároló dobozban lévő szövetszöveti minták egy ipari robotra szerelt kamera képeinek segítségével detektálásra és lokalizálására kerülnek. Ezt követően a robot a mintákat automatikusan a digitalizálásra használt szkennerbe helyezi. A deep learning alapú megközelítést a rengeteg változó tényező indokolja. Ilyen például a mintákat tároló dobozok típusa, mérete, a benne lévő minták száma, elhelyezkedése, és a megvilágítás, melyek mind változhatnak. A deep learning modellek tanítása rendkívül sok adatot igényel, melyeket még annotálni is szükséges. Megfelelő erőforrások hiányában, a deep learning modell szintetikus előállított adathalmazon tanult fel, a mintákat tároló üveg-

lemezek detekciójára. A szintetikus adatok előállítását a Blender [14] 3D számítógépes grafikai szoftver által történt. A szövettani mintákról, az azokat tároló eszközökről és a környezetről melyben a robot dolgozni fog, 3D modellek készültek. Az így létrejövő 3D környezet képes szimulálni a változatos környezeti faktorokat. A Blender lehetővé teszi az elkészített 3D környezetről fotorealistikus renderelt képek készítését. Ezekhez a képekhez a Blender Annotation Tool [15] alkalmazásával annotációs maszkok készültek a deep learning modell tanításához. A deep learning modell egy módosított Keypoint R-CNN, ami az üveglemezek sarokpontjainak detektálására tanult fel.

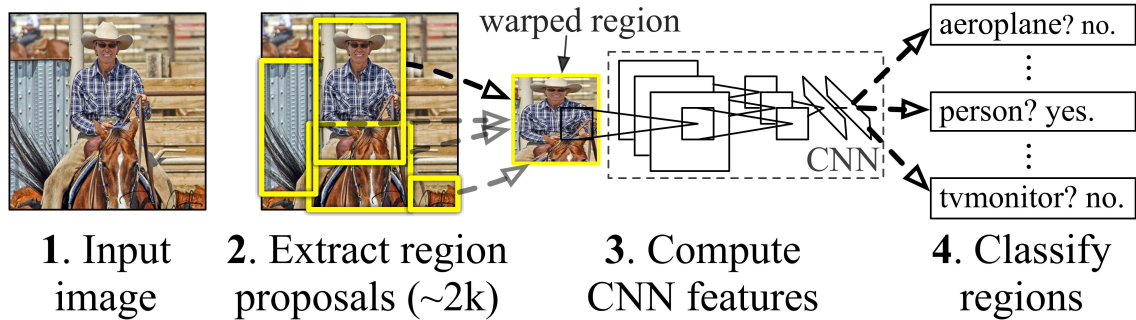
1.2 Kapcsolódó munkák

Az objektumdetekció kulcsfontosságú eljárás a képfeldolgozás és gépi látás területén, ami előre meghatározott osztályba tartozó objektumok előfordulásának észlelésével, vagy helyzetének meghatározásával foglalkozik. Ez a technológia az elmúlt években rohamos fejlődésen ment keresztül és mára már számos fontos területen alkalmazzák, például az önvezető autókban, video megfigyelő rendszerekben, orvosi képfeldolgozásban és a robotikában. Az objektumdetekcióra egyaránt alkalmaznak klasszikus algoritmusokat és gépi, vagy a mély tanulási módszereket is. Klasszikus algoritmusok közé tartozik például a hisztogram orientált gradiens módszer (HOG) [16] vagy például a SIFT [17], habár az ezeken alapuló objektumdetekciós eljárások mára már a konvolúciós hálóknak köszönhetően háttérbe szorultak. Az 1990-es évek során a konvolúciós hálókat széles körben alkalmazták, viszont a support vector machine modellek (SVM) [18] megjelenésével kimentek a divatból. 2012-ben Krizhevsky et al. [19] az ImageNet [20] Large Scale Visual Recognition kihíváson elért kiemelkedő eredményeikkel mutatták meg a konvolúciós hálóban rejlő kiaknázatlan potenciált.

Ross Girshick et al. [21] konvolúciós hálók alkalmazásával nagyobb objektum detekciós pontosságot értek el a Pascal VOC adathalmazon, az addig alkalmazott klasszikus eljárásokkal szemben. Ezen eredmény eléréséhez két fő problémára fókuszáltak. Az objektumok lokalizálása mély neurális háló alkalmazásával és nagy kapacitású modell tanítása kis méretű adathalmazzal. Ezen feladatok megoldására egy régió alapú konvolúciós hálót javasolnak, a Region Based Convolutional Neural Network modellt, röviden R-CNN. A modell elsőként a bemeneti képből 2000 egymástól független régiót készít. Ezek a régiók különböző objektum jelölteket tartalmaznak a detektor számára. Ezt követően egy nagy konvolúciós neurális háló képi jellemző vektorok kinyerését végzi ezekből különböző régiókból. Végül SVM modellek használatával a kinyert jellemzőket különböző osztályokba sorolja. A modell teljesítményét az ILSVRC2013 detekciós adathalmazon és a már említett PASCAL VOC adathalmazon mérték ki. Az első 31.4-os mAP-t ért el, ami szignifikánsan magasabb a második legmagasabb értéket (24.3) elérő OverFeat-nél [22]. A PASCAL VOC-on [23] is számottevő javulás figyelhető meg a nem lineáris SVM

modellekkel szemben. A state-of-the-art mAP 35.1-ről 53.7-re nőtt.

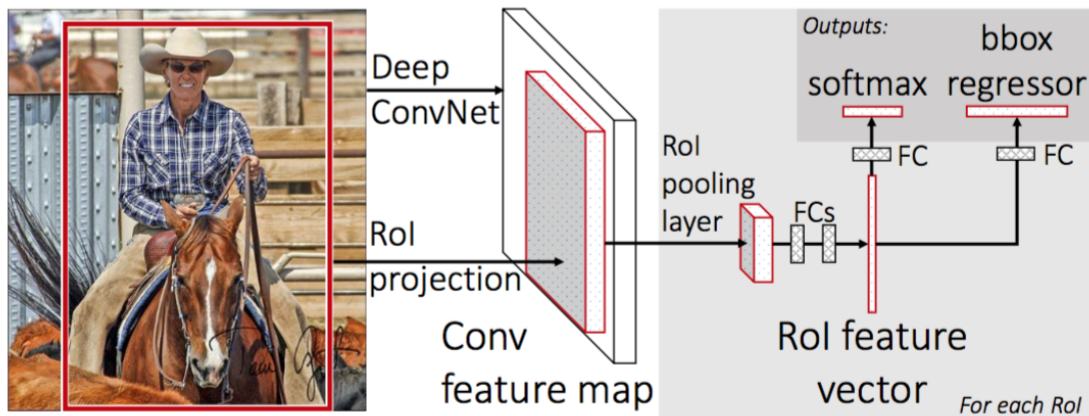
R-CNN: *Regions with CNN features*



1.3. ábra: R-CNN [21]

Az R-CNN [21] több jelentős hibájára, hátrányára a Fast R-CNN [24] mutatott rá és küszöbölt ki. Ilyen hiba például, hogy a modell tanítása egy többlépcsős folyamat, ami sok időt és erőforrást vesz igénybe. Minden bemeneti kép esetén az előállított régiókból képi jellemzőket nyer ki. Nagyon mély neurális hálók (például VGG16) tanítása 5000 képen akár 2.5 napot is igénybe vehet. Emellett az így kinyert jellemzőket tárolni is szükséges, ami akár több száz GB tárhelyet is igényelhet. Mindazonáltal az objektumokat 47 másodperc alatt képes detektálni képenként, ami lassúnak mondható. Az R-CNN ezen lassúságát első sorban az okozza, hogy a CovNet minden régió feldolgozása esetén végrehajtja a forward pass-t anélkül, hogy a már elvégzett számításokat megosztaná. A Spatial pyramid pooling háló (SSPnet) [25] ezen lehetőséget használja ki. A modell a bemeneti képből konvolúciós jellemzőtérképet készít, majd ezt felhasználva osztályozza az összes régió javaslatot. SSPnet teszt során 10 akár 100 szoros gyorsulást is képes elérni, az R-CNN-el szemben. Mindemellett a tanítási idő a harmadára csökkent, viszont az R-CNN-hez hasonlóan a tanítás egy többlépcsős folyamat és a kinyert képi jellemzőket ki kell menteni, amelynek jelentős tárhely igénye van. A Fast R-CNN egy új tanítási algoritmust alkalmazásával képes az R-CNN és az SSPnet-nél jelentkező szűk keresztmetszetek elkerülésére. Ezenfelül a Fast R-CNN magasabb detekciós pontosságot képes elérni az említett modellekkel szemben. Ennél a modellnél a tanítás egy egylépcsős folyamat, ami többfunkciós veszteség függvényt alkalmaz. Tanítás során lehetőség van a háló értekeinek frissítésére és ezenfelül nem szükséges eltárolni a kinyert képi jellemzőket. A modell bemenete egy kép, amelyet több konvolúciós és max pooling réteg dolgoz fel és több konvolúciós jellemző térképet állít elő. Ezt követően minden objektum javaslati régióra a RoI pooling réteg kinyer fix hosszúságú vektorokat. Az így kapott jellemző vektorokat teljesen összekötött rétegek sorozata kapja meg, ami végül két ágra bomlik. Az egyik ág a softmax aktivációs függvény alkalmazásával valószínűségi értéket rendel az objektum osztályokhoz. A másik ág pedig megadja a bounding-box koordinátáit minden osztály esetén. Az így létre jött háló state-of-the-art eredményt ért el mAP tekintetében

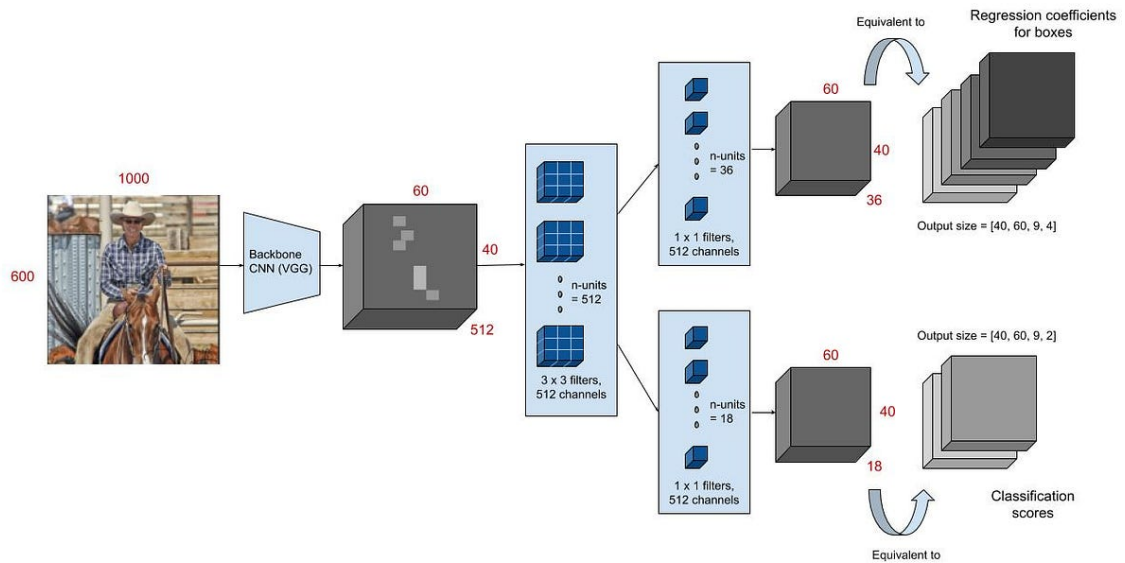
a VOC2007, VOC2010, VOC2012 [23] adathalmazokon. A tanítás gyorsasága és hatékonysága nagyságrendekkel nőtt az R-CNN és SPP-net-hez képest.



1.4. ábra: Fast R-CNN [24]

A Faster R-CNN [26] a Fast R-CNN-t veszi alapul és fejleszti tovább. A modell készítői azt a megfigyelést tették, hogy a konvolúciós jellemző térképek nem csak az objektum detektálásnál használhatóak hanem a régiók meghatározásánál is. Korábban a régiók megállapítása egy a hálózattól különálló régió becselő módszer és a max pooling réteg alkalmazásával történt. A modell két részből tevődik össze. Az első modul Region Proposal Network (RPN) ami egy mély, teljesen konvolúciós háló, ez felelős a régió ajánlásokért. A másik része pedig az előző bekezdésben említett Fast R-CNN, ami az ajánlott régiók alapján végzi a detekciót. Az utóbbi időben gyakran előforduló attention mechanizmust használva az RPN megmondja a Fast R-CNN-nek merre nézzen. Az RPN réteg egy képet kap bemenetként és téglalap alakú objektum ajánlásokat ad kimenetül "objectness" értékkel ellátva. Az objectness érték megmondja az objektum osztályokhoz való hozzátartozás mértékét, a háttérrel szemben. Az RPN által hatékonyan létrehozott régió javaslatok mérete és képaránya széles skálán mozog. Korábbi modellekkel ellentétben itt anchor boxokat alkalmaznak, mint referencia minden mérethez és képarányhoz. A Faster R-CNN magasabb mAP értéket ér el a [0.5-0.95] IoU tartományban, mint az alapját képező modell. Ebből azt a következtetést lehet levonni, hogy az RPN réteg nagyban növeli a lokalizációs pontosságot magasabb IoU tartományokban. Az MS COCO [27] adathalmazán a Faster R-CNN nagyságrendekkel nagyobb mAP értékeket ér el, mint az elődje és lehetővé teszi, hogy a rendszer közel valós időben működhessen.

A képfeldolgozás területén az objektum detekció mellett, a szemantikus szegmentáció a másik alapvető feladat. Célja, az adott képen minden pixelhez hozzárendelni egy osztálycímkét, ezzel meghatározva és elkülönítve a különböző objektumokat. Rendkívül fontos szerepet játszik a gépi látás területén, hiszen alapvető része lehet olyan alkalmazásoknak, mint az önvezető járművek, a képfelismerés vagy az orvosi képfeldolgozás.

