



M Ű E G Y E T E M 1 7 8 2

Budapesti Műszaki és Gazdaságtudományi Egyetem

Közlekedés és Járműmérnöki Kar

Közlekedés- és Járműirányítási Tanszék

Tanítóminta értékbecslő módszerek alkalmazási lehetőségeinek vizsgálata a járműipari környezetérzékelés területén

Pap Bence Szilárd

Konzulens: Kővári Bálint

2024. November 7.

Tartalomjegyzék

1	Bevezetés	3
1.1	Motiváció	3
1.2	Kapcsolódó irodalom	4
1.3	Hozzájárulás	6
2	Elméleti háttér	7
2.1	Mély neurális hálók	8
2.2	Megerősítéses Tanulás	10
2.3	Felügyelt Tanulás	11
2.3.1	Gépi látás	11
2.4	Objektum detektálás	13
2.4.1	Metrikák	15
3	Fejlesztett módszer	18
3.1	Label Change Error alkalmazása objektum detektálásra	18
3.1.1	Label Change Error számítása	19
3.1.2	Localization Change érték számítása	20
3.1.3	Class Change érték számítása	22
3.1.4	Label Change Error jelentésének szemléltetése	22
3.1.5	Tanítóminták mintavételezési valószínűségének frissítése	24
3.1.6	Adat priorizáló módszer működése	26
3.2	Kísérletek	27
3.2.1	Benchmark adatok	27
3.2.2	Modell tanítása	28
3.2.3	Adat priorizálás hatásának vizsgálata	29
4	Eredmények	33

5 Összegzés	37
Irodalomjegyzék	38

1 Bevezetés

Manapság a biztonságos közlekedés fejlesztése során kiemelkedő szerepet kap az önvezető járművek fejlesztése, melynek elengedhetetlen része a számítógépes látás, és ezáltal a mesterséges intelligencia alkalmazása is. A gépi tanulás új módszerei az intelligens érzékelés számos területét forradalmasították az elmúlt évtizedben, ugyanis a legfrissebb kutatások alapján már egyértelművé vált a gépi látás területén, hogy a mélytanuló neurális hálózatok [17] felülmúlják a hagyományos módszerek teljesítményét. Ennek köszönhetően egyre jelentősebbé válik a járműipari környezetérzékelés területén a mélytanuló neurális hálózatokra épülő megoldások kialakítása, mind az egyszerűbb feladatokál, mint a klasszifikáció, mind a komplexebb feladatok esetén, mint például a szegmentáció vagy az objektum detektálás.

A gépi látás alapú objektum detektálást számos területen alkalmazzák a járműiparban. Elsősorban az önvezető járművek fejlesztése során, a forgalomban résztvevő járművek, gyalogosok, közlekedési jelzések felismerésére. Ezen kívül előszeretettel használják még a közlekedési hálózat rendszer szintű monitorozására is. Ilyen megoldás lehet például egy autópályás útdíjfizető rendszer kialakítása, amiben kamera kép alapján objektum detektálás segítségével felismerjük az autópályán közlekedő járműveket, és meghatározzuk azok díjkategóriáját. Ezzel a módszerrel nyomonkövethetők a forgalomban résztvevő járművek, és automatizálható a büntetések kiszabásának folyamata. Ugyanezen eljárás egy parkoló megfigyelésére is alkalmas. Ilyenkor a kamera képen alkalmazott objektum detektálás segítségével társíthatjuk parkolóhelyhez a leparkolt járműveket. A járműiparban alkalmazott rendszereknél elsődleges szempont a biztonságos és megbízható működés, amelyhez az alkalmazott neurális modellekre épülő algoritmusok robusztus és lehető legpontosabb működése szükséges.

1.1 Motiváció

Kulcskérdés azonban az alkalmazott intelligens rendszerek teljesítménye, pontossága és megbízhatósága. Elengedhetetlen szempont ezeknél a megoldásoknál, hogy a kialakított módszerek robusztus működéssel rendelkezzenek, és a lehető legkevesebb hibával üzemeljenek a gyakorlatban. Ezen rendszerek teljesítményét elsősorban az alkalmazott modellek teljesítménye határozza meg, ami számos összetevőtől függhet. Ilyen meghatározó tényező lehet többek közt a neurális hálózat architektúrája, hiperparaméterei illetve a tanításuk során felhasznált adatok minősége is. A tanító adatokban rejlő mintázatok alakítják az optimalizálási teret, így a tanító adatok minősége alapvetően határozza meg a betanított modellek működését. Emiatt gyakran a fejlesztési folyamatok szűk keresztmetszetét is

képezhetik a rendelkezésre álló adatok. A problémát a tanító adatok beszerzése és feldolgozása is tovább nehezíti, ugyanis gyakran kihívást jelenthet a felügyelt tanuláshoz tartozó új adatokat annotálása, ami egy rendkívül költséges és időigényes feladat. Ezért is elsődleges szempont a mélytanuló neurális hálózatok fejlesztése során a tanító adatok alapos szelekciója. A modellek teljesítményét nem csak a megfelelő adatok kiválasztásával, de ezen adatok hatékony használatával is lehet támogatni. Ez történhet akár dinamikus, adaptív módon, a tanulás során létrejövő újabb információk figyelembevételével. Ilyenkor alkalmazhatunk úgynevezett adat priorizálási technikákat.

Az adat priorizálási módszerek segítségével a tanítás ideje alatt dinamikusan változtathatjuk, hogy melyik adatpontokkal találkozson gyakrabban a neurális háló. Így befolyásolhatjuk a háló tanításának menetét azzal, hogy a modell szempontjából kedvezőbb tanítómintákat előnyben részesítjük, és gyakrabban mutatjuk meg a neurális hálónak a tanítás során. Ezáltal az adat priorizálási technikával előreláthatólag magasabb teljesítményű modellek taníthatóak, melyekben az adott neurális hálót felhasználó rendszer megbízhatósága és hatékonysága is növekszik.

1.2 Kapcsolódó irodalom

A gépi tanulásban a neurális hálók teljesítményének optimalizálása szempontjából kulcsfontosságú kérdés az adatok hatékony felhasználása a tanítás során. Ennek érdekében számos módszert dolgoztak ki és publikáltak az adat priorizáló eljárások területén, amelyek célja, hogy a modell tanulási folyamatát olyan adatpontok felhasználásával javítsák, amelyek a legnagyobb hatást gyakorolják a tanulási teljesítményre.

A [4] cikkben leírtak alapján az adat priorizáló algoritmusok alkalmazása kiemelt jelentőségű lehet például kiegyensúlyozatlan tanító adathalmazok esetén. Ilyenkor a tanító adathalmaz aránytalan mennyiségű mintát tartalmaz a különböző címkékhez tartozóan. Az alulreprezentált adatok túlmintavételezése javíthatja a modell teljesítményét. Azonban a túlmintavételezés statikus jellege megmarad, mivel a kiegyensúlyozatlanság kezelésére általában csak az osztályokhoz tartozó adatpontok számát veszik figyelembe.

Az adatok priorizálásához szorosan kapcsolódó hasonló terület az úgynevezett aktív tanulás. [26] Az aktív tanulás egy speciális gépi tanulási módszer, amely során a modell célzottan választja ki azokat az adatokat, amelyeket a leginkább érdemes annotálni vagy további tanulásra használni. Aktív tanulás során hatékony megoldásnak bizonyulhat az adatok bizonytalanság alapú mintavételezése, mely során a modell kiválasztja azokat a mintákat amelyeknél a legkevésbé biztos a helyes címkézésben és amelyeket nem tudja egyértelműen megkülönböztetni. Ehhez a [28] cikkben dolgoztak ki egy metrikát ami

alkalmas a tanítóminták bizonytalanságának meghatározására címkézetlen adatok esetében is. Ez nemcsak a címkézési folyamat hatékonyságát javítja, hanem egy szélesebb körben alkalmazható megoldást is kínál, amely különféle gépi tanulási feladatokhoz adaptálható. Az aktív tanulásnál alkalmazott metrikák mélyebb megértése érdekében a [2] cikk összehasonlította a különböző bizonytalanságbecslési módszerek hatékonyságát konvolúciós neurális hálók (CNN-ekkel) végzett képklasszifikációs feladatok során.

Azonban az adatok priorizálása viszonylag feltáratlan terület a felügyelt tanulás, és így a gépi látás kontextusában. Megerősítési tanulás esetében viszont léteznek különböző módszerek a tanítóminták hatékony priorizálására. Ennek köszönhetően a dolgozat során fejlesztett módszer a megerősítési tanulás területén már ismert adat priorizálási stratégiákból merít inspirációt, és alkalmazza azokat a felügyelt tanulás, azon belül is objektum detektálás esetében.

A megerősítési tanulás esetében az adatok tanítása során alkalmazott priorizálás terén kiemelkedő előrelépést jelentett a Prioritized Experience Replay (PER) algoritmus [25]. Ennek a módszernek az újdonsága abban rejlik, hogy a tapasztalatokat (experiences) a várható tanulási előrehaladás alapján választja ki az újrajátszáshoz (replay), szemben az egyenletes eloszlású mintavételezéssel. Az eredeti PER számos változatban létezik. Például a [12] cikkben egy elosztott architektúrát mutattak be, amely különválasztja a cselekvés és a tanulás folyamatait. Egy másik tanulmányban [3] bevezették a Prioritized Sequence Experience Replay-t, amely a hagyományos PER egy kiterjesztése, és teljes állapotátmenet-szekvenciák feldolgozására képes. További PER algoritmusra épülő megoldást mutat be a [15] cikk, amely az eredeti a módszert fejleszti tovább az exploration tag bevezetésével az adat priorizáló metrikába. Ezzel csökkenti a túltanulás kockázatát az eredeti PER algoritmus alkalmazásakor, és segít megtalálni az egyensúlyt a már meglévő tudás kihasználása (exploitation) és az új információk felfedezése (exploration) között.

A felügyelt tanulás területén is léteznek már hatékony adat priorizálási technikák, mint például a [1] cikkben bemutatott módszer, amely a korábban ismertetett megerősítési tanulás területén alkalmazott adat priorizáló módszereken alapszik. Az eljárás az adatpontok információs tartalma alapján történő mintavételezést alkalmazza klasszifikáció esetében. Ehhez az úgynevezett label change hibetag számításával dönti el, mely adatpontoknak a legmagasabb az információs tartalma, és ez alapján melyik tanítómintákat részesíti előnyben a neurális hálók tanítása során.

1.3 Hozzájárulás

A dolgozat adat priorizálási technikák alkalmazásával foglalkozik objektum detektálás területén. A kidolgozott módszer fő célja, hogy javítsa a tanítás során rendelkezésünkre álló adatpontok felhasználásának hatékonyságát, és ezzel növelje a modellek detektálási pontosságát. Ezzel a megközelítéssel a dolgozat a gépi tanulás egyik kiemelkedő kihívását, az adatok mintavételezési hatékonyságának javítását célozza meg.

Ehhez az adat priorizáló módszer kidolgozásához a megerősítéses tanulás területén már létező technikákból merít inspirációt, és az ott megismert koncepciókat alkalmazza a felügyelt tanulás azon belül is az objektum detektálás esetében. A dolgozat az ötlet alapját képző koncepciók bemutatásán keresztül feltárja a felügyelt és megerősítéses tanulás közötti összefüggéseket, megvilágítva azokat az analógiákat és metodológiai átfedéseket, amelyek hozzájárulhatnak a gépi tanulási módszerek egységesítéséhez és továbbfejlesztéséhez különböző alkalmazási területeken. Továbbá a dolgozat bemutatja a kidolgozott metrikát és részletesen elemzi annak működését, valamint azt, hogy hogyan ér el kiemelkedő eredményeket az objektum detektálás összetett problémakörében. A kutatási eredmények hatékonyságát két YOLO alapú architektúrán és két fundamentálisan különböző adatszetten is demonstráljuk, ezáltal alátmasztva az eredmények általánosságát és megbízhatóságát.

A módszer kidolgozása során nemcsak a tudományos hozzájárulás volt fontos szempont, hanem az is, hogy a kifejlesztett adat priorizáló technika a gyakorlatban is alkalmazható legyen. A gépi látás, azon belül is az objektum detektálás különösen fontos szerepet tölt be a járműipari környezetérzékelés területén, ami az autonóm járművek fejlesztésének alapvető eleme. Ezzel az adat priorizáló módszerrel hozzájárulhatunk olyan robusztus rendszerek kialakításához, amelyek növelik például az autonóm járművek érzékelési pontosságát, támogatva a biztonságosabb és megbízhatóbb közlekedési megoldások kialakítását.

2 Elméleti háttér

A gépi tanulás a mesterséges intelligencia egyik alapvető területe, amely lehetővé teszi, hogy számítógépes algoritmusok adatokból tanulva különböző feladatokat hajtsanak végre anélkül, hogy explicit módon programoznánk őket a megoldásukra. A gépi tanulás a tanítás módszere és az adatok felhasználása alapján három fő csoportba sorolható, amit az alábbi ábra mutat be.

