

ÓBUDAI EGYETEM
ÓBUDA UNIVERSITY

Alba Regia Műszaki Kar

Geoinformatikai/Mérnöki/
Természettudományi és Szoftvertchnológiai Intézet



SZAKDOLGOZAT

**Pontfelhő automatizált osztályozása Deep Learning
alkalmazásával ArcGIS Pro környezetben.**

OE-AMK

2024

Hallgató neve: Bondár Máté

Hallgató törzskönyvi száma: T000835/FI12904/A-G



ÓBUDAI EGYETEM
ÓBUDA UNIVERSITY

Óbudai Egyetem
Alba Regia Műszaki Kar
Geoinformatikai Intézet

SZAKDOLGOZAT FELADATLAP

Hallgató neve: Bondár Máté

Szakedolgozat száma: SZD2402091215023811

Törzskönyvi száma: T000835/FI12904/A-G

Neptun kódja:

Szak: geoinformatikai szakember

Specializáció:

A dolgozat címe: Pontfelhő automatizált osztályozása Deep Learning alkalmazásával ArcGIS Pro környezetben.

A dolgozat címe angolul: Automated point cloud classification using Deep Learning in ArcGIS Pro.

A feladat részletezése:

1. Mutassa be a pontfelhő és a LIDAR tulajdonságait!
2. Mutassa be mit jelent a Deep Learning technológia!
3. Mutassa be a LIDAR-os állományok feldolgozásának módszereit ArcGIS Pro környezetben!
4. Foglalja össze eredményeit.

Intézményi konzulens neve: Pődör Andrea Dr.

A kiadott téma elévülési határideje: 2026. 12. 31.

Beadási határidő: 2024. 05. 15.

A szakdolgozat: Nem titkos.

Kiadva: Székesfehérvár, 2024. 05. 13.

P.H

Intézetigazgató

A dolgozatot beadásra alkalmasnak találom: 2024. 05. 15.

belső konzulens



külső konzulens



ÓBUDAI EGYETEM
ÓBUDA UNIVERSITY

Alba Regia Műszaki Kar
Geoinformatikai/Mérnöki/
Természettudományi és Szoftvertechnológiai Intézet

HALLGATÓI NYILATKOZAT

Alulírott hallgató kijelentem, hogy a dolgozat saját munkám eredménye, a felhasznált szakirodalmat és eszközöket azonosíthatóan közöltem. Az elkészült dolgozatban található eredményeket az Óbudai Egyetem és a feladatot kiíró intézmény saját céljára térítés nélkül felhasználhatja, a titkosításra vonatkozó esetleges megkötések mellett.

Kelt: BUDAPEST..... (hely), 2024.05.14...... (dátum)

Bouda Máté

hallgató aláírása

Tartalomjegyzék

1. BEVEZETÉS.....	6
2. PONTFELHŐK.....	6
2.1. Pontfelhő definíciója	6
2.2. 2.5D és a 3D közötti különbség	7
2.3. LIDAR.....	7
2.3.1. Time-of-flight (Tof).....	7
2.3.2. Háromszögelést használó.....	8
2.3.3. Adatrögzítés fajtái.....	8
2.4. Pontfelhő előállítása fotogrammetriai eljárással	9
2.5. Pontfelhők tárolása.....	10
2.5.1. Bináris tárolás	10
2.5.2. ASCII-alapú tárolás	12
2.6. Szoftverek	12
2.7. Hardver.....	13
2.8. A munkához felhasználható adatok	13
3. DEEP LEARNING.....	14
3.1. Mi is ez?	14
3.2. Felhasználásuk	15
3.3. Működésük.....	15
3.4. Modellek	17
3.5. Betanítás.....	18
3.6. Rendezett és rendezetlen pontfelhők feldolgozása	19
4. KEZELÉS ÉS OSZÁLYOZÁS.....	19
4.1. Pontfelhők kezelése ArcGIS Pro-ban.....	19

4.2.	Kézi osztályozás.....	20
4.2.1.	Első mód	21
4.2.2.	Második mód	23
4.3.	Harmadik mód, egyben az automatizált osztályozás	23
5.	GYAKORLATI PÉLDA	24
5.1.	Felhasznált adatok	24
5.2.	LASTools működése	25
5.3.	Pontfelhő betöltés az ArcGIS Pro-ba	26
5.4.	Pontfelhő kicsinyítés	27
5.4.1.	Thin LAS	27
5.4.2.	Extract LAS	27
5.5.	Pontfelhő osztályozás.....	28
5.6.	Objektum exportálás	29
5.7.	Tanuló modell előkészítés.....	31
5.8.	Modell tanítás.....	32
5.9.	Modell futtatás	33
5.10.	Konklúziók	34
6.	ÖSSZEFOGLALÓ	35
7.	SUMMARY	36
8.	IRODALOMJEGYZÉK	36
9.	ÁBRAJEGYZÉK	39

1. BEVEZETÉS

A mesterséges intelligencia (angol rövidítéssel AI, magyarul MI) rohamos fejlődése és térnyerése számos innovációt hoz magával az élet minden terén. Ezekben a modern időkben a gyors fejlődés lehetővé teszi számunkra az újabbnál újabb adatfeldolgozási formák használatát. Ebben persze benne van az is, hogy hamar túllépünk bizonyos megoldásokon, viszont az MI nem ilyen. Hatékonyan bevált különféle dolgok felismerésében/azonosításában, ide tartoznak az arcfelismerő rendszerek, önvezető autókban nagyon fontos összetevő, kutatásoknál különféle minták felismerésében hasznos és még sorolhatnám. Az MI mellett elengedhetetlen beszélni az eszközeink fejlődéséről is, egyrészt, ami lehetővé teszi a futtatását, vagyis a számítógép. Másrészt a különféle felvételeket készítő gépek, mint a fényképezőgépek (hagyományos vagy ipari, amiket például légifelvételek készítéséhez használnak), geodéziai mérőállomások vagy, például a LIDAR készülékek. Precízebb, élesebb felvételekkel és jobb szoftveres háttérrel magasabb minőségűvé válik a munkavégzés, ez növeli a költséghatékonyságot, ami mindenkinek kedvez. Az munkám egyik tárgya is ez, pontosabban a „mélytanulás”, amely a térinformatikában is egyre nagyobb szerepet kap. Ezt a folyamatot ebben a dolgozatban egy példaterületen szeretném bemutatni. Betanítás után pontfelhőkből szeretnénk alakzatokat kinyerni miután „begyakoroltattam” a géppel, hogy mit ismerjen fel. Az ESRI ArcGIS Pro szoftveres környezetben fogom kipróbálni a mélytanulós osztályozást és objektumfelismerést, amit egy mintaterületen/tanítóterületen betanítok kiválasztott objektumok felismerésére, majd egy teszterületen ezt kipróbálom és értékelem az eljárást.

2. PONTFELHŐK

2.1. Pontfelhő definíciója

Diszkrét pontok térbeli halmazát hívjuk pontfelhőnek, ami reprezentálhat egy térbeli objektum alakját, vagy egy földrészletet. A pontfelhő egy sokoldalúan felhasználható 3D-s adatforma, ezért számos területen alkalmazzák, mint például 3D térkép készítéséhez, magassági modellek elkészítéséhez, közművek üzemeltetéséhez, virtuális valóság alapjához (emberi alakok, vagy valóságos helyszínek virtuális megjelenítéséhez), vagy

akár városfejlesztéshez is. Ahogy a nevéből is kiderül, a pontfelhő sok (diszkrét) pont halmaza, amelyek különféle térbeli alakzatokat, tárgyakat ábrázolnak, például épületeket, fákat, domborzatot. Descartes-féle koordináta-rendszerben ábrázoljuk ezeket, minden egyes ponthoz tartozik X, Y, és Z tengelyen értelmezhető koordináta. A pontfelhő jellemzően egy átmeneti vagy adatcsere-formátum, mivel ebből többféle levezetett terméket készíthetünk: DDM-et, poligonokat, felületeket, BIM elemzést és még sok más.

2.2. 2.5D és a 3D közötti különbség

El kell különíteni egymástól a 2.5D és 3D fogalmát. Amikor vektoros vagy raszteres adatokat használunk, azokat 2D-ben jelenítjük meg és X, Y koordinátákat használunk. Egy hagyományos digitális magassági modell esetében, mivel minden pontjának összesen egy magassági értéke van, nem lehet 3D-ben értelmezni, csak 2.5D-ben. Jó példa erre, hogy ha egy négy szintes épület harmadik emeletéről szeretnénk megtudni milyen magas, akkor erre nincs lehetőségünk, mivel csak egy magassági értékünk van. A ténylegesen 3 dimenzióban tárolt pontnak több Z értéke is lehet, tehát egy falnál függőlegesen lehet akár tíz pontja is egymás alatt ugyanazon X és Y értékek alatt.

2.3. LIDAR

A pontfelhő mint adatformátum a lézerszkennerekkel jelent meg. A LIDAR (Light Detection and Ranging), lézersugarat kibocsátó radar, ami mellett egyidőben egy GPS (Global Positioning System) és egy inerciális mérőegység (IMU – Inertial Measurement Unit) . Ez a három eszköz együttesen képes megbecsülni a pontok magasságát és elhelyezkedését. Azért mondhatjuk radarnak, mert a kibocsátott jelek visszaverődését, pontosabban annak az idejét és irányát méri. A lézersugarat, mint jelet kibocsátja a felmérni kívánt tárgyra/területre, arról az visszaverődik és ezt a rendszer rögzíteni fogja egy x, y, z koordinátájú 3D-s pontként [1]. Többféle elven működnek ezek a rendszerek, de két elterjedtebb típusát célszerű kiemelni:

2.3.1. Time-of-flight (Tof)

A fényforrás (lézersugarat kibocsátó eszköz) és a fényérzékelő szenzor egymás mellett van. Itt a távolság meghatározásának elve a kibocsátott fénysugár utazási idején (kiindulásától a visszaverődéséig) és a fény sebességén alapul [2].

2.3.2. Háromszögelést használó

A fényforrás/lézersugarat kibocsátó eszköz és a fényérzékelő szenzor nem közvetlenül egymás közelében van, hanem egymástól egy meghatározott távolságra. A két lézersugár egymással egy szöget fog bezárni, aminek segítségével a távolságmérést el lehet végezni [3].

2.3.3. Adatrögzítés fajtái

A LIDAR-ok kétféleképpen rögzítenek adatokat: vagy diszkrét pontokat vesznek fel, vagy pedig a kibocsátott energia teljes egészét. A különbség az, hogy az utóbbi mód a visszatérő energia teljes egészét rögzíti, információ-gazdagabb és ezért nehezebb a feldolgozása. Az előbbi pedig ennek a töredékét veszi fel, pontosabban a hullámhossz görbén a hullámhegyeket, ahol nagyobb intenzitást észlel, oda kerül egy pont [4].

A pontokat egy táblázatban rögzítjük minden leíró adatukkal együtt, mint például a felvétel dátuma, első/második visszatérés, intenzitás stb. Ha bármilyen műveletet szeretnénk velük végezni, akkor azt érdemes vetületi rendszerben végezni, mert azok mértékegysége a méter, nem pedig a fok. Az olyan egyedi pontokat, amelyeknek z koordinátája van (tehát 3D-s), és az RGB-D színskálán értelmezhető színe van, azokat nem pixeleknek, hanem voxeleknek nevezzük [5]. Az ezekből készített rácsháló a voxelrác, amit a feltöltés után egyből több algoritmus néz át, mivel szétszedik és újra alkotják a képet [6]. Színhelyes pontfelhőt úgy lehet létrehozni, hogy vagy már alaphoz RGB-színekkel rögzítjük a pontokat, vagy ugyanarról a területről készült képet, képeket illesztünk a pontfelhő pontjaira. Utóbbinál lehet eltérés a lézeres felmérés és az elkészült képek dátumát tekintve, ez pár másodperctől pár napig, hónapig, esetleg évig is terjedhet. Veszélye ennek, hogy nem fogja ugyanazt visszaadni, amit a LIDAR felmérés és esetleg más is belekerülhet a képbe, ami nem illik oda [7].

Ezeket a felvett adatokat kétféleképpen tárolódnak: rendezett és rendezetlen adathalmazban. Amikor rendezett adatfelvételtől beszélünk, a pontok különböző adatait táblázatos formában tároljuk, tudjuk minden különálló érték helyzetét és a pontok számát. Ez előnyösebb a későbbi műveletek elvégzéséhez, mivel könnyebb velük számolni. Az előbb említett letapogató berendezésekkel általában így mérnek fel. A rendezetlen adatfelvételnél ezek a plusz információk nincsenek meg, a felvevő eszköznek nem kell

pontosan, kimérten használnia a lézersugarát. Egy előny, hogy kisebb lesz a fájl mérete, nem rögzülnek potenciálisan felesleges attribútumok, viszont minden más hátrány, mivel sokkal nehezebb feldolgozni. Több a rendezetlen állomány, mint a rendezett [8].