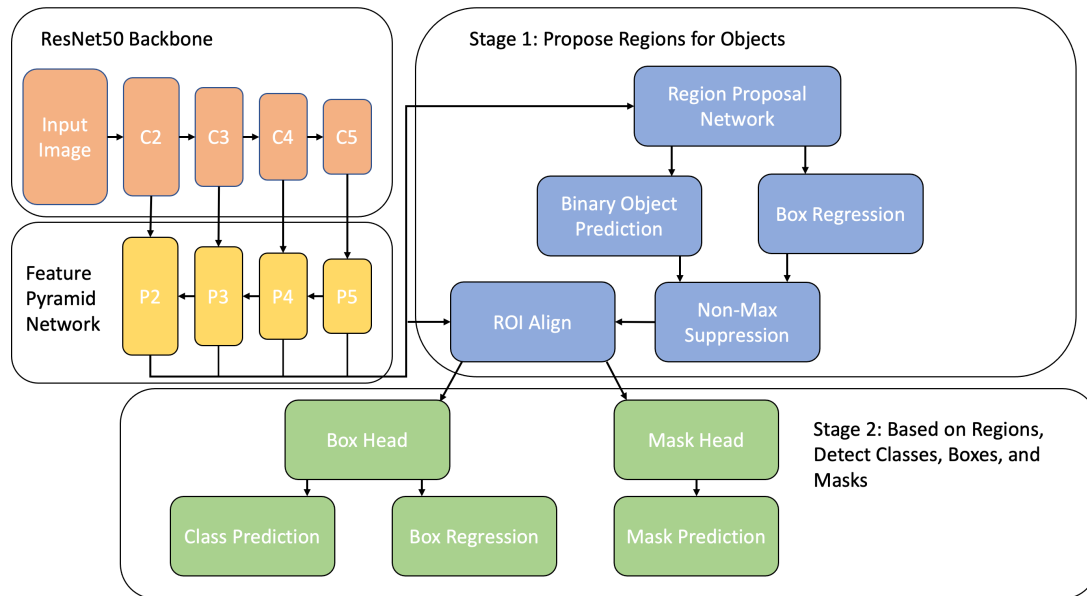


1.5. ábra: *Faster R-CNN*²

Ezzel szemben az instance szegmentáció ötvözi a két alapvető képfeldolgozási feladatot. Szükséges a képen található összes objektum megfelelő detekciója és emellett az objektumok precíz szegmentációja minden instance esetén. Ezen feladat megoldására ad választ a Mask R-CNN [28], ami a nagy sikereket elérő Fast és Faster R-CNN [24] [26] modelleket veszi alapul és bővíti ki egy új ággal. Ennek feladata szegmentációs maszk készítése minden Region of Interest (RoI) esetén. Ezzel párhuzamosan történik a bounding box regresszió és az osztályozás a Faster R-CNN alkalmazásával. A háló maszkot készítő ága egy teljesen konvolúciós neurális háló. A Mask R-CNN könnyen implementálható és tanítható, ami az alapot képző Faster R-CNN-nek köszönhető. Mindemelett a modell megtartja elődje hatékonyságát, mivel az új ág csak kevés számítási többletet jelent. Mivel a Faster R-CNN nem pixel a pixelhez igazításra tervezték különböző háló kimenetek és bemenetek közt ezért apró módosításokra volt szükség. A RoIPool térbeli kvantálást alkalmaz képi jellemzők kinyerésére, ezzel elveszítve azok hely információját. Ezt a problémát a RoIAlign küszöböli ki, ami egy kvantálás mentes réteg. Ezen réteg alkalmazása 10% -tól egészen 50% -ig képes növelni a maszk pontosságát. A modell által

²<https://towardsdatascience.com/faster-r-cnn-for-object-detection-a-technical-summary-474c5b857b46>

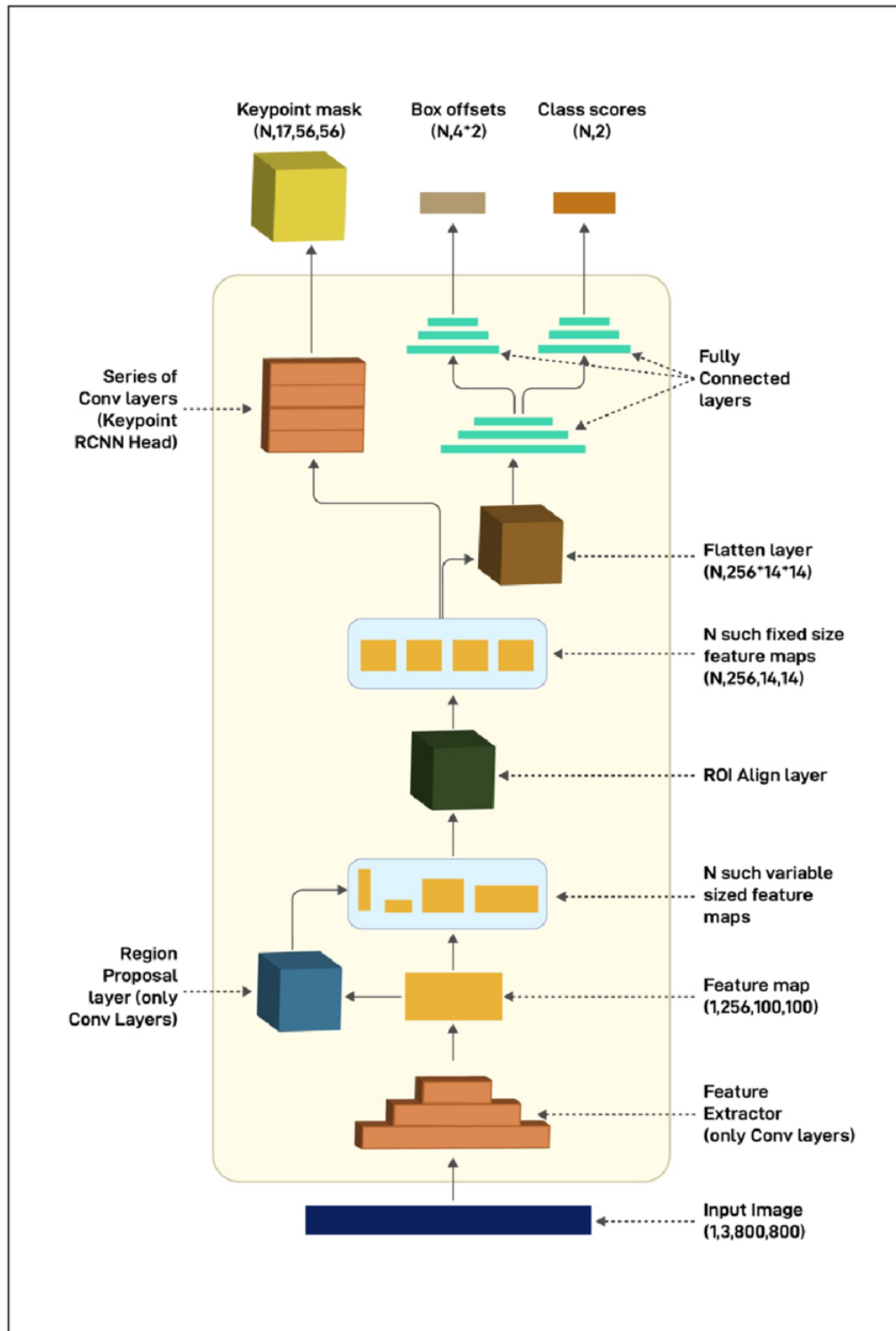
alkalmazott másik fontos módszer, hogy külön kezeli a maszkokat és az osztály predikciókat. Minden osztály számára egymástól függetlenül bináris maszkot prediktál osztályok közötti versengés nélkül, emellett a hálózat RoI osztályozó ágára támaszkodik a kategória meghatározásához. A Mask R-CNN túlszárnyalja az összes state-of-the-art modellt a COCO [27] instance szegmentációs adathalmazon és jeleskedik a COCO objektum detekciós adathalmazán is. A Mask R-CNN képenkénti 200ms-os teljesítményre képes egy GPU-n. A COCO-n való tanítás egy-két napot vesz igénybe egy 8 GPU-s gép használatával.



1.6. ábra: Mask R-CNN³

A Mask R-CNN gyorsasága flexibilitása és magas detekciós pontossága mellett, az általánosítása is rendkívüli. Ezt a COCO keypoint detekciós adathalmazon elért kiemelkedő eredmények is bizonyítják. A kulcspontok olyan jellegzetes pontok a képeken, amelyek fontosak lehetnek az objektumok azonosításához, követéséhez, vagy éppen azok térbeli viszonyainak megértéséhez. Ezek a kulcspontok lehetnek, például sarokpontok, élek, vagy más jellegzetes pontok. A kulcspont felismerő deep learning módszerek fejlődése jelentős előrelépést hozott számos területen, beleértve az objektumfelismerést, a pontfelhők regisztrációját, az emberi gesztusok vagy arcfelismerést, valamint a strukturális elemzést. A Mask R-CNN egy apróbb módosításon átesett változata túlszárnyalta a 2016-os nyertes modell eredményeit COCO keypoint detekciós adathalmazán 5 fps-es futási idővel [28]. A modell a keypointokat egy one-hot bináris maszkként reprezentálja. Így a Mask R-CNN nem csak egy feladat specifikus modellnek, hanem egy általános keretrendszernek is tekinthető.

³<https://aws.amazon.com/blogs/machine-learning/distributed-mask-rcnn-training-with-amazon-sagemakercv/>



1.7. ábra: Keypoint R-CNN⁴

⁴https://www.researchgate.net/figure/The-architecture-of-keypoint-R-CNN-The-architecture-of-keypoint-RCNN-resembles-the_fig1_367582888

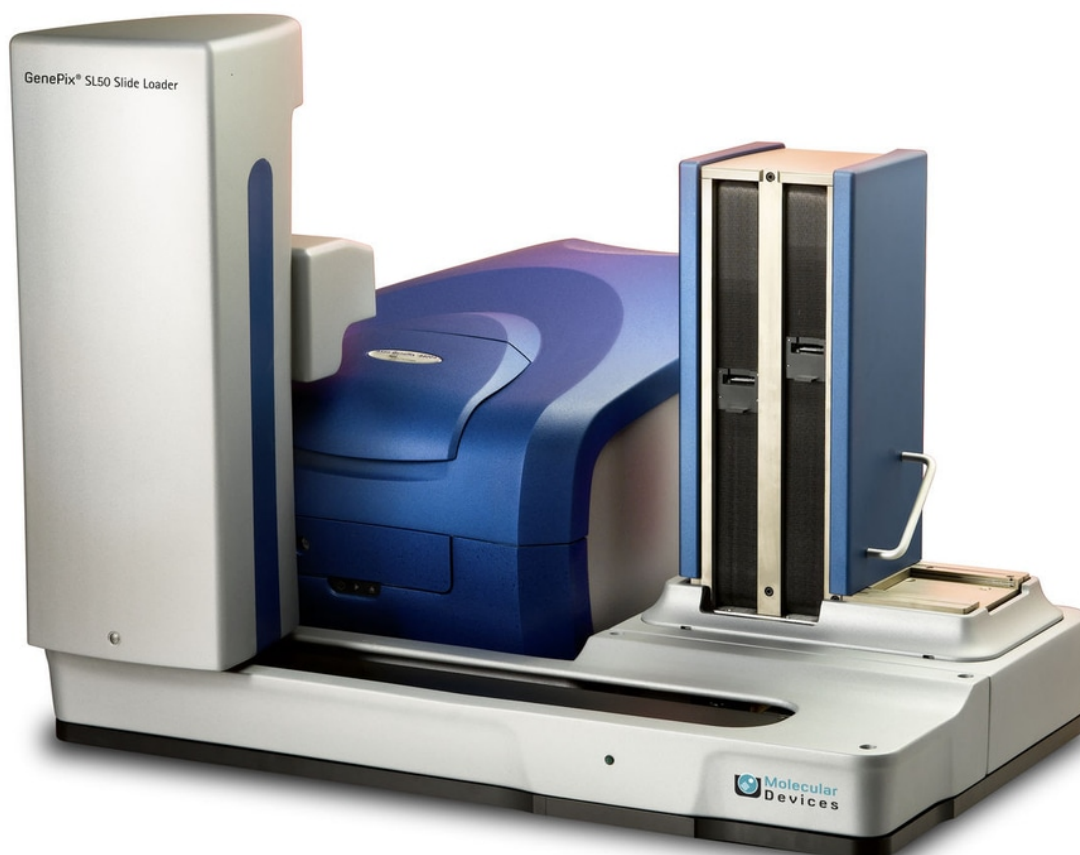
Ezen dolgozat patológiai területen történő keypoint detekciós problémára kínál megoldást, ahol a feladat mintákat tároló üveglemezek sarkainak lokalizálása, ami az említett módosított Mask R-CNN modell használatával történik. A patológiában már korábban is alkalmaztak deep learning modelleket. Egyik fő célja a szakemberek által végzett diagnosztikai feladatok segítése. Minden rákos szövetmintát egy képzett patológusnak kell megvizsgálnia és kijelentenie a tumor jelenlétét. Az ehhez hasonló alapvető repetitív feladatok elvégzésében a DL modellek alkalmazása hatalmas segítséget nyújthat a szakemberek számára. Ilyen modelleket alkalmaztak már rákos elváltozások előrejelzésére prosztata szövetekben [29]–[31], vagy éppen tumor geon típusának meghatározására [32], [33] is. Az onkológia területén a döntéshozást jelentősen nehezíti az ugrásszerűen megnőtt biomarkerek száma, ami alapján az orvosok meghatározzák a szükséges kezelést. A rengeteg biomarker bonyolult döntési fákhoz vezethetnek ezzel bonyolítva a szakemberek munkáját. Ebben az esetben is a deep learning alkalmazása megkönnyíti az egyébként rejtett információk kinyerését [34]. A daganatos betegek kezelése során biopsziával, vagy sebészeti beavatkozással rákos szöveteket nyernek ki, preparálják, majd digitalizálják azokat. Ennek köszönhetően, a patológiában rengeteg információdús és könnyen elérhető adat áll rendelkezésre melyek felhasználhatóak a DL modellek tanítására. Ezen adatok mérete általában nagy és a jelentős részén nem rákos szövetek vannak, ezért intenzív előfeldolgozásra van szükség. Az így létrejövő adathalmazon különböző deep learning modellek taníthatók fel és tesztelhetők [34]. Ezen modellek képesek eldönteni a bemene-ti szövetről, hogy az daganatos vagy sem. Több a MEDLINE adatbázisban megtalálható tanulmány is bizonyítja a DL alapú tumor detektálók robusztusságát [34]. Ezen osztályozó modelleknek teljesítményét egy úgy nevezett AUROC görbével mérik [35]. A ROC egy valószínűségi görbe, az AUC pedig az elválaszthatóság fokát, vagy mértékét jelenti, ami megmondja, hogy a modell mennyire képes az osztályok megkülönböztetésére. Minél magasabb az AUC annál jobban képes a modell az osztály meghatározására. A deep learning alapú tumor detektálók 0.99-nél nagyobb AUROC értéket érnek el, ami azt jelzi, hogy a patológusok és a DL-hálózatok eredményei szinte teljes összhangban vannak [34]. A megközelítés hatásosságát az is jelzi, hogy már többször is sikeresen detektáltak agyi tumort deep learning modellek alkalmazásával [36]. A javasolt megoldás elősegíti az ilyen típusú megközelítések megvalósítását azáltal, hogy a tanításukhoz szükséges digitalizált szövetminták digitalizálási folyamatát automatizálja.