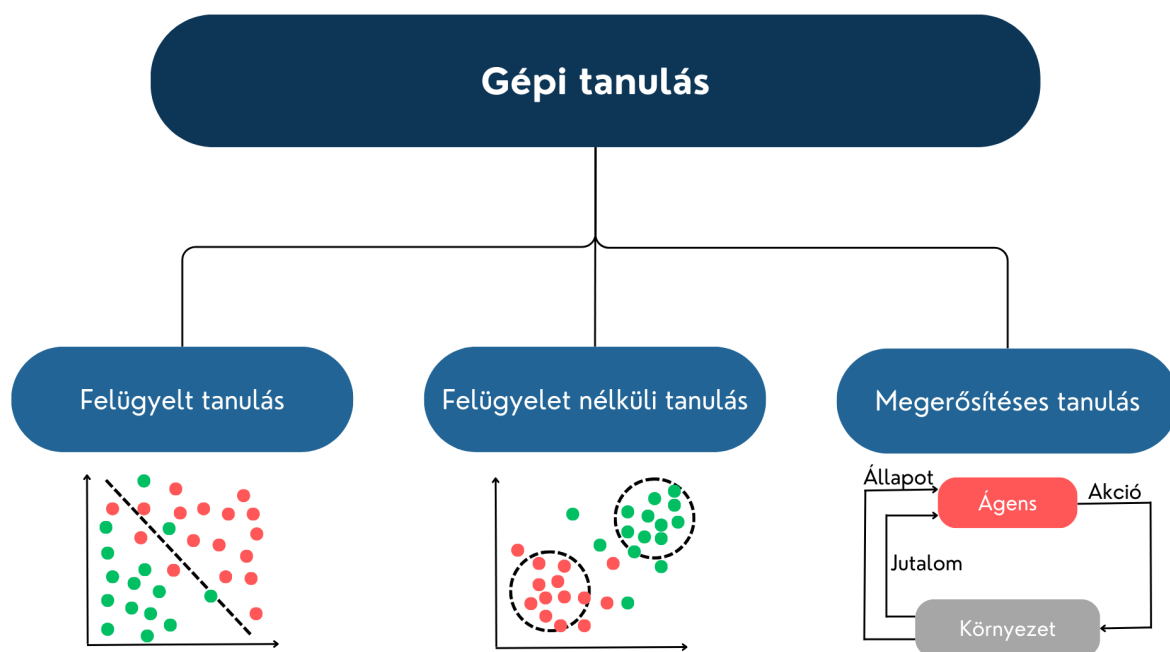


Figure 2.1: A gépi tanulás típusai

A felügyelt tanulás során az algoritmus előre meghatározott bemenet-kimenet párok segítségével tanul. Az algoritmus célja, hogy a bemeneti adatok alapján minél pontosabban meghatározza az elvárt kimenetet. Ezzel szemben a felügyelet nélküli tanulás esetében a tanító adathalmaz csak bemeneti adatokat tartalmaz, és a kimenet nem ismert. Ebben a módszerben az algoritmus feladata az, hogy a bemeneti adatok közötti rejtett struktúrákat és mintázatokat tárja fel, amelyek segítenek az adathalmaz leírásában. Végül, a megerősítési tanulás alapja az ágens és a környezete közötti kölcsönhatás. Ebben az esetben az algoritmusnak egymást követő döntéssorozatokat kell meghoznia, amelyek minőségéről a környezet visszajelzései alapján kap információt. Ezeknek a módszereknek a tanításához gyakran neurális hálókat alkalmaznak, amelyek képesek bonyolult összefüggések és mintázatok megtanulására az adatokból.

2.1 Mély neurális hálók

A mély neurális hálózatok a gépi tanulás egyik legfontosabb területét képviselik, amely az emberi agy idegsejtjeinek működését utánozza. Kimagasló teljesítményük révén számos területen találkozhatunk velük a gyakorlatban, legyen szó képfeldolgozásról, beszédfelismerésről, természetes nyelvfeldolgozásról, orvosi diagnosztikáról vagy robot és járműirányítási feladatokról. Széleskörű népszerűségüket annak köszönhetik, hogy képesek komplex problémákat megoldani, és rendkívül sokoldalúan alkalmazhatók, hiszen alkalmasak nagymennyiségű adat feldolgozására és ezen adatokban összetett mintázatok felismerésére.

A neurális hálók több rétegből épülnek fel. Az első réteg, a bemeneti réteg, ami felveszi a feldolgozandó adatokat, amelyekkel a hálózat dolgozik. A legutolsó réteg, a kimeneti réteg, amely megadja a hálózat végső válaszát vagy előrejelzését. A kettő között helyezkednek el a rejtett rétegek, amelyek feladata a minták felismerése és a magasabb szintű jellemzők kinyerése. A neurális hálókban a rétegeken belül a legkisebb egységet a neuron alkotja. A neuron olyan matematikai függvényként írható le, amely a bemeneti adatok és a hozzájuk rendelt súlyok lineáris kombinációjaként állítja elő a kimeneti értékeket. A neuronok közti súlyok határozzák meg, hogy a bemenetek mennyire befolyásolják a kimenetet, vagyis mely információk lesznek relevánsak az adott feladat szempontjából.

A neuronok által végzett számítások összetettsége révén a hálózat képes felismerni mintákat és összefüggéseket a bemeneti adatokban. Azonban a világ komplex összefüggései gyakran túl bonyolultak ahhoz, hogy egyszerű lineáris kapcsolatokkal írjuk le őket. Emiatt alkalmaznak aktivációs függvényeket, amelyek kulcsfontosságúak a kimeneti értékek meghatározásában, hiszen ezek vezetnek be a rendszerbe a nemlinearitást, lehetővé téve a hálózat számára a komplex viselkedések és mintázatok modellezését. Gyakran használt aktivációs függvények például a ReLU, a sigmoid, és a tanh, amelyek mind más módon befolyásolják a kimenetet.

A neurális hálózatok működésének másik alapvető komponense a tanulási folyamat, amely során a hálózat a bemeneti adatokhoz rendelt célértékek alapján iteratív módon finomítja a súlyparamétereit, hogy a probléma specifikus megoldására egyre jobban optimalizált súlybeállításokat érjen el. Ennek eléréséhez alkalmazzák a visszaterjesztéses algoritmust (backpropagation), amely során a kimeneti rétegben keletkező hibát rétegenként visszavezetik a hálózat korábbi rétegeibe, lehetővé téve a súlyok fokozatos korrekcióját. A hiba minimalizálása érdekében gyakran a gradiens alapú optimalizációs módszereket használják, amelyek a súlyok iteratív módosításával közelítik meg az optimális paramétereket, ezzel biztosítva a hálózat egyre pontosabb előrejelzéseit.

A neurális hálók töretlen népszerűsége és sikeres alkalmazása ellenére számos kihívást is felvet a használatuk. A hálózatok komplexitása sok esetben jelentős nehézségeket okoz, amelyek befolyásolhatják a hálózat teljesítményét és általánosító képességét. Az egyik legjelentősebb probléma a túlillesztés (overfitting), amely akkor következik be, amikor a hálózat túlzottan rátanul a tanító adathalmaz specifikus mintáira és zajra, így nem képes megfelelően általánosítani az új, ismeretlen adatokra. Ilyen esetben a modell nem a valódi mintákat tanulja meg, hanem az adathalmaz sajátos részleteit, ami jelentősen rontja a hálózat teljesítményét új helyzetekben.

A túlilleszkedés elkerülésére érdekében számos regularizációs technikát alkalmaznak. Az egyik ilyen módszer a dropout nevű eljárás, amely véletlenszerűen inaktiválja a neurális háló egyes neuronjait a tanítás során, így megakadályozva, hogy a modell túlságosan függjön bizonyos neuronoktól vagy kapcsolatoktól. Ezen kívül használják az L2 regularizációt is, amely a súlyok nagyságát korlátozza, megelőzve a túlzott paraméternövekedést, amely gyakran a túlillesztés forrása.

További kihívás az eltűnő vagy a felrobbanó gradiens problémája (vanishing and exploding gradient), amely különösen a mély hálózatoknál fordulhat elő. A problémát az okozhatja, hogy a visszaterjesztés során, ahogy a gradiensértékek visszafele terjednek a hálózat rétegein keresztül, azok egyre kisebbé válhatnak (vanishing gradient) vagy drasztikusan megnövekedhetnek (exploding gradient). Ez jelentősen lelassíthatja a hálózat tanulását, vagy teljesen megakadályozhatja a konvergenciát, különösen a rétegek növekvő száma esetén. Ennek elkerülése érdekében különböző technikákat alkalmaznak, mint például a gradiens-klipelés (gradient clipping), amely a gradiensértékeket előre meghatározott határok között tartja, így megakadályozva azok túlzott növekedését vagy túlzott csökkenését, ami a gradiens kinullázódásához vezethetne.

A neurális hálózatok tanítása során egy másik fontos kérdés, hogy milyen típusú adatok állnak rendelkezésre. A legelterjedtebb módszer a felügyelt tanulás, ahol a bemenetekhez tartozó elvárt kimenetek is rendelkezésre állnak. Ebben az esetben a modell feladata, hogy megtanulja, hogy az adott bemenetekhez milyen kimenet társítható. Ezzel szemben a felügyelet nélküli tanulás során a hálózat kizárólag bemeneti adatokat kap, és az elvárt kimenet nem ismert. Ilyen esetekben az algoritmusnak az a feladata, hogy valamilyen belső struktúrát fedezzen fel az adatokban, és ezt képes legyen kompakt módon leírni. Végül a harmadik tanulási forma a megerősítéses tanulás, amely jelentősen eltér az előző két módszertől. Ebben az esetben az algoritmusnak döntési sorozatokat kell meghoznia, de nem kap közvetlen visszajelzést minden döntés után. Ehelyett csak egy későbbi értékelést kap a döntések összességéről.

2.2 Megerősítéssel Tanulás

A megerősítéssel tanulás egy olyan gépi tanulási paradigma, amelyben az ágens tanulási folyamata nem igényel előre címkézett adatokat, mint a felügyelt tanulás esetében. Ehelyett a tanulás alapját az ágens és a környezete közötti kölcsönhatás képezi. Az ágens a környezet állapotának megfigyelése után akciókat hajt végre, és visszajelzéseket kap jutalmak formájában. E visszajelzések alapján az ágens feladata olyan stratégia kialakítása, amely maximalizálja a hosszú távú kumulatív jutalmat. A megerősítéssel tanulás célja tehát az optimális döntéshozatali folyamat megtanulása a környezettel folytatott kölcsönhatások révén.

A megerősítéssel tanulás matematikai leírását egy Markov döntési folyamatként (MDP) [21] szokták megadni.

$$(S, A, R, P, \gamma)$$

- $s_t \in S$ a t időpillanathoz tartozó állapot, ahol S az állapotok halmaza.
- $a_t \in A$ az akció, amit a t időpillanathoz választanak az s_t állapotból, ahol A az akciók halmaza.
- r_{t+1} a jutalom, amely az s_t állapotból az a_t akció választása miatt történő állapotváltozás minőségét méri.
- $P(s_{t+1}|s_t, a_t)$ annak a valószínűsége, hogy az a_t akciót választjuk az s_t állapotból a t időpillanatban, amely a_t akció az s_{t+1} állapotba vezet a következő $t+1$ időpillanatban.
- γ a diszkont faktor, amivel a korábbi jutalmakat exponenciálisan súlyozzuk. A diszkont faktor meghatározza, hogy a jövőbeli jutalmak milyen mértékben befolyásolják a döntéseket. Ennek célja, hogy a korábbi akciók hatását csökkentsük az aktuális jutalomra.

Az egyik kulcsfontosságú kihívás a megerősítéssel tanulásban az exploration-exploitation tradeoff kezelése. Az ágensnek döntenie kell arról, hogy felfedezzen új, potenciálisan magasabb jutalmat hozó akciókat (exploration), vagy inkább a már ismert, bevált akciókat alkalmazza a biztosabb jutalom érdekében (exploitation). A túlzott felfedezés időigényes lehet, és alacsonyabb jutalomhoz vezethet rövid távon, míg a kizárólag bevált akciók követése megakadályozhatja az ágenst abban, hogy felfedezze az optimális stratégiát hosszú távon. Ennek az egyensúlynak a megfelelő kezelése kritikus fontosságú a sikeres tanulási folyamat érdekében, és különböző algoritmusok használhatók a tradeoff dinamikus kiegyensúlyozására.

2.3 Felügyelt Tanulás

A felügyelt tanulás egy olyan gépi tanulási paradigma, amelyben a modellt előre meghatározott bemenet és a hozzá tartozó kimenet párok segítségével tanítják. Ahhoz, hogy ezek a bemenet-kimenet párok kialakuljanak, a meglévő feldolgozatlan adatokat annotálni kell. Annotálás során az adathalmaz bemeneti elemeihez manuálisan vagy automatikusan hozzárendelnek egy kimeneti értéket. A folyamat során az adatokon meg kell adni azokat a részeket, amiket az adott feladathoz kapcsolódóan a modellnek meg kell tanulnia. Ennek megfelelően a tanító adathalmaz tartalmazza az egyes bemenetekhez tartozó helyes kimeneti értékeket, amelyek alapján a modell megtanulja, hogy a bemenetekből hogyan állíthatja elő a megfelelő kimeneteket. A tanítási folyamat során a modell igyekszik megtanulni a bemenetek és kimenetek közötti kapcsolatokat, és ezen kapcsolatokat kifejezni egy általánosított módon. A cél az, hogy a modell a jövőbeli, ismeretlen bemenetekre is képes legyen helyes kimeneteket jósolni.

A felügyelt tanulási algoritmusoknak két fő típusa van a klasszifikáció és a regresszió. Klasszifikáció során a modell célja, hogy egy adott bemenetet egy vagy több előre meghatározott diszkrét kategóriába sorolja, míg regresszió esetén a kimenet folytonos értékű.

A felügyelt tanulást számos területen alkalmazzák a gyakorlatban, ahol előre meghatározott kimeneti címkék állnak rendelkezésre. Például a pénzügyi szektorban hitelezési kockázatbecslésre használják, míg az egészségügyben orvosi diagnózisok előrejelzésére és betegségek osztályozására alkalmazzák. Ezen kívül a természetes nyelvfeldolgozás területén is széles körben elterjedt, például szövegek automatikus kategorizálása vagy beszédfelismerés céljából. A felügyelt tanulás egyik kiemelt alkalmazási területe a gépi látás, amelynek célja képek és videók automatikus elemzése és értelmezése.