2.4. Pontfelhő előállítás fotogrammetriai eljárással

Ezt az adatpontokból álló halmazt légi- vagy földi fotogrammetriai eljárással is előállíthatjuk. Ez egy olcsóbb és egyszerűbb folyamat, mint a LIDAR felmérések, mert ezekhez nem kell speciális felszerelés és a szoftveres háttere is megvan.

TABLE 1. PROJECT PARAMETERS FOR A LIDAR AND A DIGITAL PHOTOGRAMMETRIC CAMERA; STATUS 2009	
LIDAR	DIGITAL PHOTOGRAMMETRY
170 scans per second (190 kHz),	GSD 25 cm
30° FOV;	
8 points/m ²	16 points/m ²
Flying height 750 meters	Flying height 4,188 m
Aircraft speed 60 m/sec	Aircraft speed 141 m/sec
Strip width 403 m	Strip width 4328 m
20% side-lap between flight lines	60% side-lap between flight lines
Effective strip width at 322 m	Effective strip width 1,731 m

2.1. ábra A LIDAR és a hagyományos fotogrammetriai felmérés összehasonlítása [9]

A fotogrammetriai eljárásra többféle szoftveres megoldás áll rendelkezésre. Egy lehetőség az Agisoft MetaShape (fizetős)[10], vagy a másik a Visual SFM (ingyenes)[11] program alkalmazása, amelyek egymást átfedő sztereokópikus képekből képesek 3D pontfelhőt készíteni. Hátránya ennek a módszernek, hogy rosszabb felbontású pontból kapható ebből. Ennek oka a kép felbontása, mert a kép pixeleiből hozzuk létre a pontfelhőt, amely kisebb felbontású, mint amilyen pontsűrűségre a LIDAR-ok alkalmasak. A pontfelhők földi koordinátarendszerbe illesztéséhez és a mérethelyességhez földi illesztőpontok mérése szükséges, emellett nagy számú, egymást nagy mértékben (kb. 80-90%-ban) átfedő fénykép kell hozzá. Számomra ilyen formában ez nem alkalmazható, de példának okáért hogyha egy épületet akarnánk 3D-ben újraépíteni, akkor érdemes legalább 8-12 (vagy több) irányból képeket készíteni róla, hogy legyen elég alapanyaga a programnak, amiből lehet egy 3D pontfelhőt készíteni [12]. Itt nincs többes visszaverődés (mint a LIDAR-nál 1., 2., stb visszaverődés), ami azt

jelenti, hogy egy homogén felszínt kapunk, amiből nem tudjuk elkülöníteni a felszínt a vegetációtól, vagy az egyéb tereptárgyaktól.

További pontfelhőelőállítási módszerek léteznek. Ilyen a Structured Light, vagyis egy bekalibrált 3D szkennelőrrel fehér vagy kék fény mintákat vetítünk egy tetszőleges objektumra. Ezeket a változó mintákat, mint képkockákat a szkennelő szoftvere automatikusan real-time kiértékeli, amiből pontfelhőt kapunk. Mérettől függően változik minden tárgy felvételi ideje, a kész 3D kép minősége pedig tőlünk függ, hogy mennyi ideig tartottuk egy pontban az eszközt. Ezt kisebb projektekhez lehet használni, itt arra gondolok, hogy nem egy várost, hanem inkább egy vázát, esetleg valamilyen alkatrészt akarunk rögzíteni, hogy később például újra tudjuk alkotni 3D-s nyomtatóval [13].

Szintén lehetséges pontfelhő előállítása szonárral, amelyet elsősorban a víz alatti alkalmaznak. Kevésbé, vagy egyáltalán nem átlátható zavaros vízben és nagyobb mélységekben van fontos szerepe, mikor a szokványos fényképezés és a lézeres eszközök hatékonysága nagy mértékben lecsökken. A modern szonár készülékek több irányba is képesek hanghullámokat kibocsátani, sőt, még a felvett adatokat át is tudják koordinátarendszerbe transzformálni. A legelterjedtebb fajtája a Frontális Szkennelő (Forward Scan), amely önmaga elé küldi ki a jeleket. 3D-s felvételeket vagy két ilyen eszközzel sztereóban, vagy egy GPS segítségével tudunk készíteni, mivel alaphoz 2D-s szonárképeket tudunk felvenni [14].

2.5. Pontfelhők tárolása

A pontok tárolására a két legelterjedtebb és elfogadottabb formát nézzük meg, amit a közösség is és a fejlesztők is elfogadottnak tekintenek.

2.5.1. Bináris tárolás

A pontfelhők tárolásának egyik formája a LAS-állomány (LASer szóból ered), ami alapvetően egy bináris fájlformátum. Az Amerikai Fotogrammetriai és Távérzékelési Társaság (ASPRS) specifikálta, majd vezette be 2003-ban. Előnye az ASCII-alapú fájlokkal szemben, hogy kisebb méretű és a magassági adatokat is jobban képes megjeleníteni [15]. De mivel sokszor több millió pontunk van egy-egy állományban, ezért ezeket a jobb tárolhatóság érdekében tömörítjük. Akár több gigabyte méretű állomány is

keletkezhet egy pár millió pontot tartalmazó pontfelhőből. Egy algoritmus segítségével az eredeti méret akár 13–17%-ára képes veszteségmentesen redukálni ezeket, a létrehozott állomány neve pedig LAZ lesz. Az ehhez szükséges szoftver a LASTools, ezzel lehet be- és kitömöríteni a LAS állományainkat [16]. A legtöbb geoinformatikai rendszer használja ezt a két formátumot, lehet programon belül kiexportálás során tömöríteni, de lehet az előbb említett eszközzel is.

Egy LAS állománynak négy része van: publikus fejléc (Public Header Block), változó hosszúságú rekordok (VLR – Variable Length Record), pontadat-rekordok (Point Data Records), kiterjesztett változó hosszúságú rekordok (Extended VLRs). A publikus fejléc általános információkat tartalmaz a fájlról, mint például a létrehozás ideje, a pontfelhő határai, pontok száma stb. A változó hosszúságú rekordok többféle adatot tartalmaznak, ezek között megtalálhatóak a metaadatok, a vetületi rendszer és a

- X
- Y
- Z
- Intensity
- Return Number
- Number of Returns (Given Pulse)
- Edge of Flight Line
- Classification
- Scan Angle Rank (-90 to +90)
- Point Source ID
- Minimum PDRF Size

2.2. ábra A 0. formátum rekordjai

hullámforma. Összesen 65 535 byte méretű „hasznos” adat terhet hordozhatnak a rekordok. Az EVLR-ek annyival többek a sima VLR-eknél, hogy nagyobb adattömeget hordoznak és közvetlenül a LAS fájl végére csatolható, ami lehetővé teszi a könnyebb vetület-hozzáadást. A pontadat-rekordok a pontfelhő pontjainak adatait tartalmazzák, ezeknek tizenegy típusa van 0-10-ig terjedő sorszámozással. Különböző formátumok vannak használatban, amik fejlődésével plusz rekordtípusok kerültek bevezetésre.

Még ezen kívül van 10 rekordtípus, például a Format 2-be bekerültek az RGB színekódok, a Format 3 visszahozta a GPS-időt és az RGB csatornák intenzitásértékeit is tartalmazta.

Jelenleg az 1.4-es verziója a legfrissebb, de van visszafelé kompatibilitás, így képesek vagyunk 1.1-es verziójú fájlt 1.4-es formátumba helyezni. 0-5 rekordot tartalmaz a régi, ezért duplikálni kell a meglévőket, amik nulla értékek lesznek, cserébe amit tartalmaz, az a megfelelő helyen megjelenik és továbbra is használható[17].

Az ESRI szintén készített egy saját LAS tömörítést, ami nem nyílt formátumú és egyelőre csak a saját szoftverükkel lehet kezelni. Ez a zLAS állomány, amit nagy hatékonyságúnak írnak le, veszteségmentes tömörítést ígér és megalkotásának fő szempontja a hatékony elérés volt. Arra a kérdésre, hogy „Mi szükség volt erre, mikor szinte mindenki a LAZ tömörítést használja?”, azt a választ kapjuk, hogy idézem: „Az adatok elmentése és mozgatása szempontjából a képesség, hogy korrump fájlokat felkutassunk nagyon fontos, amely problémákat a zLAS formátum figyelembe vesz.” (ESRI zLAS Github dokumentáció[18]).

2.5.2. ASCII-alapú tárolás

A másik formája az ASCII fájlformátumot használ, ebből kettő van, a PLY és a PCD (Point Cloud Data). A PLY (PoLYgon) formátumot elsősorban 3D hálók tárolására használják, text, azaz főleg szöveges adatok vannak benne, emiatt egyszerű kezelni és módosítani. A PCD-t a Point Cloud Library (PCL) nyílt forráskódú kiterjesztésének hozták létre, kifejezetten 3D pontfelhők tárolására. A PCL-ben különféle módszerek vannak a pontfelhők feldolgozására (kulcspont meghatározás, szegmentálás stb.) Előnye ennek, hogy szabadon felhasználható forráskódja miatt sokan használják. Emiatt kialakult egy közösség, amely foglalkozik a karbantartással és a fejlesztéssel, így nagy a népszerűsége [19].

2.6. Szoftverek

A térinformatikában használható szoftverek közül a legelterjedtebb az ArcGIS ArcMap Desktop (ArcMap 10.7), illetve az ezt felváltó újabb ArcGIS Pro termékvonallal (3.2-es jelenlegi verziója[20]). Ebben mindezen adatok feldolgozhatóak, mivel ennek fejlesztésével került nagyobb hangsúly a 3D-s adatfeldolgozásra, ami nagy előrelépés volt az addigi lehetőségekhez képest, ezért rendelkezik natív 3D-s modellel, így nem kell a pontfelhők megjelenítéséhez semmilyen kiegészítő modul. Megtalálható még benne a PDAL, amit kifejezetten a pontfelhők kezelésére hoztak létre, hasonló a GDAL-hoz, ami

raszteres és vektoros állományokkal dolgozik [21]. Összevetve más hasonló programokkal, mint pl. a QGIS vagy a MapInfo, egyértelműen az ArcGIS rendelkezik a legfejlettebb képességekkel a pontfelhők terén. Ehhez hozzá kell tenni, hogy sokkal nagyobb vállalat, több befektetővel, fejlesztővel és nagyobb felhasználói közösséggel rendelkezik, mint a versenytársai. Kiemelendő, hogy csak Windows-os rendszeren fut, így aki bármilyen Linux alapú, vagy macOS operációs rendszert futtat a számítógépén, sajnos nem tudja használni ezt a programot. Végül a MATLAB LIDAR Toolbox-át lehet még ide sorolni, mivel képes pontfelhőket nemcsak különféle algoritmusokkal manipulálni, hanem meg is képes jeleníteni azt. A pontfelhők feldolgozásához a MATLAB függvényei állnak rendelkezésre, de ez nem a pontfelhő feldolgozására szolgáló eszköz, hanem egy fejlesztésre szolgáló programozható mérnöki rendszer[22] Két open source szoftvert emelek még ki, a MeshLabot, amely pontfelhőből TIN felszínmodellt készít és a CloudComparet, amely két pontfelhő összehasonlításához kell.

2.7. Hardver

Fontos kitérni a szoftverek mellett a hardveres igényre is. Ezidáig az egyetlen gyártó, akinek mélytanulásban felhasználható videokártyája (GPU) van, az az NVIDIA. A benne található grafikus processzor és a CUDA párhuzamos számítási platform teszi lehetővé, hogy a nagy teljesítményt kifejezetten komplex algoritmusok és mesterséges intelligencia futtatására is fel tudjuk használni [23]. Az AMD és az Intel is gyárt videokártyákat, viszont ebbe az irányba még náluk zajlik a fejlesztés. Ehhez az AMD áll közelebb és a közeljövőben ők is befognak szállni ebbe a versenybe. Persze ez egy komponens, amiből legalább 4 gigabyte szükséges, emellé még kell elegendő memória, legalább 32 gigabyte (RAM), egy jó többmagos (4–10 db) processzor (CPU), hogy minél simábban és gyorsabban dolgozzon a szoftver.