Patológiai minták szkennelésre már készült több olyan berendezés is melyek használatával ezen folyamat többnyire automatizálható. Ilyen például a Modular Devices cég által fejlesztett GenePix SL50 is, mely egy robotikai megközelítés. Ezeknél a berendezéseknél a patológiai tárolókból az üveglemezeket az eszköz tárolójába kell áthelyezni. A patológiai laborokban az üveglemezeket többféle és fajta dobozban, mappákban tárolják sokszor rendezetlenül. Ezeknél az eszközöknél szükséges egy személy, aki a mintákat egyesével a gép tárolójába helyezi ezzel az automatizáció lényegét meghiúsítva. Ezen probléma meg-

oldására deep learning alapú megközelítést NEM publikáltak még, főleg nem olyat mely szintetikus adatokat használt volna. Így a kísérleti összehasonlítás más releváns megközelítésekkel, releváns publikációk hiányában elmarad. A manuális rendezéssel szemben egy egyszerű becslés adható a szükséges manuális beavatkozás idejének csökkentésére, ami akár 30–60-szoros is lehet, ami a teljes tároló tartalmának manuális áthelyezéséhez szükséges időből (~ 60 s) és a tároló általunk javasolt rendszerben való elhelyezéséhez szükséges ($\sim 1\text{--}2$ s) időből adódik. Ez a különbség az évek alatt gyűjtött nagy archívumok digitalizálása során jelentős mértékben csökkentheti a szükséges manuális munka mértékét (például csupán 1000 tároló esetén 2 munkanap helyett nagyjából 15–20 percnyi manuális munka szükséges). A szintetikus adatok alkalmazásával egy olyan könnyen bővíthető rendszer jön létre melyhez egyszerűen és gyorsan adható hozzá új tároló eszköz melyet a megoldás képes kezelni, ezzel biztosítva a megoldás általánosságát. A dolgozat által javasolt megközelítéshez nem szükséges egy személy, aki a mintákat egyik tárolóból a másikba helyezné. A rendszer képes kezelni a rendezetlenül tárolt mintákat, így csak a tárolót szükséges az ipari robot elé helyezni.

Deep learning modellek szintetikus adatokon való tanítása már egy korábban is kutatott terület. Ezen célból több szintetikus generált adathalmaz is létrejött, mint például a Flying Chairs [37], FlyingThings3D [38], MPI Sintel [39], UnrealStereo [40], SceneNet [41], SceneNet RGB-D [42], SYNTHIA [43], GTA V [44], Sim4CV [45] és Virtual KITTI [46]. Ezen adathalmazok főként geometriai problémák megoldására jöttek létre, ilyen például optikai áramlás, jelenet áramlás, kamera helyzetének meghatározása. Habár, közülük néhány adathalmaz tartalmaz annotációt objektum detekcióhoz vagy szegmentációs szegmentációhoz, csak néhány modell készült ezen feladatok elvégzésére. A Hinterstoisser et al. [47] által készített modell tisztán szintetikus adatokon tanult fel objektum detekcióra. A publikációjukban azt vizsgálták, hogy a különböző képi jellemzőket kinyerő rétegek lefagyasztása tanítás során mennyire lehet előnyös a detekció pontosságára. Johnson-Roberson et al. [48] munkájuk során egy autók detektálására alkalmas modellt készítettek mely fotórealisztikus adatokon tanult fel. A háló eredményességét a KITTI adathalmazon tesztelték le. A kísérleteikből azt konklúziót vonták le, hogy a szintetikus adatokon tanított deep learning modellek magas teljesítményt érnek el, mely akár vetekszik a hagyományosan tanított modellekével. Mindezt anélkül, hogy valós adatokat keverték volna a tanító adatok közé. Ezzel ellentétben, Jonathan Tremblay et al. [49] eredményei azt mutatják, hogy a valós adatokkal való finom hangolás előnyös és a modell pontosságának növelését eredményezheti. A publikáció főként a domain randomizációval [50] foglalkozik szintetikus adatokon. Ez technika véletlenszerűen zavarja a környezetet, nem fotorealistikus módokon, például véletlenszerű textúrák hozzáadásával. Ezzel kényszerítve a hálózatot, hogy csak a kép lényeges tulajdonságainak megtanulására összpontosítson. A technika eredményességét 3 state-of-the-art objektum detektorral értékelték ki melyek először a Virtual KITTI, másodszor a domain randomizáció technikával készített

adathalmazon tanultak fel. Az eredmények azt mutatják, hogy a technika képes a modell pontosságának növelésére. Későbbiekben ezen technika alkalmazása a dolgozat által javasolt megoldásban egy jó lehetőség lehet a teljesítmény növelésére.



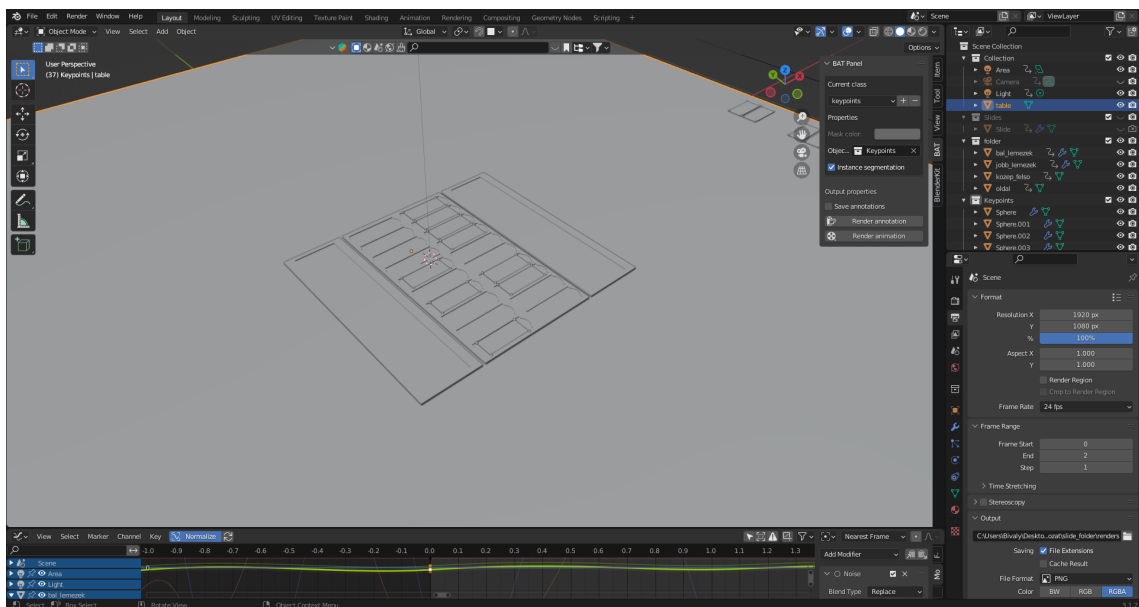
1.8. ábra: GenePix SL50⁵

⁵<https://trends.medicaexpo.com/molecular-devices/project-84611-413189.html>

2 FELHASZNÁLT ESZKÖZÖK ÉS MÓDSZEREK

2.1 Blender

A Blender egy ingyenes, nyílt forráskódú háromdimenziós grafikai szoftver [14]. Alkalmas animációk, vizuális effektek, motion graphics, interaktív 3D applikációk, virtuális valóság és akár még videójátékok készítésére is. Az alkalmazás széleskörű felhasználhatósága miatt igen népszerű és rengetegen használják többek közt, művészek, tervezők, animátorok, játékfejlesztők és még tudományos kutatók is. Az alkalmazással 3D objektumokat hozhatunk létre, szerkeszthetjük azokat. Lehetőség van karakterek, objektumok és a környezet keyframe alapú animációjára is. Az alkalmazás által használt képkocka alapú animálás rengeteg mozgási formát és effektus használatát teszi lehetővé. Emellett létrehozhatóak, más néven renderelhetők fotorealistikus képek és videók a felhasználó által létrehozott 3D környezetből. A Python szkript nyelven készített különböző bővítmények készítését és használatát is támogatja. Mindemelett akár számítógépes játékok megalkotására is alkalmas, az integrált Game Engine által.



2.1. ábra: Blender szoftver felülete

2.1.1 Blender főbb funkciói

- Modellézés: A szoftver lehetőséget ad fejlett poligon alapú modellezésre.
- Modifiers: Különböző módosító hatású effektek állnak a felhasználók rendelkezésére, amelyek renderelés vagy exportálás során alkalmazhatók, mint például a felületek felosztása.

- Sculpting: A szoftver többfelbontású digitális szobrászati funkciókkal is rendelkezik, amelyek magában foglalják a dinamikus topológiát, az újrachálózást, az újrashimmetrizációt és a tizedelést.
- Geometry nodes: Ezen funkció lehetővé teszi különböző objektumok procedurális manipulációját.
- Szimuláció: Az alkalmazással szimulálhatunk különféle fizikailag is pontos interakciót például: füst, por, eső, ruha, haj és még sok más.
- Animáció: Objektumok és a környezet keyframe alapú animációja.
- Rendering: Az alkalmazásba több renderelő motor is be van építve: Cycles, Workbench, Eevee. A Cycles egy fizikai alapú path tracing rendermotor ami támogatja a CPU és a GPU alapú renderelést is. Az Eevee egy fizikai alapú valós-idejű renderelő motor. Workbench pedig egy valós-idejű renderelő motor, ami gyors renderek elkészítésére hivatott modellezés és animációk készítése során.
- Anyag modellek: Meghatározzák a poligon hálók és az objektumok megjelenését.
- Utómunka: Renderelés során lehetőség van utómunkára a node alapú compositor használatával.
- Plugin/Addon és szkriptek: Blender támogatja a különböző python szkripteket amik lehetővé teszik különböző feladatok automatizálását például import/export, prototípusok készítése.
- File formátum: A szoftver egy belső file rendszerrel rendelkezik, amik össze csomagolhatóak egyetlen .blend fájlba.
- Blender Game Engine: Egy beépített valós-idejű grafikus és logikai motor.

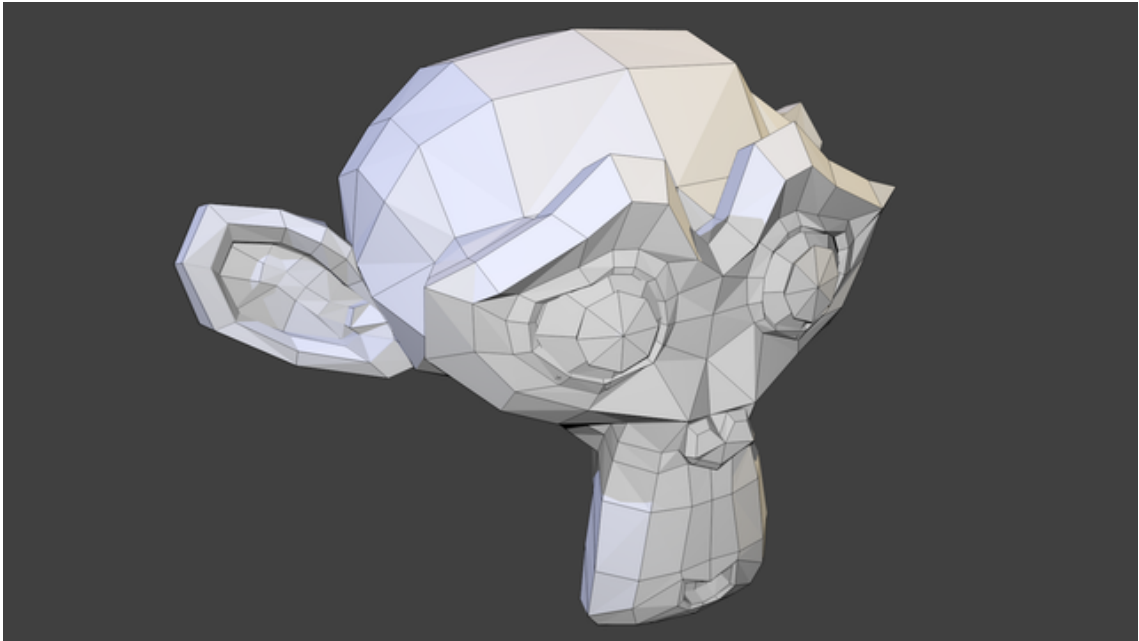
2.1.2 Blender workflow

Az alkalmazásban egy 3D környezet áll a felhasználó rendelkezésére, ahol létrehozhat különböző objektumokat, módosíthatja azok geometriáját, meghatározhatja mozgásukat. Az anyag modellek használatával tudjuk kontrollálni az objektumok megjelenését ezzel fokozva azok realisztikusságát. Továbbá 2D képekkel (textúrákkal) szimulálhatjuk az anyag színét, érdességét, göröngyösségét. Az alkalmazásban megtalálható egy kamera objektum, ami egy fundamentális eleme a 3D környezetnek. Ez határozza meg a nézőpontot és a perspektívát, amiből készül a 3D környezet renderelése. Továbbá lehetőségünk van az érkező megvilágítás változtatására is, ezt pedig a light típusú objektumok módosításával érhetjük el.

2.1.3 Suzanne

2002-ben világosság vált a Blendert fejlesztő NaN szoftver fejlesztő cég bezárása, viszont előtte még egy utolsó kiadást jelentettek meg amibe egy úgy nevezett easter egg-ként egy

csimpánz fej 3D modelljét csempészték a fejlesztők és a művészek, egy fajta személyes emléket hagyva maguk után. Az alkotója Suzannenak keresztelte a Kevin Smith féle Jay és néma Bob című film orángutánja után. Suzanne egy alacsony poligon számú modell, pontosan 500 poligonból áll. Későbbiekben ez a 3D objektum egy széleskörben elterjedt teszt objektum lett különböző anyag modellek, animációk, textúra és világítási beállítások gyors és egyszerű tesztelésére. A Blendert használók körében ennek az egyedülálló 3D modellnek szignifikáns jelentősége van és még a legnagyobb Blender verseny fődíját is Suzanne díjnak hívják.



