



ÓBUDAI EGYETEM
ÓBUDA UNIVERSITY

ORSZÁGOS TUDOMÁNYOS DIÁKKÖRI KONFERENCIA DOLGOZAT

AKADÁLYÉSZLELÉS VÁROSI KÖTÖTTPÁLYÁS KÖZLEKEDÉSBEN

Szerző: Németh Viktor
mérnökinformatikus BSc. szak, III. évf.

Konzulens: Dr. Vámosy Zoltán Imre
egyetemi docens

Budapest, 2023. január 10.

Tartalomjegyzék

1.	BEVEZETÉS	1
1.1.	A KÖZÚTI KÖZLEKEDÉS VESZÉLYEI ÉS KEZELÉSE	1
2.	HASONLÓ KUTATÁSOK ÉS FEJLESZTÉSEK	4
2.1.	A PROBLÉMA MEGKÖZELÍTÉSE	4
2.2.	HASONLÓ RENDSZEREK ISMERTETÉSE	5
2.3.	RENDSZEREK ÉRTÉKELÉSE	8
3.	RENDSZERTERVEZÉS	12
3.1.	MODULOK ÉS KÖVETELMÉNYEK	12
3.2.	VASÚTI SÍNPÁROK AZONOSÍTÁSA	13
3.2.1.	ELŐISMERETEK FELHASZNÁLÁSA	13
3.2.2.	PÁLYAAZONOSÍTÁS TECHNIKÁJA	17
3.3.	OBJEKTUMOK DETEKTÁLÁSA	23
3.3.1.	KONVOLÚCIÓS NEURÁLIS HÁLÓZATOK	23
3.4.	AKADÁLYÉSZLELÉS	26
3.4.1.	ÉSZLELÉSI FOLYAMAT	28
3.5.	RENDSZER ARCHITEKTÚRA	31
4.	EREDMÉNYEK ÉS ÉRTÉKELÉSÜK	34
4.1.	PÁLYADETEKTÁLÁSI EREDMÉNYEK	35
4.2.	AKADÁLYÉSZLELÉSI EREDMÉNYEK	42
4.3.	VÉGREHAJTÁSI SEBESSÉG EREDMÉNYEK	47
4.4.	TOVÁBBFEJLESZTÉSI LEHETŐSÉGEK	50
5.	ÖSSZEFOGLALÁS	53
	IRODALOMJEGYZÉK	55
	ÁBRA- ÉS TÁBLÁZATJEGYZÉK	57

1. BEVEZETÉS

Fővárosunk tömegközlekedésének szerves részét képezik a villamosok. Az utasok biztonságát szakképzett járművezetők biztosítják. Ám vannak olyan veszélyhelyzetek, melyekre nem lehet felkészülni és az ember néha kevés ahhoz, hogy meg tudjon akadályozni egy balesetet.¹ Dolgozatomban egy olyan támogató rendszer megvalósítását mutatom be, mely a közúti vasúti közlekedést célzott biztonságosabbá tenni. A rendszerrel szembeni átfogó követelmény, hogy a veszélyhelyzeteket képes legyen akár a járművezető előtt is felismerni és jelezni mindezt valós időben. A megvalósított rendszert a legkülönbözőbb körülmények között készült videofelvételeken teszteltem, végeztem kiértékelést, és összehasonlítottam eredményeimet a szakirodalomban fellelhető hasonló megvalósításokkal. Az eredmények rávilágítanak, hogy egy hagyományos színes kamera segítségével is elvégezhető a feladat, különböző célszenzorok használatát mellőzve.

Rendszertervezés során figyelembe kell venni a hasonló kutatásokat, fejlesztéseket (2. fejezet), illetve a szakirodalom nyújtotta lehetőségeket. Bemutatásra kerül számos kutatás és fejlesztés, melyek technikai hozzájárultak a rendszer megvalósításához.

A továbbiakban ismertetem a rendszer részletes vázlatát (3. fejezet), milyen modulok határozzák meg a rendszer működését és milyen követelményeket állítok szembe velük. Bemutatásra kerül, milyen technikák használatával megy végbe a pálya- és objektum detektálás, illetve hogyan valósul meg az akadályészlelés.

A megvalósított rendszer eredményeit (4. fejezet) pályadetektálási, akadályészlelési és rendszerteljesítményi szempontból ismertetem, összevetem hasonló rendszerek eredményeivel és továbbfejlesztési javaslatokat teszek.

1.1. A KÖZÚTI KÖZLEKEDÉS VESZÉLYEI ÉS KEZELÉSE

Az évszázad kezdetétől számítva hazánkban, a meglévőkön túl több millió közúti gépjármű állt forgalomba [1], melyek fokozott veszélyt jelentenek nem csak a közlekedésben résztvevő emberekre, de a környezetünkre is egyaránt. A megnövekedett közúti gépjárművek számával arányosan a közúti balesetek száma is jelentős mértékben emelkedett [2]. Azon személyek, akik a tömegközlekedésben részt vesznek, mint utasok, kifejezetten nagyobb veszélynek vannak kitéve. Utazáskor ki vannak szolgáltatva a körülményeknek, az adott

¹ Az állítás személyes szakmai tapasztalaton alapszik

tömegközlekedési eszközt kezelő személy döntéseinek. A tömegközlekedési eszközöket kezelő személyek magas szintű képzéseken vesznek részt, hogy a veszélyes helyzeteket a lehető legjobban tudják kezelni, ezzel együtt életet mentsenek. Sajnos vannak helyzetek, amikor nem elég a rutin, mert az ember nem biztos, hogy időben meg tudja hozni a helyes döntést².

A közúti közlekedésben folyton jelen van az emberi tényező³, mely befolyásolja a közlekedés eseményeit. Ezen események egyike akár egy szerencsétlen helyzetben balesetként is megnyilvánulhat. A baleset bekövetkezését előre megjósolni szinte lehetetlen, melyhez leginkább hozzátesz az emberi tényező kiszámíthatatlansága. Annak érdekében, hogy a balesetek számát valamilyen formában csökkenteni tudják, nem csak az infrastruktúrán hajtottak végre módosításokat, fejlesztéseket (mint például veszélyes helyre felhívó útfelfestések, jelzőtáblák), hanem a közlekedési eszközökbe is építettek számos biztonsági berendezést. Ezen berendezéseket két kategóriába lehet sorolni. Az elsőbe soroljuk az aktív eszközöket, melyek célja, hogy a veszélyes közlekedési helyzetek kialakulását csökkentsék. Ilyen eszközök például a blokkolásgátló vagy a csúszásvédelem⁴. A második kategóriába soroljuk a passzív eszközöket, melyek a baleset bekövetkezését követően próbálják csökkenteni a károkat, személyi sérülést. Ide soroljuk a légzsákokat vagy a biztonsági övet. Az aktív és passzív eszközök együttesen nyújtanak védelmet balesetekkel szemben. A biztonsági eszközök hatékonysága nagymértékben függ, hogy milyen helyzetben léptek működésbe (például a blokkolásgátlónak milyen útfelületen kell teljesítenie), illetve a jármű tulajdonságaitól.

A közlekedésben résztvevő járművek számos tulajdonságban eltérnek egymástól, melyek veszélyességi faktor aránya nagyban differenciált egyes járműfajták és járműtípusok között. A tömegközlekedésben résztvevő járművek fő tulajdonsága, hogy egy átlagos személygépjárműnél nagyobb mérettel és tömeggel rendelkeznek, melyek miatt nehezebben kontrollálhatók. Egy nem megfelelő időben megkezdett fékezésnek akár súlyos következménye is lehet. Éppen ezért a tömegközlekedésben résztvevő járműveket szakképzett kezelő személyzettel látják el a közlekedési vállalatok.

A járművezetőnek kellő figyelemmel kell defenzív módon közlekednie, mely során előtérbe szükséges helyezni a biztonságot a menetrend tartásával szemben. A biztonsághoz elengedhetetlen a kellően kipihent elme, mely fenn tudja tartani a figyelmet. A szakképzett járművezetők fokozottan ki vannak téve a közúti forgalom okozta feszültségnek, ami intenzív

² Az állítás személyes tapasztalaton alapszik.

³ Adott helyzetben résztvevő személy hozhat rossz döntést, mely negatív hatással lehet az esemény kimenetére.

⁴ Vasúti közlekedésben például a sínfék.

mentális terhet ró az elmére. Továbbá a hosszú munkaórákból adódóan fizikailag és szellemileg is kimerülnek. A fáradt elme nagyobb baleseti kockázattal jár, ami ellen az aktív biztonsági eszközök sem képesek megfelelő védelmet nyújtani. Személy szerint a tömegközlekedés közúti vasúti ágazatában voltam munkavállaló, és munkám során számtalan olyan szituációban találtam magam, ahol a járműbe épített aktív biztonsági eszköz jelentősen csökkentette a baleset bekövetkezésének esélyét. A menetrendtartás és a defenzív vezetés együtt nehezen működik. A forgalomból adódóan nagyon könnyen össze lehet szedni késést, ami később nagyobb gondot okoz, ezért a járművezető igyekszik elkerülni azt. Azonban, hogy a menetrendet tartsa, szükséges a forgalom tempójával haladni, amely fokozott veszélyt jelent a jármű hosszabb fékútja és kötött haladása miatt.

A járművezetőkre rótt intenzív teher enyhítése végett érzett motivációval terveztem meg egy olyan aktív, automata támogató rendszert, ami képes felismerni akár a járművezető előtt adott forgalmi helyzetben a jármű haladását akadályozó tényezőket. A folyamat során egy kamera színes képét használom fel, mely a jármű elejére van rögzítve a haladási iránynak megfelelően. A rendszer meghatározza a haladás útvonalát és a környezetében fellépő releváns objektumokat, melyek akadályt képezhetnek.

Melyek azok az akadályt képző szituációk, amikre folyamatosan figyelni kell a villamossal történő közlekedéskor? Érdemes megkülönböztetni két csoportot. Az egyikbe soroljuk azokat az ürszelvény részét képző elemek (például sínpálya és részei, felsővezetékhalózat) meghibásodását, melyek gátolják a továbbhaladást. Ide soroljuk példának okául a síntörést⁵, mely járművezetői szemmel is csak nagyon nehezen, közelről vehető észre. Az imént említett kategóriába sorolható akadályt képző jelenségek vizsgálatával nem foglalkozik a rendszer. A másik csoportba sorolandók azon helyzetek, melyek a közlekedésben résztvevő személyek/eszközök helyzetváltoztatásának következtében történnek. Ilyen szituációnak kell tekinteni a hibás parkolást, amikor a jármű egy része az ürszelvénybe ér vagy akár a hibás sávtartást, mely során egy álló kocsisorban tartózkodó gépjármű egy része átlóg a saját sávján túl az ürszelvénybe. Ezen forgalmi helyzetek csúcsidőben rendszeresen jelentkeznek, kiszűrésük nem könnyű feladat. A járművezető munkáját támogató rendszer az ilyen és ehhez hasonló helyzeteket célzott felismerni és megállapítani az akadály jelenlétét.

⁵ Síntörésről akkor beszélünk, ha a sínszál felépítésében sérülés keletkezik és annak hatására a sínszálban anyagfolytonossági hiba lép fel.

2. HASONLÓ KUTATÁSOK ÉS FEJLESZTÉSEK

Rendszertervezés során szükséges megvizsgálni a célterületre szánt hasonló fejlesztéseket, melyek stabil kiindulási alapot nyújthatnak a fejlesztés során. Mivel a közúti vasúti közlekedés egy nagyon specifikus terület, ezért a problémámra konkrét megoldásokat nem sikerült felfedezni publikációk terén. Érdekes más irányból megközelíteni a problémát, miszerint a közúti vasút a vasutak egyik alkategóriájába tartozik. A vasút, mint „ősosztály” témakörben viszont fellelhető számos olyan megoldás, melyek részben lefedik az eredeti problémát. A továbbiakban pár nagyon hasonló megoldást nyújtó fejlesztést fogok ismertetni, illetve a végén értékelem azokat. Bemutatásra kerül, mely technikákat érdemes átvenni a tervezés során.

2.1. A PROBLÉMA MEGKÖZELÍTÉSE

A Sensors folyóiratából egy 2021-es cikk⁶ [3] összefoglal számos olyan fejlesztést, melyek az akadályészlelést hivatottak megvalósítani. A cikk kiemeli és kategorizálja a fejlesztések során alkalmazott módszereket, melyek alapján vizsgálom az akadályészlelés megvalósításának lehetőségeit. Nulladik lépésként felbontom az akadályészlelést különböző részproblémákra.