2.8. A munkához felhasználható adatok

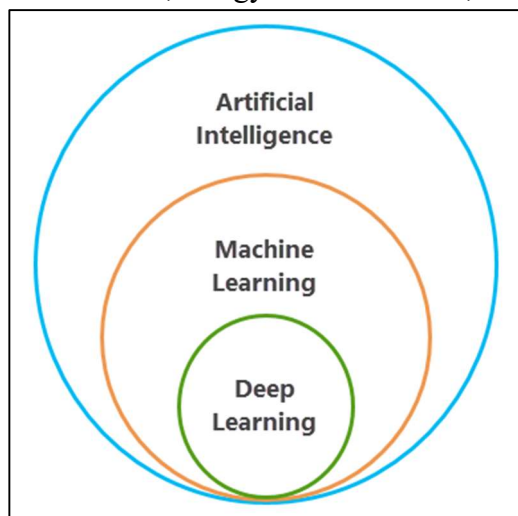
A dolgozat elkészítéséhez megfelelő tanítódát, illetve a betanított modellek kipróbálására tesztadatok beszerzésére volt szükség. A munkához szabadon, ingyenesen elérhető adatokat igyekeztem beszerezni. Számos olyan weboldal elérhető, ahonnan lehetőség van többféle célból készített pontfelhők letöltésére. Jól használhatónak tűntek

az USGS (United States Geological Survey)[24] nagy adattárából származó adatok. Itt műholdképektől kezdve kutatási anyagokon át a pontfelhőig sokféle téradat található. Szintén hasznos adatbázis az OpenTopography weboldala[25]. Itt leginkább csak pontfelhők érhetőek el és az USGS-en fellelhető adatokon kívül sok más forrásból tartalmaz adatokat. További forrás az Archaeology of Slovenia weboldala[26], ahol Szlovénia régészeti fontos helyeiről felmért adatok található. Emellett a világ sok helyére kiterjedő DDM (Digitális Domborzatmodell) és LIDAR adatokhoz kapcsolódó linkek vannak listázva, amelyek nagy része még elérhető. További lehetőség az Észti Földhivatal (Maaamet), ahol Észtországról készült hivatalos LIDAR felvételek érhetőek el [27]. Itt különböző minőségű felvételek vannak, mivel a projekt nem egy év alatt lett kivitelezve. A legkorábbi az 2008-as, a legfrissebb az 2021-es felmérés eredménye. Összességében elmondható, hogy a meglévő adatok teljesen használhatóak, viszont nehezen lehet letölteni a kiszemelt szelvényeket és néhol a pontsűrűség nem kielégítő.

3. DEEP LEARNING

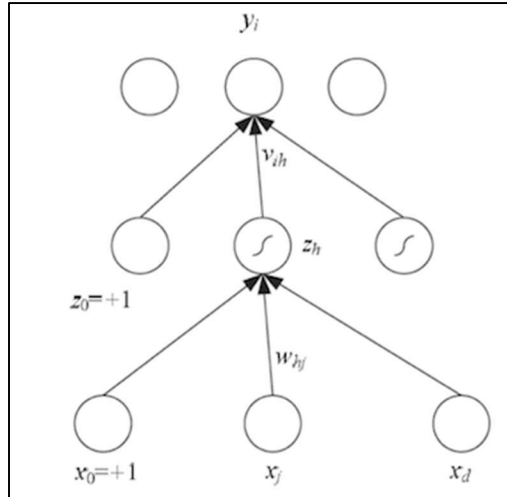
3.1. Mi is ez?

A mélytanulás (Deep Learning) olyan gépi tanulás (Machine Learning), amely mesterséges neurális hálózatokkal teszi lehetővé, hogy a digitális rendszerek tanuljanak és döntéseket hozzanak rendezett, avagy strukturálatlan, címkézetlen adatok alapján



3.1. ábra A Deep Learning helyzete [<https://pro.arcgis.com/en>]

[28]. Neurális hálózatokkal működnek, amelyek az emberi ideghálózathoz hasonlóan épülnek fel, rétegekből és azokon belül neuronokból állnak, amelyek csomópontokba tömörülnek (hidden layers). Ezek a neuronok a számolás alapjai, amelyek egy-egy hidden layerbe tömörülnek. Az információ vagy előre áramlik (bemenettől a kimenet felé) vagy hátrafele (kimenettől a bemenet felé), de réteget nem ugorhat át. [29]



3.2. ábra Egy három rétegű neurális háló felépítése

3.2. Felhasználásuk

Ezen modellek szerepe napjainkra egyre nagyobb, és a geoinformatikában is elterjedten alkalmazzák már. Nagyon sokoldalúan lehet felhasználni kutatásokhoz, egyéb munkafolyamatok automatizálásához, vagy akár a mindennapokban, amit az ember nem feltétlenül vesz észre. Elsősorban kamerákban és okostelefonokban (pl. Samsung Galaxy AI) lehet megtalálni, de az internet böngészése közben is tartalmak szűrésénél tölt be kiemelt szerepet [30]. A legkorszerűbb autókban az önvezető funkció, ami használja a mély tanulást, arcokat felismerni is hatékonyan lehet ezzel és a digitális segédek is ez alapján ismerik fel a hangunkat, amikor megszólítjuk és kérjük egy feladat elvégzésére.

3.3. Működésük

Ezeket elsősorban mintafelismerésre fogjuk használni, mivel nyers pontfelhők feldolgozásánál elsősorban betanítás útján tudok eredményt elérni mintaadatok segítségével. Itt arra kell törekedni, hogy minél több releváns adatot vigyünk be, annak

érdekében, hogy biztos egyezést kapjunk. Ez alatt azt értem, hogy ha a géptől elvárjuk, hogy mindig felismerjen valamilyen formát.

Ehhez megfelelő számú tanító adat kell, hogy rendelkezésére álljon a döntéshozatalhoz, amiből a gép egyértelműen ki tudja zárni a nem odaillő tényezőket. Ezért a magasabb számú minta kizáró erőként funkcionál, mert egyértelműsíti a döntést. Ez a módszer elsősorban az osztályozás orientált feladatoknál hasznos, amikor is megadjuk a gépnek, hogy miszerint sorolja be az adatokat [30]. Pontfelfőknél nem elég egy pontot megnézni és csak azt önmagában egy kategóriába besorolni – ha domborzat modellel dolgozok, akkor elfogadható, mert ott más szempontok szerint dolgozunk –, mivel a pontok egymástól számított helyzetét fogom keresni és azok térbeli kapcsolatát vizsgálom annak tekintetében, hogy mit rajzolnak ki (pl. háztető). Mélytanulással, avagy algoritmus alapján történő betanítás során lehet szegmentálni (egyedi formákat elkülöníteni), ezután osztályozni, ami elsőre nem hangzik sokkal könnyebbnek, mint a kézi módszer, viszont van pár előnye. Kevesebb idő ráfordításával sokkal nagyobb területet lehet feldolgozni, viszont ehhez kell egy jó modell és megfelelő tanulóterületek, ami bármilyen tetszőleges pontfelhő, de akár másmilyen raszteres réteg is lehet. A számunkra kivehető alakzatokat be tudjuk azonosítani, ezt kell a géppel is felismertetni és begyakoroltatni, hogy amikor másik bemeneti állománnyal találkozik, önmagától fel tudja ismerni a már megismert sémákat. Olyan ez, mint amikor az emberek tanulnak, bennünk is neuronok vannak, ahogy mély tanulás során is a neuronokkal dolgozó gépben is. Annyi a különbség, hogy a gép neuronjai bonyolult architektúrával/felépítéssel számolnak, míg az emberi agy biokémiai reakciók sorát hajtja végre. A kezdeti időszakban ('60-as évek) főként „igen/nem” eredmények születtek, mivel „Linear classifierst”, avagy egysíkú osztályozást alkalmaztak. ha ezzel dolgoztunk, akkor felismerte a modell, hogy például a képen egy állat (pl. egér) van, de hogy tényleg egér-e vagy patkány, azt már nem. Csak akkor tudott különbséget tenni, hogyha nagyon jó „feature extraktor-t”, azaz jó elem kinyerőt használtunk, hogy egyértelmű legyen a döntés a gépnek. Ezt fejlesztették tovább, úgy, hogy önmagától képes legyen szelektálni az oda nem illő aspektusokat (például, hogy ha egy madár repül, vagy éppen egy fa ágán pihen), és jó döntést hozni [30].

3.4. Modellek

Sokfajta modell létezik változatos felhasználásra, viszont nekünk azok a fontosak, amelyek pontokkal, 3D pontfelhőkkel és az azok közti kapcsolatokkal foglalkoznak. Guo és munkatársai egy szacikkben összeszedték a legtöbb létező modell típust, amiből csak azt a párat emeljük ki, amelyekkel mi is dolgozunk [31].

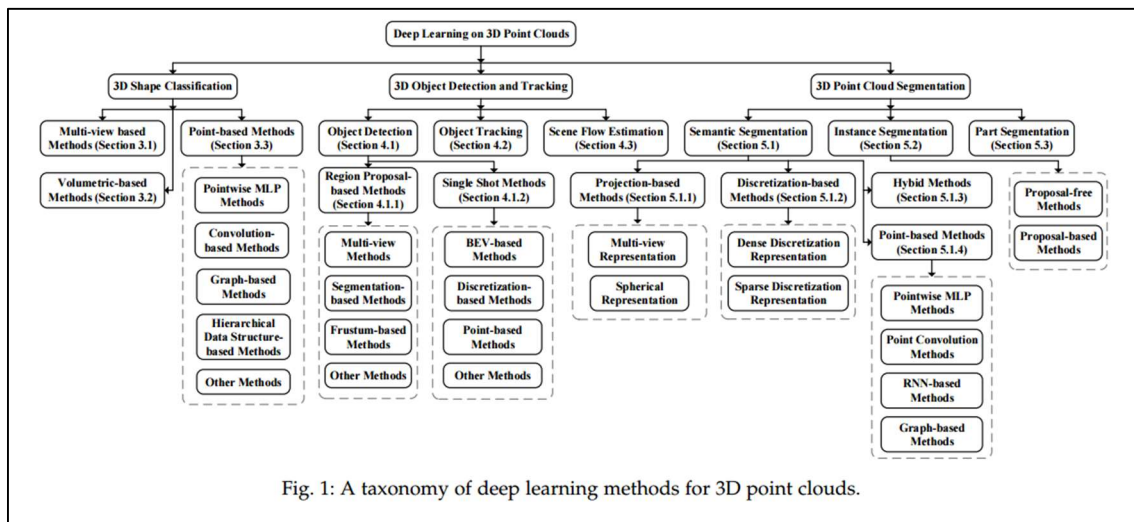


Fig. 1: A taxonomy of deep learning methods for 3D point clouds.

3.3. ábra A 3D pontfelhők mélytanulással való feldolgozásának esetei [24]

Alakosztályozó, objektumkereső és -szegmentáló modelleket különböztetnek meg, ezek közül mindegyikkel egy-egy példa erejéig fogunk foglalkozni. Az ArcGIS-ben megtalálható modellek közül három osztályozásra és egy objektumok felkutatására alkalmas. Ezek az alapértelmezettek, amelyeket egy kifejezetten erre létrehozott könyvtárcsomagból tudunk telepíteni, erről a hivatalos oldalon van egy dokumentáció, hogyan kell csinálni [32]. A pontokból alakzatok felismerése történhet többszintű észlelőket (Multi Layer Perceptrons), konvolúciós neurális hálózatot, gráf alapú keresést és adathierarchia struktúrát felhasználó modellekkel. Példának a Point CNN-t emeljük ki, amely egyszerre alkalmazza az előbb említett módok közül az első kettőt. MLP-ken keresztül transzformációt hajt végre a bemeneti adatokon, majd konvolúciót (különböző szűrőket használ eltolásos technikiával) alkalmaz azokon[31]. Objektumok felismerésénél a SECOND-t (Sparsely Embedded CONVolutional Detection) tudjuk használni. Az eddigiekhez képest (pl. VoxelNet) jobb képességekkel rendelkezik, mivel hatékonyabban ki tudja a 3D-s információkat használni. Először a z-tengelyről gyűjti be az adatokat, majd csak ezután alul-mintavételezi a 3D állományt 2D-be, ami feldolgozási sebesség-

növekedést jelent (ezért is nevezik „ritkásnak”)[33]. Szegmentálásra a RandLA-Net áll a rendelkezésünkre, ami nagy hatékonysággal képes végezni a feladatát. A véletlenszerű mintavételezéssel sok fontos adat elveszhet, habár így képes sok erőforrást megtakarítani és ezt egy „helyi elem-aggregáló” modul beiktatásával küszöböli ki[34]. Egy kevésbé ismert modell az SQN (Semantic Query Network), ami két részből áll: egy elemkiválasztóból mintafelismerés miatt (ehhez a RandLA-net-et vették alapul) és egy szemantikus (választékos) lekérdezőből, ami minél több releváns elemet gyűjt össze [35].