2.2. ábra: Suzanne⁶

2.2 Blender Annotation Tool

Tapasztaltabb felhasználók számára számtalan lehetőség áll rendelkezésre Python programozáson keresztül a Blender API-t használva. Ezen aspektus hatalmas segítség lehet a felhasználók számára, akik szeretnének speciális eszközöket készíteni, vagy személyre szeretnék szabni a munkakörnyezetüket. Mindemelett a Python integráció lehetőséget nyújt DL technológiák implementálására is. A dolgozat során használt add-on a Blender Annotation Tool (BAT) is ezt a Python API-t alkalmazza [15]⁷. Ezen kiegészítővel egyszerűen készíthetünk szegmentációs maszkokat a renderelt képekhez.

A gépi látás területén általában valós világból származó adatokon tanítják a modelleket. Az ilyen célra használt adathalmazoknak megfelelő méretűeknek kell lenni ahhoz,

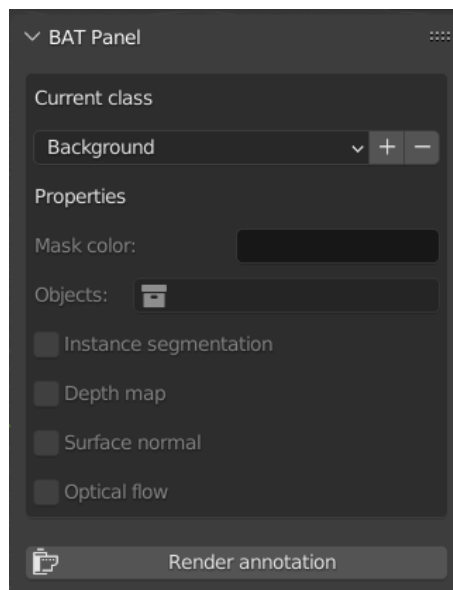
⁶<https://www.wikidata.org/wiki/Q64830346>

⁷https://github.com/karolyartur/blender_annotation_tool

hogy megbízhatóan reprezentálják a megoldani kívánt problémát. Ez sokszor egy nehezen kivitelezhető feladat, magában csak a megfelelő méretű adatgyűjtés is hatalmas erőbefektetést igényel. Ha felügyelt tanítást szeretnénk végezni, akkor az adathalmazt még annotálni is szükséges, ami egy elengedhetetlen feladat. Csakúgy, mint az adathalmaz elkészítése, az annotálás folyamata is egy hosszadalmas és erőforrás igényes feladat. Még nagyobb cégeknek is meggyűlik vele a problémája, sokszor fel kell bérelniük egy külső partnert, akik a rendelkezésre álló adatok annotációját készítik el. A Blender Annotation Tool egy olyan kiegészítő a Blender 3D modellező alkalmazáshoz, amely lehetővé teszi annotált szintetikus adathalmaz készítésének automatizálását. Ezzel lerövidítve és megkönnyítve ezen feladat elvégzését. Felhasználóbarát felületével Python nyelv ismerete nélkül, egyszerűen és automatikusan generálhatunk szegmentációs maszkokat a Blenderben készített 3D környezetünkhöz.

BAT használatának lépései:

- Használni kívánt objektumok egy gyűjteménybe helyezése
- BAT felületén a megfelelő osztály definiálása
- Osztályok és objektumgyűjtemények társítása
- Az annotáció típusának kiválasztása

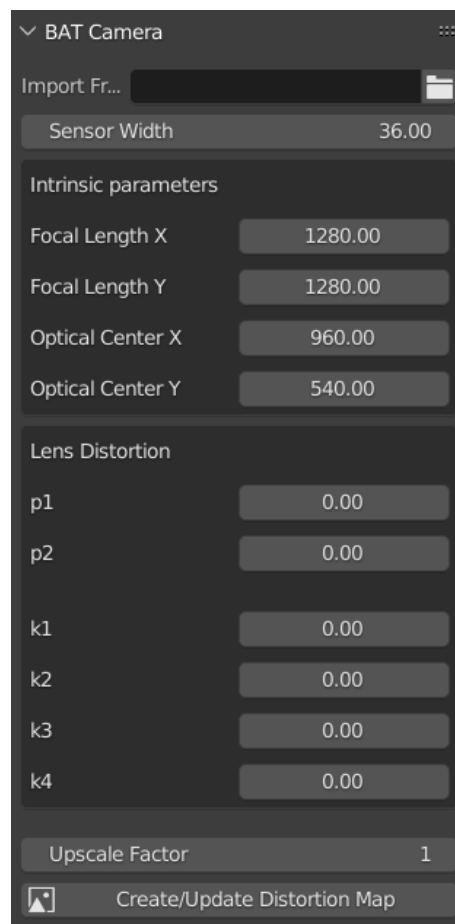


2.3. ábra: BAT Panel

A Blender Annotation Tool két részből áll. A BAT Panel fülön a Current class opciónál tudunk hozzáadni törölni vagy módosítani osztályokat. A Properties-nél tudunk a különböző osztályokhoz különböző objektumokat rendelni, valamint itt határozhatjuk meg azoknak a színét az annotációs maszkokon. Ezt követően check box-ok segítségével tudjuk megadni, hogy milyen feladatot végezzen el a BAT. Lehetőség van instance

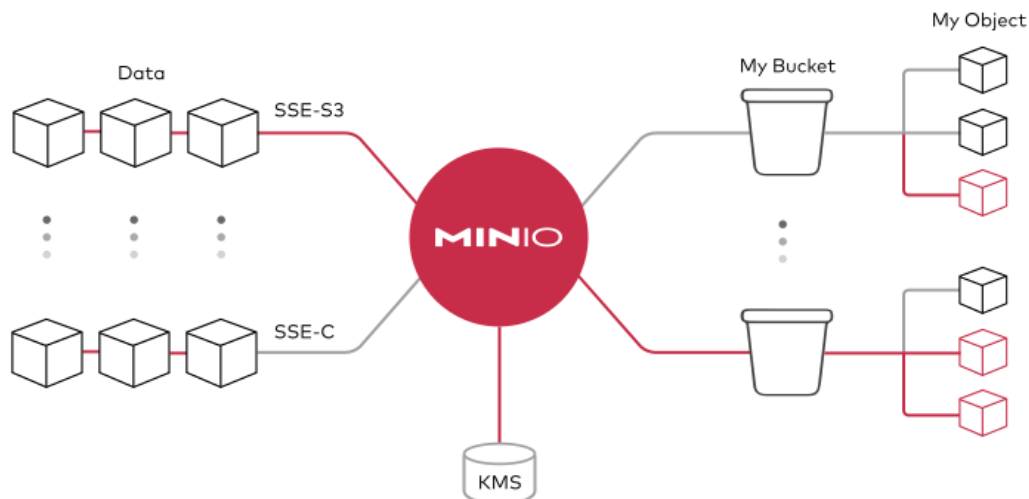
szegmentáció, mélység térkép, felületi normális és optikai áramlás készítésére is. Ezt követően a render annotation gombbal tudjuk elkészíteni az annotációt, amelyet az addon a Blender output properties fülén megadott mappába ment el.

A BAT Camera lehetővé teszi különböző kamera tulajdonságok befolyásolását. A Sensor Width opciónál beállíthatjuk a használt kamerának a szenzor méretét. Az Intrinsic parameters fülön megadhatjuk a kamera belső paraméter mátrixának értékeit is. A Lens Distortion pedig lehetővé teszi a megjelenő lencse disztorziók befolyásolását. Emellett az Upscale Factor állításával tudjuk a felskálázás mértéket meghatározni. Végül pedig a Create/Update Distortion Map gombbal tudjuk létrehozni vagy frissíteni a disztorziós térképet.



2.4. ábra: BAT Camera

Ennek a kiegészítő szoftvernek az alkalmazásával a fotórealisztikusan renderelt képekhez automatikusan tudunk szegmentációs maszkokat készíteni.



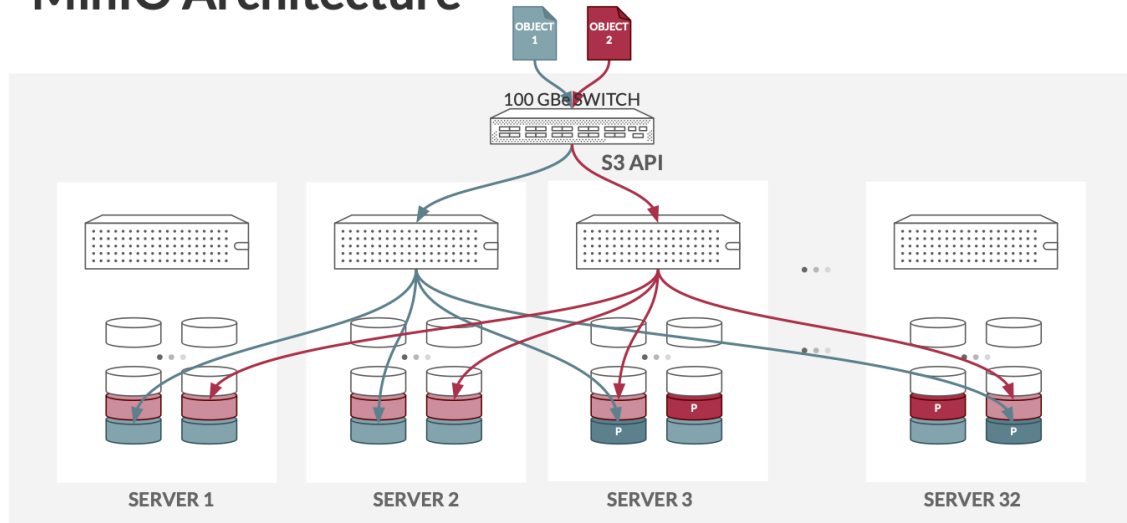
2.5. ábra: MinIO sematikus ábra⁸

2.3 MinIO

A MinIO egy nagy teljesítményű és hatékonyságú objektum tároló rendszer. Alkalmas strukturálatlan adatok, például képek, videók, log fájlok, biztonsági mentések és konténerimage-ek kezelésére. 50 TB a maximális támogatott objektum méret. A rendszer kompatibilis az Amazon S3 felhő szolgáltatásával is. A MinIO-t egy szilícium völgyi startup fejleszti, ami 2014-ben alakult és magát a szolgáltatást 2016-ban adták ki. A MinIO három fő komponensből épül fel, ezek a MinIO szerver, a MinIO kliens és a MinIO kliens SDK. A MinIO szerver egy felhő alapú tároló rendszer, amit a lehető legkisebbre terveztek, így más már létező applikációkkal könnyen működik együtt, a NodeJS-hez és a Redis-hez hasonlóan. A szolgáltatást nagyvállalati felhasználásra optimalizálták, beleértve olyan funkciókat, mint például bitrot protekció, enkódolás, identitás menedzsment, folyamatos replikáció, és több-felhős felhasználás. A MinIO szerver hardver agnosztikus és ebből kifolyólag fizikai és virtuális gépen is telepíthető. Mindemellett Docker container-ként is futtatható. A MinIO kliens remek alternatívát szolgáltat a standardnak számító UNIX parancsokhoz (pl. ls, cat, cp, mirror, diff) és biztosítja az Amazon S3 felhő szolgáltatással való kompatibilitást. Linux, Windows és macOS operációs rendszereken is egyaránt kiválóan alkalmazható. A MinIO Kliens SDK API elérést szolgáltat bármelyik Amazon S3 kompatibilis objektum tároló szerverhez. Az API több gyakran használt programozási nyelvet is támogat, ilyen például a Go, Java, Python, JavaScript, Haskell és bármelyik .NET keretrendszert alkalmazó nyelv.

⁸<https://min.io/product/overview/>

- MinIO Architecture



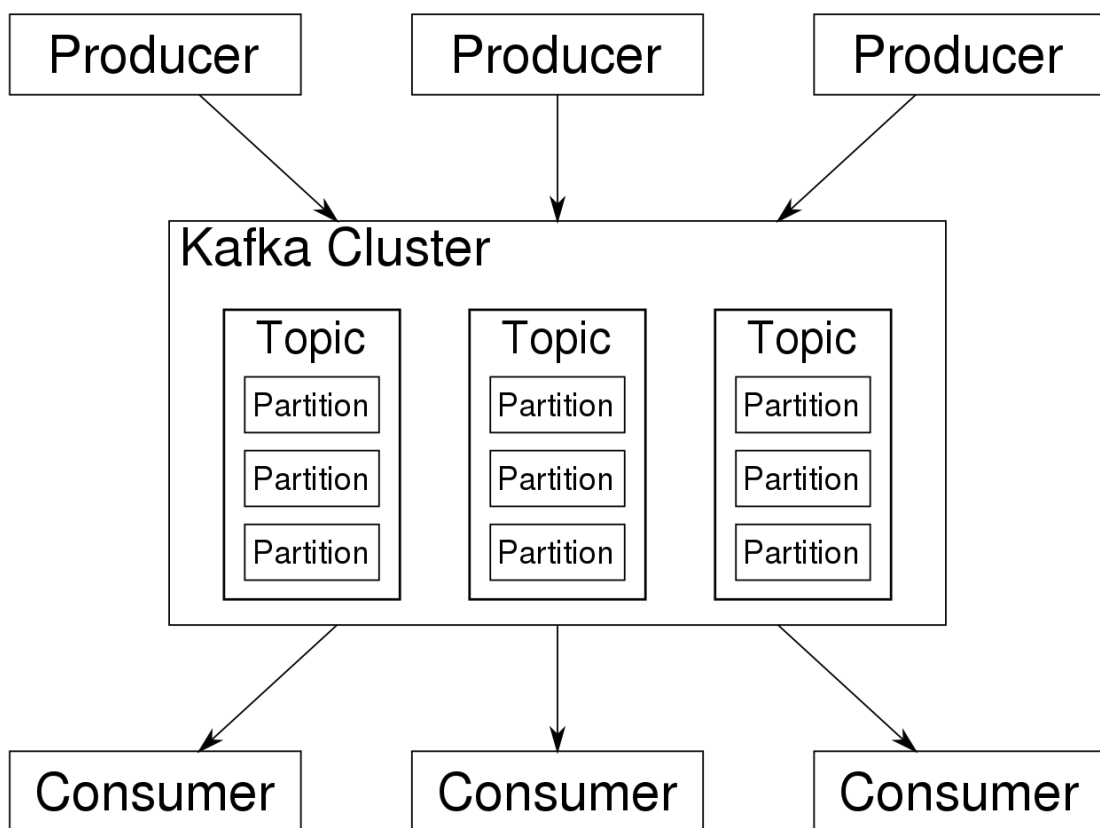
2.6. ábra: MinIO architektúra ⁹

2.4 Kafka

Az Apache Kafka egy nyílt forráskódú elosztott eseménytároló és adatfolyam-feldolgozási platform, amelyet az Apache Software Foundation fejlesztett Java és Scala nyelven. Nevét Franz Kafka íróról kapta, mivel egy írásra optimalizált rendszer és mivel a fejlesztők kedvelték Kafka munkásságát. A szolgáltatás célja egy egységes nagy teljesítményű platform létrehozása valós idejű adatfolyamok kezelésére alacsony késleltetéssel. A Kafka egy bináris TCP alapú protokollt használ, ami üzenetkészlet absztrakciót alkalmaz, az üzeneteket csoportokba szervezi, így csökkentve a hálózat oda-vissza út többletköltségét, ezzel optimalizálva a hatékonyságra. Ez nagyobb hálózati csomagokhoz, nagyobb szekvenciális lemez operációkhoz, összefüggő memóriablokkokhoz vezet, ami lehetővé teszi a szolgáltatás számára, hogy véletlenszerű üzenetek sorozatát folyamatos, lineárisra tegye. Az Apache Kafka a commit logon alapszik, lehetővé teszi a felhasználók számára, hogy feliratkozzanak rá és hogy adatokat osszanak meg tetszőleges számú rendszerrel, vagy valós-idejű applikációval. Többek közt ezen a technológián alapszik az Uber utast és sofőrt összepárosító rendszere, a British Gas valós idejű analitikáját szolgáló okos otthon rendszere és több LinkedIn valós idejű szolgáltatás is. A Kafka architektúrája alapján kulcs-érték párokat tárol, amiket különböző folyamatokból producer-ekből kap. A tárolt adat felosztható több partícióra különböző topic-okkal. Egy partíción belül az üzenetek az offset-jeik alapján vannak rendezve és timestamp-el együtt vannak eltárolva. A Kafka Stream API-ja lehetőséget nyújt adatok folyamának feldolgozására. Így olyan