Az első ilyen részprobléma a vasúti pályának meghatározása a szenzor bemenetén. A szenzor esetünkben egy kamera, mely színes képet ad. A perspektív nézet feldolgozásának két fő megközelítése van. Az első egy geometriai problémára következtethető vissza, mely során a vasúti pálya karakterisztikáit vizsgáljuk perspektív nézetből. A pályával kapcsolatos ismereteket felhasználva lehet tudni, hogy a sín párok közti távolság fix, előre kalkulálható a képen. Felülnézeti transzformáció során (idegen nevén „bird’s eye view”) a sín párok párhuzamosan haladnak. Kamera szemszögéből a sín párok egy távlatpontba konvergálnak. Ezen ismeretek felhasználásával meghatározható a pálya elhelyezkedése, melyet a 3.2 fejezetben ismertetek.

Egy másik megoldást nyújt az úgynevezett neurális hálózatok használata. A neurális háló képes olyan összefüggések felismerésére, melyekből meghatározható a vasúti pálya elhelyezkedése. A neurális hálózat pontossága részben függ, hogy milyen adatokkal végzik el a tanítást. Jelen esetben ezek az adatok nem mások, mint a vasúti pálya fekvését meghatározó képek. Ahhoz, hogy hatékony legyen egy ilyen rendszer, a tanításhoz megfelelő minőségű és

⁶ „A Review of Vision-Based On-Board Obstacle Detection and Distance Estimation in Railways” [7]

menyiségű tanítómintát kell biztosítani, melyek a lehető legszélesebb körben képesek lefedni a problémateret. Az adatoknak változatosnak kell lenniük, melyek kivitelezése nehezen valósítható meg. A neurális hálózattal történő pályafelismerés ötletét elvetem a tanításhoz szükséges adatok hiányában, valamint azért, mert villamosközlekedés esetén a pálya környezete sokkal változatosabb lehet.

Akadályészlelés során elengedhetetlen az akadályt képző, vagy képezhető objektumok azonosítása. Miután megtörtént a pálya meghatározása, az észlelés problémája leszűkíthető a pálya körüli területre. Az egyik lehetőség, hogy a sínszálak mentén keresek akadályokat olyan képfeldolgozási technikák segítségével, melyek a sínszálakat alkotó élek változását vizsgálják. Ez a technika nem terjed ki a teljes úrszelvény vizsgálatára, mely során az akadály nem feltétlen a sínszálon fekszik, hanem az úrszelvény szélén, ezért ezt a megoldást elvetem. Másik lehetőség a mozgó objektumok vizsgálata az előzményképek segítségével. Előnye, hogy ha egy objektum egy korábbi képkockához képest elmozdul, úgy az aktuális és az előző képkocka különbségén a változás megjelenik. A probléma, hogy a technika álló kamera képén alkalmazható hatékonyan. A villamosra rögzített kamera nem nyújt stabil képet a pályahibák okozta mozgásokról és kamera képének vonatkoztatási rendszeréből adódóan, mely központjában a mozgó villamos áll, így a kép csak a villamos álló helyzetében lenne használható, ezért a technikát szintén elvetem. A konvolúciós neurális hálózatok egy olyan adaptív megoldást nyújtanak, mely során az objektumok meghatározása a hálózat felismerő képességén alapul. Napjainkban számos olyan, széles körben elterjedt neurális hálózat létezik, melyek a közlekedésben résztvevő emberek és eszközök azonosítását teszik lehetővé.

2.2. HASONLÓ RENDSZEREK ISMERTETÉSE

Egy hasonló rendszert a 2017-es IMECS konferencia, „*Video Analysis for Improving Transportation Safety: Obstacles and Collision Detection Applied to Railways and Roads*” [4] cikkében ismertettek, mely során gépi látással dolgoztak fel a vasúti járműre rögzített színes kamera képét. Az ötlet azon alapszik, hogy számos költséges szenzor csak rövid távon képes hatékonyan érzékelni akadályt, ezért a képi megközelítés egy stabilabb távpontot nyújthat. A rendszer megoldást ad a vasúti pályán elhelyezkedő akadályok észlelésében. Az akadály észlelés a pályaaazonosítás során történik. A pályaaazonosítást szintén egy geometriai problémára következtetik vissza, viszont nem alkalmaztak felülnézeti transzformációt. Előismereteket felhasználva következtetéseket vontak le, hogy a sínszálak közti távolságnak konstansnak kell lenniük. Ebből következik, hogy a sínszálak perspektív nézetben egy pontba konvergálnak. A

sínszálak kezdeti pontjai a kép alsó szegmensében nem mozdulhatnak el. A megvalósított rendszer az alábbi lépésekkel dolgozik; elsőnek elvégzi a képkocka előfeldolgozását, mely során Canny [5] éldetektorral meghatározza a képen szereplő éleket. Ez követően azonosítja a megfelelő sínszálak kezdő pontjait, melynek az alapját a sínszálak közti konstans távolság biztosítja. Csúszó ablakpár segítségével a kép alsó szegmensén végig haladva lokalizálják a sínszálakat alkotó éleket. Ezután iteratív módon létrehozzák a pályát vizsgáló ablakokat⁷ a kép aljától a távlatpont felé. Segítségükkel nem csak a sínpárok elhelyezkedését határozzák meg, de minden egyes ablakban külön vizsgálják, hogy szerepel-e akadály. A vizsgálat alapját az ablakban szereplő élek kapcsolata adja, miszerint, ha a sínszálak mentén nem található akadály, akkor az élek az ablak alsó részétől a felső részéig megszakítás nélkül futnak. Ha az ablakban lévő intenzitásösszeg érték átlaga kevesebb, mint az adott küszöbérték, akkor azt törésnek veszik és az ablak további vizsgálatot igényel akadály gyanánt. Ha a töréspont konzisztens hossza meghaladja a sín 20%-át, akkor ott akadály észlelhető. Akadály esetén az ablakok további felépítését nem folytatják tovább, az iteráció megszakad. Ellenkező esetben az iteráció a távlatpont vizsgálatában ér véget. A vágányok közti akadályok vizsgálata során az ablakok közti szakaszokat veszik figyelembe, melyeken az előzmény képkocka segítségével a háttereket kivonják egymásból. A rendszer tesztelését követően megállapították, hogy a tesztvideón elhelyezett mind a 10 akadály azonosítása sikerrel járt. A rendszer szenzitivitása 100%, fals negatív azonosítás nem történt, ezért a rendszer biztonságosnak tekinthető. Viszont történt számos fals pozitív azonosítás is, mely során olyan objektumokat detektált a rendszer, melyek valójában nem voltak jelen a tesztkörnyezetben.

Hasonló megközelítés pályadetektálásra, a 2010-es Szingapúri ICARCV⁸ konferencián megjelent „*Obstacle-Free Range Determination for Rail Track Maintenance Vehicles*” [6] cikkben olvasható. A rendszer távolságtartás biztosítására szolgál a konvojban közlekedő vasúti pályakarbantartó járművek között. A pályaaazonosítás alapját Hough-transzformáció [7] nyújtja, mely során a képen lévő éleket egy polárkoordináta-rendszerbe képzik le. Első lépésként meg kell határozni a vasúti pálya, kamerához közeli szegmensét egy trapézzal, aminek négy sarokpontjával számítják ki azt a transzformációs mátrixot, melynek segítségével képzik le a felülnézeti képet. Az eredményképen a vasúti sínpárok már párhuzamosan haladnak, konstans távolsággal egymástól. Az éldetektálás Canny operátor használatával történik. Az operátor küszöbértékeit egy algoritmus határozza meg, mely során a magas küszöbértéket addig

⁷ „Bottom-up windows”

⁸ International Conference on Control, Automation, Robotics and Vision (ICARCV)

csökkentik, amíg nem lesz elegendő mennyiségű él a képen, hogy elvégezhető legyen a Hough-transzformálás. A pálya meghatározása ezután letről felfelé történik, iteratív módon létrehozott ROI⁹ ablakokban. Minden iteráció során azon élpárokat keresi az algoritmus, melyek az előző iterációban meghatározott sínszalak irányszögéhez hasonló iránnyal bírnak és az élek közti távolság konstans. Akadályészlelés során az élek gradiens irányát vizsgálják. Ha a gradiens irányja nem merőleges a sínpárhoz tartozó élek irányára, akkor a vizsgált ROI területét akadállynak veszik.

Akadályészlelés terén sajátos megközelítéssel álltak elő a 2016-os IEEE ICIP¹⁰ konferencián megjelent „*Moving camera background-subtraction for obstacle detection on railway tracks*” [8] cikkben. A megalkotott rendszer a nagyvasúti közlekedésben történő akadályészlelésre javasolt egy megoldást, mely az előzményképek felhasználásával próbálja lokalizálni az objektumokat. Ahhoz, hogy előzményképeket tudjanak felhasználni az aktuális képpel történő összehasonlításhoz, pixel szinten kell hozzáigazítani a referencia képet az aktuálishoz. Egy mozgó szerelvényre rögzített kamera képét fontos térben és időben is egymáshoz igazítani. A probléma megoldására a rendszer elsőnek az aktuális képhez hasonló referencia képet keres a korábban rögzített képkockák közül. A referencia képen ezután elvégzi minden pixel szerint az igazítást, mely során úgy mond ráigazítja az aktuális képre a referencia képet. Végül a két kép különbségét képzik és az akadályészlelést elvégzik a kimenet vizsgálatakor.

Az elmúlt években nagy népszerűségnek örvendenek a neurális hálózatokkal használata különféle objektumdetektáló problémára. A 2019-es ITSC konferencián bemutattak egy olyan konvolúciós neurális hálózatot, mely célja a nagyvasúti környezetben jelentkező általános akadályt képző objektumok detektálása. A „*Real-time Obstacle Detection Over Rails Using Deep Convolutional Neural Network*” [9] cikkben bemutatott rendszer a Single Shot Multibox Detection (SSD) [10] és a You Only Look Once (YOLO) [11] sikereire építve nyújt egy hasonló, ám a célterületre fókuszáló problémamegoldást. A rendszer viszonylag kis adathalmazon lett tanítva és tesztelve, mely 5107 képet tartalmaz tanításra és 1277-et tesztelésre. A modellt két összetevőre lehet bontani: a fő hálózatra, illetve egy jellemzőket egyesítő összetevőre. A hálózat fő része felel a képen szereplő jellemzők kinyeréséért. A jellemzőegyesítő réteg két különféle konvolúciós kernelből áll, melyek feladata, hogy kinyerje az alacsony és magas szintű jellemzőket, majd egyesítse őket. Utolsó lépésként a nem-

⁹ Region of interest (ROI)

¹⁰ International Conference on Image Processing (ICIP)

maximumok elnyomásával¹¹ (NMS) kiválasztják a magasabb konfidenciaszinttel bíró behatároló téglalapokat. A rendszer tesztkörnyezetben kimagasló eredményeket ért el 91,61%-os pontossággal és 26,3 FPS¹²-sel, míg a YOLO 75,67%-kal és 28 FPS-sel teljesített.

2.3. RENDSZEREK ÉRTÉKELÉSE

A bemutatott fejlesztések mindegyike valamilyen sajátos megközelítésben nyújt megoldást a pályadetektálásra, vagy éppen az akadályok azonosítására. A pályadetektáló alrendszer felelős a vasúti sín párok azonosításáért a fényviszonyoktól függetlenül. A fényviszonyok az adott napszaknak megfelelően és az időjárási jelenségek tükrében változnak. Hóval vagy esővízzel fedett pályaszakaszok lokalizálása kimondottan nehéz feladat. A hasonló rendszerek ezt a problémát nem, vagy csak részben érintik. Nagyvasúti környezetben a pályát nem fedi esővíz vagy hó, a magasvágány egyértelműen kitűnik, melynek meghatározása jelentősen könnyebb az úttestbe épített pályával szemben. Havas nagyvasúti pályára a [6] cikkben látható példa. A városi környezetet nem fedik le ezen rendszerek, ezért a probléma hagyományos képfeldolgozási technikákkal nagyon nehezen oldható meg, az eltérő időjárási jelenségekben tesztelt rendszerem eredményeit az 4.1. fejezet végén ismertetem. Felmerül a kérdés, hogy a hasonló rendszerek mégis milyen javaslattal bírhatnak a pályadetektálás és akadályészlelés megoldására. Továbbiakban ezen szempontok szerint vizsgálom részletesen a megvalósítás lehetőségeit.