3.5. Betanítás

Többféle módon lehet betanítani ezeket a mély tanulási modelleket. Ezek közül az egyik a felügyelt betanítás, ami a legelterjedtebb formája a modell tanításnak. Összeválogatunk olyan adatokat, amelyekkel a betanítást szeretnénk elvégezni. Ezeket az adatokat kategorizáljuk és a kívánt kategóriát felcímkézzük, utána a nyers címkézetlen adatokat betöltjük a modell bemeneti részébe [30]. Kis részleteket kell nekünk elkészíteni, ezt angolul chips-nek, azaz darabkáknak hívják (raszteres környezetben „image chips” lesz). Amint ezek elkészültek, prezentálni tudjuk az algoritmusnak és az ki fog adni egy eredményt, ami először nem lesz tökéletes, ezért mindig meg kell vizsgálni a kidobott értéket. A modell ezután kategóriánként kiír egy-egy pontszámot, amit összehasonlítatunk a már előre megcímkézett kategória értékeivel és azt várjuk, hogy a legmagasabb pontszámú értéket az általunk felcímkézett kategória érje el. Addig kell folytatni ezt a folyamatot, amíg el nem érjük, hogy jól működjön a program az összes felhasznált adaton. Mindezek alapján a betanított modellel a nagyszámú adatra támaszkodva egyre pontosabb eredményt kapunk. Ezt egy úgynevezett „objektív funkcióval” kiszámoljuk, hogy a várt és a kimeneti érték között mekkora a kimeneti hiba. Ezt a gép a következő tanulási fázisnál fel tudja használni arra, hogy csökkentse ennek a hibának mértékét, úgy, hogy az eredmények vektorát különféle súlyokkal állítja. Ha úgy értékeli ki a modell, hogy rosszabb eredmény fog születni, akkor az ellenkező irányba módosítja a súlyt [30]. Addig finomítjuk és tanítjuk a modellt, amíg a kívánt eredmény meg fog születni, persze a legkevesebb hibával és ezentúl akármilyen bemeneti adatot kap a modell, a betanultak alapján el tudja látni a feladatát. Ezt strukturális minta felismerésnek hívjuk, ez az egyik típusa az adatok felismerésének, mivel itt mi mutatjuk meg a modellnek, hogy mit szeretnénk eredményül kapni. A másik típus a temporális

minta felismerés, ahol az algoritmusnak el kell tudnia tárolni mindazt, amit mi megtanítottunk neki és hogyha ismerős alakzattal találkozik, akkor automatikusan ismerje fel azt [28].

3.6. Rendezett és rendezetlen pontfelhők feldolgozása

A Deep Learning modelleknek két fajtája létezik, amelyek a rendezett, avagy a rendezetlen pontfelhőket dolgozzák fel. Rendezett lévén következtethetünk arra, hogy az adataink valamilyen szempontból szabályosan helyezkednek el. Ezekből egyszerűbben kivehetőek alakzatok, mivel olyan hasznos plusz információkat tartalmaznak, mint a pontok egymáshoz való kapcsolata és egyéb felépítési információk. A rendezetlen pontfelhőknél a legnagyobb probléma az, hogy nemcsak inkonzisztens a pontfelhő sűrűsége, hanem más fontos elemek kivehetősége is elveszik, például az épületek sarkai. Mindemellett sok zajt tartalmaznak, ami rontja az összképet és a feldolgozás hatékonyságát[36]. A PointNET ilyen céllal lett létrehozva, mivel kevés rendelkezésre álló adattal is képes geometriai alakzatokat kinyerni [37]. Ez a modell két másikat vesz alapul, a VoxNet-et és a 3DCNN-t, amelyek egyesítésével voxel alapon lesznek a pontok megvizsgálva. Pontosabban pontsűrűség alapján hozza létre a voxeleket, amikből megbecsüli, hogy egy-egy 3D rácspont mennyire zsúfolt. Ebből egy úgynevezett zsúfoltsági hálót (Occupancy Grid) hoz létre valószínűségszámítással. Ez tartalmazza az összes 3D információt, amelyet betöltenek a konvolúciós neurális hálóba és az osztályozza a bevitt értékeket [38].

4. KEZELÉS ÉS OSZÁLYOZÁS

LIDAR adatokat leginkább a föld felszínének feltérképezésére használjuk, majd ezekből különböző digitális magassági modelleket (felszín, terep) készítünk belőlük. Ezeket később kiértékeljük és különböző statisztikákat hozunk létre. Osztályozásnál az egyes pontok valamely attribútumát, vagy a pontfelhőben való elhelyezkedésének jellemzőit használjuk fel az egyes kategóriákba besoroláshoz.

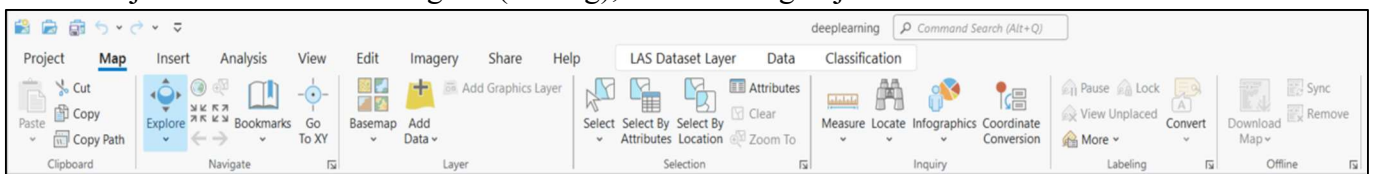
4.1. Pontfelhők kezelése ArcGIS Pro-ban

A pontfelhőket célszerű 3D-s térben megjeleníteni. ArcGIS Pro-ban dolgozva Scene-ben, azon belül is Local Scene-ben tökéletesen tudjuk kezelni a térbeli objektumunkat,

vagy pontfelhőnk. Meg kell tudnunk, hogy milyen adatokkal dolgozunk, mert pl. a DJI UAV-val (ez a legelterjedtebb drónmárka) készített felvételek magassági értékei eltérhetnek a program referencia-értékétől. Ezért érdemes az Elevation Surface-t kikapcsolni, mert alákerülhetnek a felvételeink. Fontos, hogy megfelelő vetületi rendszerben legyen a pontfelhőnk, mivel másképp nem tudja kezelni a szoftver. Érdemes például UTM vetületet használni, mert ez egy derékszögű koordináta-rendszer, amiben számolni tudunk az adatokkal.

A megfelelő fájlokat kiválasztva, azokat egy mappába rakva és az ArcGIS Pro futtatásával meg tudjuk kezdeni a műveleteket.

A kezdéshez a standard Map-ről át kell váltani Scene nézetre, amiben majd meg tudjuk jeleníteni a 3D-s elemeket. Ahogy behoztuk ezt, a következő lépés megkeresni a jobb oldalon a katalógust (catalog), amivel megtudjuk keresni az előbb elkészített



4.1. ábra Az ArcGIS Pro Ribbon menüsora

fájljainkat. Ezeket Drag-and-Drop-pal hozzáadjuk a nézetünkhöz és amint megtettük, megnyílik 3 új fül felül az eszköz sávunkon (Ribbon), a help mellett, ezek a LAS Dataset Layer, Data és a Classification. Ezekkel tudjuk manipulálni a megjelenését az adatoknak, emellett számos más lehetőséget is felkínál, mint például az adatainkkal magasabb szintű elemzések végzése, pontsűrűség beállítása, osztályozás, pontkiválasztás stb. Elsősorban ezekre fogok támaszkodni, majd a komolyabb feladatoknál más „tool”-okat is fogok használni.

4.2. Kézi osztályozás

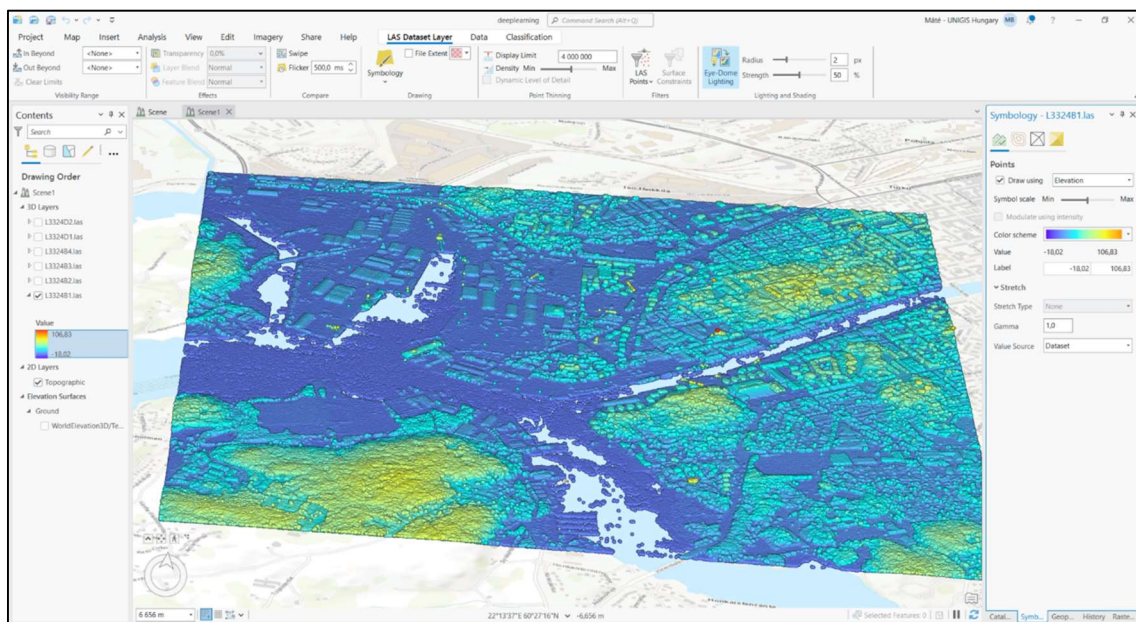
Különböző algoritmusok és eszközök lettek kidolgozva és ezek kerülnek bemutatásra. Ilyen szempontból nézve lehet kézileg, vagy manuálisan pontfelhőket feldolgozni, ami azt jelenti, hogy minden egyes folyamatot lépésről lépésre mi csinálunk. A másik oldalon pedig az automatizált megoldás található, ez a Deep Learning.

Manuális feldolgozásnál érdemes egy munkafolyamatot (workflow) kialakítani, amely magában foglalja, hogy az elejétől a végéig mit és milyen sorrendben kell

elvégezni. Ha egy nagyobb területről van LiDAR felmérésünk, akkor az egész terület egyszerre magasság alapján tudjuk osztályozni, így például a domborzati elemek jól kirajzolódnak. Ha tereptárgyak szerint kell bekategorizálni a pontfelhő részeit, akkor generalizáltan tudjuk alak alapján, például a fákat zölddel, az épületeket fehérrel, esetleg szürkével, az oszlopokat barnával jelölni, és ezzel egy jobban elkülöníthető ábrázolást kapunk. Persze be lehet minden pontot színhelyesen is jelölni, hogy visszaadja a valós világban látható kép/tájat.

4.2.1. Első mód

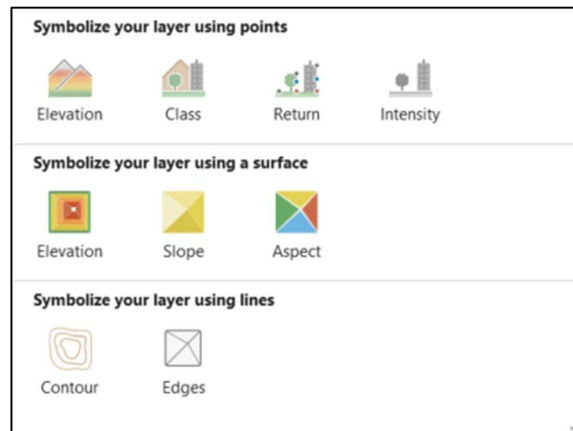
Az alapértelmezett megjelenítés pontok szerint, a magasság alapján kiértékelt osztályozást fogja megmutatni nekünk.



4.2. ábra Egy betöltött LAS állomány Turku városáról

A program három csoportot ajánl fel a pontok ábrázolására: pontszerűen, felületszerűen és vonalasan. Mindegyiknél az értékek egy lekisebbtől a legnagyobb értékig tartó skálán vannak megjelenítve, amihez színt, színeket rendelünk. Ezt az interpretáció igénye szerint módosíthatjuk.

Az első csoport négyet tartalmaz, ez a 2. módnál lesz kifejtve.



4.3. ábra Pontok ábrázolási lehetőségei

A második csoport hármat tartalmaz (felületet hoz létre): Magasság (Elevation); Lejtők generálása (Slope); Aspect. Itt egy 2D-s felületet kapunk

A harmadik csoport tartalmaz (a pontokból kirajzol): Kontúrt (Contour); Sarkokat (Edges)

Külön meg tudom még határozni, hogy a teljes felhőből minden pontot, csak az első visszatérés pontjaik (ahol van második, vagy több visszatérés, ez ott lehet fontos), csak a földfelszíni pontokat vagy ezeken kívül minden mást.

Két látványosabb elemosztály elemeinek kiexportálását emeli ki a program, a villamosvezetékekét és az épületekét, ezért erre van is ezeknek dedikált kiértékelő eszköz. A mélytanulás segítségével az általunk beprogramozott elemtípusokat is meg tudjuk keresni, de ezt majd a gyakorlati részben fejtem ki.