⁹<https://www.penguinpunk.net/blog/minio-not-your-fathers-object-storage-platform/>

Java alkalmazások fejleszthetők, amik a Kafka által szolgáltatott adatokat feldolgozzák és az eredményt vissza is írják. A Kafka egy vagy több szerveren fut melyeket általában brokereknek hívnak. A topicok partíciói ezen szerverek közt vannak elosztva. Ez az architektúra lehetővé teszi a szolgáltatás számára, hogy nagy mennyiségű üzenetek folyamatát tudja kezelni nagy hibatűréssel, ezzel leváltva a konvencionális üzenő rendszereket, mint például a Java Message Service (JMS), Advanced Message Queuing Protocol (AMQP) stb. A Kafka általános és tömörített topicokat támogat. Az általános topicok esetében azokat a rekordokat, amelyek idősebbek az adott megőrzési időnél, vagy ha egy partíció túllépi a tárhely limitet, akkor a Kafka kitörölheti tárhely felszabadítás céljából. 7 nap megőrzési idő az alapértelmezett de lehetőség van határozatlan ideig való tárolásra is. A tömörített topic-ok ezzel szemben nem rendelkeznek ehhez hasonló megkötésekkel. Az üzeneteket a korábbi üzenetek frissítéseiként kezeli ugyanazzal a kulccsal, és garantálja, hogy soha nem törli a legújabb üzeneteket.



2.7. ábra: Kafka áttekintése ¹⁰

Kafka 5 fő API-val rendelkezik:

- Producer API

¹⁰https://en.wikipedia.org/wiki/Apache_Kafka

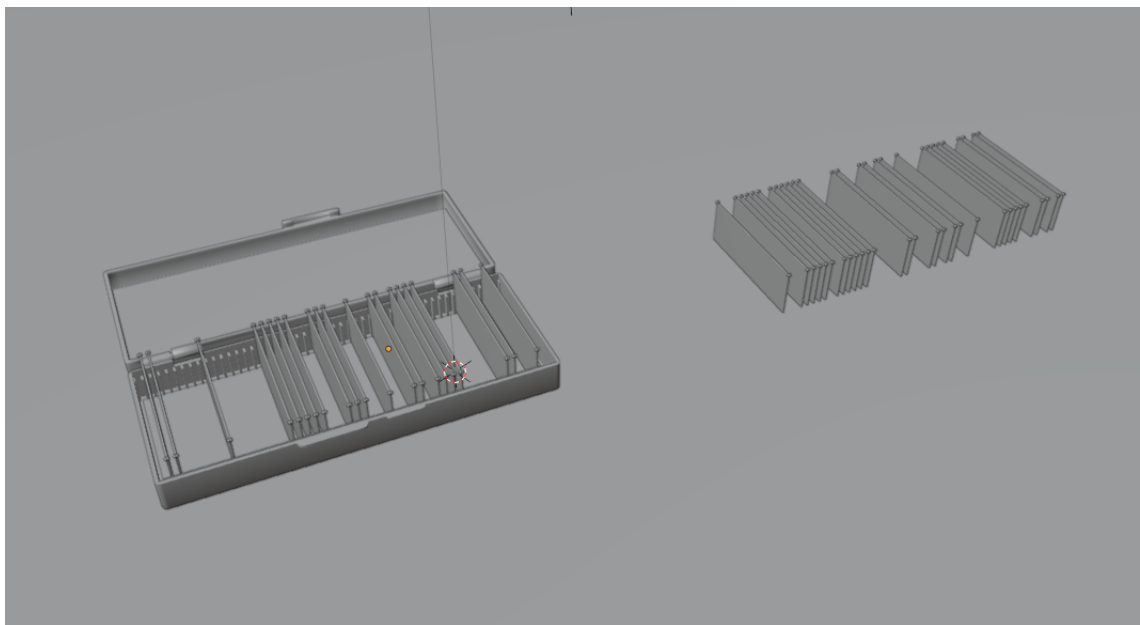
- Consumer API
- Connect API
- Streams API
- Admin API

A Producer API lehetővé teszi egy alkalmazás számára, hogy rekordok folyamát tehesen közzé. Consumer API-val egy alkalmazás különböző topic-okra feliratkozva rekordok folyamát dolgozza fel. A Connect API végrehajtja a producer és a consumer API-t ezzel összekötve topic-okat és a már létező applikációkat. A Streams API a bemeneti adat folyamot kimenetivé konvertálja és előállítja az eredményt. Az Admin API a Kafka topic-ok, brokerek és egyéb Kafka objektumok menedzselését teszi lehetővé.

3 MEGVALÓSÍTÁS ÉS EREDMÉNYEK

3.1 Szintetikus keypoint detekciós adathalmaz

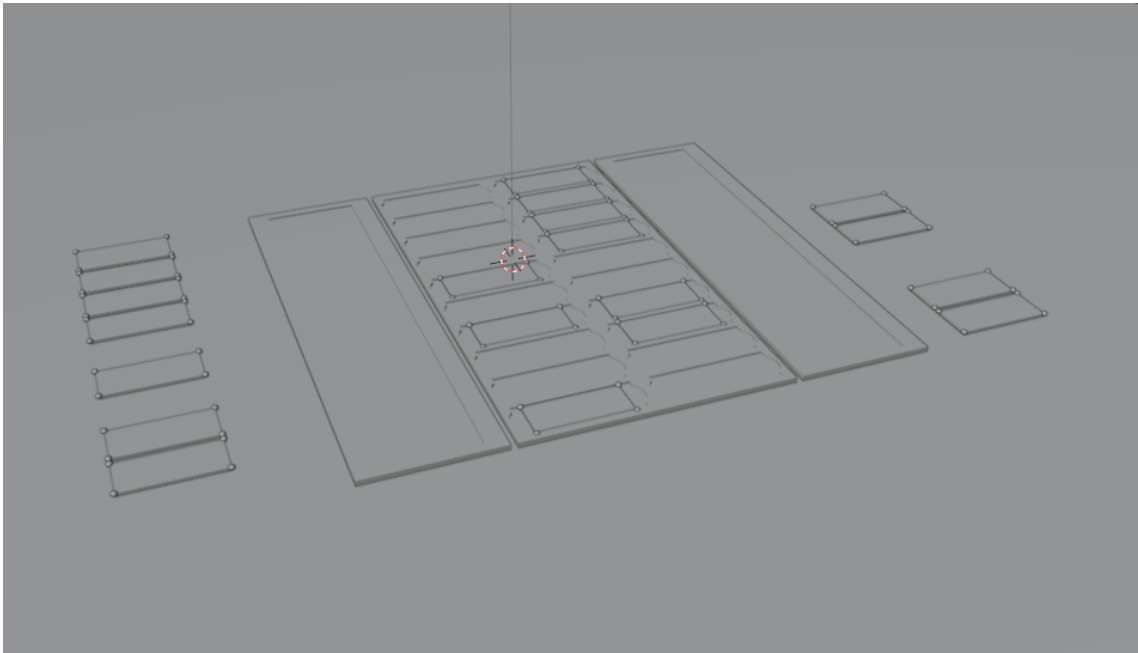
A feladat megoldására alkalmazott DL modell szintetikusán generált adathalmazon tanult fel. A döntést a szintetikus adathalmaz mellett az indokolja, hogy egy DL modell tanításához nagy mennyiségű adatra van szükség, amely megbízhatóan reprezentálja a megoldani kívánt feladatot. A patológiai mintákat különböző típusú dobozokban, mappákban tárolják. Az alkalmazott típusok laboronként különbözőek lehetnek, így nehezítve az adatgyűjtés folyamatát. Mindemelett, ha azt szeretnénk, hogy a robot különböző helyzetekben is jól használható legyen, akkor figyelembe kell venni a környezeti tényezők változását is. Ilyen például a tároló elhelyezkedése, illetve a benne lévő minták száma. Ezen kívül változhat a környezet, amelyben a megoldást használják és még a megvilágítás is. Ezen tényezők nagyban nehezítik a megbízható adathalmaz gyűjtését. Továbbá ezen adatok annotációja sem maradhat el, ami hatalmas erőforrást igényel, ha azt manuálisan kell elvégezni. Ezen okokból, sajnos valós adatok gyűjtésére nem mindig van lehetőség. Ilyenkor a szintetikus adatok generálása nyújthat megoldást.



3.1. ábra: Doboz 3D modellje

Két szintetikus keypoint detekciós adathalmaz készült el, két különböző típusú patológiai tároló eszközökről. Az egyik egy doboz melybe maximálisan 50 darab szövetszövetmintát tartalmazó üveglemez helyezhető el, a másik pedig egy két részes fedőből álló mappa, amibe 20 minta fér. A két adathalmazhoz a Blender alkalmazás használatával készültek el, a tárolók 3D modelljei, melyek a 3.1. ábrán és a 3.2. ábrán láthatóak. A doboz 3D

modelljének teteje és az alja külön objektumként kezelhetők, melyekben egy-egy papírlapot reprezentáló objektum lett elhelyezve a valóságérzet növelése végett. A mappa 2 részből áll, a két oldalon lévő fedlap és a mintákat tároló rész, amik szintúgy külön-külön kezelhető objektumok. Ezek a modellek az adathalmazok létrehozásához, külön-külön, de tulajdonságaikban megegyező 3D környezetekbe lettek elhelyezve. A 3D környezeteknek része az asztalt szimbolizáló felület, a megvilágítás és a kamera. A kamera objektum a Blenderben azt a perspektívát reprezentálja, ahonnan a renderelt fotorealistikus kép készül.

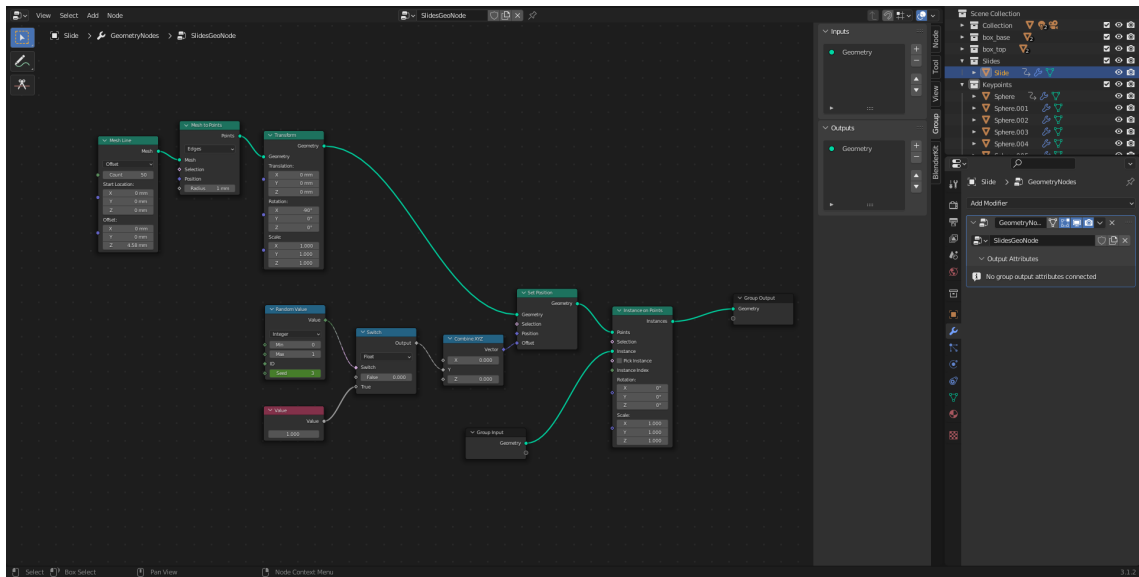


3.2. ábra: Mappa 3D modellje

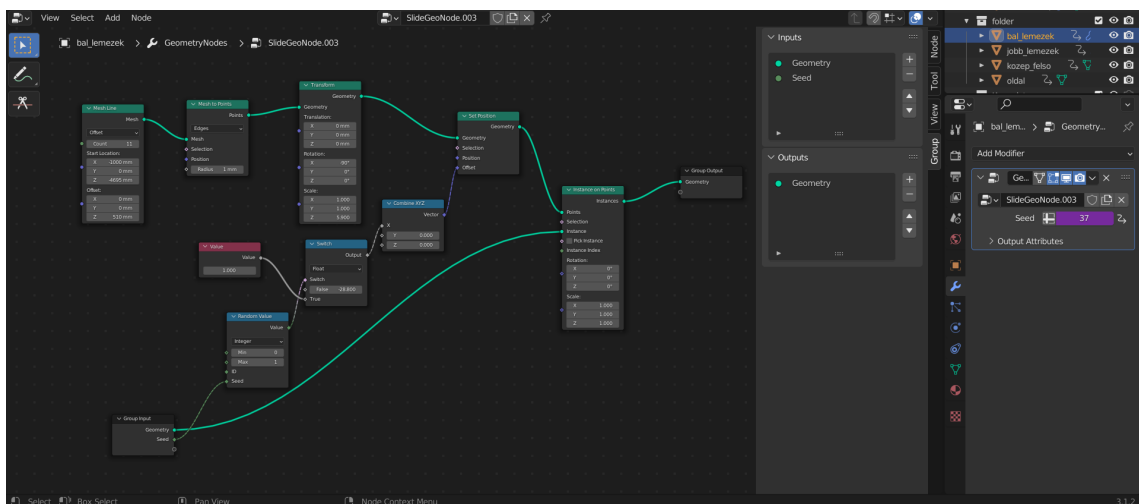
3.1.1 Üveglemezek

Mindkét esetben az üveglemezeket képviselő objektumok létrehozása a Blender Geometry Node rendszerével történt, melyek a 3.3. ábrán és a 3.4. ábrán láthatóak. Ezen funkció használatával lehetőség nyílik 3D objektumok geometriájának módosítására és komplex procedurális manipulálására. Ez egy csomópont alapú interface a Blender Geometry Node szerkesztőjében, ahol azok összeköttetésével és komplex rendszerével lehet az objektumok tulajdonságait befolyásolni. A Node-ok különböző funkciókkal rendelkeznek, változtathatják például az objektumok méretét és elhelyezkedését, vagy azok alakját, végezhetnek összeadást, kivonást, akár logikai VAGY műveletet, visszaadhatják az objektumok bizonyos tulajdonságait és emellett még sok egyéb feladatot is elláthatnak. Az üveglemezek megvalósításához is ilyen csomópontok komplex rendszerét használtuk. Egységnyi térközzel rendelkező pontok lettek létrehozva, melyek a globális vízszintes