A[4] cikkben bemutatott rendszer a hagyományos gépi látás technikáit alkalmazva lokalizálta a sín párokat, közben akadályt keresve a sínszalak adott környezetében és a sín párok között. A hagyományos gépi látás előnye, hogy kiforrott technikák állnak rendelkezésre, végrehajtási sebességük előre, szinte pontosan meghatározható és viszonylag könnyen testre szabhatók. Érdeemes megemlíteni a Canny operátort, mely több, egymást követő képfeldolgozási technikával dolgozik. A bemeneti kép elmosását követően a Sobel [12] operátorral létrehozott gradiens kép képes kiemelni a képen szereplő éleket. A nem potenciális éleket a nem-maximumok elnyomásával (NMS) szűrik és csak a valós éleket tartják meg. Kettős küszöböléssel behatárolják a keresni kívánt éleket, melyeket hiszterézises küszöbölés segítségével összekötnek. Az operátor rengeteg erőforrást igényel és a technikák egy része nem párhuzamosítható. Az eredmény képen fals élek is megjelenhetnek, melyek nem kapcsolódnak a valódi sín pár éleihez, köszönhető a nem-maximumok elnyomásának. A kettős küszöbölés

¹¹ Idegen kifejezéssel „Non-maximum suppression” (NMS)

¹² Másodpercenkénti képkockaszám (Frames per second)

értékeinek automatizálása szükséges a változó fényviszonyokhoz való igazodás érdekében, mely egy újabb alrendszer bevezetését igényelné. Számomra, ami érdekes, az a Sobel operátor által létrehozott gradiens kép, mely a fényviszonyoktól függetlenül is elfogadható eredményt tud nyújtani. A Canny operátor bináris képet ad vissza, melyen az élek erőssége elvész és egy esetleges fals él kiemelése nem várt irányba terelheti a sínpálya keresést, melyre a „gyenge” élek megtartása megoldást nyújthat, ám a kettős küszöbölés miatt erre nincs mód.



1. ábra. Bazaltkőkockás pályán alkalmazott Canny operátor előtte és utána

Az 1. ábra szemlélteti, hogy a Canny operátor egy bazaltkőkockás burkolattal ellátott pályán milyen zajos eredményt produkál. Jól látható, hogy nem tesz különbséget a „gyenge” és „erős” élek között. A Canny abban esetben megfelelő választás, ha a küszöbértékek változtatása nem szükséges és a sínpárok szemmel is jó látható módon kitűnnek gyengén zajos környezetben is.

Első meggondolásként mégis csak jó megközelítés volna, ha lehetne párhuzamosakat keresni zajos környezetben. Erre a problémára nyújt megoldást a kép felülnézeti transzformációja, melyre példát a [6] cikkben mutattak be. A megközelítés, miszerint az eredeti képen látható sínpárok felülnézetbe történő transzformálása párhuzamosakat fog eredményezni, kifejezetten jó eredményt nyújt. A probléma a teljesítménnyel van. A projektív transzformálás során az eredményképet interpolációs technikával szükséges meghatározni, mivel a mátrixszorzás¹³ eredménye cellánként nem biztos, hogy egész szám lesz. A pixelenkénti interpolálás erőforrásigényes¹⁴, mely során elképzelhető, hogy a koordináták leképzése többször is azonos eredményt fog adni. Egy párhuzamos számítás során, ha több szál is ugyan

¹³ A perspektív transzformációs mátrix és a bemeneti kép pixelei alkotta mátrixának szorzata.

¹⁴ OpenCV implementáció CPU-n és GPU-n egy GTX 1080-as GPU-val egyaránt ~12 ms végrehajtási idővel rendelkezett a tesztkörnyezetben.

azon eredménycellába kívánja a számítás eredményét beírni, úgy versenyhelyzet alakul ki, mely hosszabb számítási időt eredményez a transzformálás során. Egy korai pályadetektáló megvalósítás olvasható a 4.1. fejezet végén, mely implementálja az említett felülnézeti számítást. Az 1. táblázatban összesítésre kerültek a [4] és [6] rendszer előnyei és hátrányai.

Cikk (rendszer)	Pályaazonosítás módja	Előnye	Hátránya
<i>Video Analysis for Improving Transportation Safety: Obstacles and Collision Detection Applied to Railways and Roads</i> [4]	Canny operátor eredményét csúszó ablakok segítségével elemzi	Viszonylag gyors, Canny eredménye egyértelműen meg tudja határozni a pályát	Canny küszöbértékei fényviszonyoktól és pálya körüli zajtól függenek
<i>Obstacle-free range determination for rail track maintenance vehicles</i> [6]	Felülnézeti transzformáció után Hough operátor segítségével azonosít	Precíz pályaazonosítás, fényviszonyok nehezen befolyásolják	Felülnézeti transzformáció és Hough operátor lassú művelet

1. táblázat. Hasonló rendszerek összegzése pályaazonosítás módja szerint

Akadályészlelés terén részletesen ismertettem a [4], [8] és [9] cikkekben használt módszereket, melyek az ismertetett problémákra optimális megoldást nyújtottak. Esetemben szükséges megvizsgálni, hogy a megközelítések mennyire tekinthetők sikeresnek objektum azonosítás során. A [6] cikkben bemutatott pályaazonosítási módszer részben tartalmazott minimális akadályészlelést is, mely során a gradiens irányát vizsgálták a sínpálya mentén. Esetemben akár az útburkolatba épített pályát is rengeteg zaj veheti körül, melyeket leginkább az öreg burkolat mentén megjelenő repedések és felületi hibák okozzák, ezért a gradiens iránya többszöri változást fog eredményezni. Így a [6] módszer nem jól alkalmazható és csak magára sínre terjed ki, nem pedig az úrszelvényre. A [4] komplex rendszer nem csak a sínszál kisebb környezetét vizsgálja, nevezetesen a kereső ablakok területét, hanem a sínszálak közti szakaszt is figyelemmel kíséri. Mint korábban a [6] rendszer kapcsán is felvetettem, a pálya közvetlen környezetében lévő zaj fals eredményt adhat a keresőablak vizsgálatakor, ezért érdemes nem a keresőablakot vizsgálni, hanem a teljes képet potenciális akadályok kapcsán.

Az úrszelvény és környezetének vizsgálatához kézenfekvőbb megoldást nyújt a [8] és [9] rendszere. A [8] cikknél ismertetett előzményképek felhasználására alkalmas módszer futás ideje 10 FPS-re tehető és számos esetben fals negatív eredményt mutatott, melyeket a képkülönbséget képző modul okozta a szerzők szerint. Egybefüggő objektumok problémát

jelenthetnek szegmentáláskor, mely egy álló kocsisor esetén nem előnyös. A rendszer megközelítését leginkább a nem valós idejű végrehajtási sebessége miatt vetem el.

A neurális hálók térnyerésének köszönhetően optimális megoldást nyújtana számomra a [9] cikkben ismertetett rendszer. A vizsgálatot a hálózat nem csak az úrszelvényre, de az egész képre kiterjesztve végzi. Az eredmény pontossága a hálózat tanításának menetétől és a felhasznált tanítómintától függ. A hálózat leírását mellékelték a cikkben, ami szerint reprodukálható, de mivel a tanítóminta egyedi, publikusan nem elérhető és a hálózat helyes paraméterekkel történő tanítása nagyon sok időbe telhet, ezért nem kivitelezhető a rendszer implementálása.

Cikk (rendszer)	Objektum detektálás módja	Előnye	Hátránya
<i>Video Analysis for Improving Transportation Safety: Obstacles and Collision Detection Applied to Railways and Roads</i> [4]	Keresőablakok és köztes szakasz vizsgálata gradiens intenzitás alapján	Gyors, fényviszonyoktól független	Csak a sín és sínszálak közti területet vizsgálja
<i>Obstacle-free range determination for rail track maintenance vehicles</i> [6]	Keresőablakon belüli gradiens irányok vizsgálata	Gyors, fényviszonyoktól független	Csak a sínszálakra terjed ki a vizsgálat
<i>Moving camera background-subtraction for obstacle detection on railway tracks</i> [8]	Előzményképek és aktuális kép különbségével végzett szegmentálás	Nagy távolságra alkalmas, fényviszonyoktól független	Lassú, szegmentálás pontatlanságot eredményez
<i>Real-time Obstacle Detection Over Rails Using Deep Convolutional Neural Network</i> [9]	Neurális hálózat segítségével	Relatív gyors, teljes képet vizsgálja	Hosszú tanítási idő, tanítóminta nem publikus

2. táblázat. Hasonló rendszerek összegzése objektumdetektálás alapján

A 2. táblázat eredményeiből következtetve a neurális hálózat használata tűnik a legjobb megoldásnak objektumdetektálásra. A 6. fejezetben részletesen tárgyalom, hogyan oldottam meg az objektumdetektálás problémáját valós időben. Az 1. táblázatban feltüntetett pályadetektálási technikák csak részben nyújtanak megfelelő lehetőségeket. A [4] cikkben használt Canny operátor részfolyamatát képző gradiens kép előállítás egy jobb éldetektálást biztosít fényviszonyok, zaj és párhuzamosítás tekintetében, illetve a csúszó ablakok felhasználhatóak az egyes pályaszakaszok azonosítására, melyet a 3.2.2 fejezetben ismertetek.

3. RENDSZERTERVEZÉS

Egy kamera képén történő akadály megállapítása összetett gépi látással kapcsolatos problémaként értelmezhető, mely során olyan technikákat kell alkalmazni, ami a kép hatékony feldolgozását és értelmezését teszi lehetővé. Az algoritmusnak optimális megoldást kell nyújtania a környezeti hatásoktól függetlenül, nem kockáztatva a végrehajtási sebességet. A következőkben ismertetem az rendszer moduljaival szembeni követelményeket, az alkalmazandó technikák kiválasztását, egyes modulok működését és végül a vázlatos rendszertervet.

3.1. MODULOK ÉS KÖVETELMÉNYEK

Az akadályészlelés összetett probléma, melyet kisebb alproblémák együttese határoz meg. Két alproblémát kell megvizsgálni; a pálya azonosításának problémáját és az objektumok detektálását. A kettő eredményét felhasználva azonosíthatók az akadályok. A program futása során a pálya azonosítás és objektum detektálás két külön folyamatként jelennek meg, melyek közötti kapcsolat egyedül a bemenet, ezért azt lehet mondani, hogy az egyes folyamatok függetlenek egymástól, felmerül a párhuzamos végrehajtás lehetősége. Az implementálás során azonban nem alkalmazok modul szintű párhuzamosítást, melynek okát a 4.3. fejezetben részletezem. A rendszer iteratívan dolgozza fel a kamera képeit, ezért cél a lehető legkisebb iterációs idő elérése, úgy, hogy a feldolgozás minőségének rovására ne menjen.

Szükséges meghatározni az egyes modulokkal szembeni követelményeket. Minden modul esetén elmondható, hogy törekedni kell a legjobb teljesítmény elérésére, biztosítani kell a valós idejű feldolgozást. Általánosságba véve valós idejű feldolgozásról akkor beszélünk, ha az algoritmus futási ideje megegyezik a forrás (bemenet) által szállított képek sebességével. Esetemben a forrást a kamera jelenti, ezért azt lehet mondani, hogy valós idejű feldolgozási sebességgel bír a rendszer, ha a kamera képeit legalább olyan vagy annál nagyobb sebességgel képes feldolgozni, mint ahogy az közvetítve van a rendszer felé.

A pályadetektálás és objektumdetektálás során függetlenedni kell az időjárási jelenségek és napszak tekintetében vett fényviszonyoktól. A két modul közti kapcsolat egyedül az azonos bemenet jelentheti, függőségek nem lehetnek köztük. A pályadetektálásnak eredményül olyan pontthalmazt kell visszaadnia, mely képre vetítve egyértelműen meghatározza a villamos pálya ürszelvényének elhelyezkedését. Az objektumdetektálásnak vissza kell térnie az észlelt objektumok gyűjteményével, melyek a képre vetítve egyértelműen meghatározhatóak, mind a

pozíciójuk, mind a méretük. Az érzékelt objektumokat típusuk szerint eltérő módon vizsgálom, ezért az objektumok klasszifikálása szükséges. Visszatérésként továbbá szükséges az objektum típusa is. Az akadályészlelő modultól elvárt, hogy a két modul eredményét felhasználva azonosítsa az űrszelvényben érzékelt objektumokat és visszatérésül az akadályt képző objektumot szolgáltatassa.