2.3.1 Gépi látás

A gépi látás területének alapvető célja magas szintű információk kinyerése a képekből vagy videókból. Ez történhet hagyományos módszerek vagy neurális hálók segítségével. Napjainkra egyértelművé vált, hogy a hagyományos módszerek teljesítményét nagy mértékben felülmúlják a neurális hálók, főként változatos körülmények között. Erre a célra a gépi látásban jellemzően konvolúciós neurális hálókat [16] használnak, mivel ezek a hálók hatékonyan kezelik a képek nagyszámú paraméterét, és képesek elsajátítani a képek térbeli struktúráinak felismerését.

A konvolúciós neurális hálók alapvető építőeleme a konvolúciós réteg, amely a hagyományos fully connected rétegekkel szemben csak lokális kapcsolatokat használ, azaz a

bemeneti kép kisebb részleteit dolgozza fel egyszerre. Minden konvolúciós rétegben több szűrő, kernel található, amelyek mérete kisebb, mint a bemeneti kép, jellemzően egy 3×3 -as mátrix. A konvolúció során ezek a szűrők végighaladnak a bemeneti képen, és minden pozícióban kiszámítják a helyi jellemzők szorzatát, amivel egy aktivációs térképet alakítanak ki. A konvolúciós művelet eredményeként létrejövő aktivációs térkép egy új reprezentációja a bemeneti képnek, amely a kernel által azonosított jellemzőket tartalmazza. Ezek a kép lokális jellemzői, mint például élek, textúrák vagy egyszerűbb mintázatok.

A konvolúciós neurális hálók egy másik lényeges eleme a le- és felkálázást végző pooling rétegek. A pooling rétegek célja a konvolúciós rétegek által generált aktivációs térképek térbeli méretének csökkentése. Ezek a rétegek hasonlóan működnek a konvolúciós rétegekhez, csúszóablakos műveleteket hajtanak végre a képen, amelyek során a kernel által lefedett területet egyetlen értékkel helyettesítik. Ez az érték lehet a lefedett terület maximuma vagy az elemek átlaga. A pooling rétegek alkalmazása jelentősen csökkenti a hálózat számítási komplexitását, így gyorsítva a tanítást.

A konvolúciós műveletnek köszönhetően a konvolúciós neurális hálók egyik kulcsfontosságú előnye, hogy képesek hierarchikus jellemzőket tanulni. Ugyanis a konvolúciós rétegben található egyes szűrők különböző tulajdonságokat képesek felismerni. Az alacsonyabb rétegek egyszerűbb struktúrákat, mint például élek és textúrák tanulnak meg, míg a magasabb rétegek összetettebb mintázatokot, például formákat vagy tárgyakat. Ez a hierarchikus reprezentáció lehetővé teszi a konvolúciós neurális hálók számára, hogy különösen hatékonyak legyenek a képfeldolgozási feladatokban.

A gépi látás legalapvetőbb feladata a klasszifikáció, amikor a bemeneti képhez egy címkét rendelünk, amely a képen lévő objektum kategóriáját adja meg. Ebben a folyamatban a konvolúciós neurális háló rétegei hierarchikus mintázatokot tanulnak meg, amelyek lehetővé teszik a képek különböző szintű jellemzőinek felismerését, és ennek alapján az objektum osztályának pontos meghatározását. A klasszifikációs feladatokban gyakran használt architektúra a ResNet (Residual Neural Network) [10], amely mélyebb hálózatok használatát teszi lehetővé reziduális összeköttetések révén. Ez a megközelítés segít a tanulás során elkerülni a hálózat teljesítményének romlását, amely különösen mély hálózatok esetében jelenthet kihívást.

A klasszifikációra épülő, magasabb szintű számítógépes látási feladat a szegmentálás [19], amely során nemcsak az objektum osztályát határozzuk meg, hanem annak pontos helyét és kiterjedését is a képen. Az eljárás során a kép minden pixelét egy objektumhoz rendeljük, ezzel a képet felosztva több, egymástól elkülönülő régiókra. Szegmentálási feladatoknál gyakran alkalmazott architektúrák a konvolúciós rétegekből felépülő a Fully Convolutional Networks (FCN) [27] vagy Mask-R-CNN [11].

A szegmentálás után a gépi látásban még összetettebb feladatokat is megvalósíthatunk, generatív modellek használatával. A generatív modellek nemcsak elemzik és felismerik a képi adatokat, hanem képesek új képeket is előállítani, amelyek hasonlóak a betanított adatokhoz. Ezek a modellek, mint például a Generative Adversarial Networks (GAN) [9] vagy a Variational Autoencoders (VAE) [14], a bemeneti adat statisztikai eloszlását tanulják meg, majd ezt felhasználva generálnak új képeket. Az utóbbi időben egy másik figyelemreméltó megközelítés, a diffúziós modellek is előtérbe kerültek, amelyek szintén képesek magas minőségű képek generálására. A diffúziós modellek a kép generálása során zajos adatból indulnak ki, és a zaj fokozatos eltávolításával generálnak végül éles és részletes képeket. Egy kiemelkedő példa erre a megközelítésre a Stable Diffusion [24] modell, amely képes realisztikus és részletes képeket előállítani nagy felbontásban.

Ezek a modellek különösen jól teljesítenek komplex struktúrájú képek generálásában, és egyre szélesebb körben alkalmazzák őket a képgenerálás, javítás és szintetikus adatok létrehozása területén. A generatív modellek mellett a gépi látás egyik legelterjedtebb alkalmazási területe az objektum detektálás, amely kritikus szerepet játszik számos ipari alkalmazásban is.

2.4 Objektum detektálás

Az objektum detektálás a számítógépes látás egy meghatározó területe, mely eljárás célja a képen lévő objektumok felismerése, osztályuknak és helyzetüknek meghatározása. Ebből kifolyólag az objektum detektálás egy összetett feladat, ami két részből áll klasszifikációból és lokalizációból. A módszer kivitelezésére hagyományos megoldások is rendelkezésre állnak, azonban a mélytanuló neurális hálózatok fejlődésével ezek a módszerek meghaladták a hagyományos megközelítéseket. Ezen módszereket fejlődéséről [29] ír. A legkorszerűbb objektum detektáló algoritmusokat általában két csoportra szokták osztani az egylépcsős és a kétlépcsős detektorokra.

A kétlépcsős detektorok a detektálás folyamatát két részre bontják. Először a képből összefüggő régiójavaslatokat állítanak elő, majd ezt követően ezeken a régiókon belül határozzák meg az objektumokat befoglaló téglalapokat illetve azok osztályát. Ebbe a kategóriába sorolható például az R-CNN [8] módszer. Az R-CNN működése során minden régiójavaslaton lefuttat egy konvolúciós neurális hálót, amely kimenete megadja a bounding box-okat és az objektumok osztályait. Legfőbb hátránya, hogy az összes régióon lefut a konvolúciós neurális háló, ami rengeteg számítást igényel, aminek eredményeképpen lassú lesz a hálózat futási ideje. Ennek a problémának a kiküszöbölésére hozták létre a Faster R-CNN-t [23], ami az R-CNN egy továbbfejlesztett változata. A módszer lényege, hogy

az egész képen futtattunk egy konvolúciós rétegekből álló neurális hálót, létrehozva egy aktivációs térképet, amiből a modell meghatározza a régiójavaslatokat. Ezzel a technikával csökkenthető a régiójavaslatok előállításának ideje, aminek köszönhetően jelentős csökken a háló futási ideje is.

A kétlépcsős detektor hálók és régió alapú módszerek legnagyobb hátránya a lassú futási idő. Ennek a problémának a kiküszöbölésére jelentenek megoldást az egylépcsős detektorok mint például a YOLO [22] vagy az SSD [18]. Ezek a megoldások nem használják a régiójavaslatokat, hanem közvetlenül a képből kinyert jellemzők alapján határozzák meg az objektumok osztályát és az ezen objektumokat határoló téglalapokat. [13] Ennek köszönhetően jelentősen gyorsabb működés érhető el, mint a régió alapú módszerekkel [7], és így már valós időben is alkalmazhatóak ezek a neurális hálók. Önvezető járművek fejlesztése, valamint számos gyakorlati alkalmazás során elengedhetetlen szempont az alkalmazott modellek futási ideje, többek között ezért is választottuk a YOLO architektúrát a dolgozatban.

A YOLO algoritmus a bemeneti képet cellákra osztja fel. Minden cella egy előre meghatározott B számú bounding boxot predikál, ahol a predikciók a bounding boxból és a box-hoz tartozó konfidencia értékből állnak. A konfidencia érték meghatározza, hogy a háló mennyire biztos abban, hogy a bounding box tartalmazza az objektumot, és, hogy mennyire tartja pontosnak a predikált bounding box-ot. Egy adott objektum detektálásáért az a cella lesz a felelős, amely cellába esik az objektum középpontja. A bounding box-ok predikciója az úgynevezett anchor box-ok segítségével történik. Az anchor box-ok különböző méretű és arányú referencia téglalapok, amik javítják háló detektálásának pontosságát. Az anchor box-okat az adott adathalmaznak megfelelően határozzák meg, általában klaszterezést alkalmazva. Használatuk lehetővé teszi a változatos méretű és arányú objektumok pontosabb detektálását. A háló predikciója során ezeket az anchor box-okat igyekszik minél pontosabban ráilleszteni az objektumra.

Problémaként merül fel azonban, hogy cellánként B bounding box-ot becsülve a predikció során rengeteg befoglaló téglalap keletkezik, amelyeknek nagy részére nincs szükségünk. Ennek megoldására szokták alkalmazni a predikció végső lépéseként a Nonmaximum Suppression-t (NMS) [20]. Az NMS használatával csökkenthetjük a bounding box-ok számát úgy, hogy közben megkaphatjuk az objektumra legjobban illeszkedő bounding box-okat, és kiszűrhetjük a kiválasztott box-al átlapolódó maradék befoglaló téglalapokat is. Az algoritmus során először meghatározunk egy konfidencia és egy IoU küszöbértéket. Ezután konfidencia küszöbérték alapján csökkenő sorrendbe rendezzük a box-okat, és eldobjuk azokat, amelyek konfidencia értéke kisebb mint a küszöbérték.

Ezt követően végigmegyünk az összes box-on a legnagyobb konfidencia értékű téglalaptól

kezdve, és kiszámoljuk az IoU értékét az összes adott osztályhoz tartozó bounding box-al. Ha az IoU érték nagyobb, mint az IoU küszöbérték, akkor a kisebb konfidencia értékkel rendelkező box-ot eldobjuk. Ezeket a lépéseket addig végezzük, amíg végig nem értünk az összes box-on. Az NMS-t általában a head után, utófeldolgozásként szokták alkalmazni, kiszűrve vele az átlapolódó, felesleges box-okat, ezzel kiválasztva objektumonként az adott objektumhoz legjobban illeszkedő bounding box-ot. A YOLO működési elvét jól szemlélteti a 2.2 ábra.

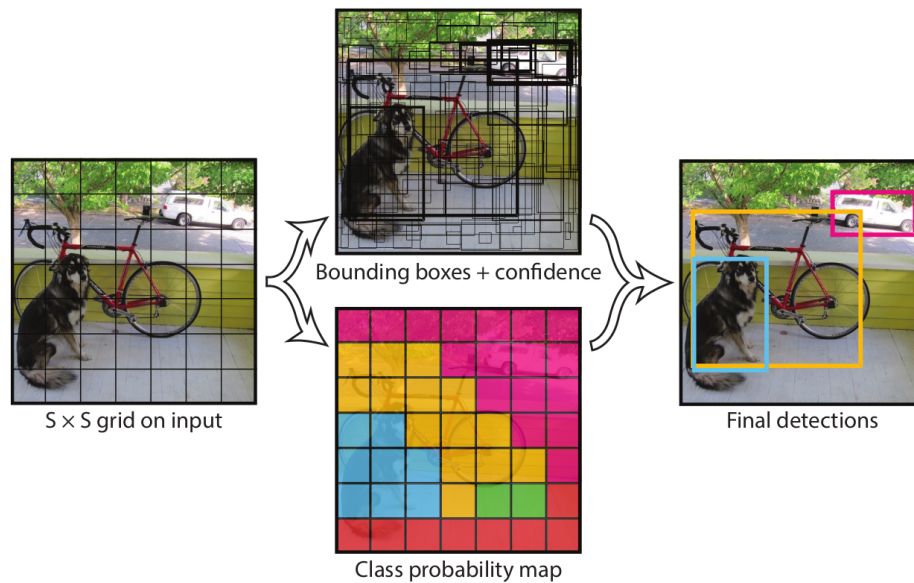


Figure 2.2: YOLO működési elvének szemléltetése [22]

2.4.1 Metrikák

Az objektum detektálást végző modellek teljesítményének mérése elengedhetetlen része a fejlesztési folyamatnak. Ezért is fontos a megfelelő metrikák alkalmazása a kiértékelés során, ami a lehető legjobban tükrözi a háló pontosságát a gyakorlatban. Ilyen bevett metrikák közé tartozik a precision, recall, F1 score, Intersection over Union (IoU) vagy a mean Average Precision (mAP) számítása. Az említett metrikák kiszámításához a háló predikcióit az alábbi kategóriákba sorolhatjuk.

- **True Positive (TP):** A háló predikciója helyesen megadta egy objektum adott helyét a képen a felcímkezett adatnak (ground truth) megfelelően.
- **False Positive (FP):** A háló tévesen predikálta az objektum helyzetét a képen, az objektum nincs ebben a pozícióban a ground truth alapján.