tengely körül 90 fokban el vannak forgatva. A pontok száma az adott tároló kapacitásától függ. Ezekhez lett hozzá rendelve az üveglemezek 3D modellje, így megfelelő mennyiségű lemezt generálva. A tárolóban megjelenő lemezek számának randomizálásához egy véletlen számot generáló nodeot használtunk, ami vagy 1-et vagy pedig 0-t ad értékül. A node seedje képkockánként változik, ezzel biztosítva a véletlenszerűséget. Ezen értéktől függ az adott lemez eltolása, ezzel biztosítva, hogy az nem fog a tárolóban szerepelni. Mindkét esetben ezen megközelítés használatával hoztuk létre az üveglemezeket, a doboznál egy, míg a mappánál kettő objektumként.



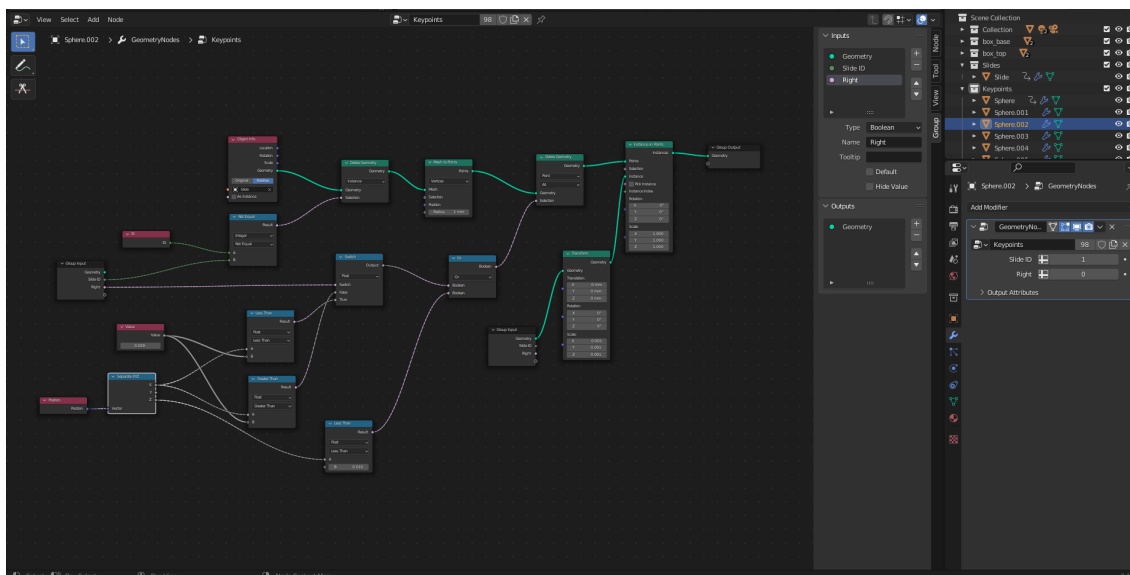
3.3. ábra: Üveglemez geometry node rendszere (doboz)



3.4. ábra: Üveglemez geometry node rendszere (mappa)

3.1.2 Keypointok

A feladat megoldására használt deep learning modell az üveglemezek sarkainak detektálására tanul fel. Az elkészült adathalmazt úgy szükséges annotálni, hogy a lemezek sarkai jól látható módon legyenek megjelölve a kész annotáción. Így a modell különböző gépi tanulási módszerek alkalmazásával képes lesz rátanulni ezen pontok detekciójára. Az üveglemezek sarkainak jelölésére a 3D környezetben is létre lettek hozva pontok (keypointok), a Blender geometry node rendszerével. Ezek a 3.5. ábrán és a 3.6. ábrán láthatók. A geometry node rendszereknek 3 bemenete van. A slideID meghatározza, hogy az adott keypoint melyik lemezhez tartozik. A top és a right bemenetek segítségével pedig azt tudjuk meghatározni, hogy a lemez melyik sarkához rendeljük a keypointot. A doboz esetén a felső két sarok egyikéhez, a mappa esetén mind a 4 sarokhoz szükséges volt egy-egy keypointot rendelni. Ezek a pontok, míg modellezés során láthatóak, addig a renderelt tanításra használt fotórealisztikus képeken nem. A pontokat a Blender Annotation Tool használja fel az annotáció elkészítésére.

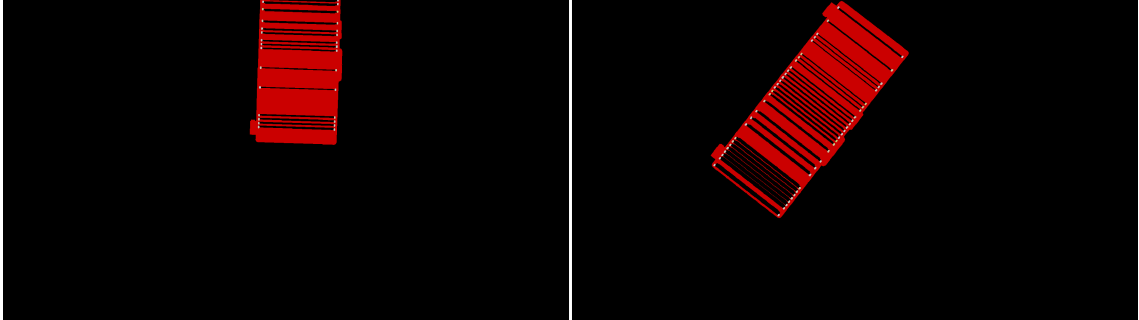


3.5. ábra: Keypointok geometry node rendszere (doboz)

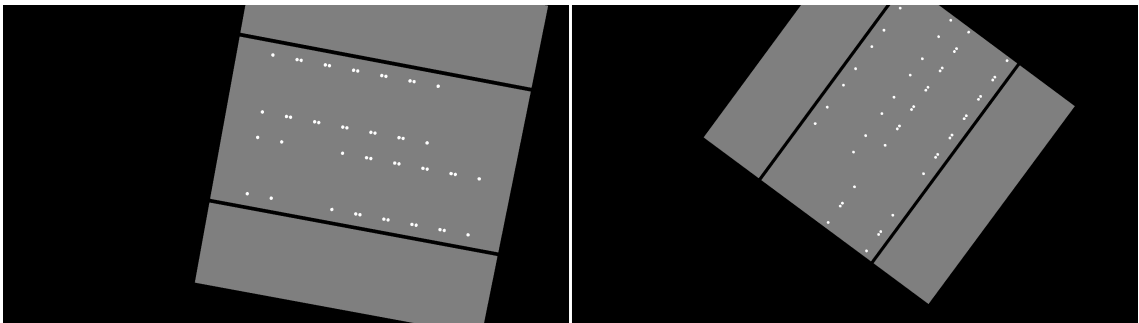
3.1.3 Változások reprezentálása

A Blender lehetővé teszi, hogy különböző objektumokhoz anyagmodelleket rendeljünk, ezzel meghatározva azok felületének megjelenését. A felületekhez olyan anyagmodell lett rendelve, hogy az fotórealisztikusan reprezentálja a folyamat során használt eszközöket. A doboz helyzetének, az asztal textúrájának, a megvilágításnak és a kamera helyzetének képkockánkénti randomizálása a Blender által szolgáltatott graph modifier funkció alkalmazásával történt. Ezzel az objektumok x és y-ban felvett helyzetéhez zaj módosítót

pedig világosabb szürkék. A keypointokhoz tartozó annotációkat az instance segmentation checkbox bepipálásával hoztuk létre, ami azt jelenti, hogy minden egyes keypoint egyedi árnyalatú szürke színnel van jelölve. A kamera belső paraméterei és a lencse disztorziós értékek az alapértelmezett értékektől nem tértek el. Az így létrejött annotációkról néhány példa a 3.8. ábrán és a 3.9. ábrán látható.



3.8. ábra: Doboz annotációs maszkok



3.9. ábra: Mappa annotációs maszkok

3.1.5 Adathalmazok generálása

Mindkét adathalmaz esetében a 3D környezeteket tartalmazó .blend kiterjesztésű Blender fájlokat a MinIO alapú fájlserverre töltöttük fel. A képek és az annotációk az Apache Kafka-t használó kafka runners repository¹¹ használatával lettek legenerálva. Ez a szolgáltatás a bemeneti Kafka topic üzenetéből parancssorból futtatható parancsokat készít, melyek végrehajtása után egy másik Kafka topicban választ üzenetet küld. Ezen szolgáltatás alkalmazásával automatizálható a fotórealisztikus képek és az annotációk generálása a .blend fájlokból. A két .blend fájlból egyenként 2000 képet és annotációt generáltunk. Az így létrejövő annotációs maszkok OpenEXR fájlként lettek elmentve, ami egy nagy dinamikatarományú, többcsatornás fájlformátum. Ez a fájl formátum több információ eltárolását teszi lehetővé, mint például a PNG formátum. Az elkészített OpenEXR fájlok

¹¹https://github.com/karolyartur/kafka_runners

csatornáit tartalmazzák a képi információkat. Ebből kinyerhetők a tároló eszközt határoló négyzet (bounding box röviden bbox) és a keypointok koordinátái, illetve azok láthatósági. Ahhoz, hogy ezek a fájlok a háló tanításra felhasználhatóak legyenek, szükséges kinyerni fájlok csatornáiból a releváns információkat. Ehhez egy python szkript készült, mely a MinIO-ból olvassa be az annotációkat (OpenEXR fájlok), majd a kinyert koordinátákat egy json fájlba menti, amit aztán fel is tölt a MinIO tárhelybe. A hálózat tanításához ezeket a json fájlokat használtuk fel.



3.10. ábra: Szintetikus adatok a dobozról



3.11. ábra: Szintetikus adatok a mappáról



3.12. ábra: Mappa szintetikus képe és az annotációja együtt

3.2 Tanítási folyamat

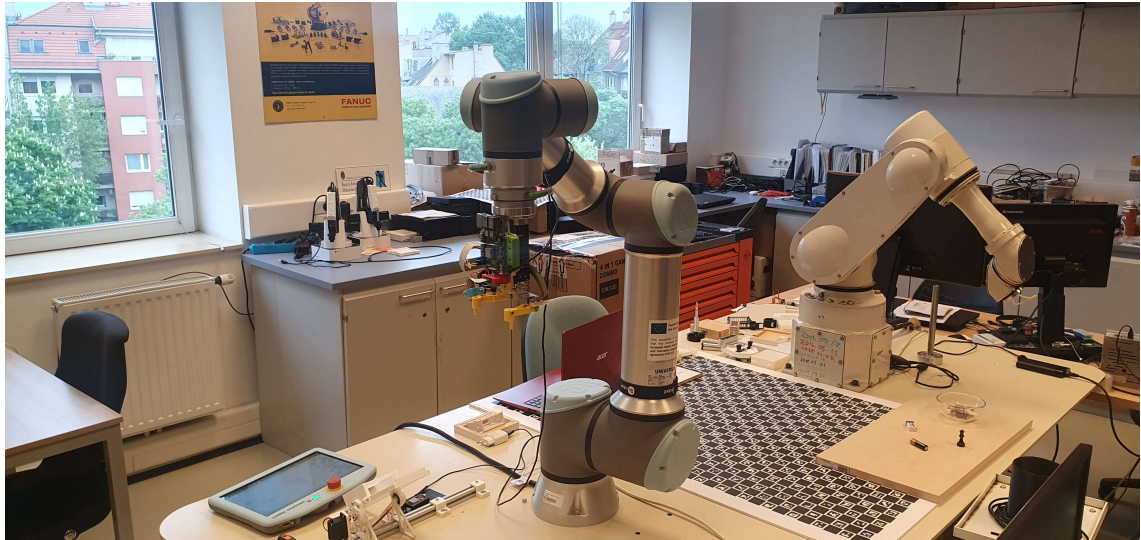
A szintetikusan előállított adathalmazokon, az üveglemezek detektálására, egy keypointok detektálására módosított Mask R-CNN modellt tanítottunk fel. A Keypoint-RCNN egy, a Torchvision Python programcsomagba implementált modell, mely keypointok detektálását végzi, úgy hogy a keypointokat egy one-hot bináris maszkként értelmezi. A modell tanítása során bemenetként a tanító kép mellett, alapigazsággként, az objektumot határoló bounding box helyzetét és a detektálandó keypointok koordinátáit is szükséges megadni. A bounding box helyzete meghatározható 4 érték által, melyek a téglalap által az x és y tengelyen felvett maximális és minimális értékek. Egy keypointot az x és y tengelyen felvett értékeivel és egy láthatósági értékkel szükséges megadni. A modell alapvetően viszont tanulás során csak a keypointok helyzetét tanulja meg. A láthatósági érték a kimeneten mindig 1-esre van állítva. Ahhoz hogy a Keypoint R-CNN felhasználható legyen a dolgozatban meghatározott problémára egy kisebb módosításon esett át¹². Egy extra fully connected réteg lett hozzáadva a modellhez, sigmoid aktivációs függvényvel, melynek kimeneteinek száma megegyezik a detektálandó keypointok számával. Az új réteg minden keypointhoz egy láthatósági értéket prediktál. Ezen módosításra azért van szükség, mivel a keypointok becsült láthatósága dönti el, hogy egy üveglemez benne van-e a tároló eszközben vagy sem.