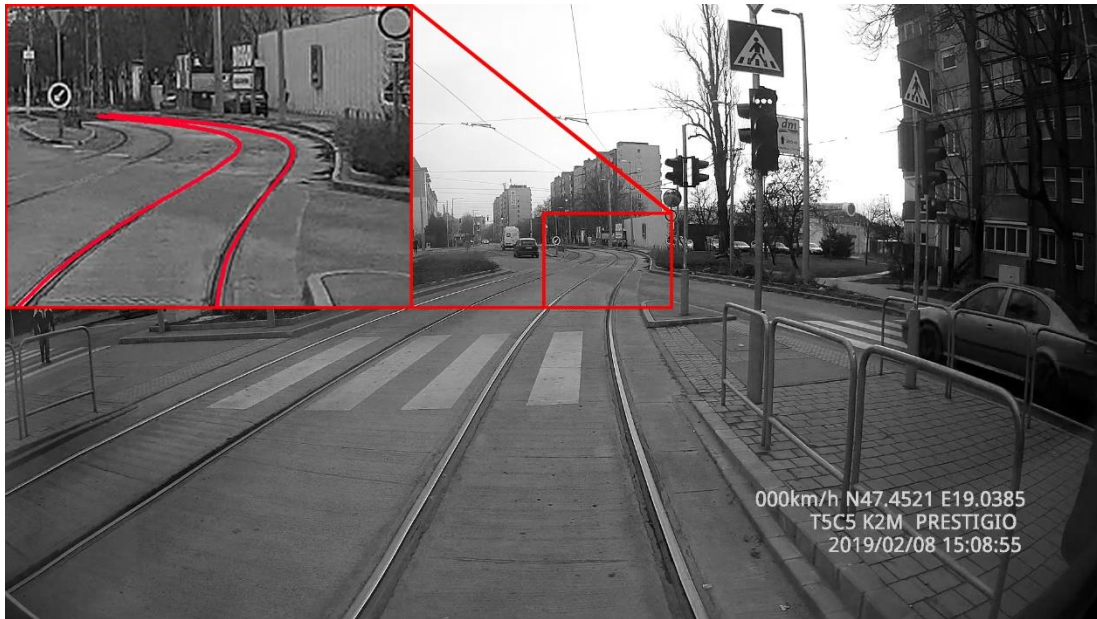
3.2. VASÚTI SÍNPÁROK AZONOSÍTÁSA

A pályadetektálásért felelős modul feladata, hogy a vasúti sínpárokat kamerától mérve a lehető legnagyobb távolságban azonosítsa. A probléma megfeleltethető egy képfeldolgozási és egy gépi látási feladatnak, mely során a digitális kép tartalmát különböző eljárások segítségével manipulálom úgy, hogy a végeredmény alkalmas legyen a sínpárok egyértelmű meghatározására. Ahhoz, hogy az algoritmus képes legyen meghatározni a pálya elhelyezkedését, szükséges definiálni melyek azok a képi jellemzők, amik leszűkítik a keresési teret. Ezen jellemzők meghatározásához felhasználhatók a korábban szerzett vasúti pályával kapcsolatos ismeretek.

3.2.1. ELŐISMERETEK FELHASZNÁLÁSA

A villamospálya számos jellemzővel bír, melyek alkalmassá teszik a környezetből való kitűnésre. A következőkben ezen vizuális jellemzőket fogom ismertetni, illetve, hogy milyen jelenségek befolyásolják a jellemzők „jóságának” mértékét.

Elsősorban elmondható, hogy a vasúti pálya részét képző sínszálak közti távolság konstans, melyet a szerelvény abroncsai közti távolság határoz meg. A pálya ezen tulajdonságára viszont nem minden esetben lehet számítani, mely az adott nézőponttól függ. Ha felülnézetből tekintünk le a pályára, a sínszálak párhuzamos módon haladnak egymás mellett minden esetben. Ha a vezetőfülkéből történik a szemlélet, akkor látható, hogy a sínszálak egy pontba tartanak (vanishing point), mely a horizonton helyezkedik el. A sínszálak közti távolság ebben az esetben is konstans, viszont ezt az értéket a szemlélőtől való pálya távolságához kell kötni. Fizikailag minél távolabb esik a vasúti pálya, annál kisebbnek látni a sínszálak közti távolságot. A pálya ezen tulajdonságára lehet építeni, miszerint a pálya és a szemlélő közti fizikai távolsághoz rendelhető egy változót, mely meghatározza a sínszálak közti távolságot. Ezt a változót egy távolságfüggvény írja le, mely bemenete és kimenete közt arányosság áll fenn.



2. ábra. Sínszálak összefutása

Érdemes megemlíteni, hogy a távolságot leíró függvény nem nyújt pontos értéket, ha a pályaszakasz ívet ír le. Az ív a felülnézeti képen mért távolságokat nem befolyásolja, viszont perspektív nézetből a távolságfüggvénnyel meghatározott értéknél is kisebb eredmény mérhető (2. ábra). A jelenség modellezéséhez szükséges bevezetni egy szorzót a távolságfüggvény kimeneti értékére, mely szemlélő és az ívhez tartozó irány közt mérhető szög függvényében csökkenti a kimeneti értéket. A valóságban megfigyelhető, hogy ha kellő távolságra helyezkedik el az ívelt pályaszakasz, akkor a görbület mentén a pálya egy sínszálát fog képezni (2. ábra). A jelenség a szemlélőtől mért ív kezdeti távolságának függvényében jelentkezik. A kép feldolgozása során a későbbiekben figyelembe kell venni ezen jelenséget.

A fentiekből következik, hogy a szemlélő nézőpontjából, azaz a kamera képén a sínszálak kiindulási pontjai a kép alján fix távolságra helyezkednek el egymástól. A pozíciók azonban a szerelvény jellegéből adódóan nem mindig egyértelműek. A kamera látószögétől függően a sínszálak kiindulási pontjai a kép bal alsó szegmensétől egészen a jobb alsó szegmenséig elhelyezkedhetnek. A jelenségre a szerelvény felépítése ad magyarázatot, miszerint minden forgóváz a szerelvény eleje és vége között helyezkedik el, ezért az alváz/kocsiszekrény két vége az ívelt pályaszakaszon való közlekedéskor csak részben helyezkedik el a pálya felett (3. ábra).



3. ábra. Ívelt pályán haladó villamos

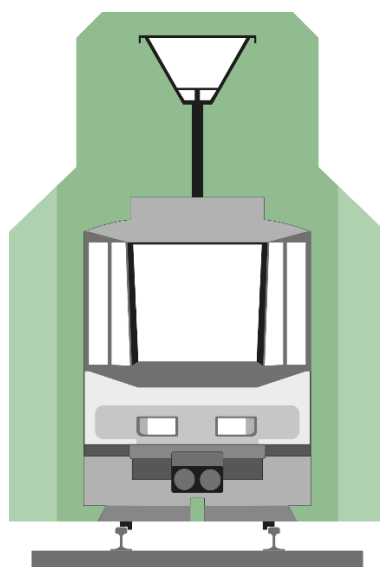
Egy másik jellegzetesség a sínszálak anyaghasználatából adódó fémes/szürke felület. A fémes hatásból adódóan a sínszálak a rávetődő fénysugarak egy részét visszaverik, ezért a felület színértéke az eredeti szín körüli ε tartományban változhat a betekintési szög és fényviszonyok függvényében. A sínszálak szintartományai egyértelműen meghatározzák a pályát, amennyiben a pálya burkolatát képző felület színe eltér.



4. ábra. Nyomcsatorna, tiszta (fent), homokolt (lent)

Egy másik jellemző a sínszálak között kialakított, nyomkarima futását biztosító nyomcsatorna, mely többnyire egy jól látható sötét területként szerepel a képen. A csatorna jellegéből adódóan, a nap állásától függően, egy árnyékolt terület, mely az esetek nagy részében sötétebb a pályaburkolatnál és a sínkoronánál (4. ábra felső kép). A szerelvény csúszásvédelmét képző homokszóró a csatornát képes feltölteni homokkal, mely után a csatorna elveszítheti sötét jellegét (4. ábra alsó kép).

A felsorolt vizuális jellemzők „jóságát” az időjárási viszonyok, mint például az eső vagy a hó képes rontani, ezzel pontatlanabb pályaaazonosítást eredményezve. Míg ezek a jellemzők egyértelműen meghatározzák a sínszálak elhelyezkedését, a gyakorlat során mégis úgy bizonyult, hogy nem lehet rájuk alapozni a detektálás során. A fémes felület esetén a problémát a betekintési szög okozza, mely során a közeli sínszál színe gyakran eltér a messzebb elhelyezkedő szakasztól, illetve sokszor megegyezik az útburkolat színével. A nyomcsatorna a gyakorlat során többnyire olyan mértékben van megtelve homokkal és más szennyeződéssel, hogy nem emelkedik ki kellő képen a sínszál környezetéből. Az említett problémákból kifolyólag nem lehet építeni detektálás során a kép színvilágából adódó jellemzők kinyerésére.



5. ábra. A vágányra merőleges űrszelvény (zöld)

A pálya másik fontos jellemzője az úgynevezett űrszelvény (5. ábra), ami a vágány mellett és fölött szabadon tartandó térnek a vágányra merőleges metszete, melyen belül a jármű a legkedvezőtlenebb körülmények között is biztonságosan elfér [13]. Az űrszelvény „szabadon” tartása kritikus része a közúti vasúti közlekedésnek. Az elsodrások balesetek többnyire az űrszelvényben történő tartózkodásból adódnak, éppen ezért az űrszelvény meghatározása kritikus pontot képez a rendszer megvalósítása során.

3.2.2. PÁLYAAZONOSÍTÁS TECHNIKÁJA

Pályadetektálási folyamat során a bemeneti képen szereplő sínpárok leképzése történik egy ponthalmazba. A folyamat során két egymásra épülő probléma megoldása történik. Az első probléma a kép alkalmassá tétele ahhoz, hogy a pályát leíró részleteket ki lehessen nyerni. Ezt számos alacsony szintű feldolgozási eljárás segítségével valósítható meg. A második probléma az előfeldolgozott állomány értelmezése, mely során az algoritmusnak a pályaleíró pontokat kell meghatároznia a kép pixelei alapján. A következőkben a két problémát megoldó folyamatokat ismertetem.

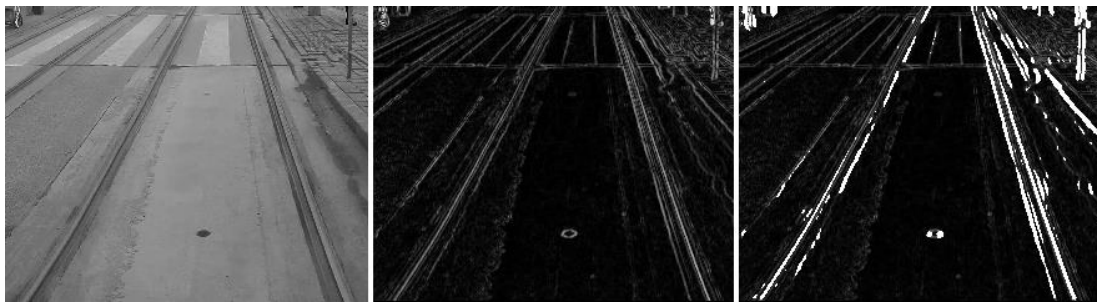
I. ELŐFELDOLGOZÁSI FOLYAMAT

Első lépésben előállítom a kamera képének szürkeárnyaltos változatát. A kamera egy három színcsatornás RGB képet állít elő, melynek pixelértékeit egy $M \times N \times C$ dimenziós mátrixban tárolom, ahol M a kép magassága, N a kép szélessége és C a színcsatornák száma. Szürkeárnyaltos konverzió során a színeket meghatározó csatornákat egyesítem, melynek eredményéül egy $M \times N \times 1$ dimenziós mátrix áll elő, ahol a pixelek értékeinek reprezentálása egy 0-tól 255-ig terjedő diszkrét tartományban történik. A konverzió eredményeképpen a további feldolgozási eljárások komplexitása csökken, ezért jobb futási idő érhető el.

Ahhoz, hogy a sínszalakat egyértelműen meghatározó jellemzőket ki lehessen emelni, elsősorban szükséges az élek meghatározása. A folyamat eltér a klasszikus éldetektáló eljárásoktól olyan tekintetben, hogy magukat az éleket nem címezzük meg, helyette az intenzitásváltozások nagyságának értékeinek meghatározása történik. Képfeldolgozási szempontból maga az él egy jelentős helybeni intenzitásváltozás, azaz a képpontok egy adott környezetre vetített értékváltozásának mértéke. Ezt az értékváltozást reprezentálja a kép gradiensének a nagysága. Minden pixelre vetített intenzitásváltozás mértéke alkotja a gradiensképet, mely alkalmas a különböző élek által meghatározott jellemzők kinyerésére. A gradienskép pontjait konvolúciós eljárás során állnak elő. Konvolúciós kernelekkel dolgozik számos éldetektálásra szolgáló szűrő. G. M. H. Amer és A. M. Abushaala az „*Edge detection methods*” [14] cikkében ismertetett számos éldetektáló technikát. A technikák során az éleket előállító konvolúciós kernelek eredményeit küszöbölték. Az adott küszöbérték feletti éleket vették valós éleknek, illetve további feldolgozások alá vetették. Esetemben minden élre szükségünk van, ezért a küszöbölés előtti állapot előállítása a cél. Két gradiens alapú éldetektáló eljárás jöhet szóba. Az első a *Laplace* [15], a másik a *Sobel* [12] operátor. A Laplace operátor

egy konvolúciós kernellel állítja elő a másodrendű deriváltakat. Hátránya, hogy érzékeny a zajra, ezért célszerű az operátor használata előtt a képet elmosni. A *Sobel* operátor egy X és egy Y kernellel együttesen állítja elő a gradiens képet. Elsőfokú deriválttal történő számítás végett nem érzékeny a zajra, viszont az eredmény előállítására a két kernel miatt lassabb művelet a *Laplace*-szal szemben. A *Sobel* nem tesz különbséget a gyenge és erős élek között, ezért fogom alkalmazni.