- **False Negative (FN):** A háló tévesen nem predikálta azt, hogy objektum van egy helyen, miközben a ground truth alapján ott van.
- **True Negative (TN):** A háló helyesen nem predikálta azt, hogy objektum van egy adott helyen, ahol valójában nincs objektum a ground truth alapján.

Ezeket az értékeket felhasználva a kiértékelés során leggyakrabban alkalmazott metrikák a következőképpen definiálhatóak:

- **Precision:** A precision értéke megadja százalékosan, hogy az összes detektált, szegmentált objektum közül mennyi volt a helyes predikció a ground truth alapján.

$$\text{Precision} = \frac{TP}{TP + FP}$$

- **Recall:** A recall értéke megadja százalékosan, hogy az összes ground truth objektum közül mennyit detektált, szegmentált helyesen a háló.

$$\text{Precision} = \frac{TP}{TP + FN}$$

- **Intersection over Union (IoU):** Meghatározza a predikált és ground truth detektálás esetén a bounding box, szegmentálás esetén a szegmentált terület metszetének és uniójának hányadosát, ezzel egy mérőszámot adva a lokalizáció jószágára. A háló predikcióinak kiértékelése során az IoU értéket felhasználva egy meghatározott IoU küszöbszint felett tekintjük az adott predikciót helyesnek.

Míg az klasszifikációs feladatokban elegendő lehet az accuracy metrika használata a modell teljesítményének kiértékeléséhez, az objektum detektálás sokkal összetettebb kihívásokat tartalmaz. Mivel az objektum detektálás két részből áll, klasszifikációból és lokalizációból, ezért nem elég csak a modell osztályozási képességét értékelni, hanem azt is figyelembe kell venni, hogy mennyire pontosan képes meghatározni az objektumok pozícióját a bounding boxok segítségével. Annak érdekében, hogy az objektum detektáló hálók teljesítményét egy kompakt mérőszámmal is jellemezni lehessen, megalkották az F1 score, average precision és a mean average precision mérőszámokat.

- **F1-score:** Az F1-score a precision és a recall harmonikus átlagaként számítható ki:

$$F1 = 2 \cdot \frac{\text{Precision} \cdot \text{Recall}}{\text{Precision} + \text{Recall}}$$

- **Average Precision (AP):** A precision-recall görbe alatti terület nagyságát adja meg.

A gyakorlatban téglalapos közelítéssel, vagy interpoláció segítségével számítják ki.

$$AP = \int_{r=0}^1 p(r) dr$$

- **Mean Average Precision (mAP):** Az osztályonként számított average precision-ok átlagaként határozható meg.

$$mAP = \frac{1}{k} \sum_i^k AP_i$$

3 Fejlesztett módszer

A fejlesztett módszer adat priorizálási technika segítségével igyekszik javítani az objektum detektáló algoritmus teljesítményét. A tanítás során felhasznált adatok nagymértékben határozzák meg a neurális hálók teljesítményét. Alapvetően a tanítás során azt feltételezzük, hogy a rendelkezésünkre álló tanítóminták információs tartalma megegyezik, és egyformán hasznosak a háló számára. Ennek értelmében a tanítás alatt felhasznált batch-eket egyenletes eloszlás alapján állítjuk össze az adatokból. Ez a feltételezés azonban nem igaz a gyakorlatban. Ennek a problémának a kiküszöbölésére nyújthat hatékony megoldást a tanítóminták információs tartalma szerinti mintavételezése tanítás közben.

A tanító adatok információs tartalmának meghatározása azonban nem egyértelmű feladat. A fejlesztett módszer során elsősorban a tanítóminták bizonytalansága alapján szeretnénk felmérni azok információs értékét. Ehhez a mintavételezés alatt a legbizonytalanabb adatpontokra összpontosítunk, ahol a modell nehezen tud egyértelmű különbségeket tenni. Ezek olyan tanítóminták, amik a döntési határ közvetlen közelében helyezkednek el, és amelyek esetében a neurális háló gyakran téves predikciót szolgáltat a tanítás során. Az ilyen adatpontok azonosítása és megfelelő kiemelése a dolgozat fő célja. Hasonló, klasszifikáció területén alkalmazott módszert mutat be a [1] cikk, mely az úgynevezett Label Change hibatag számításával dönti el, mely adatpontoknak a legmagasabb az információs tartalma.

Ezen metrika az adatok információ tartalmát méri fel. A modell tanítása közben dinamikusan kerül meghatározásra, hogy melyik tanítóminták esetében vált legtöbbször osztályt a háló predikciója. Ezáltal kiderül, hogy melyik adatoknál a legbizonytalanabb a modell, és így melyik mintákra érdemes nagyobb hangsúlyt fektetni a tanítás során. Az eljárás a kiszámolt Label Change Error alapján súlyozza a kezdetben egyenletes eloszlású tanítóminták valószínűségeit, amivel magasabb kiválasztási valószínűséget rendel azokhoz az adatokhoz, amelyekről feltételezhető, hogy egy adott időpontban nagyobb információs tartalommal rendelkezik. Ezzel a módszer célja, hogy ezekre a kritikus adatpontokra fókuszálva a modellt a bizonytalanabb minták hatékonyabb felismerésére tanítsa, és ezzel összességében növelje a neurális háló általánosító képességét és teljesítményét.

3.1 Label Change Error alkalmazása objektum detektálásra

A Label Change Error szerinti adat priorizálási módszer alkalmazása objektum detektálás esetében nem triviális. A Label Change Error kifejezetten a klasszifikáció feladatkörére van tervezve, azonban a mostani egy komplexebb feladat. Ez a problémakör túlmutat

a klasszifikáción, mert a képen lévő objektumok osztályán kívül azok helyzetét is meg kell határozni. A feladatot tovább nehezíti, hogy míg klasszifikáció esetében csak a felismerendő osztály szerepel a képen, addig objektum detektáláskor több objektum és több osztály is előfordulhat. Ezért a Label Change Error számításának módszerén különböző módosításokat kell végrehajtanunk ahhoz, hogy objektum detektálásnál lehessen alkalmazni.

Egy kézenfekvő megoldás lehet objektum detektálás esetén a Label Change Error számítására az alábbi képletet alkalmazni.

$$LC_i = \sum_{k=1}^n \max(c_{c,k}, l_{c,k})$$

A képlet egy adott tanítómintára adja meg a Label Change Error értékét, ahol n a képen lévő objektumok száma, c_c a Class Change és l_c a Localization Change értéke az egyes objektumoknak. A Class Change értéke megadja, hogy hányszor változott egy objektum osztálya a modell predikciója szerint, ezzel kifejezve a detektáló háló klasszifikációs bizonytalanságát. A Localization Change pedig azt jelzi, mennyire változott az adott objektum helyzetét leíró bounding box pozíciója a különböző predikciók során, és ezáltal a detektáló háló lokalizációs képességének bizonytalanságát méri fel.

Ez alapján a Label Change Error értéket alkalmazva mintavételezés során azokra az adatpontokra összpontosítunk, ahol az objektum osztálya vagy bounding box-ának helyzete gyakran változott a modell predikciója szerint, vagyis ahol a modell bizonytalanságot mutatott az objektumok osztályának vagy helyzetének meghatározásában. A Label Change Error alkalmazásával nemcsak az ilyen bizonytalan adatpontokat azonosíthatjuk, hanem egyszerre javíthatjuk a modell klasszifikációs és lokalizációs képességét is, mivel ezáltal a háló figyelme azon területekre irányul, ahol a legnagyobb hibák fordultak elő. Ezáltal ezeken a kritikus mintákon szereplő objektumok hatékony felismerésére tanítjuk a neurális hálót, javítva az objektum detektálás osztályozási és lokalizációs részét és összességében növelve a detektor modell teljesítményét is.

3.1.1 Label Change Error számítása

Minden tanítómintához a modell tanítása során kiszámoljuk a Label Change Error értékét, hogy felmérjük az adott adatponthoz tartozó predikciók bizonytalanságát. A számítás az aktuális és a korábbi predikciók összehasonlításán alapul. Minden alkalommal, amikor a háló tanul egy mintán, újra prediktálunk arra a képre inference módban, majd az így kapott eredményt összevetjük az előzőleg elmentett predikcióval. Az inference mód

használata azért különösen fontos, mert a modellt a valós környezetben is ebben a módban alkalmazzuk, így szimulálhatjuk, mennyire bizonytalan a modell jelenlegi tudása alapján a valós helyzetekben. Ez lehetővé teszi, hogy nyomon kövessük, mennyiben tér el az aktuális predikció az előzőtől, és meghatározzuk mennyire bizonytalan a modell a predikcióiban.

A Label Change Error meghatározásához először ki kell számolni a Class Change és a Localization Change értékeket az adott tanítómintára. A Class Change kiszámításához a képen lévő objektumok előző és aktuális predikált osztályait hasonlítjuk össze. Az eltérések számát tekintjük a Class Change értékének, tehát ahány objektum esetén különbözik az előző és az aktuális osztály predikciója, annyi lesz a Class Change. A Localization Change esetében az előző és az aktuális bounding boxok helyzetét vetjük össze. Ez alapján amennyi bounding boxnak megváltozott a pozíciója, annyi lesz a Localization Change értéke.

Ennek megfelelően egy adott objektum esetében a Label Change Error értéke 0 vagy 1 lehet, ami azt jelzi, hogy megváltozott-e a modell predikciója az adott objektumra vonatkozóan. A változás kétféleképpen következhet be. Az objektum predikált osztályának változásával, amit a Class Change mutat meg, vagy az objektum bounding box helyzetének eltéréseivel, amit a Localization Change értéke tükröz. Egy adott tanítóminta Label Change Error értékét a képen lévő összes objektum Label Change értékének összege adja. Ha egy adatpont Label Change Error értéke magas, az arra utal, hogy a modell bizonytalan az adott minta predikciójában, és nehézséget okoz számára a képen lévő objektumok pontos osztályozása és lokalizálása.

A Label Change Error számítása a képen lévő objektumok osztályainak és a bounding boxok helyzetének összehasonlításán alapul, azonban ezek összehasonlítása nem mindig egyértelmű feladat. A legnagyobb kihívást az jelenti, hogy hogyan rendeljük össze az előző és az aktuális predikált bounding boxokat. Előfordulhat ugyanis, hogy a modell új pozícióba predikál bounding boxot, amely az előző predikciókban nem szerepelt, vagy éppen az, hogy az aktuális predikciók között nem található olyan bounding box, amely az előző predikciókban még jelen volt. Ahhoz, hogy ilyen esetekben is kiszámíthassuk a Label Change Error értékét, alkalmazzuk a Localization Change mutatót, amely a predikciók lokalizációs bizonytalanságát jellemzi, segítve ezzel az előző és aktuális predikciók közötti különbségek felmérését.

3.1.2 Localization Change érték számítása

A Localization Change értéke egy tanítóminta predikcióinak lokalizációs bizonytalanságát mutatja. Ennek kiszámításához az előző és aktuális bounding box-okat hasonlítjuk össze,

és a pozícióik közötti eltérések alapján határozzuk meg az értéket. A folyamat során kiszámítjuk az előző és aktuális bounding boxok közötti IoU (Intersection over Union) értékeket, amely egy mátrix formájában ábrázolja a különböző boxok közötti átfedéseket. A mátrix alapján azokat a bounding boxokat tekintjük összetartozónak, amelyek IoU értéke meghalad egy előre meghatározott küszöbértéket. Egy tanítóminta Localization Change értékét azoknak a bounding boxoknak a száma adja meg, amelyek IoU értéke alacsonyabb volt a küszöbértéknél, vagyis amelyek esetében nem találtunk megfelelő boxot a korábbi predikciók között.

A Localization Change számítására mutat példát a 3.1 ábra, ahol pirossal jelöljük a modell korábbi predikciójának bounding boxát, kék színnel pedig az aktuális predikciót. Az ábrán a bal oldali képen látható, hogy jelentős átfedés van a két box között, így ezek összetartozónak tekinthetők. Ennek következtében a Localization Change értéke erre az objektumra 0 lesz, ami azt jelenti, hogy a háló aktuálisan predikált bounding box helyzete csak kis mértékben változott meg az előzőhöz képest, jelezve, hogy a modell magabiztos az objektum helyzetének meghatározásában. Ezzel szemben a jobb oldali ábra azt mutatja, amikor a két box közötti IoU érték kisebb a küszöbértéknél. Itt látható, hogy a modell aktuálisan predikált bounding box pozíciója jelentősen eltér az előzőhöz képest, ezért a Localization Change értéke 1 lesz, ami a predikció bizonytalanságát tükrözi.



Figure 3.1: Localization Change szemléltetése

Ezt a módszert könnyen alkalmazhatjuk a nehézséget okozó false positive és false negative bounding boxokra is. A false positive boxok olyan új boxok, amelyek korábban nem szerepeltek a háló predikcióiban, de az aktuális predikcióban már megjelennek, így nincs hozzájuk tartozó korábbi box. Ezzel szemben a false negative boxok olyan boxok, amelyek a korábbi predikciókban még jelen voltak, de az aktuális predikcióban már nem szerepelnek, így nincs aktuális box, amely megfelelne a korábbi predikciónak. Mind a false positive, mind a false negative boxok növelik a Localization Change értékét, mivel ezek a változások jelzik a modell bizonytalanságát és a predikciók közötti eltéréseket.

3.1.3 Class Change érték számítása

A Class Change értéke egy tanítóminta predikcióinak osztályozási bizonytalanságát jellemzi. A kiszámításhoz az előző és aktuális predikciókban szereplő objektumok osztályait hasonlítjuk össze, és az eltérések alapján határozzuk meg az értéket. A Class Change számítása során azonban annyiban könnyebb a feladat, hogy a Localization Change számításakor már azonosítottuk az összetartozó bounding boxokat. Ezt az információt felhasználva a Class Change kiszámításakor elegendő ezeknek az összetartozó boxoknak az osztályait összevetnünk. Végül egy tanítóminta Class Change értékét azoknak az objektumoknak a száma adja, amelyeknél az aktuális predikció osztálya eltért a korábbi predikció osztályától.