A rendelkezésre álló, két különböző patológiai tároló eszközt reprezentáló adathalmazon, egy-egy tanítást hajtottunk végre. A dolgozatban a mappát reprezentáló adathalmazon végzett tanítás körülményeit és eredményeit ismertetem. Az adathalmaz 2000 képből áll melyből 1800-at tanításra, és a maradék 200-at pedig tesztelésre és metrikák számítására használtuk. Emellett a háló valós környezetben való tesztelésére 10 képet készítettünk a valós tároló eszközről, melyen később a modell predikciót végez. A képek a 3.13. ábrán látható ipari robot segítségével készültek. A robot megfogóján egy Raspberry Pi található, melynek kamerájával készültek a felvételek. A képeken a mappa különböző mennyiségű üveglemezt tartalmaz és különböző pozíciókat vesz fel.

A tanítás 100 epochon keresztül történt, ami azt jelenti, hogy a teljes tanító adathalmazt a modell 100 alkalommal dolgozta fel a tanítás során. A detektálandó keypointok száma az adott tároló eszközöknek a kapacitását figyelembe véve lett meghatározva. A mappa 20 üveglemez tárolására alkalmas, melyek fektetve helyezkednek el, így egy üveglemeznek mind a 4 sarka látható. Ez azt jelenti, hogy ennél a tároló eszköznél a modell által detektálandó keypointok száma 80. A modell tanításához szükséges megadni a szintetikus képekhez tartozó alapigazságokat. Ezek a tároló eszközt határoló bounding box értékei és a keypointok koordinátái, láthatóságai. Mindez az előre kimentett json fájlok-

¹²<https://github.com/pytorch/vision/compare/main...karolyartur:vision:main>

ból lett beolvasva. A neurális háló optimalizálásához a Stochastic Gradient Descent [51] (SGD) algoritmust használtuk. A modellt 0.001 kezdeti learning rate-el, 0.9 momentum-mal, 0.0005 weight decay-értékkal tanítottuk. A modell az előtanított ResNET 50 hálót [52] használja backbone-jaként. Ezeken kívül a modell többi paramétere nem tér el az alapértelmezettől. A tanítás egy NVIDIA A100 GPU-val rendelkező számítógépen történt és 12 órát vett igénybe.



3.13. ábra: Ipari robot a valós adatgyűjtéshez

3.3 Kiértékelés

A keypointokat detektáló modellek pontosságának meghatározásához az Object Keypoint Similarity (OKS) metrikát szokás használni, ami nem terület, hanem távolság alapú megközelítést alkalmaz. Kiszámításához az alapigazság és a prediktált keypointok euklideszi távolságát (d_i), egy nem normalizált Gauss-eloszláson szükséges meghatározni adott szórással. A szórás az objektum szegmentált területének négyzetgyökével (s) és egy keypointokként változó k értékkel határozható meg. Ezek alapján keypointonként egy KS értéket számolhatunk. Minden keypoint esetén az OKS egy 0 és 1 közötti érték.

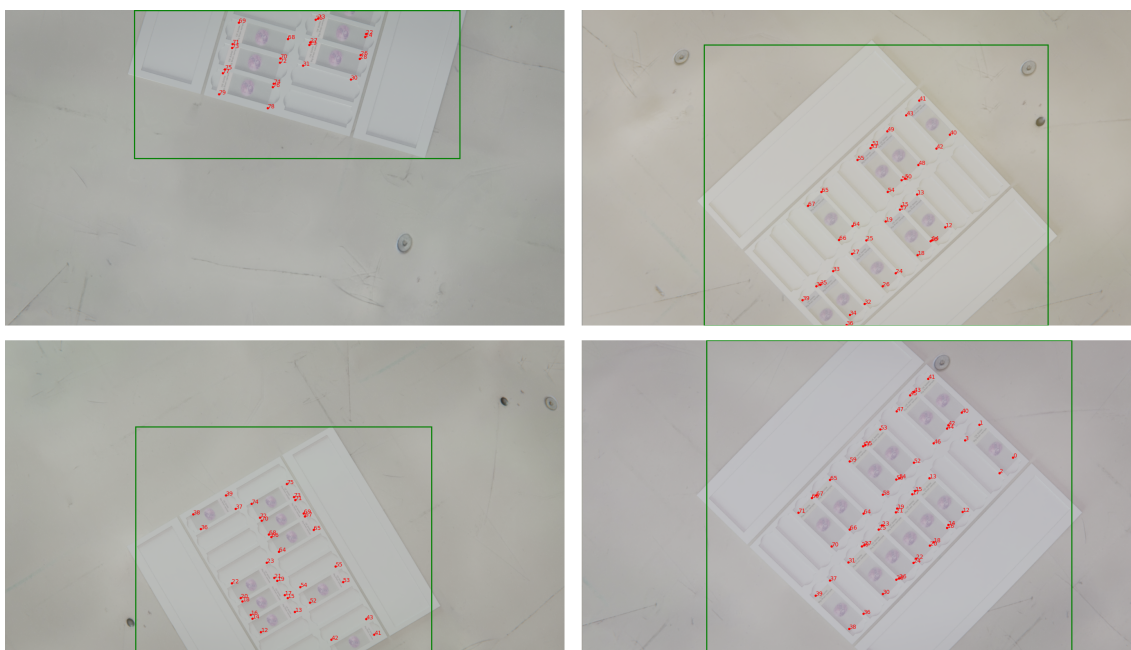
$$KS_i = \exp\left(\frac{-d_i^2}{2s^2k_i^2}\right) \quad (3.1)$$

A metrikát az MS-COCO úgy határozta meg, hogy a keypointok rendelkeznek egy láthatósági értékkel v . Ez három értéket vehet fel 0-t, 1-et, és végül 2-öt. Ezeknek jelentése sorban: a keypoint nem szerepel a képen, szerepel, de ki van takarva és végül a keypoint szerepel. Az Object Keypoint Similarity (OKS) végül egy példány összes annotált és prediktált keypointjai közötti hasonlóságok (KS) számtani átlagaként kerül kiszámításra a

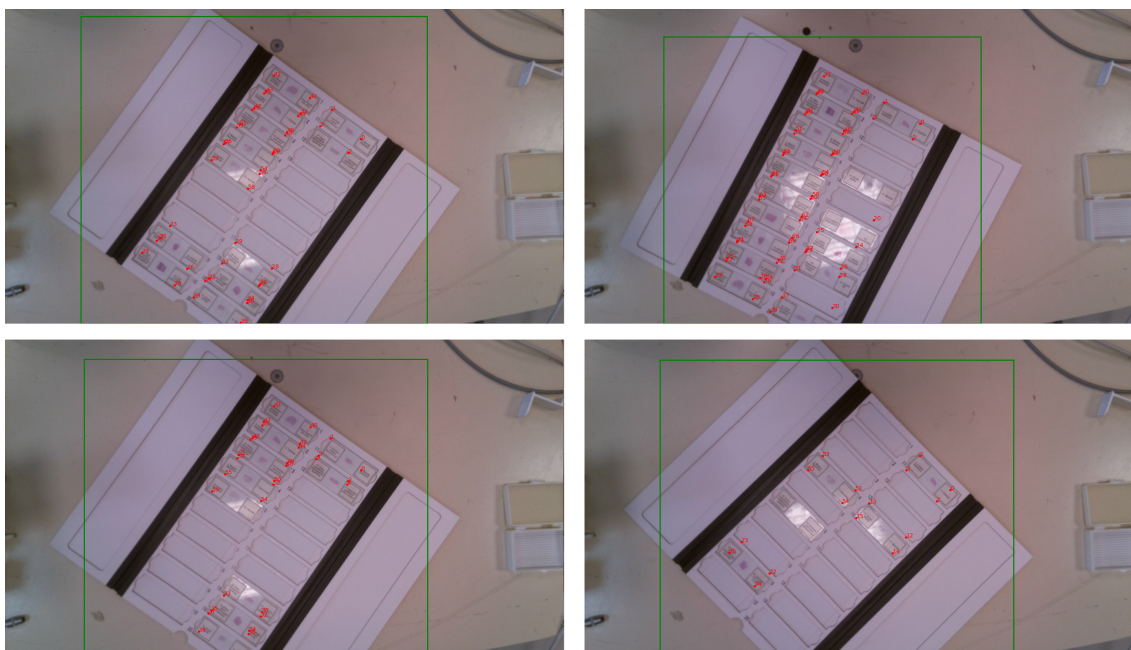
láthatósági értékek figyelembevételével. Ez összhangban van a dolgozatban használt láthatósági értékkel mivel csak akkor nem számítható a metrika, ha a keypoint nem látható.

$$OKS = \frac{\sum_i K S_i \cdot \delta(v_i > 0))}{\sum_i \delta(v_i > 0)} \quad (3.2)$$

A modell 200 szintetikusan előállított képen lett kiértékelve, melyből pár példa predikció a 3.14. ábrán látható. Észrevehető, hogy a modell sikeresen határozza meg a mappát határoló bounding box értékeit. Mindemellett, sikeresen detektálja az üveglemezek mind a 4 sarkát és azoknak láthatósági értékeit is. Az utóbbit onnan tudjuk megállapítani, hogy a képen csak azok a keypointok szerepelnek, ahol ténylegesen vannak üveglemezek. Az implementációhoz szükséges idő hiányában az OKS helyett az euklideszi távolságot (L2 távolság) használtuk a tanítás kvantitatív kiértékeléséhez. Ezen metrika OKS kiszámításához is szükséges és emellett alkalmas a modell teljesítményének és a tanítás eredményességének reprezentálására. Megmondja a valós és a prediktelt keypointok közötti távolságot pixelben mérve. Az L2 távolságok átlaga a 200 szintetikus képen **3.7586 pixel**, ami azt jelenti, hogy átlagosan nagyjából csak 4 pixel eltérés van a modell kimenete és a valóság között. Ezen eredmény kifejezetten jónak tekinthető és igazolja, hogy az adott problémára megfelelő modell és azon megfelelő módosítások lettek végrehajtva. A modell kvalitatív kiértékeléséhez valós képeken végzett predikciókat hajtottunk végre, amik a 3.15. ábrán láthatóak. Ezeken észrevehető, hogy a mappát minden esetben megtalálja a modell, viszont a keypointokat sokszor nem sikerül megfelelően elhelyeznie. Ezekből az eredményekből az a következtetés vonható le, hogy a szintetikus adatok nem kellő mértékben reprezentálják a valóságot. Több eltérés is felfedezhető a valós és szintetikus képek közt, például a mappa fedlapjainál jelenlévő fekete rész. Emellett a szintetikus képeken a mappa más anyagúnak látszik, mint a valósakon. Észrevehető még a lemezeken egyfajta becsillanás is, amit a laborban lévő lámpák okoznak. Ezek reprezentálása szükséges lesz a későbbiekben ahhoz, hogy a modell a valós életben is kiváló eredményeket érjen el, ne csak szintetikus adatokon.



3.14. ábra: Predikció szintetikus képeken



3.15. ábra: Predikció valós képeken

4 ÖSSZEFOGLALÁS ÉS JÖVŐBELI FELADATOK

Ezen dolgozat a patológiai minták digitalizálásának automatizálására, egy DL alapú robotos manipulációt használó megoldást kínál. A használt modell a Keypoint R-CNN Torchvisionbeli implementációja, amely kisebb módosításokon esett át, hogy a detektálandó keypointok láthatóságát is figyelembe vegye. A modell tanításához két adathalmaz lett létrehozva, két különböző szövettani mintákat tároló eszközzel. Ezen adathalmazok Blender 3D modellező alkalmazás segítségével, szintetikusán lettek létrehozva. A két adathalmaz fotórealisztikusan reprezentálja a lehetséges környezeti változásokat. Ilyen például a tároló eszköz helyzete, a benne szereplő minták száma, a megvilágítás, az asztal textúrája. Az adatokhoz a BAT Blender addon segítségével annotációs maszkok készültek. Ezeken a maszkokon a mintákat tároló lemezek sarkaira keypointok lettek elhelyezve, ezen pontok detektálására tanult fel a modell. Az adathalmazok létrehozásának automatizálásához Apache Kafka alapú szolgáltatást használtunk. Ezen szolgáltatás használatával tároló eszköz típusonként 2000 képből és annotációból álló adathalmazok lettek legenerálva. Az így létrejövő adathalmazok egy MinIO alapú fájlserveren lettek eltárolva. Az annotációs maszkokból egy python szkript segítségével a keypointokat és a bounding box koordinátáit json fájlokba mentettük ki. Ezeket a fájlokat a modell tanításához használtuk.

A dolgozat megírásának pontjái a két szintetikus adathalmaz közül, a mappát ábrázoló adathalmazon lett tanítás végrehajtva. A 2000 képből 1800 tanításra, a maradék 200 pedig tesztelésre lett felhasználva. A tanítás egy NVIDIA A100-as GPU-val rendelkező számítógépen futott, megközelítőleg 12 órán keresztül. A tanítás eredményessége kvalitatív és kvantitatív módon is kiértékelésre került. A dolgozatban használt kvantitatív metrika a valós és a prediktelt keypointok távolságának átlaga a 200 szintetikus képen. Ennek eredménye **3.7586 pixel** lett, ami egy rendkívül jónak mondható érték. Ez azt jelenti, hogy a modell által prediktelt keypointok átlagosan nagyjából 4 pixellel térnek el a valóságtól. A szintetikus teszt adatokon végzett predikcióról pár példa a 3.14. ábrán látható. A tanítás kvalitatív kiértékelése valós tároló eszközzel készült képeken történt melyet az 3.13. ábrán látható robot segítségével lettek elkészítve. A valós képeken történt predikciók eredményei a 3.15. ábrán láthatóak.