A *Sobel* X és Y komponense felhasználható finomhangolás céljából is. A vasúti sínszálak perspektív nézetből egy pont felé haladnak letről felfele, ahol megfigyelhető, hogy a sínszálakat alkotó élek leginkább függőleges irányba dominálnak, melyet a vízszintes szűrést végző kernel érzékel. Ezért a gradienskép létrehozásakor az X és Y komponensek súlyozása szükséges, melyek közül az X komponens nagyobb értékkel bír, ezzel jobban kiemelve a sínszálak függőleges éleit.



6. ábra. Balról jobbra: Eredeti kép, *Sobel* 3×3 , *Sobel* 3×3 és 5×5 kombinálva

A 6. ábra közepén látható a *Sobel* operátor eredménye. Megfigyelhető, hogy a sínszálak körüli intenzitásváltozás mértéke nagyon hasonló értékkel bír, mint a sínszál környezete. A jelenséget a sínszálak gyenge kitűnésével magyarázható. Egy megoldást nyújt a *Sobel* operátor módosítása, mely alatt az eredeti 3×3 -as kernel 5×5 -ös kiterjesztett változatát értem. Eredményképpen egy nagyobb területet bejárása történik, mely során a differenciák nagyobb értéket tudnak felvenni, ezzel felerősítve az erős éleket, a gyengéket ezzel egyidőben elnyomva. A technikát önmagában nem érdemes alkalmazni, az élek felerősítése csak abban az esetben fog megfelelő eredményt nyújtani, ha kellően nagy változás történik a konvolúciós kernel területén. A tapasztalatok szerint útburkolattal ellátott felületen nem biztosít pontos élkimielést, ezért a 3×3 -as *Sobel* kiegészítéseként érdemes használni. Az eredmény a 6. ábra jobb oldalán látható.

Az 5×5 -ös *Sobel* kernel mellékhatása, hogy nem csak a sínszalakat erősíti fel, de a környezetét is, ezzel olyan jelenségeket eredményezve, mint például a zúzottköves felépítmény túlzott megjelenése. A hatás csökkentése érdekében a szürkeárnyaltos képen homályosítást szükséges végezni az operátor használata előtt. Homályosítással, vagy másnéven életlenítéssel kapcsolatban érdemes megemlíteni két szűrőt. Az egyik a *Gauss*-féle operátor, a másik a *Median* szűrő. A *Median* nem lineáris szűrő, mely az ablakban lévő elemek sorrendbe tétele során kiválasztja a tömb középső elemét, mely a szűrés eredményét képezi. A technika célja a zaj csökkentése, miközben bizonyos kondíciók alatt megtartja az éleket. A *Gauss*-féle szűrő egy lineáris, konvolúciós kernelt használó szűrő, mely a *Gauss*-függvényen alapszik. A függvény normáeloszlást követ, ezért a kernel közepétől kifelé haladva az elemek értékei csökkennek. A szűrő biztosítja a kép gyors, egyenletes életlenítését, viszont eredményképpen nem csak a zajt csökkenti, de az éleket is gyengíti. A technikák összehasonlítását bemutatta A. Kumar és S. S. Sodhi a „*Comparative Analysis of Gaussian Filter, Median Filter and Denoise Autoencoder*” [16] cikkében. A tesztkörnyezetben elért eredmények alapján a *Median* szűrő jobb eredményt biztosított kevesebb iterációs idő alatt. Korábbi teszteredményeim alapján a *Median* szűrő valóban képes volt a zúzottkő ágyazat éleit gyengíteni, miközben a sínszalak éleit megtartotta, de csak kedvező fényviszonyok között. A kőkockák anyagukból adódóan kiemelt módon képesek visszaverni az erős fényt, ezért a *Median* szűrő a zajcsökkentés helyett kiemeli az ágyazat nemkívánatos éleit, melyet az 5×5 -ös *Sobel* kernel felerősít és következményképp a sínszalak nem tudnak kellő képen kitűnni. Ez a jelenség a *Gauss*-féle életlenítés során nem tapasztalható jelentős mértékben, ezért a fejlesztés során a *Gauss* életlenítéssel vittem véghez a kép elmosását.

Az előfeldolgozás végén egyesítem a *Sobel* 3×3 -as és 5×5 -ös kernelével alkotott gradiens képet, melynek eredménye a 6. ábra jobb oldalán látható.

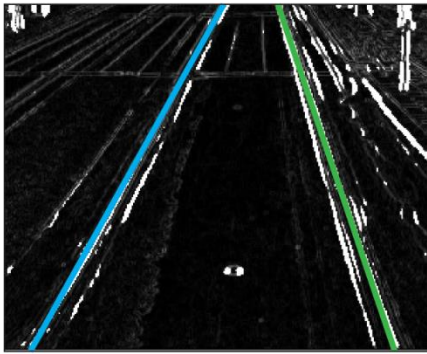
II. PÁLYALEÍRÓK MEGHATÁROZÁSA

Az előfeldolgozási folyamatban a módosított kamera képen kiemelésre került minden olyan erős és gyenge él, melyek szemléltethetik a vasúti pályát. A feltételes kijelentés oka, hogy maga a pálya élerőssége nem csak a sínszálak mentén van jelen, hanem a környezetében is, például a sínszálak alátámasztását biztosító aljak, vagy a bazaltkőkockás úttest mentén, melynek felszíne számos esetben erősebb éllel rendelkezik. Ezeket a nemkívánatos jelenségeket zajnak tekintettem. A zaj kiküszöbölésére előfeldolgozás során a *Gauss*-féle elmosás technikáját vettem igénybe, melyet legfeljebb olyan erősséggel alkalmaztam, hogy magát a sínszálak éleit ne rontsa a felismerhetetlenségig.

A pályaleírók meghatározása során nem konkrét élék keresése történik, hanem élcsoportok vizsgálata. A megközelítésnek két oka van; Az egyik, hogy számos esetben a pálya nehezen vehető ki még szabad szemmel is a képről, ezért gyenge élék formájában jelenik meg. A gyenge élék több esetben is megszakadnak, esetleg zaj formájában jelennek meg a sínszálak körül. Feltételeztem, hogy ahol több gyenge él van jelen, ott található valójában a sínszál. A másik ok, hogy élcsoportokat vizsgálok, visszavezethető az 5×5 -ös kernel egyik problémájára, miszerint csak az erős éleket tartja meg, a gyengéket elveti, mely szintén élszakadások formájában jelentkezik, esetek többségében zajmentesen. A probléma kiküszöbölésére alkalmaztam a 3×3 -as kernel kiegészítésére az 5×5 -öt, mely során az erős élék szakadásait a gyenge élék „feltöltötték”. Mivel a pályát különféle erősségű élcsoportok alkotják, ezért küszöbölést korábban se alkalmaztam élék szűrésére. A *Canny* éldetektor, illetve *Hough* transzformáció használatát ezért is vettem el.

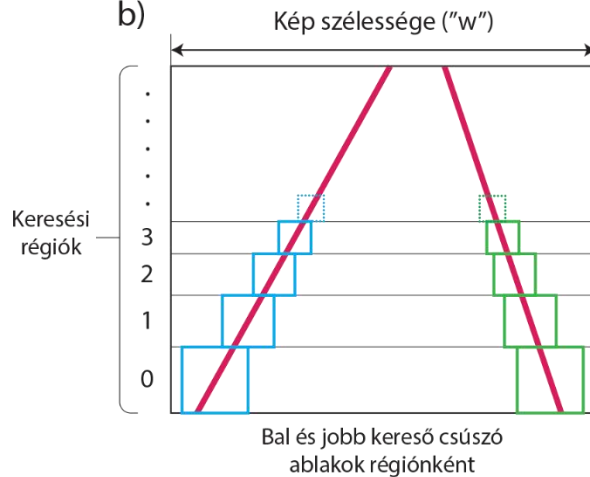
A megfelelő élcsoportokat keresőablakkal lokalizálom, melyekben a gradienskép élintenzitásait összegzem. A keresés alapjául azt vettem, hogy ahol magasabb intenzitásösszeggel rendelkezik a keresőablak területe, ott nagy valószínűséggel sínszál található. A korábban ismertetett távolságfüggvény felhasználásával meghatároztam, hogy milyen magasságban kell keresni a megfelelő sínszálak közti távolságokat. A képet felbontom horizontálisan keresési régiókra és meghatározom a szükséges csúszó ablak méreteket (7. ábra, „b” pont). A régiókon belül előre tudni lehet, hogy milyen régió magassághoz milyen keresett távolság tartozik, melyet később az akadály távolságának meghatározására felhasználok. A régiókon belül a sínszálak keresése párban történnek, fix távolságra egymástól.

a)



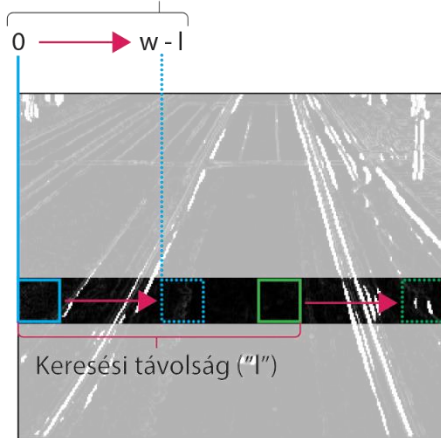
Bal (kék) és jobb (zöld) sínszál előfeldolgozott képen

b)



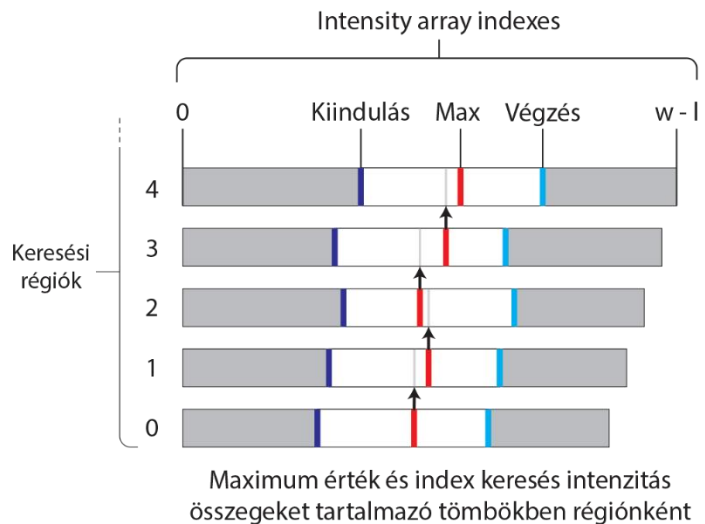
c)

Intenzitás összegeket
tartalmazó tömb indexei



Pixel intenzitás összegzés bal és jobb csúszó ablakon belül adott régió területén

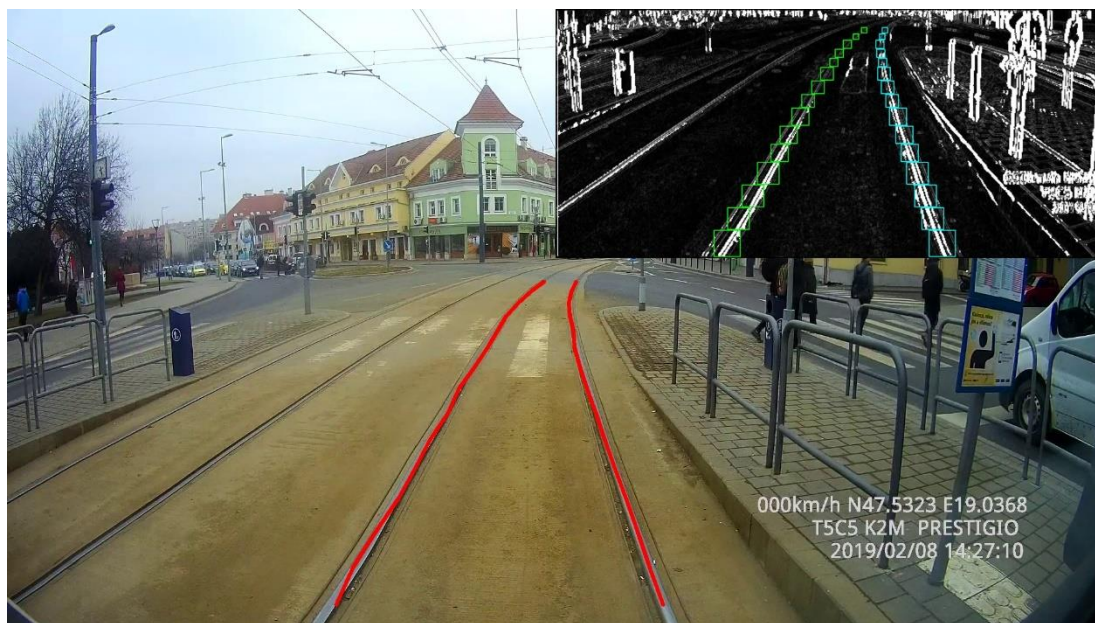
d)



7. ábra. Pályadetektálás lépései előfeldolgozás után

A keresés során a csúszó ablakokat végig csúsztatom a régió területén és intenzitás összegeiket eltárolom (7. ábra, „c” pont). Ahol a két ablak intenzitásösszege a legnagyobb, a vasúti pálya nagy valószínűséggel ott található, ezért egy maximum keresést végzek az intenzitás összeget tároló tömbökben, régióként. A számítási folyamatot iteratív módon kell elvégezni, melynek lépésszáma függ a csúszó ablakok méretétől és a régiók számától (7. ábra, „d” pont). A megtalált maximális értékhez tartozó indexszám jelenti a következő tömbben a keresési terület közepét. Ez egy kifejezetten lassú folyamat tud lenni, ezért a felgyorsításának érdekében az előfeldolgozás végén meghatározom az előfeldolgozott képre vett integrálképet, mely lehetővé teszi az intenzitásösszeg számítását három lépésből. A keresés során a pályát lentről felfele építem fel.