A 3.2 ábra a Class Change számítására mutat példát, ahol pirossal jelöljük a modell korábbi predikciójának osztályát, kék színnel pedig az aktuális predikciót. A bal oldali képen látható, hogy a két predikált osztály megegyezik. Ennek következtében a Class Change értéke erre az objektumra 0 lesz, ami azt jelenti, hogy a háló magabiztosan azonosította az objektum osztályát, mivel nem történt változás a klasszifikációban az előző predikcióhoz képest.



Figure 3.2: Class Change szemléltetése

Ezzel ellentétben a jobb oldali kép azt szemlélteti, amikor az objektum osztálya eltér az előző predikcióhoz képest. Az ábrán látható, hogy a modell aktuálisan predikált osztálya különbözik a korábbtól, ami arra utal, hogy a háló bizonytalan az objektum osztályának meghatározásában. Ennek következtében a Class Change értéke 1 lesz, ami a predikció osztályozási bizonytalanságát mutatja.

3.1.4 Label Change Error jelentésének szemléltetése

A 3.3 ábra a Label Change Error számításának egy összetettebb példáját mutatja be. A képen látható, hogy a bal oldalon álló autó pozíciójának meghatározása kihívást jelent

a háló számára, feltehetően a részleges kitakarás miatt. A modell aktuálisan predikált keretének helyzete eltér az előzőtől, így ennek az autónak a Localization Change, és ez alapján Label Change Error értéke is 1 lesz. A kép jobb oldalán egy biciklit támasztó gyalogos látható, aki szintén nehézséget okoz a modell számára. Korábban a háló gyalogosként azonosította, azonban a jelenlegi predikció szerint biciklisként ismeri fel. Ennek eredményeként megváltozott a bounding box helyzete és osztálya is, így az objektum Class Change és Localization Change értéke is 1 lesz, vagyis a Label Change Error értéke 1. Ez a változás jól szemlélteti, hogy ez az objektum egyértelműen nehéz eset, hiszen a döntési határon helyezkedik el, és nehéz meghatározni, hogy gyalogosként vagy biciklisként kell-e kategorizálni.

További nehézséget jelent a modell számára az úttest mentén parkoló furgon, amit korábban autóként osztályozott, most azonban furgonként detektált. Ez szintén egy összetett példa, hiszen a két osztály között nincs nagy különbség. Ezzel szemben a kereszteződésben álló gyalogos és az úton haladó autó esetében a modell magabiztosan megadta az objektumok helyzetét és az osztályukat is. Ezek az egyszerűbb példák közé tartoznak, mivel jellemzőik egyértelműen látszódnak, amik alapján a háló könnyen tud döntést hozni.

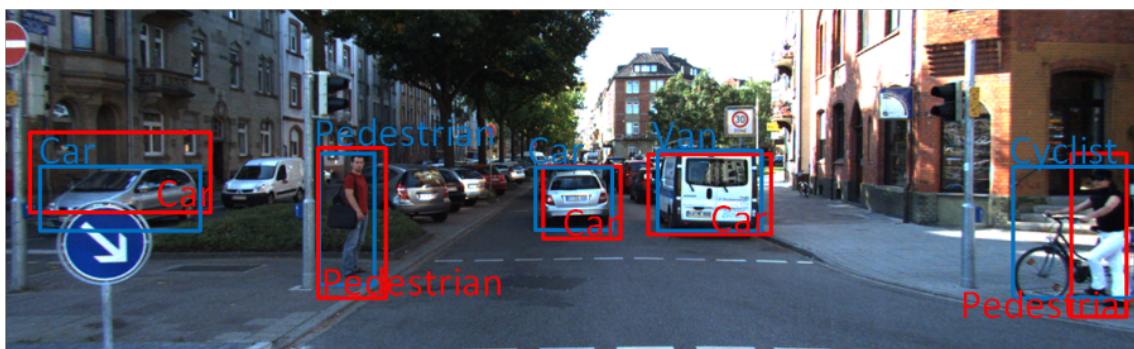


Figure 3.3: Label Change Error szemléltetése

Összességében a képen három objektum esetében történt változás a bounding boxuk helyzetében vagy az osztályban, így a tanítómintához tartozó Label Change Error értéke 3 lesz. Ez az érték jól mutatja, hogy a metrika alapján ez egy olyan tanítóminta, amelynél a modell bizonytalanságot mutat a tanulási folyamat során. A képen ugyanis több olyan objektum is található, amelyek detektálásakor a modell nem tudott egyértelmű döntést hozni, és az előző fit-elés során kapott predikcióhoz képest változás történt. Ennek következtében a Label Change Error értéke magasabb, jelezve, hogy a háló számára ez a minta összetett és nehezebben értelmezhető példákat tartalmaz.

3.1.5 Tanítóminták mintavételezési valószínűségének frissítése

Az adat priorizálási módszer alkalmazásával a tanítás során a magasabb információs tartalmú minták kiemelése érdekében frissítjük az adatpontok mintavételezési valószínűségeit. Ehhez a Label Change Error értéket felhasználva a bizonytalanabb tanító mintákhoz nagyobb kiválasztási valószínűséget rendelünk, aminek eredményeképpen ezekre a kritikus adatpontokra fókuszálva a modellt a bizonytalanabb minták hatékonyabb felismerésére tanítjuk. Az adat priorizálás folyamata során az adatpontok kiválasztása nagyban hasonlít a 2.2. fejezetben ismertett megerősítéses tanulás során jelentkező exploration-exploitation tradeoff problémájára, ami felügyelt tanulás esetén az alul- és túltanulás problémájának felel meg.

A felügyelt tanulás kontextusában az exploration-exploitation tradeoff jelensége úgy értelmezhető, hogy ha túlzottan a felfedezésre (exploration) összpontosítunk anélkül, hogy a megszerzett ismereteket kellően finomítanánk, alultanulás (underfitting) léphet fel. Ilyen esetben a modell nem tudja megfelelően elsajátítani az adatok mögötti struktúrákat, túlságosan általános marad, és nem képes felfedezni azokat a mintázásokat, amelyek elengedhetetlenek a pontos predikciókhoz. Ezzel szemben, ha túl nagy hangsúlyt fektetünk a már megszerzett, szűkebb tudásra (exploitation), a túltanulás (overfitting) problémája jelentkezhet. Ilyenkor a modell túlzottan igazodik a tanító mintákhoz, ami csökkenti az általánosító képességét új, eddig nem látott adatokon, és végső soron gyengébb teljesítményt eredményezhet a valós környezetben.

Az adat priorizálás során az alul- és túltanulás közötti egyensúly megtalálásához alkalmazhatjuk azokat a módszereket, amelyek a megerősítéses tanulásban hatékonyan kezelik az exploration-exploitation tradeoff problémáját. Az ilyen eljárások közé tartozik például a [15] cikkben ismertetett Upper Confidence Bound (UCB) stratégia, amely matematikai leírása alábbi képlettel adható meg.

$$UCB_i = \frac{p_i^\alpha}{\sum_k p_k^\alpha} + c_p \cdot \sqrt{\frac{2 \cdot \ln(\max_k n_k)}{n_i + \epsilon}}$$

A képlet egy adott tanító mintára adja meg az UCB értékét az exploration és exploitation tagok összegeként. Az exploitation tag számításához a bizonytalanabb tanító minták kiemelése érdekében a Label Change Error metrikát használjuk fel az alábbi képlet alapján.

$$exploitation_i = \frac{LC_i}{\max_k (LC_k)}$$

A képlet egy adatponthoz rendeli az exploitation tag normalizált értékét, ahol LC_i az

adott tanító mintához tartozó Label Change Error értékek összegét jelenti. Mivel az LC_i értéke függ a képen található ground truth bounding boxok számától, a tanító minták bizonytalanságának összehasonlíthatósága érdekében célszerű a normalizált Label Change Error-t alkalmazni. Ennek a normalizációnak az a célja, hogy az exploitation tag értékei jobban kifejezzék a tanító minta bizonytalanságát, figyelembe véve az összes többi mintát is. Ennek érdekében az LC_i -t elosztjuk az összes adatponthoz tartozó Label Change Error értékek maximumával, aminek eredményeképpen megkapjuk a normalizált exploitation tag értékét. Ez az érték pontosabban tükrözi a tanító minta bizonytalanságát, és ezzel lehetővé teszi a minták közötti hatékonyabb összehasonlítást.

A bizonytalanabb tanító minták kiemelésével azonban problémát okozhat a túltanulás kockázata. Mivel a neurális háló gyakran találkozik bizonyos adatokkal tanítás közben, ezért előfordulhat, hogy túlságosan specializálódik ezekre a mintákra, ami csökkentheti a modell általánosító képességét új környezetekben. A túltanulás kockázatának csökkentése érdekében adat priorizálás alatt alkalmazzuk az exploration tagot, ami korlátozza a mintavételezés során az adatpontok kiválasztási valószínűségeit befolyásoló súlyokat.

$$exploration_i = c_p \cdot \sqrt{\frac{2 \cdot \ln(\max_k n_k)}{n_i + \epsilon}}$$

A képletben n_i azt mutatja meg, hányszor használta a modell az adott tanító mintát a tanulás során a neurális háló súlyainak frissítésére. Ezt az értéket ahhoz viszonyítjuk, hogy az összes adatpont közül melyik mintával találkozott a legtöbbször a modell a tanulás során. Az adatpontok tanítás közben történő felhasználásának számát azért viszonyítjuk, mert ez lehetővé teszi az egyes tanító minták felhasználási gyakoriságának normalizálását. Így összehasonlíthatóvá válik, hogy egy-egy adatpont milyen mértékben járult hozzá a modell tanulási folyamatához.

A nullával való osztás lehetőségének elkerülése érdekében bevezetünk egy konstans ϵ értéket. Továbbá, az exploration tagban egy új c_p hiperparaméter is szerepel, amely egy simító tagként funkcionál. A c_p hangolása révén lehetőség van az exploitation hatásának csökkentésére szélsőséges esetekben, amikor az egyensúly túlzottan az exploitation felé tolódik el. Ez különösen fontos az olyan minták kiemelésében, amelyek az adat priorizálás során kevesebb figyelmet kaptak, így csökkentve a túltanulás kockázatát, és javítva a modell általánosító képességét.

Végül az exploitation és exploration tagokat felhasználva minden alkalommal frissítjük az adott tanító mintához tartozó mintavételezési valószínűséget, amikor a modell az adott adatponton tanul. A Label Change Error értékére épülő exploitation tag alkalmazásával megnöveljük a kiválasztási valószínűséget azokhoz az adatokhoz, amelyeket a hálózat a

tanulás során bizonytalanabbnak ítél. Ezzel kiemeljük ezeket a mintákat, és hatékonyabban tanítjuk a modellt arra, hogy felismerje ezeket az összetettebb eseteket. Eközben az exploration tag, a c_p hiperparaméter segítségével, ellensúlyozza exploitation tag hatását annak érdekében, hogy csökkentsük a túltanulás kockázatát az adat priorizálás során.

$$W_i = UCB_i = exploitation_i + exploration_i$$

Ezzel a kiszámolt UCB értékkel módosítva a tanító minta mintavételezési valószínűségének súlyát egyensúlyt teremtünk az adat priorizálás során az exploration és exploitation között. Így a modell a két stratégia előnyeit kihasználva képes optimalizálni a tanulási folyamatot, ami hozzájárul a neurális háló teljesítményének maximalizálásához és jobb általánosító képesség eléréséhez.

3.1.6 Adat priorizáló módszer működése

A 3.4 ábrán az adat priorizálási folyamat látható, amely a tanítás során az egyes minták kiválasztását befolyásolja.