Az elért eredmények alapján a megközelítés jónak tekinthető, de még messze nem optimális. A modell által a szintetikus adathalmazon elért nagyszerű eredmények alapján azt a következtetést lehet megtenni, hogy az általunk választott modell és az azon végrehajtott módosítások megfelelőek. Mindemellett tanítás során használt hiperparaméterek is jónak tekinthetőek, az eredmények fényében. A kvalitatív eredmények alapján azt a következtetést tehetjük meg, hogy a szintetikus adatoknak valósághűbben kell reprezentálnia a tároló eszközt. Ezek alapján lehet tovább indulni és a jövőben megtenni a szükséges módosításokat. Ilyen például, hogy szintetikus adatokon a mappa anyaghasz-

nálata, a becslés szimulálása, több fajta üveglemez használata és még sok más. Jó megközelítés lehet egy kisebb valós képekből készült adathalmazt manuálisan annotálni. Ez több területen is felhasználható, például tanításhoz belekeverni az adathalmazba, ezzel növelve annak sokoldalúságát, továbbá a domain randomizáció [50] technika használata is előnyös lehet a szintetikus adatok augmentálására (erről bővebben a kapcsolódó munkák szekciójában). Ezek a megközelítések elősegíthetik a modell könnyebb általánosítását. Mindemellett egy kézzel annotált adathalmaz felhasználható lenne kvalitatív kiértékelés során is metrikák számítására. Ezek mellett nagyobb adathalmaz használata tanítás során is egy olyan megoldás, ami elősegítheti a modell pontosságának növelését. Például egy 20000 szintetikus képen történő tanítás, ahol 18000 a tanító és 2000 a validációra használt képek száma. Mindemelett a patológiai laborokban használt tároló dobozok típusa igencsak változatos, így a további modellek elkészítése és az abból készült adatokon való tanítás még várat magára. Ezen aspektusok alapján lehet tovább indulni és tovább fejleszteni a dolgozat által ajánlott megoldást.

5 IRODALOMJEGYZÉK

- [1] S.-C. Wang és S.-C. Wang, „Artificial neural network”, *Interdisciplinary computing in java programming*, 81–100. old., 2003.
- [2] S. Grigorescu, B. Trasnea, T. Cocias és G. Macesanu, „A survey of deep learning techniques for autonomous driving”, *Journal of Field Robotics*, 37. évf., 3. sz., 362–386. old., 2020.
- [3] H. A. Pierson és M. S. Gashler, „Deep learning in robotics: a review of recent research”, *Advanced Robotics*, 31. évf., 16. sz., 821–835. old., 2017.
- [4] R. Miotto, F. Wang, S. Wang, X. Jiang és J. T. Dudley, „Deep learning for health-care: review, opportunities and challenges”, *Briefings in bioinformatics*, 19. évf., 6. sz., 1236–1246. old., 2018.
- [5] D. W. Otter, J. R. Medina és J. K. Kalita, „A survey of the usages of deep learning for natural language processing”, *IEEE transactions on neural networks and learning systems*, 32. évf., 2. sz., 604–624. old., 2020.
- [6] M. Raghu és E. Schmidt, „A survey of deep learning for scientific discovery”, *arXiv preprint arXiv:2003.11755*, 2020.
- [7] R. K. Sinha, R. Pandey és R. Pattnaik, „Deep learning for computer vision tasks: a review”, *arXiv preprint arXiv:1804.03928*, 2018.
- [8] Z. Li, F. Liu, W. Yang, S. Peng és J. Zhou, „A Survey of Convolutional Neural Networks: Analysis, Applications, and Prospects”, *IEEE Transactions on Neural Networks and Learning Systems*, 33. évf., 12. sz., 6999–7019. old., 2022. DOI: 10.1109/TNNLS.2021.3084827.
- [9] L. O. Schwen, T.-R. Kiehl, R. Carvalho, N. Zerbe és A. Homeyer, „Digitization of pathology labs: A review of lessons learned”, *Laboratory Investigation*, 100244. old., 2023.
- [10] M. G. Hanna, V. E. Reuter, M. R. Hameed és tsai., „Whole slide imaging equivalency and efficiency study: experience at a large academic center”, *Modern Pathology*, 32. évf., 7. sz., 916–928. old., 2019.
- [11] D. J. Hartman, L. Pantanowitz, J. McHugh, A. L. Piccoli, M. J. OLeary és G. R. Lauro, „Enterprise implementation of digital pathology: feasibility, challenges, and opportunities”, *Journal of digital imaging*, 30. évf., 555–560. old., 2017.
- [12] O. Kusta, C. V. Rift, T. Risør, E. Santoni-Rugiu és J. B. Brodersen, „Lost in digitization—A systematic review about the diagnostic test accuracy of digital pathology solutions”, *Journal of Pathology Informatics*, 100136. old., 2022.

- [13] G. Lujan, J. C. Quigley, D. Hartman és tsai., „Dissecting the business case for adoption and implementation of digital pathology: a white paper from the digital pathology association”, *Journal of Pathology Informatics*, 12. évf., 1. sz., 17. old., 2021.
- [14] Blender Online Community, *Blender - a 3D modelling and rendering package*, Blender Foundation, Stichting Blender Foundation, Amsterdam, 2018.
- [15] A. I. Károly és P. Galambos, „Automated dataset generation with blender for deep learning-based object segmentation”, *2022 IEEE 20th Jubilee World Symposium on Applied Machine Intelligence and Informatics (SAMi)*, IEEE, 2022, 329–334. old.
- [16] N. Dalal és B. Triggs, „Histograms of oriented gradients for human detection”, *2005 IEEE computer society conference on computer vision and pattern recognition (CVPR'05)*, Ieee, 1. köt., 2005, 886–893. old.
- [17] D. G. Lowe, „Distinctive image features from scale-invariant keypoints”, *International journal of computer vision*, 60. évf., 91–110. old., 2004.
- [18] M. Hearst, S. Dumais, E. Osuna, J. Platt és B. Scholkopf, „Support vector machines”, *IEEE Intelligent Systems and their Applications*, 13. évf., 4. sz., 18–28. old., 1998. DOI: 10.1109/5254.708428.
- [19] A. Krizhevsky, I. Sutskever és G. E. Hinton, „Imagenet classification with deep convolutional neural networks”, *Advances in neural information processing systems*, 25. évf., 2012.
- [20] J. Deng, W. Dong, R. Socher, L.-J. Li, K. Li és L. Fei-Fei, „ImageNet: A large-scale hierarchical image database”, *2009 IEEE Conference on Computer Vision and Pattern Recognition*, 2009, 248–255. old. DOI: 10.1109/CVPR.2009.5206848.
- [21] R. Girshick, J. Donahue, T. Darrell és J. Malik, „Rich feature hierarchies for accurate object detection and semantic segmentation”, *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2014, 580–587. old.
- [22] P. Sermanet, D. Eigen, X. Zhang, M. Mathieu, R. Fergus és Y. LeCun, „Overfeat: Integrated recognition, localization and detection using convolutional networks”, *arXiv preprint arXiv:1312.6229*, 2013.
- [23] M. Everingham, L. Van Gool, C. K. I. Williams, J. Winn és A. Zisserman, „The Pascal Visual Object Classes (VOC) Challenge”, *International Journal of Computer Vision*, 88. évf., 2. sz., 303–338. old., 2010. jún.
- [24] R. Girshick, „Fast R-CNN”, *Proceedings of the IEEE international conference on computer vision*, 2015, 1440–1448. old.

- [25] K. He, X. Zhang, S. Ren és J. Sun, „Spatial pyramid pooling in deep convolutional networks for visual recognition”, *IEEE transactions on pattern analysis and machine intelligence*, 37. évf., 9. sz., 1904–1916. old., 2015.
- [26] S. Ren, K. He, R. Girshick és J. Sun, „Faster R-CNN: Towards real-time object detection with region proposal networks”, *Advances in neural information processing systems*, 28. évf., 2015.
- [27] T. Lin, M. Maire, S. J. Belongie és tsai., „Microsoft COCO: Common Objects in Context”, *CoRR*, abs/1405.0312 évf., 2014.
- [28] K. He, G. Gkioxari, P. Dollár és R. Girshick, „Mask R-CNN”, *Proceedings of the IEEE international conference on computer vision*, 2017, 2961–2969. old.
- [29] G. Campanella, M. G. Hanna, L. Geneslaw és tsai., „Clinical-grade computational pathology using weakly supervised deep learning on whole slide images”, *Nature medicine*, 25. évf., 8. sz., 1301–1309. old., 2019.
- [30] W. Bulten, H. Pinckaers, H. van Boven és tsai., „Automated deep-learning system for Gleason grading of prostate cancer using biopsies: a diagnostic study”, *The Lancet Oncology*, 21. évf., 2. sz., 233–241. old., 2020.
- [31] P. Ström, K. Kartasalo, H. Olsson és tsai., „Artificial intelligence for diagnosis and grading of prostate cancer in biopsies: a population-based, diagnostic study”, *The Lancet Oncology*, 21. évf., 2. sz., 222–232. old., 2020.
- [32] N. Coudray, P. S. Ocampo, T. Sakellaropoulos és tsai., „Classification and mutation prediction from non-small cell lung cancer histopathology images using deep learning”, *Nature medicine*, 24. évf., 10. sz., 1559–1567. old., 2018.
- [33] J. N. Kather, A. T. Pearson, N. Halama és tsai., „Deep learning can predict microsatellite instability directly from histology in gastrointestinal cancer”, *Nature medicine*, 25. évf., 7. sz., 1054–1056. old., 2019.
- [34] A. Echle, N. T. Rindtorff, T. J. Brinker, T. Luedde, A. T. Pearson és J. N. Kather, „Deep learning in cancer pathology: a new generation of clinical biomarkers”, *British journal of cancer*, 124. évf., 4. sz., 686–696. old., 2021.
- [35] S. Narkhede, „Understanding AUC-ROC curve”, *Towards Data Science*, 26. évf., 1. sz., 220–227. old., 2018.
- [36] A. Işın, C. Direkoğlu és M. Şah, „Review of MRI-based brain tumor image segmentation using deep learning methods”, *Procedia Computer Science*, 102. évf., 317–324. old., 2016.
- [37] A. Dosovitskiy, P. Fischer, E. Ilg és tsai., „FlowNet: Learning optical flow with convolutional networks”, *Proceedings of the IEEE international conference on computer vision*, 2015, 2758–2766. old.

- [38] N. Mayer, E. Ilg, P. Hausser és tsai., „A large dataset to train convolutional networks for disparity, optical flow, and scene flow estimation”, *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2016, 4040–4048. old.
- [39] D. J. Butler, J. Wulff, G. B. Stanley és M. J. Black, „A naturalistic open source movie for optical flow evaluation”, *Computer Vision–ECCV 2012: 12th European Conference on Computer Vision, Florence, Italy, October 7-13, 2012, Proceedings, Part VI 12*, Springer, 2012, 611–625. old.
- [40] W. Qiu és A. Yuille, „Unrealcv: Connecting computer vision to unreal engine”, *Computer Vision–ECCV 2016 Workshops: Amsterdam, The Netherlands, October 8-10 and 15-16, 2016, Proceedings, Part III 14*, Springer, 2016, 909–916. old.
- [41] A. Handa, V. Patraucean, V. Badrinarayanan, S. Stent és R. Cipolla, „Understanding real world indoor scenes with synthetic data”, *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2016, 4077–4085. old.
- [42] J. McCormac, A. Handa, S. Leutenegger és A. J. Davison, „Scenenet rgb-d: 5m photorealistic images of synthetic indoor trajectories with ground truth”, *arXiv preprint arXiv:1612.05079*, 2016.
- [43] G. Ros, L. Sellart, J. Materzynska, D. Vazquez és A. M. Lopez, „The synthia dataset: A large collection of synthetic images for semantic segmentation of urban scenes”, *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2016, 3234–3243. old.
- [44] S. R. Richter, V. Vineet, S. Roth és V. Koltun, „Playing for data: Ground truth from computer games”, *Computer Vision–ECCV 2016: 14th European Conference, Amsterdam, The Netherlands, October 11-14, 2016, Proceedings, Part II 14*, Springer, 2016, 102–118. old.
- [45] M. Müller, V. Casser, J. Lahoud, N. Smith és B. Ghanem, „Sim4cv: A photo-realistic simulator for computer vision applications”, *International Journal of Computer Vision*, 126. évf., 902–919. old., 2018.
- [46] A. Gaidon, Q. Wang, Y. Cabon és E. Vig, „Virtual worlds as proxy for multi-object tracking analysis”, *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2016, 4340–4349. old.
- [47] S. Hinterstoisser, V. Lepetit, P. Wohlhart és K. Konolige, „On pre-trained image features and synthetic images for deep learning”, *Proceedings of the European Conference on Computer Vision (ECCV) Workshops*, 2018.
- [48] M. Johnson-Roberson, C. Barto, R. Mehta, S. N. Sridhar, K. Rosaen és R. Vasudevan, „Driving in the matrix: Can virtual worlds replace human-generated annotations for real world tasks?”, *arXiv preprint arXiv:1610.01983*, 2016.

- [49] J. Tremblay, A. Prakash, D. Acuna és tsai., „Training deep networks with synthetic data: Bridging the reality gap by domain randomization”, *Proceedings of the IEEE conference on computer vision and pattern recognition workshops*, 2018, 969–977. old.
- [50] J. Tobin, R. Fong, A. Ray, J. Schneider, W. Zaremba és P. Abbeel, „Domain randomization for transferring deep neural networks from simulation to the real world”, *2017 IEEE/RSJ international conference on intelligent robots and systems (IROS)*, IEEE, 2017, 23–30. old.
- [51] S.-i. Amari, „Backpropagation and stochastic gradient descent method”, *Neurocomputing*, 5. évf., 4-5. sz., 185–196. old., 1993.
- [52] B. Koonce és B. Koonce, „ResNet 50”, *Convolutional Neural Networks with Swift for Tensorflow: Image Recognition and Dataset Categorization*, 63–72. old., 2021.

6 ÁBRAJEGYZÉK

1.1. Átlagos konvolúciós háló felépítése	4
1.2. Digitális patológiai munkafolyamat [9]	6
1.3. R-CNN [21]	8
1.4. Fast R-CNN [24]	9
1.5. Faster R-CNN	10
1.6. Mask R-CNN	11
1.7. Keypoint R-CNN	12
1.8. GenePix SL50	15
2.1. Blender szoftver felülete	16
2.3. BAT Panel	19
2.4. BAT Camera	20
3.1. Doboz 3D modellje	25
3.2. Mappa 3D modellje	26
3.3. Üveglemez geometry node rendszere (doboz)	27
3.4. Üveglemez geometry node rendszere (mappa)	27
3.5. Keypointok geometry node rendszere (doboz)	28
3.6. Keypointok geometry node rendszere (mappa)	29
3.7. Noise modifier	29
3.8. Doboz annotációs maszkok	30
3.9. Mappa annotációs maszkok	30
3.10. Szintetikus adatok a dobozról	31
3.11. Szintetikus adatok a mappáról	31
3.12. Mappa szintetikus képe és az annotációja együtt	31
3.13. Ipari robot a valós adatgyűjtéshez	33
3.14. Predikció szintetikus képeken	35

3.15. Predikció valós képeken	35
---	----

7 RÖVIDÍTÉSEK

DL	Deep learning
BBOX	Bounding box
ROI	Region of Interest
HOG	Histogram of Oriented Gradients
RPN	Region Proposal Network
CNN	Convolutional Neural Network
SVM	Support Vector Machine
IOU	Intersection over Union
MAP	Mean Average Precision