A csúszó ablakokon belül megjelenő gyenge és erős élek néhány esetben hamis eredményhez vezethetnek, melyek a keresést akár teljesen rossz irányba is terelhetik. A probléma kiküszöbölésére egyrészt az előfeldolgozás finomhangolása szükséges, illetve a régiókon belüli keresést egy adott irányba terelni. Az irány meghatározásához az előző– két régió képezte haladási irány fogja megmondani, merre kell folytatni a keresést. A régió belüli keresést korlátozom olyan kisebb területre, mely a csúszó ablak szélességének vett kétszerese feleltethető meg. A korlátozás biztosítja, hogy amennyiben egy vélhetően rossz irányba halad a keresés, úgy vissza tudjon térni az algoritmus a megfelelőnek vélt eredeti irányra. Az előismeretek felhasználása során említésre került, hogy a sínszálak megfelelő nagyságú ívelt pályaszakaszon, horizontálisan egy sínszálként jelennek meg (2. ábra). A probléma kiküszöbölésére a keresés irányának meghatározását követően, az irányszög nagyságától függően a korlátozott keresési területet megnövelem a keresés irányába. Az ismertetett keresési területet a 7. ábra, „d” pontján látható. A megoldás biztosítani fogja, hogy a horizontális megjelenéshez közeledő pályaszakaszra a keresés ki tudjon térni.



8. ábra. Detektált pálya

Sikeres pályaaazonosítást követően a régióként azonosított pályaszakaszok bal és jobb sínszálát meghatározó pontjai visszatérési értéként szolgálnak, melyek egyértelműen behatárolják a pálya területét (8. ábra).

3.3. OBJEKTUMOK DETEKTÁLÁSA

Az objektumdetektáló modul feladata, hogy a képen szereplő objektumokat, alakzatokat felismerje. Elhelyezkedésüket és méretüket olyan formában adja vissza, melyekkel később további számításokat lehet végezni.

Akadályészlelés során szükséges meghatározni, mely objektumok képezhetnek akadályt az űrszelvényen belül. Ezek az objektumok meghatározása történhet a mozgásuk alapján, melyek pixel intenzitás változást okoznak haladásuk során. Az Optical Flow [17] technika ezeket az intenzitás változásokat követi le. A technika fő feladata az aktuális és előző kép különbsége során keletkezett flow vektorok meghatározása, melyek szegmentálását követően képesek meghatározni az objektum alakját, lehetőség nyílik az objektum követésére. A technika hátránya, hogy önmagában csak nem mozgó, fix kamerával történhet objektumdetektálás, hisz könnyen belátható, hogy a kamera mozgása során az egész képre kiterjedő flow vektorok jelenhetnek meg. Egy korábban bemutatott rendszer [8] az Optical Flow technikára építve oldotta meg a mozgó kamerával történő objektum detektálást, ám az említett technikát elvettem, mert a végrehajtási sebesség nem volt számomra megfelelő.

Amennyiben nem mozgásváltozásból adódóan kell objektumot meghatározni, úgy lehetőség van a pixelek kapcsolatainak vizsgálatára. Egy objektumot számos pixeleyüttes határoz meg. Amennyiben emberi szemmel nézzük a képet, úgy bizonyos jellemzők alapján írjuk le mik lehettek fel a képen. Ezek a jellemzők az ember számára egyértelműen meghatározzák a fellelhető objektumokat. Ezeket a jellemzőket lehetőség van kinyerni a képről különböző konvolúciós technikák segítségével. Minden keresett jellemző meghatározható konvolúciós kernelek segítségével, melyek megadják, milyen jósággal rendelkezik az adott jellemző a vizsgált területen. A megfelelő jósággal rendelkező jellemzők közti kapcsolatot neurális hálózatok segítségével meghatározhatók. Azokat a neurális hálózatokat, melyek a képen szereplő objektumok detektálására képesek a különböző jellemzők kinyerésével és közöttük fennálló kapcsolat megállapításával, konvolúciós neurális hálózatoknak nevezik.

3.3.1. KONVOLÚCIÓS NEURÁLIS HÁLÓZATOK

A konvolúciós neurális hálózatokat (CNN) széleskörben használják minta- és képfelismerési problémák során, mivel a hagyományos technikákkal szemben számos előnnyel rendelkeznek. A CNN egy speciális neurális hálózat típus, mely egy vagy több konvolúciós réteggel rendelkezik, almintavételező (subsampling) réteggel párosítva, melyet egy vagy több

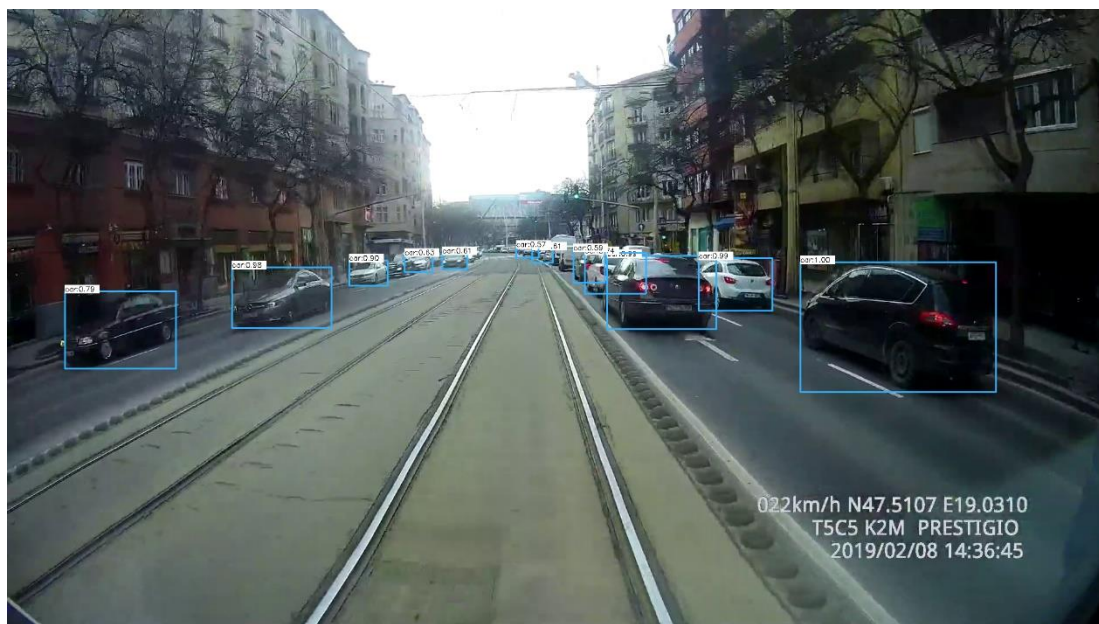
sűrűn összekötött réteggel kapcsolnak össze. A CNN alapját az agy részét képző „látókéreg” ihlette. A konvolúciós rétegek felelnek a jellemzők kinyeréséért. Konvolúció során a kép átalakítását végezzük úgy, hogy egy konvolúciós kernelt (ablakot) helyezünk a pixelekre, melyet végig csúsztatunk a képen. A kernel olyan értékek alkotta mátrix, mely mérete és értékei meghatározzák a konvolúciós folyamat eredményét. A konvolúciós réteg szűrőjének értéke, azaz a súlyait, a hálózat tanítása során állítódnak be. A konvolúciós rétegek képesek kinyerni a helyi jellemzőket, mert a helyi neuronok csak a környezetüket vizsgálják, ami a szűrő méretére¹⁵ korlátozódik. Hagyományos technikákkal szemben a szűrők értékeit nem kézzel állítják be, a tanításhoz használt képeken szereplő objektumok felismerésére fog szakosodni a hálózat. Egy megfelelően kialakított és tanított hálózat képes akár ezer különböző objektum klasszifikálására és helyzetmeghatározására.

A bemeneti és kimeneti réteg között helyezkednek el a rejtett rétegek. A rejtett rétegek állnak konvolúciós és egyesítő (pooling) rétegekből. A konvolúciós rétegek eredménye a bemeneti mátrixhoz méretéhez képest egy kisebb mátrix, mely a konvolúciós szűrők sajátossága miatt van. A konvolúciós réteget egy pooling réteg követi, mely feladata a mátrix dimenziójának további csökkentése. Általánosságban elmondható, hogy míg a konvolúciós réteg felel az objektum jellemzőinek a kinyeréséért, addig a pooling réteg az érzékelt jellemzők összekapcsolását látja el. Végül egy teljesen összekötött réteg felel a felismert objektumok klasszifikálásáért [18].

Képek klasszifikálását követően a következő előrelépést a régió alapú konvolúciós neurális hálózatok [19] (R-CNN) megjelenése jelentette. Az R-CNN szelektív kereséssel generál úgynevezett „javaslatokat”, hogy a képen hol találhatóak a hálózat számára „érdekes” területek, melyek objektumokat tartalmazhatnak. Ezeket a területeket fix méretté konvertálva átadja a CNN-nek, mely egy jellemző (feature) kinyerő funkciót lát el, ami 4096 különböző jellemző előállítására képes. Az eredményt átadja egy SVM klasszifikáló rétegnek. Maga a klasszifikáláson túl négy különböző eltolási értéket is meghatároz az egyes objektumokhoz, melyek együttesen megadják az objektumok körbe fogláló négyzetet. Az R-CNN hátránya, hogy nagyon hosszú végrehajtási idővel bír, mely nem teszi alkalmassá valós idejű alkalmazások használatához. Továbbá pontatlan javaslatokkal térhet vissza a szelektív tanulási képesség hiánya miatt. Ezen hibák kiküszöbölésre továbbfejlesztették az alap modellt, mely Fast R-CNN és Faster R-CNN névre hallgat [20].

¹⁵ Általában 3x3-as szűrő méretet alkalmaznak.

2015-ben az objektumdetektálás egy innovációjaként bemutatták a YOLO-t (*You Only Look Once*), egy egységes, valós idejű objektumdetektáló módszert. Az R-CNN-nel ellentétben a YOLO nem használ javaslatok és klasszifikáció kombinációját az objektumok észlelésére, hanem az objektumokat befoglaló téglalapokat (bounding box) és az osztályba sorolás valószínűségének értékét a pixelekből határozza meg. YOLO bemenetként megkapja az egész képet. Egyetlen konvolúciós hálózattal egyidejűleg határozza meg a befoglaló téglalapokat és az osztályvalószínűségeket. Ez a megközelítés számos előnnyel jár. Először is, leegyszerűsíti az észlelési folyamatot egy regressziós problémává, mellyel sikerül elkerülni az R-CNN bonyolult és erőforrás igényes folyamatait. Másodszor, a régiójavaslat alapú megközelítéssel szemben, a predikciók megállapítása során rendelkezésre áll a teljes kép, mint információ, ezzel csökkentve a háttérhibák számát. Az érem másik oldala viszont, hogy a YOLO nagy valószínűséggel fog pontatlanabb helyzetmeghatározásokat végezni. Kiértékelés során a kis és nagy befoglaló téglalap hibákat egyenrangúan kezeli, mely pontatlan helyzetmeghatározást eredményez. Másodszor, objektumdetektálás során képes figyelmen kívül hagyni néhány objektumot, példának okául mikor számos objektum helyezkedik el egy vizsgált cellán belül. YOLO alapvetően nem tudja a hibát elkerülni, ha minden befoglaló téglalap ugyan abban a cellában helyezkedik el és ugyan azon osztályba vannak sorolva. Idővel megjelentek a YOLO továbbfejlesztett verziói, melyek igyekeznek ezen hibákat kiküszöbölni. Az objektumdetektálásra egy példát szemléltet a 9. ábra.