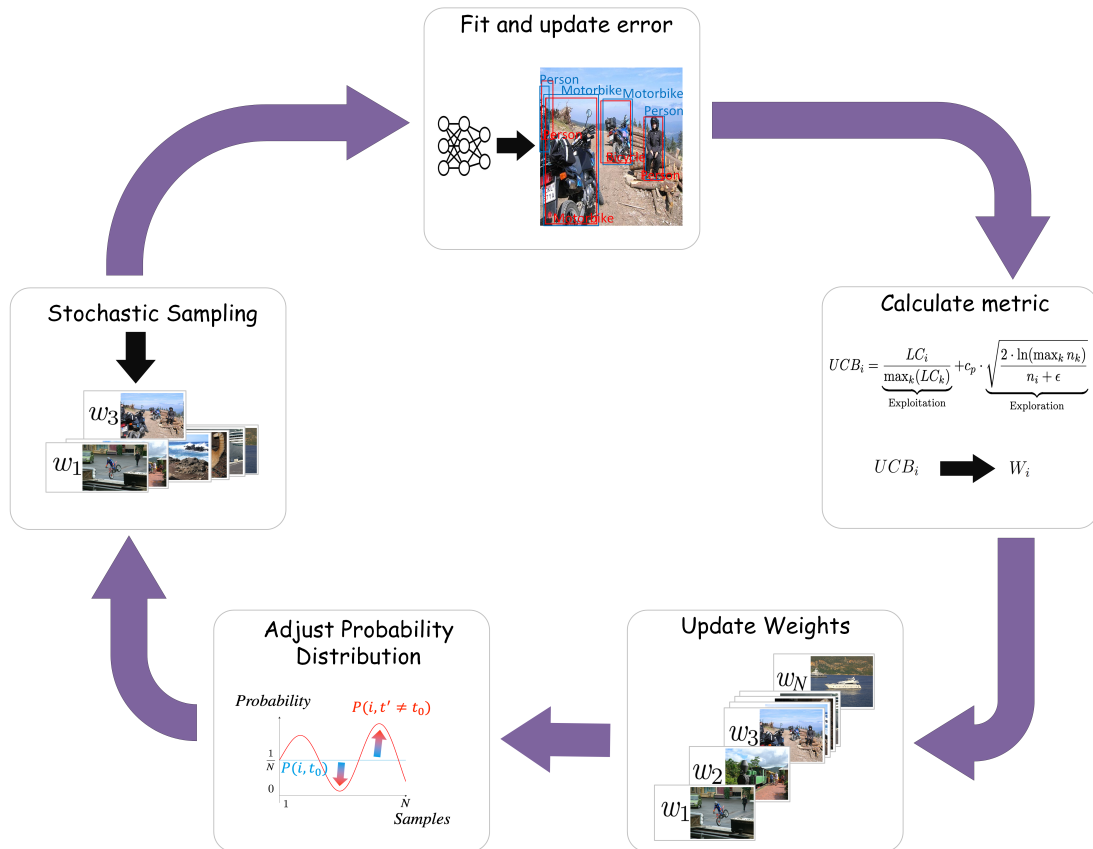


Figure 3.4: Adat priorizáló módszer működésének szemléltetése

Minden tanítási lépésben, amikor a modell tanul egy adott mintán, kiszámoljuk az adatponthoz tartozó Label Change Error-t. Ezt követően az előző fejezetben ismertetett módszer szerint meghatározzuk az UCB értéket, amely segítségével súlyozzuk az adatpontok mintavételezési valószínűségeit. Ennek hatására az eredetileg egyenletes eloszlású mintavételezési valószínűségek megváltoznak, és a következő mintákat sztochasztikus mintavételezéssel választjuk ki a háló tanításához. A folyamat során a Label Change Error metrika alapján nehezebb mintáknak magasabb valószínűséget rendelünk, így ezeket az adatokat gyakrabban választjuk ki a tanítás során, ami segíti a modell teljesítményének javítását az összetettebb példákra.

3.2 Kísérletek

3.2.1 Benchmark adatok

Minden gépi látással kapcsolatos feladatkörhöz (például klasszifikáció, objektum detektálás) tartoznak benchmark adat szettek. Ezek mindenki számára ingyenesen elérhető képeket tartalmazó adathalmazok, melyek már rendelkeznek annotációval. A szakirodalom kapcsolódó publikációi jellemzően ezeket az adathalmazokat használják az eredmények prezentálására. Ennek köszönhetően a modellek kiértékelése során kapott eredmények összehasonlíthatóak lesznek a korábbi munkákkal, és így egyértelműen láthatóvá válik a fejlődés mértéke és iránya. Annak érdekében, hogy pontos képet kaphassunk a kidolgozott módszer hatékonyságáról a modellt több adatbázison, különböző körülmények között készült képeken értékeljük ki, amivel a háló robusztussága is vizsgálható. Az adatbázisok kiválasztása során fontos szempont még, hogy a felhasználási környezethez hasonló adatokon is kiértékelhessük a betanított modelleket. Ehhez olyan adatokat használunk, amivel könnyen tesztelhető, hogyan teljesít a háló a járműipari környezetérzékelés területén.

A KITTI [6] adatbázis az autonóm járművek kutatásában használt egyik legjelentősebb és legelterjedtebb nyílt forráskódú adatbázis. A gépi látás számos feladatához biztosít tanító adatokat, mint például az objektum detektálás, szegmentálás, optical flow számítás, odometria vagy track-elés. Az adatbázis városi környezetben forgalmi szituációkról tartalmaz képeket, amiken különböző osztályok szerepelnek mint például a forgalomban résztvevő járművek, gyalogosok. A detektor modellek tanításához közel 15000 annotált tanító és teszt adat áll rendelkezésünkre, amelyeken körülbelül 80000 címkézett objektum szerepel.

A PASCAL VOC [5] egy régebbi, de a mai napig széleskörben elterjedt adatbázis az objektum detektáló neurális hálók tanításához. Az objektum detektáláson kívül lehet még klasszifikációs hálók tanításához is felhasználni. Az adatbázis közel 10000 képet tartalmaz,

amelyeken majdnem 25000 objektum jelenik meg. Az adatokon 20 osztályt jelöltek fel az annotálás során. Ilyen osztályok például többek között emberek, autó, busz, vonat vagy egyéb beltéri objektumok is mint például asztal vagy TV.

3.2.2 Modell tanítása

A dolgozatban ismertetett módszer kidolgozásakor fontos szempont volt, hogy az eljárás tudományos értéke mellett olyan megoldást alakítsunk ki, amely a gyakorlatban is alkalmazható. Ehhez elengedhetetlen, hogy valós futásidejű objektumdetektáló hálót alkalmazzunk, ezért a fejezetben leírtak alapján a YOLO architektúrát választottuk a modell tanításához. A YOLO egy state-of-the-art neurális háló architektúra, amely nagy népszerűségnek örvend az ipari alkalmazásokban, köszönhetően gyors és pontos detektálási képességeinek. Így a kidolgozott módszert a YOLO architektúrával alkalmazva olyan megoldást kapunk, amely széleskörűen felhasználható a gyakorlatban is. Emellett az objektum detektálás szakirodalmában jellemzően YOLO modelleket használnak, így a háló kiértékelése során kapott eredmények könnyen összehasonlíthatóak lesznek a korábbi munkákkal.

Az évek során számos verziója jelent meg a YOLO architektúrának, amik különböző új technológiákat vezettek be az objektum detektáló algoritmusok területén. Ezek közül a kidolgozott módszer alkalmazásához a YOLOv5 és YOLOv8-as modelleket használtuk fel a neurális háló tanításához. A YOLOv5 az egyik legnépszerűbb a YOLO modell, amit előszeretettel használnak a járműipari rendszerekben is, míg a YOLOv8 az első olyan YOLO architektúra, ami anchor-free objektum detektálást valósít meg szemben a korábbi anchor boxokra támaszkodó megoldásokkal. Ezekhez a modellekhez külön speciális backend tartozik, azonban a kidolgozott adat priorizáló módszer kiértékeléséhez a saját backendjük mellett más architektúrával is tanítottuk a YOLO modelleket. Ilyen neurális háló volt például a ResNet, amelyet a YOLO hálók mellett alkalmaztunk.

A YOLO modelleket a háló tanításához a Kitti és Pascal adatszeteken használtuk fel. A modelleket 150 epochon keresztül tanítottuk 640-es méretű képekkel és 32-es batch mérettel. A tanítás során Adam optimalizálót és cosine annealing learning rate schedulert alkalmaztunk 50 epoch hosszúságú warmup szakasszal. A tanítóadatok augmentálásához a beépített YOLO augmentálási módszereket használtuk, mint például a random scaling (véletlenszerű méretezés), horizontal flipping (vízszintes tükrözés), color jitter (színváltoztatás) és random perspective (véletlenszerű perspektíva torzítás), amely segíti a modelleket a különböző nézőpontok kezelésében.

Ha a tanítás során adat priorizálást alkalmazunk, akkor kulcskérdés a c_p simító tag

értékének megfelelő megválasztása. Ugyanis a c_p tag segítségével egyensúlyt teremtünk az exploration és exploitation között, ami nagyban befolyásolja adat priorizálás során a tanítóminták mintavételezési valószínűségeinek súlyozását, és ezáltal a modell teljesítményét. Ha a c_p értéke túl kicsi, akkor az algoritmus inkább az exploitationre fog összpontosítani, ami gyorsabb konvergenciához vezethet, de fennáll a túltanulás veszélye. Ezzel szemben, ha a c_p értéke nagy, az algoritmus több időt fordít az exploration, vagyis gyakrabban választ új vagy kevésbé használt adatmintákat ahelyett, hogy a megszerzett ismereteket kellően finomítaná. Ez alultanulást eredményezhet és lassíthatja a konvergenciát, mivel az algoritmus kevésbé összpontosít a magas prioritású mintákra. Ez alapján különböző próbálkozások után a c_p értékét 1-nek választottuk.

Azzal, hogy a kidolgozott adat priorizáló módszert különböző architektúrák és adatszetek mellett is megvizsgáltuk átfogóbb képet kaptunk a kidolgozott módszer hatékonyságáról és általánosíthatóságáról. Ennek eredményeként arra a következtetésre juthatunk, hogy egy általános megoldást hoztunk létre, ami különböző technikákkal és különböző adatokon is hatékonyan működik objektum detektálás esetén.

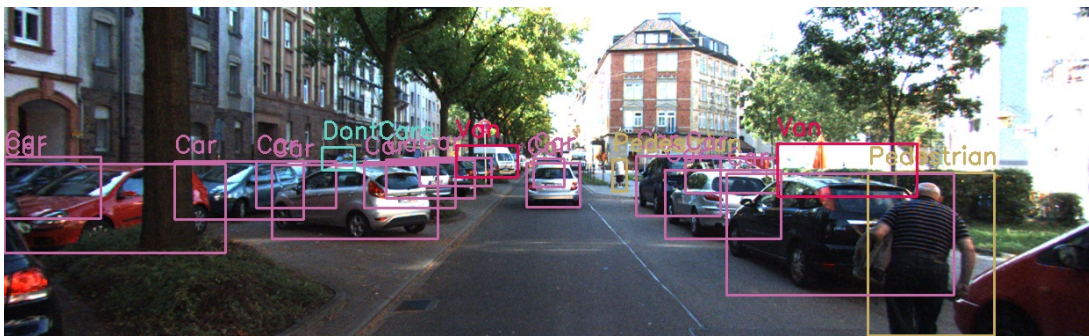
3.2.3 Adat priorizálás hatásának vizsgálata

Az adat priorizálás hatásának vizsgálatához a Label Change Error metrikát alkalmazva kiválasztottuk a modell számára nehezebb és könnyebb tanítómintákat. Nehezebb adatpontnak minősülnek azok a képek, amelyekhez tartozó Label Change Error érték magas, vagyis amelyeknél a modell bizonytalan volt a tanítás során. Ezzel szemben a könnyebb példák azok, amelyeknél alacsonyabb ez az érték. Ennek érdekében a tanítás során minden adatponthoz rögzítettük, hogy a háló hányszor tanult rajta, és mennyi volt a hozzárendelt Label Change Error értéke. Ezt követően predikciókat készítettünk az adat priorizálással és anélkül tanított modellekkel azokra a képekre, amelyeken a háló legtöbbször illetve legkevesebbszer tanult az adat priorizálás során. Ennek köszönhetően lehetőségünk nyílik összehasonlítani az eredményeket amivel betekintést nyerhetünk abba, hogyan javult az adat priorizálással tanított modell teljesítménye a nehezebb tanítóminták kiemelésével.

A 3.5 ábrán a Label Change Error alapján egy nehezebb és egy könnyebb tanítóminta összehasonlítása figyelhető meg. A tanítóminták információs tartalmának szemléltetésekképpen az ábrán megjelenítettük a képeken található objektumokhoz tartozó ground truth bounding boxokat.



(a) Label Change Error alapján könnyű tanítóminta



(b) Label Change Error alapján nehéz tanítóminta

Figure 3.5: Tanítóminták összehasonlítása Label Change Error alapján

A 3.5a ábrán látható, hogy kevés objektum szerepel a képen, és ezek jellemzői könnyen kivehetők. Ezzel szemben a 3.5b képen sokkal több objektum és különböző osztály található. További nehézséget jelent, hogy számos objektum esetében nehéz meghatározni azok helyzetét és osztályát a részleges kitakarások miatt. Ez különösen igaz a szorosan egymás mellett álló járműveknél, ahol az autók egymást részben takarva csak kis részeik láthatók. Az ilyen objektumok felismerése komoly kihívást jelenthet a háló számára, és bizonytalanságot okozhat azok detektálásában, ami ronthatja a teljesítményét zsúfolt környezetekben.

A 3.6 ábrán egy nehezebb tanítóminta predikcióit hasonlítottuk össze. Piros színnel a modell false positive predikció, míg kézzel a false negative boxok láthatóak. Mivel sok objektum található a képen, ezért a jobb átláthatóság érdekében csak a háló hibás predikciót ábrázoltuk, a helyes boxokat nem. Ennek alapján minél kevesebb false positive vagy false negative box jelenik meg a képen annál pontosabbnak tekinthető a modell objektum detektálása.



(a) Adat priorizálás alkalmazása nélkül tanított modell téves predikciói



(b) Adat priorizálás alkalmazásával tanított modell téves predikciói

Figure 3.6: Adat priorizálás hatásának szemléltetése nehéz minta esetén

A 3.6b képen látható, hogy az adat priorizálás alkalmazásával tanított modell hibás predikcióinak száma lecsökkent. A 3.6a ábrán korábban megjelenő két false positive box eltűnt, továbbá a parkolóban egymás mellett álló autók esetében a korábbi két false positive box helyett már csak egy jelenik meg, és az ott található false negative box is hiányzik az adat priorizálással tanított modell predikciói közül. Ez alapján az adat priorizálással tanított háló objektum detektálása pontosabbnak bizonyult, mivel több képen lévő objektumot talált meg, miközben kevesebb területet azonosított tévesen objektumként. Ebből arra a következtetésre juthatunk, hogy a nehezebb tanítóminták kiemelésével a modell pontosabban meg tudja tanulni az ezeken a képeken található összetettebb jellemzőket, aminek köszönhetően javul a háló teljesítménye.

Ezzel szemben a 3.5a ábra a Label Change Error metrika alapján egy könnyebb tanítómintát mutat be. Az előző példától eltérően itt nem szerepelnek téves boxok a modell predikciói között. Ez arra utal, hogy mindkét modell helyesen detektálta a képen található autót, és a hálók által predikált boxok jelentős átfedést mutatnak egymással. Ezt azt jelzi, hogy a két modell hasonló teljesítményét mutatott ennél az egyszerűbb mintánál.