4.4. ábra További kiértékelési lehetőségek

Osztályokat 2D-s, 3D-s és raszteres elemekre tudunk létre hozni, amiket saját preferenciánk szerint, amihez a pontfelfőnk és valamilyen tetszőleges elem (pl. poligon vagy épület alakja, ami lehet 3D-s is) kell bemeneti adatnak. Van automatizált osztályozásra is lehetőség, szintén felajánl számunkra 5 előre beállított osztályt, de a meglevő, rosszul, vagy akár tévesen klasszifikált pontokat is bármikor át tudjuk állítani a

megfelelőre. Sajátokat is létre tudok hozni, mindezt shapefile-ok felhasználásával érem el. Digitalizálnom kell minden egyes elemosztályt, amit látok, vagy ami szükséges számomra. Másik megoldás lehet, hogy egy digitális magasság modellt készítek a pontfelhőből és az alapján rajzolok, majd az osztály létrehozásánál be tudom állítani, hogy az elkészített poligonok/vonalak/pontok körül milyen távolságban osztályozzon az algoritmus.

4.2.2. Második mód

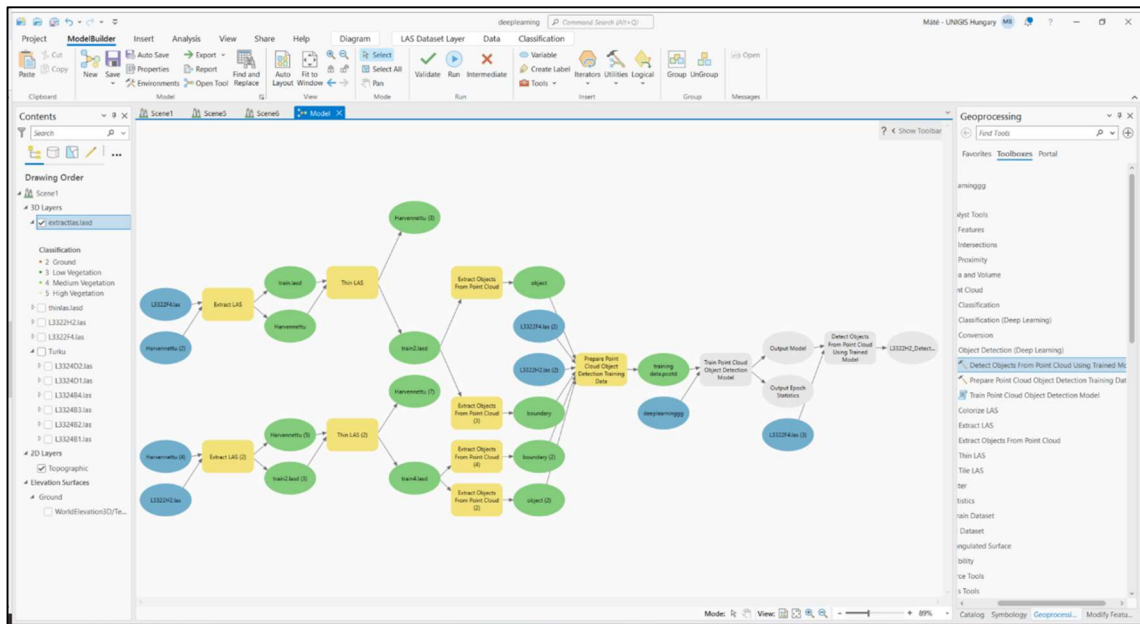
Attribútumok alapján is tudunk kategorizálni. Ez persze a rendezett pontfelhők esetében lehetséges, mivel sok egyéb tulajdonságot hordozhatnak magukkal a pontok. A program szerint felajánlott alapelehetőségek (a pontokat magukat jeleníti meg több kategória szerint): Magassági (Elevation); Osztály alapú (Class); Visszatérés szerinti (Return); Intenzitás szerinti (Intensity); Szkennelés szöge szerinti (Scan angle). Tudunk repülés iránya, felvétel szöge, GPS idő vagy RGB színkód alapján is rangsorolni. Szintén az alapértelmezett menüsávban találhatóak.

4.3. Harmadik mód, egyben az automatizált osztályozás

Vannak olyan komplexebb esetek, amikor az előbbi két mód kevésnek bizonyul, ilyenkor jön be a Deep Learning a képbe. Amikor nem elég nagy általánosságban kategorizálni a területünkön lévő pontokat, akkor lehetőségünk van a gépi tanulást alkalmazni. Például, ha a háztetőket szeretnénk kiértékelni, arra ez előbbi fejezetben bemutatott betanítási módszert alkalmazzuk annak érdekében, hogy minél pontosabban, minél jobb eredményt kapjunk. Itt rendelkezésünkre áll a Geoprocessing fülben a 3D Analyst Tools toolbox-ban a Pontfelhő (Point Cloud) eszközök alatt kettő olyan csomag, amiben kifejezetten mély tanulós módszerrel lehet osztályozni. Az egyik az osztályozással foglalkozik (Classification), ebben egy betanító, egy előkészítő, egy kiértékelő és egy osztályozó eszköz található. A másik az Objektumfelismeréshez (Object Detection) tartalmaz egy előkészítő, egy betanító és egy felismerő tool-t.

5. GYAKORLATI PÉLDA

Ebben a fejezetben be fogom mutatni a menetét egy pontfelhő feldolgozásának, ehhez össze kell gyűjtenem a szükséges eszközöket és alapanyagokat. Kell pár LAS/LAZ



5.1. ábra Egy példa a munkafolyamatra. A modelbuilder-ben hoztam létre.

fájl, amivel dolgozok, egy ArcGIS Pro (ehhez licenz is kell) és egy LASTools a gyorsabb kitömörítés miatt. Használhatom az ArcGIS beépített tool-ját is, viszont szerintem érdemesebb ezt használni, mert több állományt is egyszerre ki tudok csomagolni. Leginkább az objektumfelismerés lesz a fókuszban, de általános osztályozást be fogok mutatni. Az 5.1-es ábrán egy elméleti workflow látható, amit én hoztam létre, viszont a valóságban ezeket a toolokat külön be kell állítani, nem lehet csak úgy előre beállítva használni, mert minden helyzetre más paraméterek kellenek.

Elsősorban objektumfelismerést fogok végezni, mert ebben látom a nagyobb lehetőséget.

5.1. Felhasznált adatok

A világ több pontjáról kerestem felvételeket, amiket most be is mutatok. Mindegyik ingyenes volt, habár volt pár, amelyet kormányzati weboldalról töltöttem le és csak egy nevet és e-mail címet kértek a letöltéshez. Letöltöttem Zürich, Turku, Kilpilahti, Oslo, New York, Baltimore és Amsterdam városok szelvényeit, viszont ezek közül az utóbbit nem használtam.

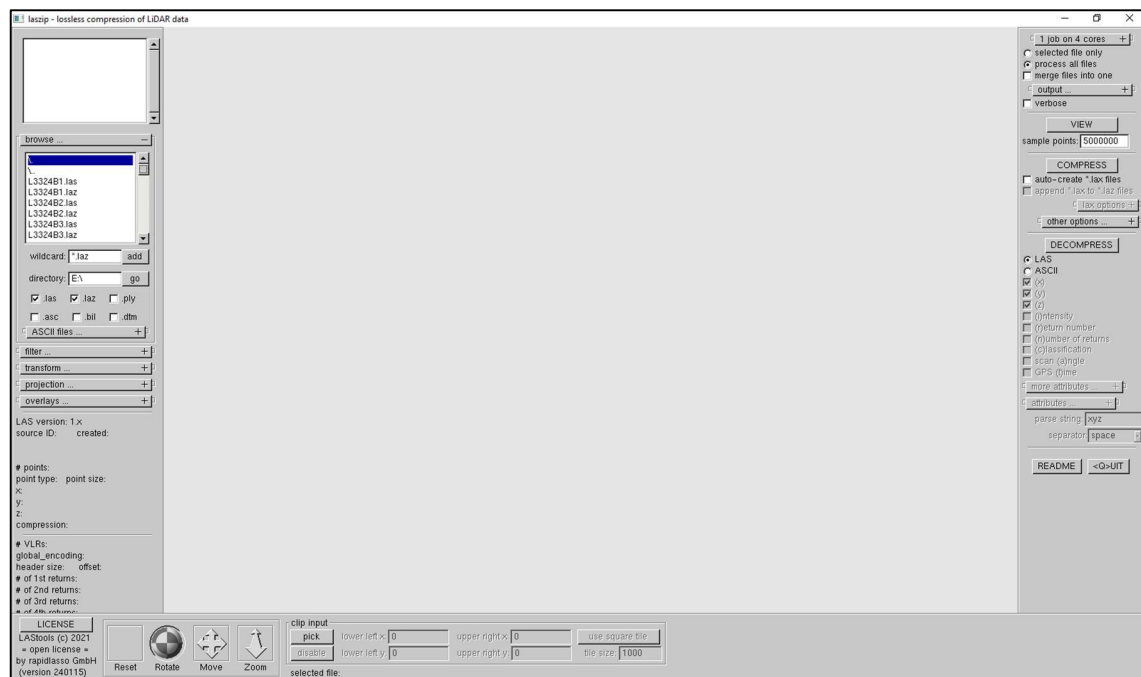
5.2. LASTools működése

Miután letöltöttük a futtatásánál figyelni kell arra, hogy rengeteg modullal rendelkezik, így a „bin” mappában lévő képpel rendelkező laszip alkalmazást futtassuk.

lasvoxel	2024. 01. 15. 19:28	Alkalmazás	1 840 KB
lasvoxel64	2024. 01. 15. 19:18	Alkalmazás	1 186 KB
laszip	2024. 01. 15. 19:28	Alkalmazás	1 332 KB
laszip64	2024. 01. 15. 19:16	Alkalmazás	932 KB
shp2las	2024. 01. 15. 19:28	Alkalmazás	1 296 KB
shp2las64	2024. 01. 15. 19:19	Alkalmazás	1 029 KB

5.2. ábra A laszip.exe

Ez egy GUI-t fog nekünk megnyitni, amin belül egy üres felületet fogunk látni, amit különböző eszközök találhatók. Ebből nekünk főleg a Browse és a Decompress rész kell, viszont a többi is hasznos lehetőségeket biztosít, mint például, a View gomb, amivel itt az üres felületen meg tudjuk jeleníteni az állományunk pontjait. Miután kiválasztottuk a



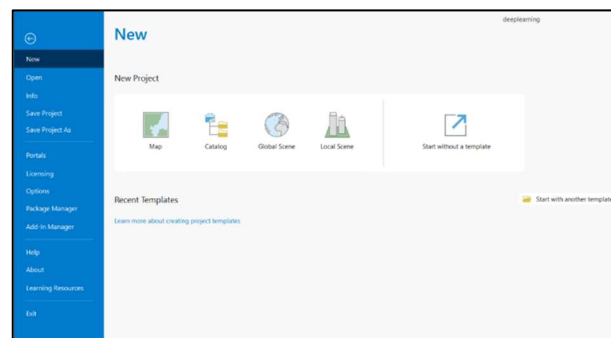
5.4. ábra A munkafelület tartalmazza az összes fontos funkciót az előkészítéshez

megfelelő .laz fájlokat és .las állományokká alakítottuk, meg tudjuk kezdeni a munkát. Megjegyzésként annyit érdemes mondani, hogy a fájlkezelőnkben nem .las, hanem HASSELL kiterjesztésben fogjuk ezeket megtalálni. Emellett szintén megjegyzés, hogy semmilyen ékezetes betű nem szerepelhet a .laz fájlok elérési útvonalában. Vagyis, ha például a „Svájc” nevű mappába mentettünk, akkor nem fogja tudni beolvasni, csak

akkor, ha „Svajc”-ra írjuk át a mappa nevét. A „wildcard” felirat mellett a kiterjesztést tudjuk módosítani, az „add” gombbal pedig az előnézethez hozzáadni. Hogyha például 12 darab szelvényünk van, akkor egyesíteni tudjuk ezeket, a könnyebb kezelhetőség kedvéért. Ezt jobb felül a „process all files” lehetőség alatt kipipálható „merge file into one” teszi lehetővé, emellé célszerű egy kimeneti mappát beállítani az „output” gombbal. A későbbiekben a szükséges kisebb területet ki fogjuk vágni, ez elsősorban teljesítményjavulást eredményez, ami nagymértékű sebességnövekedést eredményez főleg a gyengébb számítógépeknél. Az én esetemben ez fontos, mivel egy régebbi laptopom van, amin éppen elfuttatja ezt a pár programot.

5.3. Pontfelhő betöltés az ArcGIS Pro-ba

Az ArcGIS Pro-t megnyitva egy új projektfájlt kell létrehozunk, amit egy Local Scene-nel célszerű megtenni, mivel ebben egyből 3D környezet áll elő egy topografikus



5.5. ábra Az ArcGIS Pro fogadófelülete

és egy 3D világméretű felszín alaptérképpel. Ez utóbbi rendelkezik Z koordinátákkal, így amint behúztuk a pontfelhőnket, az egyből ráilleszkedik, kivéve például, ha a már említett DJI UAV-val felvett adatot szeretnénk megjeleníteni. Lesz olyan állomány, amellyel így kell dolgoznom (nem kell drónfelvételnél lennie, hogy bekerüljön alá), ehhez ki kell kapcsolni a 3D-s réteget, hogy látható legyen a pontfelhő.