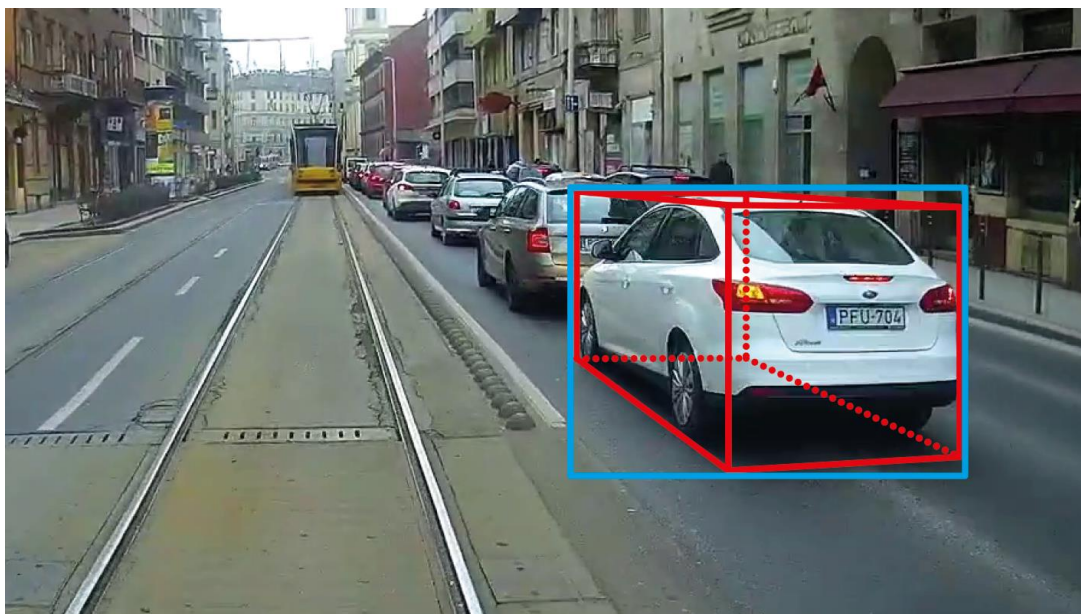


9. ábra. Objektum predikciók

Az objektumdetektálást az ONNX Runtime [21] felhasználásával a YOLO ötödik verziójával [22] (YOLOv5) végzem. Az ONNX Runtime egy nyílt forráskódú projekt, mely különböző keretrendszeren célzott a gépi tanulást gyorsítani. Az ONNX Runtime könyvtára lehetőséget nyújt a YOLO CUDA [23] környezetben történő hatékony implementálására, mely lehetővé teszi a neurális háló GPU-n történő futtatását valós időben. A CUDA az Nvidia vállalat által megalkotott számítási modell és párhuzamos számításokat támogató platform Nvidia GPU-k számára. A CUDA környezet nem csak neurális háló futtatására alkalmas, hanem számos képfeldolgozási technika és egyéb számítás gyorsítására is kiváló lehetőséget nyújt.

3.4. AKADÁLYÉSZLELÉS

Az akadályészlelést megvalósító modul feladata, hogy az érzékelt objektumokat vizsgálja meg olyan szempontból, hogy az azonosított pályából képzett űrszelvényen belül helyezkednek-e el. Ha igen, akkor térjen vissza az akadályt képző objektummal. Ha ránézünk a kamera által rögzített képre, egyértelműen meg tudjuk határozni a vizsgált tárgy helyzetét, melyből következtetésként levonjuk, hogy ránézésre hol helyezkedik például egy autó a pályához képest. Az emberi szem érzékeli a perspektív képen fellelhető mélységeket. Egy gép viszont nem rendelkezik térlátással, szenzorok segítségével határozza meg a távolságokat, melyekből következtetéseket von le. A probléma megoldására egy kamerát alkalmazunk csak, mely önmagában nem képes mélységi adatokat biztosítani. Speciális eljárások segítségével közelíteni lehet a távolságokat. Érdeemes megemlíteni a „DisNet” [24] nevet viselő neurális hálózatot, mely relatíve pontos becslést tud adni a befoglaló négyzetbe foglalt objektum távolságára. A hálózat nem térbeli becslést végez, hanem az objektumot meghatározó jellemzők méretével végez egy közelítést, mely az objektum teljes egészére határoz meg egy távolságot. A feladat megoldására viszont további mélységi adatokra volna szükség, melyet a következőkkel lehet magyarázni; Példának vegyünk egy autót, melyet a neurális háló azonosított és meghatározta az autót magába foglaló téglalapot. Ha a téglalapon belül felvesszük az autó pozícióját magába foglaló térbeli, geometriai testet, akkor egy téglatestet kapunk, melyet a 10. ábra szemléltet.

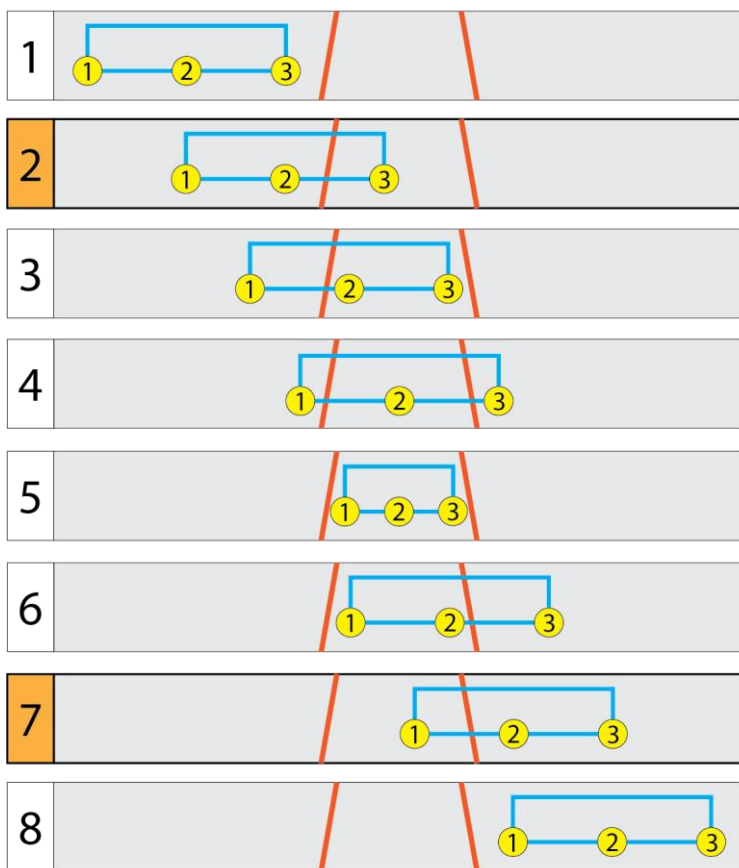
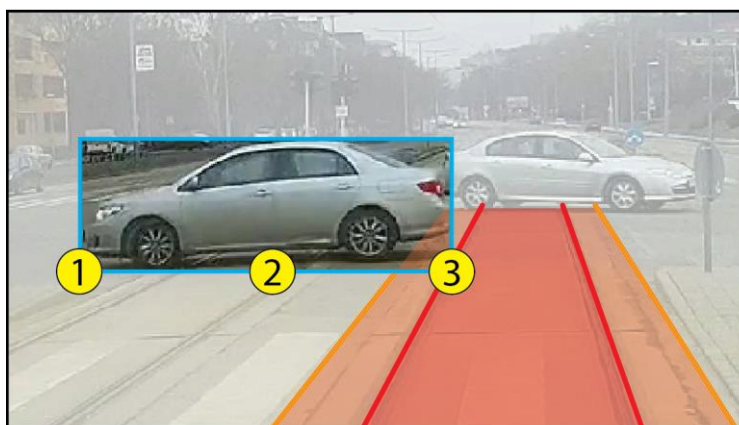


10. ábra. Objektumot reprezentáló téglatest (piros) és 2D befoglaló téglalap (kék)

A 3D hexaéder mérete viszont az autó elhelyezkedésétől függően a 2D képen változik. A 2D-ben reprezentált hexaéder egy poligonnal feleltethető meg, mely pontjait X és Y koordinátákkal lehet reprezentálni. Míg a vasúti pálya, és úrszelvény 3D elhelyezkedését egészen pontosan meg lehet határozni, addig az autót reprezentáló téglatest 3D pontjait nem lehet egy 2D képről meghatározni, ezért a befoglaló téglalap nem fogja megfelelő pontossággal szemléltetni a jármű térbeli elhelyezkedését. A DisNet csak a jármű középpontjának kamerától vett távolságát fogja megbecsülni, mely bizonyítottan nem elegendő az objektum térbeli elhelyezéséhez, ezért a térben végzett akadályészlelés elméletét elvetem. Továbbá érdemes megemlíteni, hogy a perspektív szemlélet okozta torzulás megfigyelhető a hexaéderen, mely során a befoglaló négyzetbe tartoznak olyan területek is, melyek nem képezik az objektum foglalta teret.

A problémát ezért geometriai úton közelítem meg, mely során egy durva becslést végzek az objektum formájára. Az úrszelvény pontjait 2D koordinátákként kezelem, melyek egy poligont alkotnak. A vizsgálat során azt figyelem, hogy az objektum becsült alakja a poligonon belül található-e vagy sem. Ha az objektumot leíró alakzat kis része is a poligon területén belül helyezkedik el, az objektumot akadálynak tekintem.

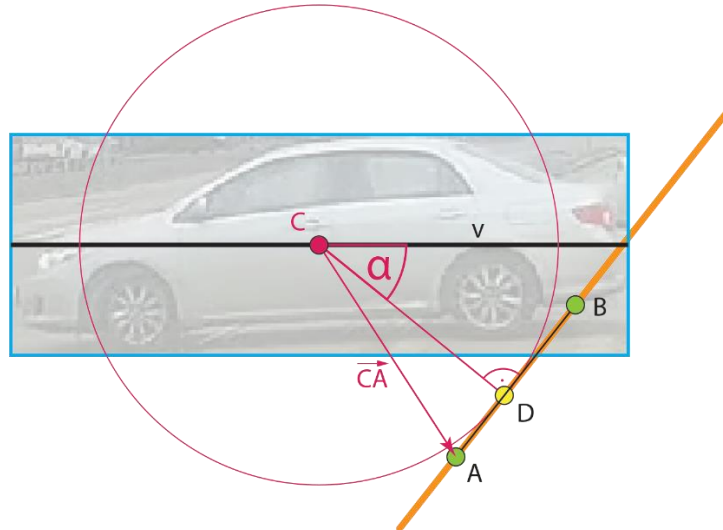
A 10. ábra során megfigyelhető, hogy az autót leíró téglatest párhuzamosai a perspektív nézet miatt egy pont felé, a „vanishing point” felé tartanak. Ez a jelenség felerősödik, minél inkább távolabb helyezkedik el a horizont közepétől az objektum.



12. ábra. Objektum helyzetének esetei az űrszelvényhez viszonyítva

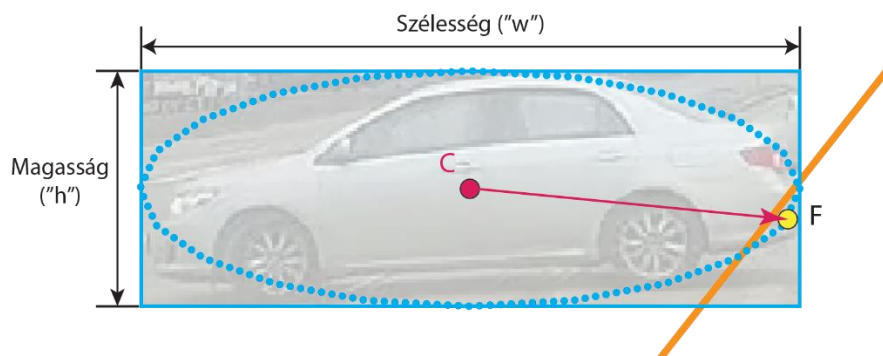
A vizsgálat során definiálok három, az objektumhoz tartozó pontot, melyet a 12. ábra szemléltet. A pontok X tengelyi értékét a befoglaló téglalap két szélé (1 és 3) és a közepe adja (2). Az Y értéket a befoglaló téglalap alsó szakasza adja. Amennyiben további vizsgálatot szükséges folytatni, úgy az Y értékét minden további régió alsó szakasza képezi. Az alapfeltevés a következő; Ha a szélső és középső pontok az űrszelvényen kívül helyezkednek el (12. ábra 1. és 8. eset), akkor az objektum nem képez akadályt, a vizsgálatot nem folytatom tovább. Ha a középső pont (2) az űrszelvényen belül helyezkedik el (3., 4., 5. és 6. eset), akkor

az objektum akadályt képez, a vizsgálatot leállítom és a maradék objektumokat kiértékelésére nem lesz szükség. Amennyiben az egyik szélső pont az űrszelvényen belülre esik (2. és 7. eset, úgy az objektumot megközelítő alakot kell vizsgálni.