Az adat priorizálással tanított modell azonban lényegesen kevesebbet találkozott a könnyebb mintákkal a tanulás során. Ugyanis az adat priorizálás alatt a Label Change

Error metrika alapján változtatjuk a tanítóminták mintavételezési valószínűségeit, amivel magasabb valószínűséget rendelünk a nehezebb képekhez és alacsonyabbat a könnyebb adatpontokhoz. Ez az előző példa alapján azt jelenti, hogy a könnyebb tanítóminták esetén a modell kevesebb idő alatt is képes megtanulni a pontos detektáláshoz szükséges jellemzőket az egyszerűbb képekről. Ezzel együtt a nehezebb tanítómintáknál bemutatott összehasonlítás során azt figyeltük meg, hogy az adat priorizálással tanított háló teljesítménye javult, mivel a nehezebb képekre több figyelmet fordítottunk. Annak köszönhetően, hogy a modell gyakrabban tanult ezeken a képeken, képes volt mélyebben elsajátítani a rajtuk található összetettebb jellemzőket, ami végső soron az objektum detektálási pontosság növekedéséhez vezetett.

Ez alapján arra a következtetésre juthatunk, hogy a Label Change Error szerinti adat priorizálási módszer javította a rendelkezésünkre álló adatok felhasználásának hatékonyságát. A módszer annak köszönhetően, hogy kiemeli magasabb információs tartalommal rendelkező tanítómintákat nagyobb hangsúlyt fektet a nehezebb képeken található összetettebb jellemzők megtanulására. Így a modell nem csak gyorsabban sajátítja el a könnyebb minták jellemzőit, hanem mélyebben megérti a komplexebb struktúrákat is, ami növeli a modell objektum detektálási pontosságát, és javítja az általánosító képességét a kihívást jelentő környezetekben is.

4 Eredmények

A betanított modellek teljesítményét a validációs adathalmazon értékeltük ki, amely olyan képeket tartalmaz, amelyekkel nem találkozott a háló a tanítás során. Ez lehetővé teszi, hogy felmérjük a modell általánosító képességét ismeretlen adatokon, és betekintést nyerjünk abba, hogyan teljesítene a háló valós környezetben. Annak érdekében, hogy átfogóbb képet kapjunk a modellek teljesítményéről és általánosíthatóságáról különböző háló architektúrákat, és adatszeteket használtunk fel a kiértékeléshez. A fejezetben leírtak alapján YOLOv5 és YOLOv8-as hálókat tanítottunk be különböző backboneokkal KITTI és Pascal VOC adatszeteken.

Továbbá a pontosabb eredmények érdekében több véletlen seedet használtunk a modellek tanításához és teszteléséhez, majd az eredményeket kiátlagoltuk. Ez a megközelítés segít csökkenteni az egyes seedekből adódó véletlen ingadozásokat, így megbízhatóbb és stabilabb képet kaphatunk a modell teljesítményéről, amely jobban tükrözi a valós képességeit. Az így kapott eredményeket az alábbi táblázatok foglalják össze.

Table 4.1: F1 score értékek összehasonlítása a validációs adathalmazon

	Dataset	YOLOv8 with YOLO backbone	YOLOv5 with ResNet backbone
Uniform Sampling	KITTI	81.18%	69.12%
	Pascal VOC	48.05%	49.41%
Prioritized Sampling	KITTI	81.63%	69.52%
	Pascal VOC	49.24%	50.38%
<i>F1 Score Gain</i>	KITTI	+0.451%	+0.393%
	Pascal VOC	+1.184%	+0.965%

A táblázatokban a Uniform Sampling értékek az egyenletes eloszlású mintavételezéssel tanított modellek eredményeit mutatják, míg a Prioritized Sampling a dolgozatban ismertett adat priorizálási módszerrel tanított háló teljesítményét tükrözi. A 4.1 táblázatban szereplő F1 score értékekből egyértelműen látható, hogy jelentősen megnőtt az adat priorizálással tanított modell teljesítménye különböző architektúrák és adatszetek mellett is.

Az F1 score metrika nagyobb hangsúlyt fektet a hálózat klasszifikációs teljesítményének szemléltetésére, ugyanis az objektum detektálási modell kiértékeléséhez szükség van egy meghatározott IoU küszöbértékre is, amely felett a predikált és a ground truth bounding boxokat összetartozónak tekintjük. Az IoU küszöb alkalmazása lehetőséget ad arra, hogy kiértékeljük a hálózat lokalizációs képességeit is, de az F1 score inkább a predikciók

precision és recall értékeit súlyozza egyetlen metrikában. A dolgozat célja elsősorban a háló F1 score értékének javítása volt, azonban az adat priorizáló módszer hatására az mAP metrika eredményei is javultak.

Table 4.2: mAP50 értékek összehasonlítása a validációs adathalmazon

	Dataset	YOLOV8 with YOLO backbone	YOLOV5 with ResNet backbone
Uniform Sampling	KITTI	84.29%	73.06%
	Pascal VOC	54.90%	52.71%
Prioritized Sampling	KITTI	84.72%	73.48%
	Pascal VOC	55.00%	53.60%
<i>mAP50 Gain</i>	KITTI	+0.429%	+0.418%
	Pascal VOC	+0.107%	+0.89%

Az mAP egy erősebb metrikának tekinthető az objektum detektálás területén, mert pontosabban fejezi ki a modell klasszifikációs és lokalizációs teljesítményét is. Az mAP számítása során figyelembe veszik a teljes precision-recall görbét, az osztályok közötti teljesítménybeli különbségeket, valamint a különböző konfidencia küszöbértékek hatását is. Ennek eredményeképpen az mAP átfogóbb képet nyújt a modell detektálási képességeiről, és hatékonyabban fejezi ki a modell lokalizációs képességének pontosságát.

A 4.2 táblázatban szereplő értékek azt mutatják, hogy az adat priorizálással tanított modell teljesítménye javult az mAP metrika alapján is. Ebből azt a következtetést lehet levonni, hogy az adat priorizálási módszer nagy mértékben képes javítani az objektum detektáló háló lokalizációs képességét, miközben az osztályozási pontosságukat is növelik.

A két táblázat eredményeit összehasonlítva megfigyelhető, hogy a KITTI adatszett esetében az adat priorizálással tanított modell teljesítményéhez kapcsolódó metrikák hasonló mértékben emelkedtek. Ezzel szemben a Pascal VOC adatszethoz az mAP érték növekedése jóval mérsékeltebb volt az F1 score javulásához képest. Ez arra utal, hogy az adat priorizáló módszer hatékonyabban javította a KITTI adatszeten tanult modell lokalizációs képességét.

Ez a jelenség azzal magyarázható, hogy a KITTI adatszett képein jellemzően több objektum található mint a Pascal VOC adatain, és a Label Change Error alapján ezek a képek magasabb információtartalomú tanítómintáknak számítanak. Az, hogy a KITTI adatokon tanult modell lokalizációs képessége nagyobb mértékben növekedett, azt jelzi, hogy a modell komplexebb jellemzőket volt képes elsajátítani a KITTI adatokon, amelyek elengedhetetlenek a lokalizációs teljesítmény javításához.

Az F1 score és mAP metrikák összehasonlítása mellett a tanulási folyamatot is megvizs-

gáltuk a kiértékelés során. Ennek érdekében ábrázoltuk az F1 score és mAP konvergenciagörbéit a validálási adathalmazon, szemléltetve a tanulási teljesítmény változását az egyes epochok alatt. A pontosabb eredmények érdekében kiszámoltuk több véletlen seeden kapott értékek átlagát és szórását, és megjelenítettük ezeket az értékeket az epochok függvényében. Az így kapott diagramok az alábbi ábrákon láthatóak.

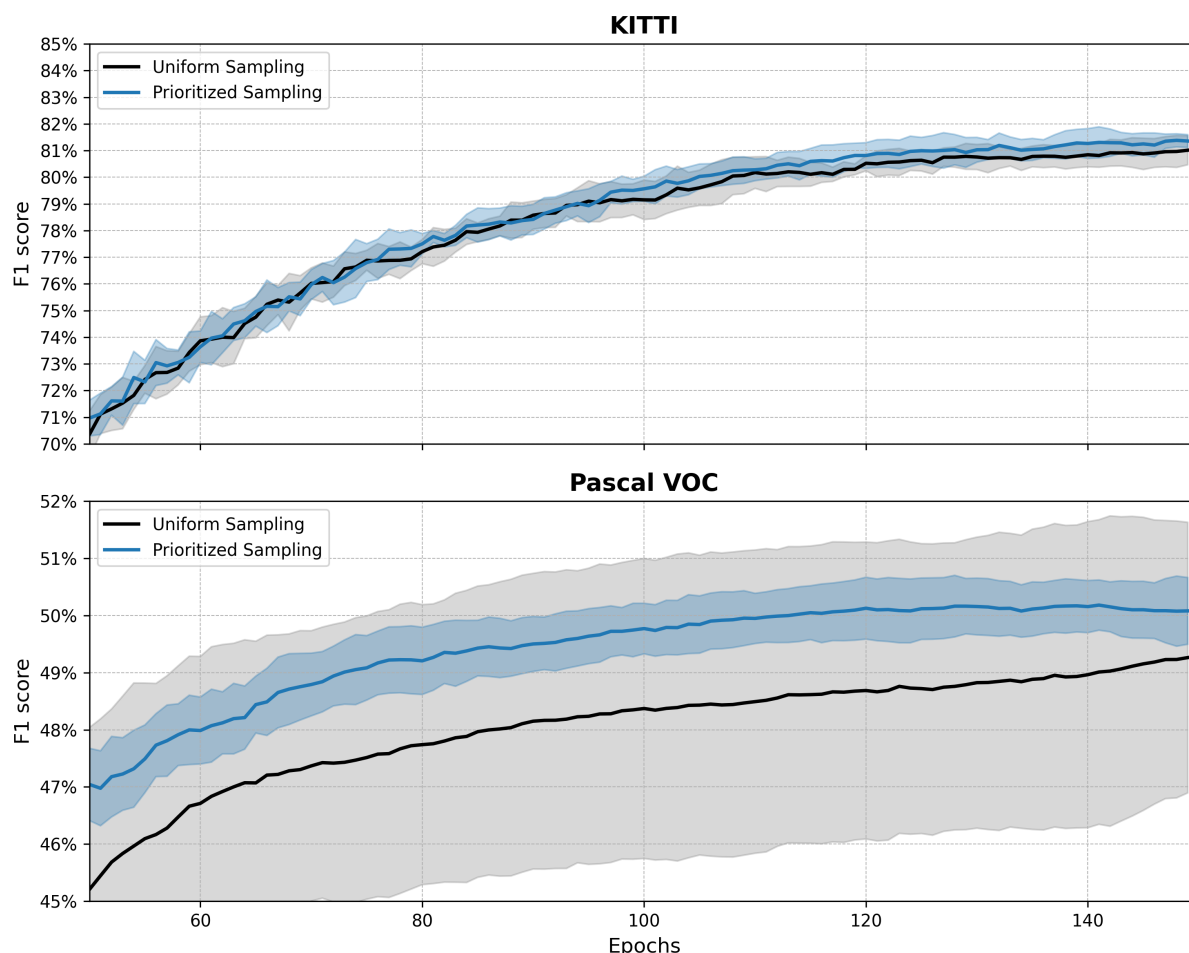


Figure 4.1: F1 score konvergenciagörbéje a validációs adathalmazon

Az ábrákon a kék és fekete vonalak az adott metrikákhoz tartozó átlagértékeket mutatják, míg a színes területek a metrikák szórását ábrázolják. Ez alapján az ábrákon megfigyelhető, hogy az adat priorizálás alkalmazása stabilabb tanulási folyamatot eredményezett, amit a csökkent variancia és a gyorsabb konvergencia is alátámaszt. A kisebb variancia arra utal, hogy a modell teljesítménye kevésbé ingadozik az epochok során, ezáltal kiszámíthatóbb és megbízhatóbb tanulási folyamatot biztosít a kidolgozott módszer. Az adat priorizálás lehetővé teszi, hogy a modell hatékonyabban sajátítsa el az összetettebb jellemzőket, ezáltal javítva a tanulási folyamat konzisztenciáját. Ennek eredményeképpen az adat priorizálás nemcsak a konvergencia sebességét növeli, hanem fokozza a modell

általánosító képességét is, hiszen a kisebb varianciájú teljesítmény azt mutatja, hogy a modell kevésbé érzékeny a tanítóminták változékonyságára.