A Catalog felületén meg kell keresnünk minden ide kapcsolódó állományunkat, amiket Drag and Drop-pal hozzá tudunk adni a térképünkhöz. A térkép vetülete automatikusan felveszi a hozzáadott rétegek vetületét, tehát ezzel nem kell foglalkoznunk. Ha mégis más vetületet szeretnénk meghatározni, akkor a Contents menüben a Scene, jobbklikk – Properties opciót választva a Coordinate System almenüben tudunk változtatni. Ahogy említettem már, UTM vetületet érdemes használni,

mert ebben sokkal számol az ArcGIS. Nekem a Finn adatok kapóra jöttek, mert ezek eleve ETRS 1989 ETRS-TM35 vetületben vannak definiálva. A pontfelhő koordinátarendszerét mind a LASTools-ban és az ArcGIS-ben a Define Projection tool-lal tudjuk módosítani. Amint behúztunk mindent, a program elkezd mindent kirajzolni, ez több millió pontnál időigényes művelet. Fentebb utaltam egy kisebb terület kivágására, ez megkönnyíti a feldolgozást. Egy másik hasznos dolog a LAS Dataset létrehozása, mivel ez úgy működik, mint a LASTools merge funkciója, viszont erre tudunk statisztikát és piramisokat is számolni. Utóbbi az eredeti állomány 1/3-át teszi ki és lehetővé teszi a nagyobb méretarányban történő megjelenítést, mivel egy bizonyos távolságból a pontok egy részét tartja meg. A többit elrejt, hogy ne kelljen minden egyes alkalommal újra feldolgoznia a programnak.

5.4. Pontfelhő kicsinyítés

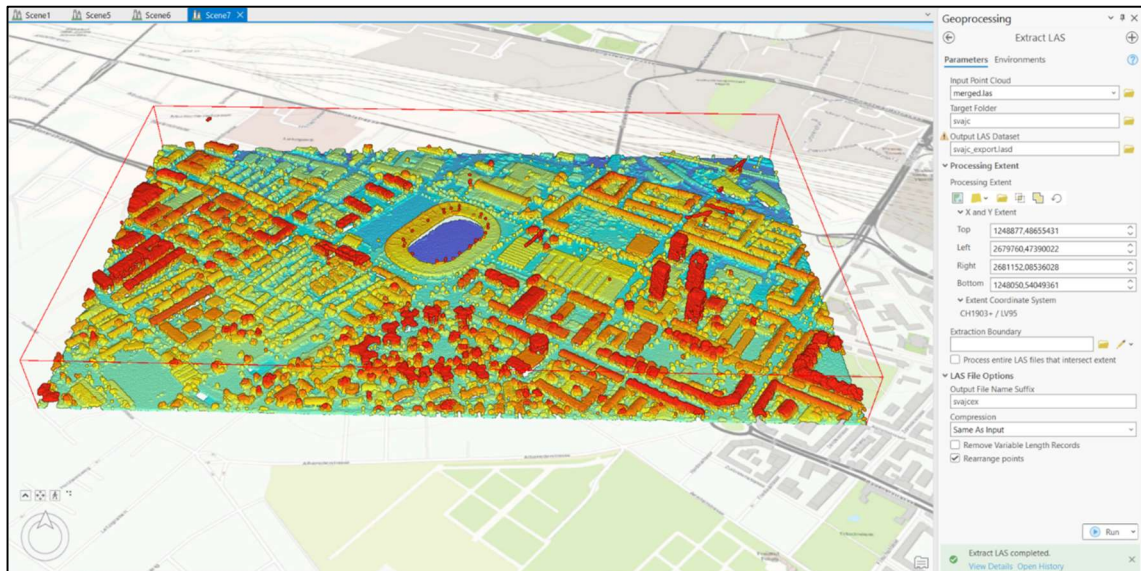
Két egyszerű tool-t fogok használni, amiket a „3D Analyst Tools” toolboxban a „Point Cloud” almenüben találunk meg.

5.4.1. Thin LAS

Ezzel az eszközzel a pontfelhő pontjait képes vagyok megritkítani, amire több lehetőség van. Ki tudom választani, hogy melyik osztályokat hagyjam meg, azt, hogy x+y, és z tengelyen hány [tetszőleges mértékegység]-et vágjak le a szelvény kiterjedéséből. Pontkiválasztási módokat is választhatok és még tömöríteni is tudok zLAS formátumba.

5.4.2. Extract LAS

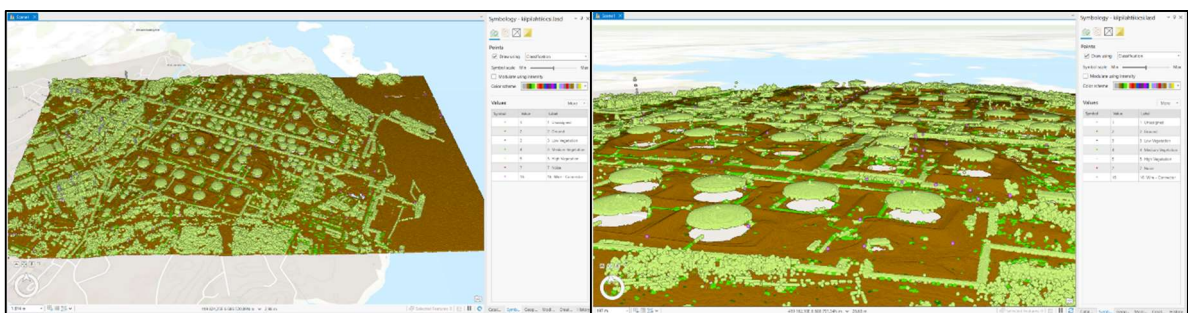
Az egyszerűsített pontfelhőt a „Processing extent” lehetőségnél kiválasztott méret alapján meg tudjuk szabni. Ehhez csak bemenetnek a megmetszendő pontfelhőnk kell és egy kimeneti célmappa.



5.6. ábra Egy Thin és Extract toollal feldolgozott pontfelhő részlet Zürichből. A pontok magasság alapján vannak megjelenítve

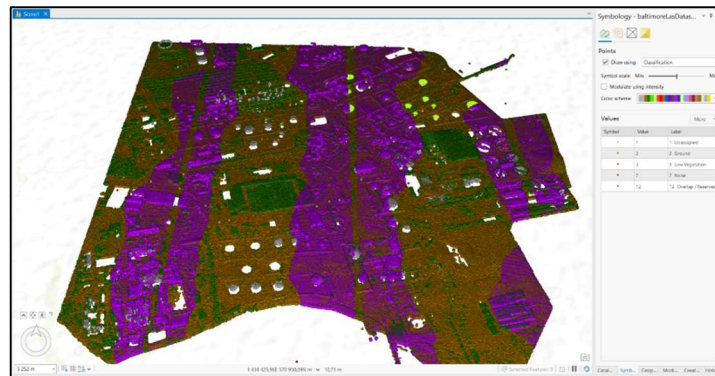
5.5. Pontfelhő osztályozás

Ahhoz, hogy ki tudjuk nyerni a számunkra megfelelő pontokat, amik potenciálisan objektumokat határolnak, elengedhetetlen az osztályozott pontfelhő. Erre a legkézenfekvőbb a mély tanulás általi betanított modellel történő klasszifikáció, de van pár egyszerűbb parancs is, amit a kézi osztályozásnál mutattam be. Ebből az „Automated classification”-ben megtalálható „Classify by height” parancssal magasság alapján el tudjuk különíteni a potenciális objektumainkat. Én több olajfeldolgozó telephelyet választottam ki erre a célra, mivel többnyire nincs zavaró vegetáció, ami elvinné a felismerő-algoritmust rossz irányba. Ezek szerintem jól kivehető objektumok, amiket jó betanításra használni, bár ki kell emelnem, hogy ezek a többnyire felületszerűen jelennek



5.7. ábra Szelvényrészlet a Kilpilahti olajfinomítóról

meg. Van példa rosszabb minőségű pontfelhőre is, például a baltimore-i szelvényem tökéletesen tükrözi, hogy nem minden tökéletes. Amikor ezt szerettem volna osztályozni, akkor a magassági osztályozó tool-ban csak Ground (2) és Model Key Point (8) osztálysorszámából enged választani, mást nem lehet beírni. Ezért jó megoldás lehet, ha a



5.8. ábra A Baltimore-i olajfinomító hibás szelvénye

Reassign Classification lehetőséget választjuk, ami bármely osztály pontjait át tudja számozni. Csak itt sajnos nem engedte magát felülírni, ezért amikor a magassági osztályozót rátettem, akkor ilyen lyukakat generált, amik nem jelennek meg. Viszont, ha például intenzitás alapján vannak megjelenítve, akkor látszódnak, mivel nem veszték el, szimplán szerintem hibát generált önmagában, amit visszavonni nem lehet.

Emiatt főleg hasonló, de mégis más területekről készült pontfelhőkkel fogok dolgozni.

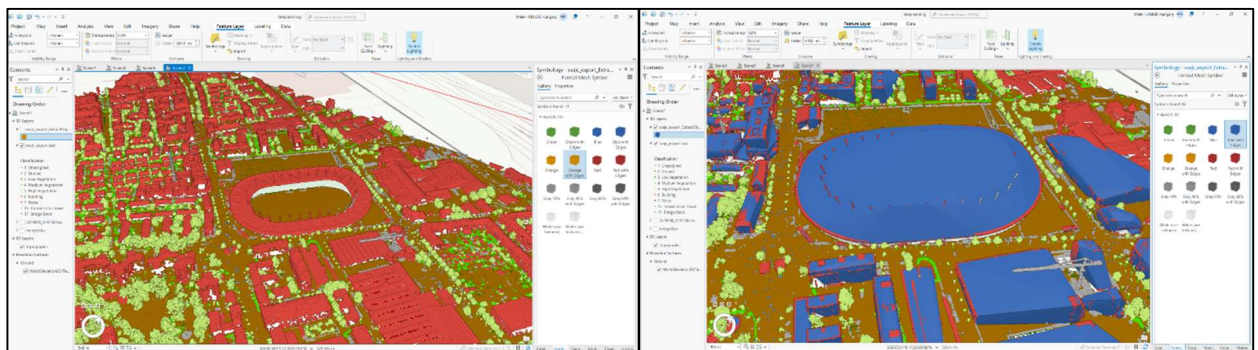
5.6. Objektum exportálás

Itt több mindent elő kell készíteni, ezért ez egy munkaigényes lépés. Az eszköz nem bonyolult, csak az előkészített pontfelhő, a kinyerendő osztály(ok) száma szükséges. Aztán hogy mit szeretnénk kirajzoltatni a programmal (objektum középpontja, egy konvex/konkáv 2D vagy 3D alakzat), a klaszterezés (csoportosítás) távolságát és a minimum pontok számát nekünk kell kikísérletezni. Leginkább az utóbbi kettőt, mivel fontos lemérni a beazonosítandó objektumokat, amiket majd egy (leginkább) 3D-s,



5.9. ábra Az általam készített piros multipatch alakzatok és alattuk szürke színű körök (balra), mellette (jobbra) az eredeti osztályozott pontfelhő a Turku keresztmetszelyéről.

úgynevezett „Multipatch” fájlal kell felvennünk. A pontok száma egy objektumon belül azért fontos, hogy kisebb, oda nem illő darabkákat ne rajzoljon be az eszköz. Itt érdemes a csoportosítást és a minimum pontok értékeit próbálgatni, mert ha túl nagy értéket teszünk be, akkor túlzott méretű poligont kapunk, ha túl kicsi, akkor meg semmiyet. Ami furcsa, hogy nem enged méter mértékegységen kívül mást választani, mindig visszaugrik rá, mintha ez lenne az alapértelmezett. Az 5.10-es ábrán látható, hogy több helyen eléggé nagy és szabálytalan alakzatok jöttek ki, persze ezt a „Modify features” ablakban bármikor módosíthatjuk.



5.10. ábra Polygonrajzolás előtt és után osztályozott pontfelhőre egy Zürich-i stadion és környékén

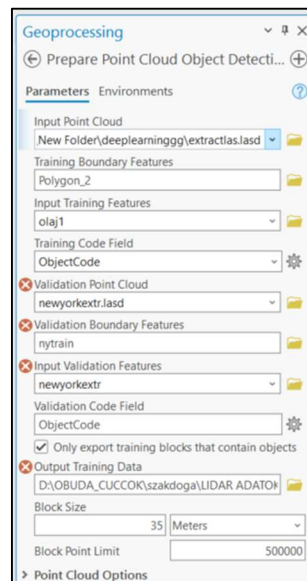
Fontos, hogy minél pontosabban visszaadja a rajzunk, azt, amit a pontfelhőn látunk, így a modell betanításakor kevesebb utómunkálattal kell végezzünk. Amit a tool rajzol ki nekünk, azt bármikor módosítani tudjuk, a program tetején található Edit fül Create Features segítségével. Még a 3D-s tárgyakat is egyszerűen szerkeszthetjük, persze igényel némi térlátást a vertexek mozgatása. Ami még furcsa lehet a két képre ránézve, hogy ezek valamilyen tárolók lennének, viszont nincs oldaluk, csak tetejük, ezért erre tekintettel kell lenni. Nagy valószínűséggel repülőgépre felerősített lézerszenkennel mérték fel a

területet, ezért leginkább a tetejüket rögzítette az eszköz. Ez igaz lesz a legtöbb általam felhasznált pontfelhőre.

Mivel az automatikus megrajzolás sokszor hibás, ezért közelebbi nézetben kell megjeleníteni. Emellett a csoportosítás (Clustering distance) és a minimum pontsűrűség értékeinek próbálgatása eredményre vezet, ha elégszer végezzük el a műveletet. Például az én esetemben 3-5 m-es csoportok és 100 db minimum pont volt a megfelelő. Az osztálykódokra is figyelni kell, mert több osztály pontjait össze is lehet vonni, ha például többől épül fel az objektum, akkor azonos csoportkódot lehet megadni. Emiatt ügyelni kell, hogy az összes betanításban résztvevő elem ugyanazokat az osztálykódokat tartalmazza. Alternatíva lehet, ha mi magunk hozunk létre nagyjából azonos objektumokat, amiket később újra tudunk méretezni, majd a megfelelő helyre bemozgatva véglegesíteni a szerkesztést. Persze nehézkes az ArcGIS Pro-ban 3D alakzatokat szerkeszteni, tapasztalatam alapján az objektum extraktálás gyorsabbnak bizonyult ezért én ezt ajánlom.

5.7. Tanuló modell előkészítés

A „Training model” előállításához szükség van egy teszt és egy validáló adatszettre. Egy-egy pontfelhő, 3D alakzatokat tartalmazó multipatch fájl és egy 2D poligonokat tartalmazó .shp fájl kell ennek elkészítésére (ez utóbbi az alakzatok kiterjedését tartalmazza). Ebből egy .pcotd (Point Cloud Object Detection Training Model) fájlt hoz létre a szoftver, ami a Deep Learning modellhez szükséges egyik bemeneti állomány. A



5.11. ábra Az eszköz paraméterei

végén még kell egy „Block size” értéket és egy maximális ponttartalmú mezőt kitöltenünk, ami a megrajzolt alakzataink méretére vonatkozik, ez az én esetemben 35 m volt.

5.8. Modell tanítás

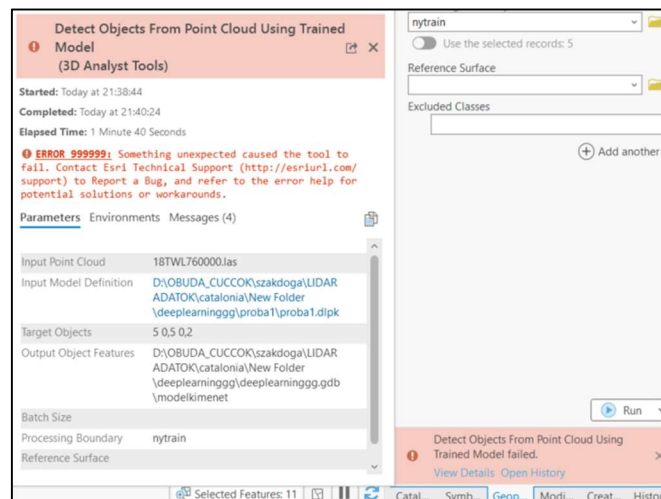
Objektum felismeréshez az előre beépített SECOND mélytanuló modellt tudom használni. Emellett meg kell adnom azt a pontfelhőt, amelyen használni szeretném és az előző alfejezetben létrehozott .pcotd fájlt. Lejjebb tudok még a meglevő osztálykódokon változtatni, meg tudom határozni a modell általános pontosságát, hány epochával mérjen, hány ciklusban tanítson és ki tudom kötni, hogy ha már nem fejlődik jobban a modell, akkor álljon le a tanítással. A processzort be kell állítani GPU-ra vagy CPU-ra, máskülönben nem fut le. Az erősebbet érdemes használni, a gyorsabb feldolgozás érdekében, de ez a modell általában voxelekkel dolgozik, így érdemes a GPU-t használni, mivel a grafikus meghajtónk felel meg jobban a célnak.

Én itt kétszer futtattam le ugyanazokra a bemeneti adatokra, először 50 epochával 2-es batch (tétel) mérettel, utána 100 epochával 4-es tétel mérettel. Ez nagyjából megfelelőnek bizonyult egy kísérleti objektum kereséshez.

A túltanítás elkerülése végett minden egyes epocha után érvényesítés zajlik le (itt 9-15 tételnyi). Veszélye, hogy csak és kizárólag a betanult sémákat fogja figyelni és az újabban bemutatott példákon már rosszabb eredményt ér el[29].

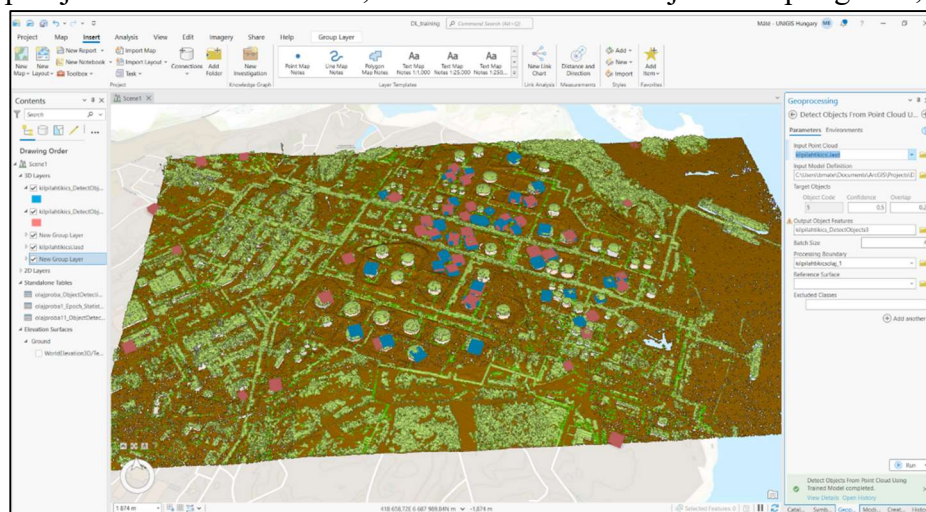
5.9. Modell futtatás

Az első próbálkozásom sikertelen volt, nem sikerült lefuttatnom a modellt, egy 999999-es kódszámú hibaüzenetet adott vissza a program, amit igyekeztem megoldni, viszont nem szűnt meg. Ekkor még minden rétegem külön vetületi rendszerben volt, pontosabban azokban, amikben elkészítették ezeket.



5.13. ábra A hibaüzenet

Elkezdtem kutatni a hiba okát és rájöttem pár dologra. Először is a tétel méretet beállítottam arra, amit a betanításkor megadtam (4), ezután a Processing boundary-t üresen hagytam, mivel ez limitálta a keresési területünket (5.14 ábra ciklámén). Itt még kipróbáltam, hogy ha a próba területünkre szintén készítek 2D határoló köröket, akkor milyen eredmény születik (5.14 ábra kék). Látható, hogy a második keresés a terület közepén jobban koncentrált, mivel oda voltak berajzolva a poligonok, viszont így



5.14. ábra Első eredmény (ciklámén) és második eredmény (kék) a Kilpilahti szelvényről

sem találta meg az összes kör feletti objektumot. Ez várható volt, mivel egy frissen betanított modellről beszélünk.

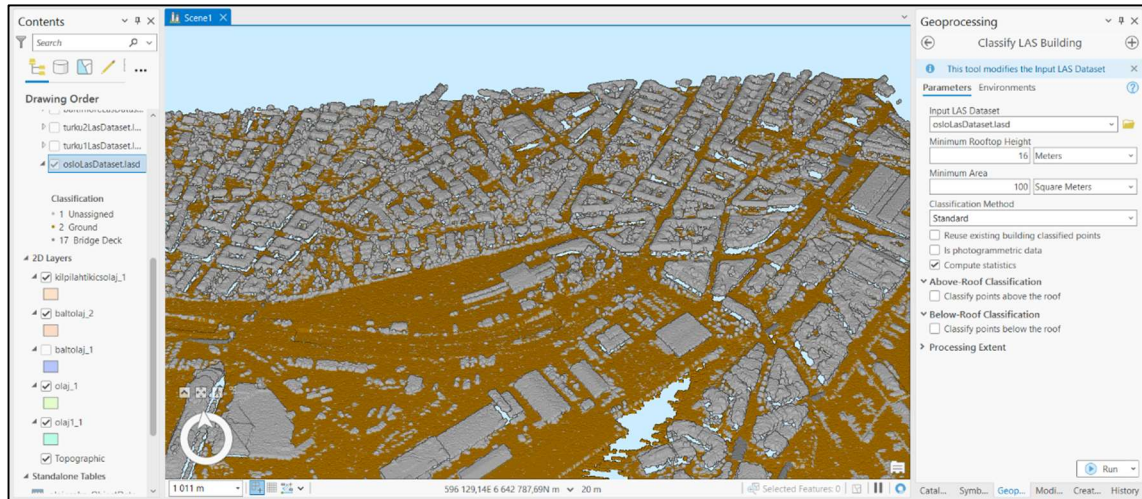
Ez után egy új projektfájlt nyitottam, amiben mindent egy Local Scene-be tettem, mivel a Global Scene-ben két darab vetület érhető el, a WGS 84 és egy kínai vetület EPSG:4490. Mivel nekem UTM vetület kellett, ezért mindent a finn vetületbe tettem és az összes pontfelhő koordináta-rendszerét ebben definiáltam. A LAS Dataset-ek létrehozásával átláthatóbb lett a TOC (Table of contents) felülete, a program teljesítménye is nőtt. Ezek után sikeresen lefuttattam az objektumkeresést, aminek az 5.10-es ábra az eredménye.

5.10. Konklúziók

Ha valaki csak úgy el szeretne kezdeni pontfelhőket feldolgozni, akkor azt javaslom, hogy először győződjön meg arról, hogy megvan a szükséges hardveres háttere. Az én esetemben ez éppen megvolt, de inkább a minimum határát súrolta, ezért ez is okozhatta a baltimore-i szelvény romlását. Hogyha nincs meg a kellő gépigény, akkor nem tudja ezt a programot futtatni, vagy hibákat fognak generálni a folyamatok (erre példa a baltimore-i szelvény). Az ArcGIS Pro-t folyamatosan fejlesztik, ennek következtében a előfordulhatnak bugok, de ez szerencsére kevészer fordul elő. Az ArcMap-hez képest egyszerre rendelkezik több és kevesebb funkcióval, például a natív 3Dkezelés ilyen funkció, a Defense Mapping egyes eszközei pedig hiányoznak belőle. A mélytanulás irodalomkutatása során nagyszámú modellel találkoztam, és ennek egy töredéke áll a rendelkezésünkre. Remélhetőleg már bővítés alatt van, hogy a későbbiekben több lehetőség legyen egy munkafolyamat elvégzéséhez.

A pontfelhő osztályozás terén nehezen tudok pozitív dolgokat mondani, mivel sok nehézségbe ütköztem. Példának az 5.15-es ábrát hozom fel, amin látszik, hogy van két nagy kategória, a Föld és bekategorizálatlan. Itt a jobb oldalon látható eszközzel szerettem volna épületeket klasszifikálni. Ez egy úgynevezett szabályrendszer alapján működő eszköz, aminek pár paramétert kell megadni és azok alapján keres, viszont miután lefutott a folyamat, minden ugyanolyan maradt. Erre nem sikerült megoldást találnom, lehet csak

a hardveres korlátok tartják vissza az eredményt. Az osztályozást visszaállító parancs több sikerrel működött, de ezt itt nem használtam.



5.15. ábra Részlet Oslo város egyik szelvényéről, a pontok klasszok alapján vannak megjelenítve

6. ÖSSZEFOGLALÓ

Ebben a dolgozatban a mélytanulás alkalmazása a téma, amit pontfelhők feldolgozásával mutatok be. Az ESRI ArcGIS Pro szoftverében megtalálható eszközök és modellek segítségével dolgozom fel ezeket az állományokat. A második fejezetben a LIDAR és LAS állományokról írok, amiben kifejtem a felhasználási módjaikat, a működésüket és a felépítésüket. Említést teszek a szoftveres és hardveres háttérrel, mik a minimális feltételek a program használatához. A következő fejezetben a mélytanulást, ezen belül a neurális hálózatokat és a konvolúciót ismertetem. A működésük mellett kitérek a különböző modellekre és némelyik működését nagyvonalakban kifejtem. A negyedik fejezetben a pontfelhők feldolgozásáról írok, bemutatom az ArcGIS Pro-ban megtalálható pont megjelenítési és osztályozási formákat. Ezzel az elméleti háttérrel előkészítettem a gyakorlati részt. Finnország néhány városából töltöttem le szelvényeket, amiket egy adatbázisba gyűjtöttem és előkészítettem a feldolgozásra. Sokféle eszközt használtam rajtuk, szerkesztő tool-lal 2 -és 3D alakzatokat készítettem, amivel behatároltam a betanításhoz szükséges régiókat. Ezeket egy objektum felismeréshez használható fájlba szedtem össze, amit a betanító modulba beillesztettem és 2 fázisban betanítást végeztem. Az elérhető egy mélytanuló modellel és ezzel a betanított

állománnyal egy kiválasztott szelvényen kétszer objektum keresést hajtottam végre. A két keresés paramétereiben enyhe eltérés volt, viszont közel hasonló eredményt kaptam. A dolgozatom végén levonok több következtetést és tapasztalatot, amit a munka során tapasztaltam.