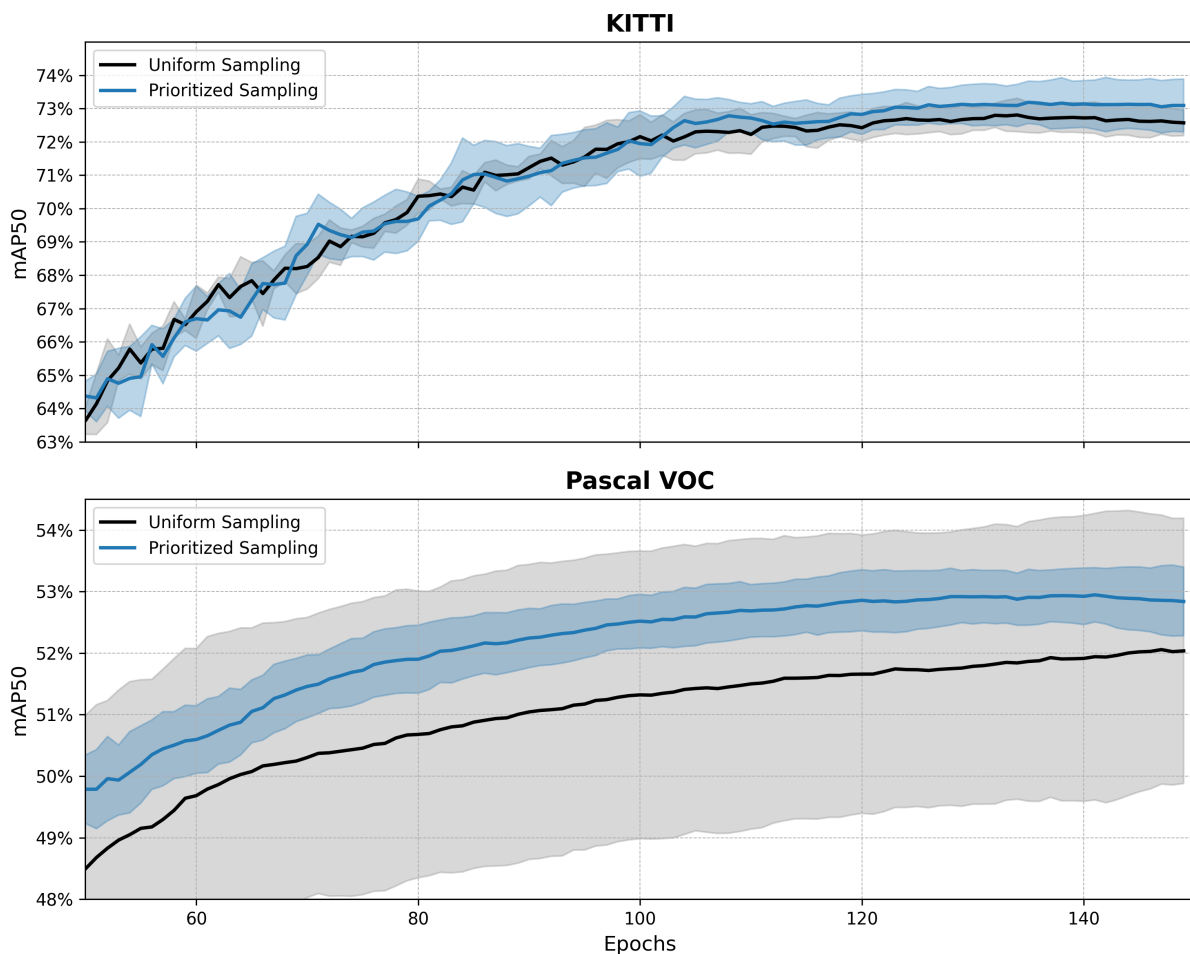


Figure 4.2: mAP50 konvergenciagörbéje a validációs adathalmazon

A kiértékelés során kapott eredmények azt mutatják, hogy a kidolgozott módszer javította a rendelkezésünkre álló tanítóminták felhasználásának hatékonyságát, és a magasabb információs tartalommal rendelkező adatpontok kiemelésével növelte a modell detektálási pontosságát. A Label Change Error metrika alapján történő adat priorizálás javította a háló klasszifikációs és lokalizációs képességét, amivel összességében növelte annak teljesítményét. Emellett a kidolgozott módszer javította a modell általánosító képességét is, amelynek köszönhetően robosztusabb és megbízhatóbb hálót kaptunk, ami változatosabb körülmények között is alkalmazható valós környezetben. Azzal, hogy különböző háló architektúrákkal és adatszettken is javítani tudtunk az adat priorizálással az detektálás teljesítményén azt jelezni, hogy egy általános megoldást hoztunk létre, ami különböző technikákkal és különböző adatokon is hatékonyan működik objektum detektálás esetén.

5 Összegzés

A dolgozat a gépi tanulás egyik mai napig kihívást jelentő területével foglalkozik, az adatok mintavételezési hatékonyságának fejlesztésével a neurális hálók tanítása során. A modellek teljesítményét alapvetően határozza meg a rendelkezésünkre álló adatok minősége, és azok felhasználása, emiatt kulcsfontosságú azok hatékony mintavételezése a tanítás folyamán. Erre a problémára nyújthat korszerű megoldást az adat priorizálási technikák alkalmazása.

Az adatok priorizálása a felügyelt tanulás és a megerősítéses tanulás határterületén helyezkedik el. A tanítóminták hatékony mintavételezése viszonylag feltáratlan terület a felügyelt tanulás, és így a gépi látás kontextusában, azonban a megerősítéses tanulás esetében léteznek már kidolgozott megoldások. Ennek köszönhetően a bemutatott módszer inspirációt merít az ismert adat priorizálási stratégiákból a megerősítéses tanulás területén, és az ott megtalálható metodológiák szellemiségét adaptálja a felügyelt tanulásban, az objektum detektálás esetére.

A dolgozatban tanítóminták információs tartalmának jellemzésére fejlesztettük ki a Label Change Error metrikát. A tanítás során a Label Change Error értékét felhasználva a magasabb információ tartalmú adatpontok kiemelésével a modellt a bizonytalanabb minták hatékonyabb felismerésére taníthatjuk, és ezzel összességében növelhetjük a neurális háló általánosító képességét és teljesítményét.

A modellek kiértékelése során azt tapasztaltuk, hogy az adat priorizáló módszer alkalmazásának köszönhetően javultak az objektum detektáló hálók validációs F1 score és mAP metrikái. Ez az eredmény azt mutatja, hogy a kidolgozott módszer nem csupán a klasszifikációs teljesítményt, de a lokáció meghatározásának hatékonyságát is javítja, és ezzel robosztusabb módon növeli a modellek detektálási pontosságát. Emellett azzal, hogy különböző háló architektúrákkal és adatszettken is javítani tudtunk adat priorizálással a detektálás teljesítményén azt jeletni, hogy egy általános megoldást hoztunk létre, ami különböző technikákkal és különböző környezetekben is hatékonyan működik.

A fejlesztett módszer alkalmazásával hozzájárulhatunk a robusztus rendszerek kialakításához a járműipari alkalmazásokban, és elősegíthetjük a biztonságos közlekedés fejlesztését. Ezenkívül az adat priorizálási módszerek vizsgálatával feltárhatjuk a felügyelt és megerősítéses tanulás közötti összefüggéseket, amivel bizonyíthatjuk a mesterséges intelligencia különböző területei fölött átívelő analógiákat. Ez nemcsak a járműipari alkalmazásokban jelenthet előrelépést, hanem tudományos kontribúcióként hozzájárulhatunk gépi tanulási módszerek jobb megismeréséhez is.

Irodalomjegyzék

- [1] Csanád Levente Balogh et al. “Strategic Data Navigation: Information Value-based Sample Selection”. In: (2024).
- [2] William H. Beluch et al. “The Power of Ensembles for Active Learning in Image Classification”. In: *2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition*. 2018, pp. 9368–9377. DOI: 10.1109/CVPR.2018.00976.
- [3] Marc Brittain et al. *Prioritized Sequence Experience Replay*. 2020. arXiv: 1905.12726 [cs.LG]. URL: <https://arxiv.org/abs/1905.12726>.
- [4] Damien Dablain, Bartosz Krawczyk, and Nitesh V. Chawla. “DeepSMOTE: Fusing Deep Learning and SMOTE for Imbalanced Data”. In: *IEEE Transactions on Neural Networks and Learning Systems* 34.9 (2023), pp. 6390–6404. DOI: 10.1109/TNNLS.2021.3136503.
- [5] M. Everingham et al. *The PASCAL Visual Object Classes Challenge 2007 (VOC2007) Results*. <http://www.pascal-network.org/challenges/VOC/voc2007/workshop/index.html>.
- [6] Andreas Geiger, Philip Lenz, and Raquel Urtasun. “Are we ready for Autonomous Driving? The KITTI Vision Benchmark Suite”. In: *Conference on Computer Vision and Pattern Recognition (CVPR)*. 2012.
- [7] Ross Girshick et al. “Region-Based Convolutional Networks for Accurate Object Detection and Segmentation”. In: *IEEE Transactions on Pattern Analysis and Machine Intelligence* 38.1 (2016), pp. 142–158. DOI: 10.1109/TPAMI.2015.2437384.
- [8] Ross Girshick et al. *Rich feature hierarchies for accurate object detection and semantic segmentation*. 2014. arXiv: 1311.2524 [cs.CV]. URL: <https://arxiv.org/abs/1311.2524>.
- [9] Ian J. Goodfellow et al. *Generative Adversarial Networks*. 2014. arXiv: 1406.2661 [stat.ML]. URL: <https://arxiv.org/abs/1406.2661>.
- [10] Kaiming He et al. *Deep Residual Learning for Image Recognition*. 2015. arXiv: 1512.03385 [cs.CV]. URL: <https://arxiv.org/abs/1512.03385>.
- [11] Kaiming He et al. *Mask R-CNN*. 2018. arXiv: 1703.06870 [cs.CV]. URL: <https://arxiv.org/abs/1703.06870>.
- [12] Dan Horgan et al. *Distributed Prioritized Experience Replay*. 2018. arXiv: 1803.00933 [cs.LG]. URL: <https://arxiv.org/abs/1803.00933>.

- [13] Licheng Jiao et al. “A Survey of Deep Learning-Based Object Detection”. In: *IEEE Access* 7 (2019), pp. 128837–128868. ISSN: 2169-3536. DOI: 10.1109/access.2019.2939201. URL: <http://dx.doi.org/10.1109/ACCESS.2019.2939201>.
- [14] Diederik P Kingma and Max Welling. *Auto-Encoding Variational Bayes*. 2022. arXiv: 1312.6114 [stat.ML]. URL: <https://arxiv.org/abs/1312.6114>.
- [15] Bálint Kóvári, Bálint Pelenczei, and Tamás Bécsi. “Enhanced Experience Prioritization: A Novel Upper Confidence Bound Approach”. In: *IEEE Access* 11 (2023), pp. 138488–138501. DOI: 10.1109/ACCESS.2023.3339248.
- [16] Alex Krizhevsky, Ilya Sutskever, and Geoffrey Hinton. “ImageNet Classification with Deep Convolutional Neural Networks”. In: *Neural Information Processing Systems* 25 (Jan. 2012). DOI: 10.1145/3065386.
- [17] Yann LeCun, Y. Bengio, and Geoffrey Hinton. “Deep Learning”. In: *Nature* 521 (May 2015), pp. 436–44. DOI: 10.1038/nature14539.
- [18] Wei Liu et al. “SSD: Single Shot MultiBox Detector”. In: *Lecture Notes in Computer Science*. Springer International Publishing, 2016, pp. 21–37. ISBN: 9783319464480. DOI: 10.1007/978-3-319-46448-0_2. URL: http://dx.doi.org/10.1007/978-3-319-46448-0_2.
- [19] Shervin Minaee et al. *Image Segmentation Using Deep Learning: A Survey*. 2020. arXiv: 2001.05566 [cs.CV]. URL: <https://arxiv.org/abs/2001.05566>.
- [20] Alexander Neubeck and Luc Van Gool. “Efficient Non-Maximum Suppression”. In: vol. 3. Jan. 2006, pp. 850–855. DOI: 10.1109/ICPR.2006.479.
- [21] Martijn Otterlo and Marco Wiering. “Reinforcement Learning and Markov Decision Processes”. In: *Reinforcement Learning: State of the Art* (Jan. 2012), pp. 3–42. DOI: 10.1007/978-3-642-27645-3_1.
- [22] Joseph Redmon et al. *You Only Look Once: Unified, Real-Time Object Detection*. 2016. arXiv: 1506.02640 [cs.CV]. URL: <https://arxiv.org/abs/1506.02640>.
- [23] Shaoqing Ren et al. *Faster R-CNN: Towards Real-Time Object Detection with Region Proposal Networks*. 2016. arXiv: 1506.01497 [cs.CV]. URL: <https://arxiv.org/abs/1506.01497>.
- [24] Robin Rombach et al. *High-Resolution Image Synthesis with Latent Diffusion Models*. 2022. arXiv: 2112.10752 [cs.CV]. URL: <https://arxiv.org/abs/2112.10752>.
- [25] Tom Schaul et al. *Prioritized Experience Replay*. 2016. arXiv: 1511.05952 [cs.LG]. URL: <https://arxiv.org/abs/1511.05952>.
- [26] Burr Settles. “Active learning literature survey”. In: (2009).

- [27] Evan Shelhamer, Jonathan Long, and Trevor Darrell. *Fully Convolutional Networks for Semantic Segmentation*. 2016. arXiv: 1605.06211 [cs.CV]. URL: <https://arxiv.org/abs/1605.06211>.
- [28] Donggeun Yoo and In Kweon. “Learning Loss for Active Learning”. In: June 2019, pp. 93–102. DOI: 10.1109/CVPR.2019.00018.
- [29] Zhengxia Zou et al. *Object Detection in 20 Years: A Survey*. 2023. arXiv: 1905.05055 [cs.CV]. URL: <https://arxiv.org/abs/1905.05055>.

Ábrajegyzék

2.1	A gépi tanulás típusai	7
2.2	YOLO működési elvének szemléltetése [22]	15
3.1	Localization Change szemléltetése	21
3.2	Class Change szemléltetése	22
3.3	Label Change Error szemléltetése	23
3.4	Adat priorizáló módszer működésének szemléltetése	26
3.5	Tanítóminták összehasonlítása Label Change Error alapján	30
3.6	Adat priorizálás hatásának szemléltetése nehéz minta esetén	31
4.1	F1 score konvergenciagörbéje a validációs adathalmazon	35
4.2	mAP50 konvergenciagörbéje a validációs adathalmazon	36

Táblázatjegyzék

4.1	F1 score értékek összehasonlítása a validációs adathalmazon	33
4.2	mAP50 értékek összehasonlítása a validációs adathalmazon	34