7. SUMMARY

In this paper, I use the application of deep learning, which I demonstrate by processing point clouds. I will process these datasets using the tools and models in ESRI's ArcGIS Pro software. In the second chapter, I write about LIDAR and LAS datasets, explaining their uses, how they work and how they are constructed. I mention the software and hardware background, what are the minimum requirements to use the software. In the next chapter I describe deep learning, including neural networks and convolution. In addition to their operation, I will discuss the different models and explain some of them in outline. In the fourth chapter, I write about point cloud processing, introducing the point representation and classification formats available in ArcGIS Pro. With this theoretical background, I prepare the practical part. I downloaded sections from some cities in Finland, collected them in a database and prepared them for processing. I used a variety of tools on them, created 2 and 3D shapes with an editing tool, which delimited the regions needed for the training. I compiled these into a file for object recognition, inserted it into the training module and performed training in 2 phases. With the one available deep learning model and this trained set, I performed two object searches on a selected section. There was a slight difference in the parameters of the two searches, but the results were almost similar. At the end of my thesis I draw several conclusions and lessons learned from my work.

8. IRODALOMJEGYZÉK

- [1] A. Diab, R. Kashef, és A. Shaker, „Deep Learning for LiDAR Point Cloud Classification in Remote Sensing”, *Sensors*, köt. 22, sz. 20, o. 7868, okt. 2022, doi: 10.3390/s22207868.
- [2] Vedaant Varshney, „LiDAR: The Eyes of an Autonomous Vehicle”, Medium. [Online]. Elérhető: <https://medium.com/swlh/lidar-the-eyes-of-an-autonomous-vehicle-82c6252d1101>

- [3] Z. Dong, X. Sun, C. Chen, és M. Sun, „A Fast and On-Machine Measuring System Using the Laser Displacement Sensor for the Contour Parameters of the Drill Pipe Thread”, *Sensors*, köt. 18, sz. 4, o. 1192, ápr. 2018, doi: 10.3390/s18041192.
- [4] Leah A. Wasser, „The Basics of LiDAR - Light Detection and Ranging - Remote Sensing”, NSF, NEON. [Online]. Elérhető: <https://www.neonscience.org/resources/learning-hub/tutorials/lidar-basics#discrete-vs-full-waveform-lidar>
- [5] A. Bacciaglia, A. Ceruti, és A. Liverani, „A systematic review of voxelization method in additive manufacturing”, *Mechanics & Industry*, köt. 20, sz. 6, o. 630, 2019, doi: 10.1051/meca/2019058.
- [6] A. Dai, A. X. Chang, M. Savva, M. Halber, T. Funkhouser, és M. Nießner, „ScanNet: Richly-annotated 3D Reconstructions of Indoor Scenes”, 2017, doi: 10.48550/ARXIV.1702.04405.
- [7] Nagy Gábor, „Digitális domborzatmodellek és pontfelhők alkalmazása a terep modellezésében”, Doktori (Phd) értekezés, Óbudai Egyetem, 2018. [Online]. Elérhető: <https://oda.uni-obuda.hu/handle/20.500.14044/10186>
- [8] Varga Dániel, „3D pontfelhő regisztrációs folyamatok és jellegzetességeiről vizsgálata”, Doktori (Phd) értekezés, Eötvös Loránd Tudományegyetem, 1117 Budapest, Bogdánfy u. 10a, 2022. [Online]. Elérhető: <https://vargadaniel.web.elte.hu/ertekezes.pdf>
- [9] F. Leberl és mtsai., „Point Clouds”, *photogramm eng remote sensing*, köt. 76, sz. 10, o. 1123–1134, okt. 2010, doi: 10.14358/PERS.76.10.1123.
- [10] „Agisoft Metashape”. Agisoft, 2024. [Windows, macOS, Linux]. Elérhető: <https://www.agisoft.com/index.php?id=54>
- [11] Changchang Wu, „Visual SFM”. 2024. [Windows, macOS, Linux]. Elérhető: <http://ccwu.me/vsfm/index.html>
- [12] *Structure from Motion with VisualSFM*. [Online Video]. Elérhető: <https://www.youtube.com/watch?v=5ceiOd8Yx3g>
- [13] Matthew McMillion, „How does structured-light 3D scanning work?”, Artec 3D. [Online]. Elérhető: <https://www.artec3d.com/learning-center/structured-light-3d-scanning>
- [14] S. Negahdaripour, „Application of Forward-Scan Sonar Stereo for 3-D Scene Reconstruction”, *IEEE J. Oceanic Eng.*, köt. 45, sz. 2, o. 547–562, ápr. 2020, doi: 10.1109/JOE.2018.2875574.
- [15] Michel Ott, „Towards Storing Point Clouds in PostgreSQL”, in *Master of Science in Engineering, Major Software and Systems*, HSR Hochschule für Technik Rapperswil: FHO Fachhochschule Ostschweiz, jan. 2012. [Online]. Elérhető: www.hsr.ch/mse
- [16] *Lesson 10e2: Using LASzip to Decompress Lidar*, (2020. július 28.). [Online Video]. Elérhető: <https://www.usgs.gov/media/videos/lesson-10e2-using-laszip-decompress-lidar>
- [17] American Society for Photogrammetry and Remote Sensing (ASPRS), „LAS Specification 1.4 - R15”. The American Society for Photogrammetry & Remote Sensing, 2019. július 9. [Online]. Elérhető: https://www.asprs.org/wp-content/uploads/2019/07/LAS_1_4_r15.pdf
- [18] ESRI, „Esri Optimized LAS (zLAS) FAQ”. ESRI, 2015. március 13. [Online]. Elérhető: <https://github.com/Esri/esri-zlas-io-library/tree/master/EzLasLib>

- [19] J. Landa, D. Procházka, és J. Šťastný, „Point cloud processing for smart systems”, *Acta Univ. Agric. Silvic. Mendelianae Brun.*, köt. 61, sz. 7, o. 2415–2421, dec. 2013, doi: 10.11118/actaun201361072415.
- [20] ESRI, „ArcGIS Pro”. ESRI, 2019. [Windows]. Elérhető: <https://pro.arcgis.com/en/pro-app/index-geonet-allcontent.html>
- [21] Andrew Bell, Brad Chambers, Howard Butler, „PDAL - Point Data Abstraction Library”. [Online]. Elérhető: <https://pdal.io/en/2.7-maintenance/>
- [22] The MathWorks, Inc., „Lidar Toolbox”. [Online]. Elérhető: <https://www.mathworks.com/products/lidar.html>
- [23] NVIDIA Corporation, „NVIDIA”. [Online]. Elérhető: www.nvidia.com
- [24] U.S. Geological Survey, „USGS”, USGS science for a changing world. [Online]. Elérhető: <https://www.usgs.gov/>
- [25] OpenTopography Facility, „OpenTopography”. [Online]. Elérhető: <https://opentopography.org/>
- [26] Ismeretlen, „Arheologija Slovenije/Archaeology of Slovenia”. [Online]. Elérhető: https://arheologijaslovenija.blogspot.com/p/blog-page_81.html
- [27] Republic of Estonian Land Board, „Elevation Data”, Estonian Geoportal. [Online]. Elérhető: <https://geoportaal.maaamet.ee/geoportaal.maaamet.ee/eng/spatial-data/elevation-data-p308.html>
- [28] S.-C. Wang, „Artificial Neural Network”, in *Interdisciplinary Computing in Java Programming*, Boston, MA: Springer US, 2003, o. 81–100. doi: 10.1007/978-1-4615-0377-4_5.
- [29] R. O. Duda, P. E. Hart, és D. G. Stork, *Pattern classification*, 2nd ed. New York: Wiley, 2001.
- [30] Y. LeCun, Y. Bengio, és G. Hinton, „Deep learning”, *Nature*, köt. 521, sz. 7553, o. 436–444, máj. 2015, doi: 10.1038/nature14539.
- [31] Y. Guo, H. Wang, Q. Hu, H. Liu, L. Liu, és M. Bennamoun, „Deep Learning for 3D Point Clouds: A Survey”, *IEEE Trans. Pattern Anal. Mach. Intell.*, köt. 43, sz. 12, o. 4338–4364, dec. 2021, doi: 10.1109/TPAMI.2020.3005434.
- [32] ESRI, „Install deep learning frameworks for ArcGIS”. [Online]. Elérhető: <https://pro.arcgis.com/en/pro-app/latest/help/analysis/deep-learning/install-deep-learning-frameworks.htm>
- [33] Y. Yan, Y. Mao, és B. Li, „SECOND: Sparsely Embedded Convolutional Detection”, *Sensors*, köt. 18, sz. 10, o. 3337, okt. 2018, doi: 10.3390/s18103337.
- [34] Q. Hu és mtsai., „RandLA-Net: Efficient Semantic Segmentation of Large-Scale Point Clouds”, 2019, doi: 10.48550/ARXIV.1911.11236.
- [35] Q. Hu és mtsai., „SQN: Weakly-Supervised Semantic Segmentation of Large-Scale 3D Point Clouds”, 2021, doi: 10.48550/ARXIV.2104.04891.
- [36] H. Zhou és mtsai., „Geometry and Learning Co-Supported Normal Estimation for Unstructured Point Cloud”, in *2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, Seattle, WA, USA: IEEE, jún. 2020, o. 13235–13244. doi: 10.1109/CVPR42600.2020.01325.
- [37] J. Zhou, H. Huang, B. Liu, és X. Liu, „Normal Estimation for 3D Point Clouds via Local Plane Constraint and Multi-scale Selection”, *Computer-Aided Design*, köt. 129, o. 102916, dec. 2020, doi: 10.1016/j.cad.2020.102916.

- [38] A. Garcia-Garcia, F. Gomez-Donoso, J. Garcia-Rodriguez, S. Orts-Escolano, M. Cazorla, és J. Azorin-Lopez, „PointNet: A 3D Convolutional Neural Network for real-time object class recognition”, in *2016 International Joint Conference on Neural Networks (IJCNN)*, Vancouver, BC, Canada: IEEE, júl. 2016, o. 1578–1584. doi: 10.1109/IJCNN.2016.7727386.

9. ÁBRAJEGYZÉK

2.1. ábra A LIDAR és a hagyományos fotogrammetriai felmérés összehasonlítása [9] ..	9
2.2. ábra A 0. formátum rekordjai	11
3.1. ábra A Deep Learning helyzete [https://pro.arcgis.com/en]	14
3.2. ábra Egy három rétegű neurális háló felépítése	15
3.3. ábra A 3D pontfelhők mélytanulással való feldolgozásának esetei [24]	17
4.1. ábra Az ArcGIS Pro Ribbon menüsora.....	20
4.2. ábra Egy betöltött LAS állomány Turku városáról.....	21
4.3. ábra Pontok ábrázolási lehetőségei	22
4.4. ábra További kiértékelési lehetőségek	22
5.1. ábra Egy példa a munkafolyamatra. A modelbuilder-ben hoztam létre.	24
5.2. ábra A laszip.exe	25
5.3. ábra A laszip.exe	25
5.4. ábra A munkafelület tartalmazza az összes fontos funkciót az előkészítéshez	25
5.5. ábra Az ArcGIS Pro fogadófelülete	26
5.6. ábra Egy Thin és Extract toollal feldolgozott pontfelhő részlet Zürichből. A pontok magasság alapján vannak megjelenítve	28
5.7. ábra Szelvényrészlet a Kilpilahti olajfinomítóról	28
5.8. ábra A Baltimore-i olajfinomító hibás szelvénye	29
5.9. ábra Az általam készített piros multipatch alakzatok és alattuk szürke színű körök (balra), mellette (jobbra) az eredeti osztályozott pontfelhő a Turku keresztshelvényről.	30
5.10. ábra Poligonrajzolás előtt és után osztályozott pontfelhőre egy Zürich-i stadion és környékén.....	30
5.11. ábra Az eszköz paraméterei	32
5.12. ábra Az eszköz paraméterei	32

5.13. ábra A hibaüzenet	33
5.14. ábra Első eredmény (ciklámen) és második eredmény (kék) a Kilpilahti szelvényről	33
5.15. ábra Részlet Oslo város egyik szelvényéről, a pontok klasszok alapján vannak megjelenítve.....	35