13. ábra. Objektum legközelebbi pontjának meghatározása az űrszelvényhez 1/2

A vizsgálat első felét a 13. ábra szemlélteti. Az űrszelvényt a régió határai szerint van értelmezve. Esetünkben a szakasz kezdetét az „A” pont, végét pedig a „B” jelenti. Meghatározom a befoglaló téglalap középpontjából („C”) az „A” pontba mutató CA vektort. Kiszámításra kerül a CA vektor vetülete az AB szakaszra nézve. A „C” pont és vetület végpontja („D”) határozza meg a CD szakaszt, mely merőleges az AB szakaszra. A „C” középpontú, CD szakasz hosszú sugárral bíró kör érintője az AB szakasz, mely egyben meghatározza a legrövidebb távolság hosszát az objektum középpontjától az űrszelvény szakaszig. Az objektum középpontja által meghatározott vízszintes („v”) és a CD szakasz által bezárt α szög megadja az űrszelvény szakaszhoz legközelebb eső pont irányát.



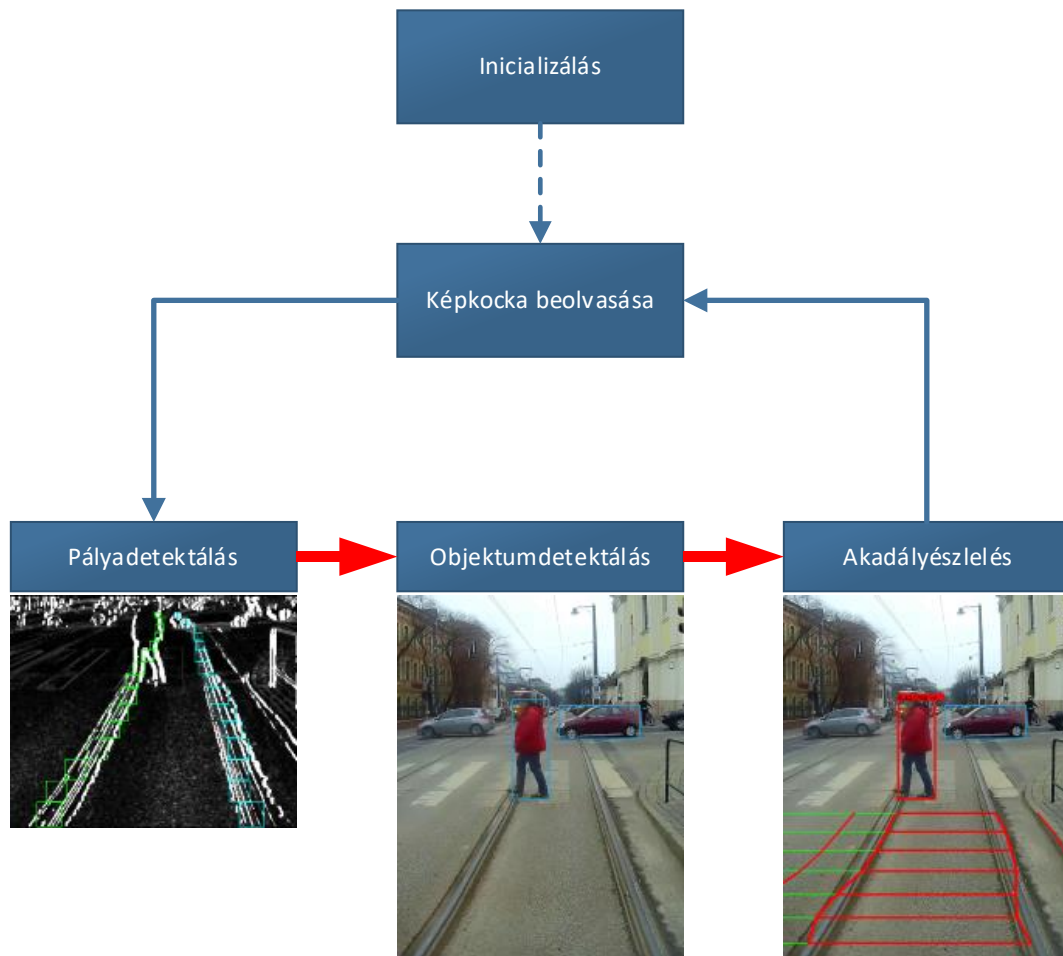
14. ábra. Objektum legközelebbi pontjának meghatározása az űrszelvényhez 2/2

Az azonosítás második felét a 14. ábra szemlélteti. A korábbiakban meghatározott α szög segítségével megkaptam az ellipszisen elhelyezkedő, űrszelvénytárcsához legközelebbeső pont koordinátáit. Ha vesszük a „C” középpontú, $w/2$ sugarú kört, akkor megállapítható, hogy a kör hasonló a korábbi CD sugarúhoz viszonyítva. Ha vesszük a $w/2$ sugarú kört és megpróbáljuk az objektumot befoglaló téglalapon belülre helyezni, akkor egy ellipszist kapunk, mely határai a téglalap szélei (14. ábra, szaggatott kék vonalú ellipszis). A kapott ellipszis feleltethető meg az objektum alakját meghatározó közelítésének, mely a 3.4 fejezetben került ismertetésre. Az ellipszisen elhelyezkedő, űrszelvénytárcsához eső legközelebbi pont („F”) koordinátáit a következők szerint számítom (14. ábra): Az „F” pont X koordinátáját az α szög koszinusza és a téglalap szélességének felével vett szorzata adja. Az „F” pont Y koordinátáját hasonlóan az α szög szinusza és a téglalap magasságának felével vett szorzata adja. Az „F” pont koordinátáihoz még hozzá kell adni a „C” középpont koordinátáit, így megkapjuk a képre vetített globális koordinátákat. Utolsó előtti lépésként meghatározom, hogy az objektumot reprezentáló téglalap mely sarkához helyezkedik el közelebb az „F” pont. A 3.4 fejezetben a 11. ábra szerint ismertetett átmenetet képezek a kiválasztott sarokpont és az „F” pont között.

Az átmenetképzés során kapott végső koordinátáról eldöntöm, hogy a „C” középpont és űrszelvénytárcsán túl helyezkedik-e el. Ha igen, akkor az objektum akadályt képez, a vizsgálatot nem folytatom le más objektumokra sem. Ha viszont a „C” pont és űrszelvénytárcsa közt helyezkedik el, akkor a következő régióban is megismétlem a vizsgálat lépéseit mindaddig, amíg nem érem el a „C” pont magasságát. Mivel a közelítő ellipszis körvonala a „C” pont magasságától felfele haladva egyre távolodik a pályától, ezért a vizsgálatot nem érdemes tovább folytatni.

3.5. RENDSZER ARCHITEKTÚRA

Első lépésben szükséges a rendszer inicializálása, mely során megtörténik a szükséges memória területek allokációja rendszer- és video memóriában. Az első képkocka felhasználásával meghatározásra kerül a pálya keresésének kiindulási pontját és az objektumdetektáláshoz szükséges maszkolási terület. Az első képkockát követően történhet meg az objektumdetektálás és akadályészlelés. A 15. ábra szemlélteti a rendszer működését folyamatok szintjén.



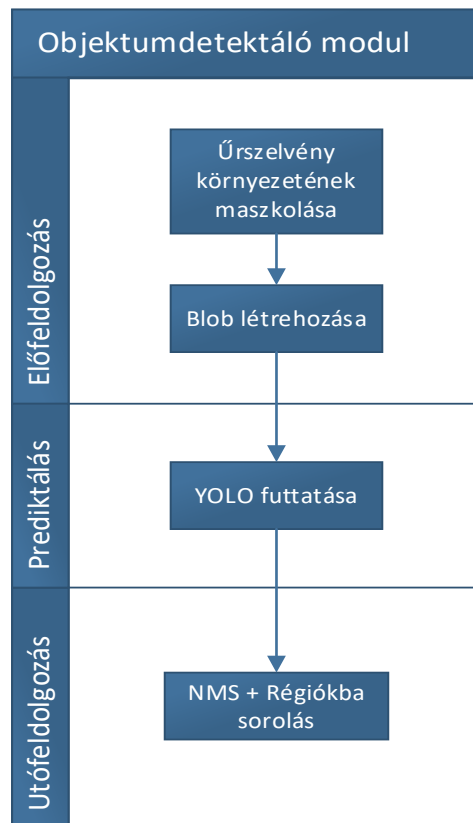
15. ábra. Rendszer működésének folyamatábrája

Inicializálást követően a rendszer iteratíván, minden képkockát feldolgoz. A pályadetektáló és objektumdetektáló modul gyakorlatilag függetlenek egymástól és ránézésre célszerű lehet külön futtatni őket, viszont a gyakorlat azt mutatja, hogy nem érdemes párhuzamosítani. Mivel a folyamatok nagy része ugyan azon GPU-t használja, ami folyamatos kihasználtság alatt áll, ezért, ha közel egyidejűleg szeretnék CUDA kerneleket indítani, akkor ezen kernelek ütemezése nagyobb késleltetéshez vezetne, mely rontaná a rendszer teljesítményét, ezért maradok a modulok egymás után történő futtatásánál.

A pályadetektálás során a kép feldolgozása teszi ki a modul legnagyobb részét. A képfeldolgozási lépések során egy művelet hajtok végre a kép összes pixelén. Ugyan azon művelet végrehajtható minden pixel esetén függetlenül egymástól. A végrehajtás során minden pixelre egy külön szál fogja elvégezni az adott eljárás műveletét. A szálak nyújtotta párhuzamosítási lehetőségek maximális kihasználásáért az eljárásokat CUDA környezetben implementáltam. A szürkeárnyaltos konverziótól egészen a *Sobel* operátorok eredményének kombinálásáig a folyamatnak nem szükséges elhagynia a GPU környezetét. Az integrál kép

CUDA környezetben történő megvalósítására egy hatékony módszert mutatott be *Berkin Bilgic, Berthold K.P. Horn és Ichiro Masaki* az „Efficient integral image computation on the GPU” [25] cikkükben. A gyakorlat során arra a következtetésre jutottam, hogy a CPU-n futtatott integrálkép sebessége elegendő, a bonyolultabb GPU implementációval elért teljesítmény növekedés nem számottevő.

A régiók vizsgálatának további gyorsításának érdekében az integrálképen elvégzem a csúszó ablakokban számított integrálösszegek meghatározását GPU-n. A folyamat egy $N \times M$ méretű, kétdimenziós tömböt eredményez, ahol „N” a régiók száma és „M” az intenzitásösszegek száma, mely régiónként eltér. A pályaleírók meghatározása során csak az intenzitásösszegek között kell maximum keresést végezni a 3.2.2 alfejezetben ismertetett módon.



16. ábra. Objektumdetektáló modul folyamatábra

Az objektumdetektálás (16. ábra) első lépésében megtörténik a képkocka maszkolása, majd a kapott képet átméretezem és létrehozom belőle a neurális háló input tenzorával kompatibilis blob-ot. Ezt követően futtatom a neurális hálót. A prediktálást követő eredménymátrixon elvégzem a küszöbölést, amit követően csak azokkal a predikciókkal foglalkozok, melyek elérték a megadott küszöbértéket. Az objektumjavaslatok leginkább

egymást fedve jelentkeznek különböző konfidencia értékkel. A tényleges objektumot reprezentáló predikció kiválasztása a nem-maximumok elnyomásával történik, mely a predikciók alkotta javaslatklaszterek közül kiválasztja a legmagasabb konfidencia értékkel rendelkező predikciót. Javaslatklasztereket azok a predikciók alkotnak, melyek befoglaló téglalapjai egy megadott küszöbérték felett fedik egymást. Két befoglaló téglalap fedésének arányát meghatározó számot Jaccard index-nek [26], másnéven Jaccard hasonlósági együtthatónak nevezik. A folyamat végén előállnak a detektált objektumok, melyeket a pályadetektálás során ismertetett (3.2.2 fejezet) régiókba sorolok, gyorsítva későbbi vizsgálatukat.

Az utolsó lépésben megtörténik az akadályészlelés a 3.4.1 fejezetben ismertetett módon, mely során nem végzek párhuzamosítást.

4. EREDMÉNYEK ÉS ÉRTÉKELESÜK

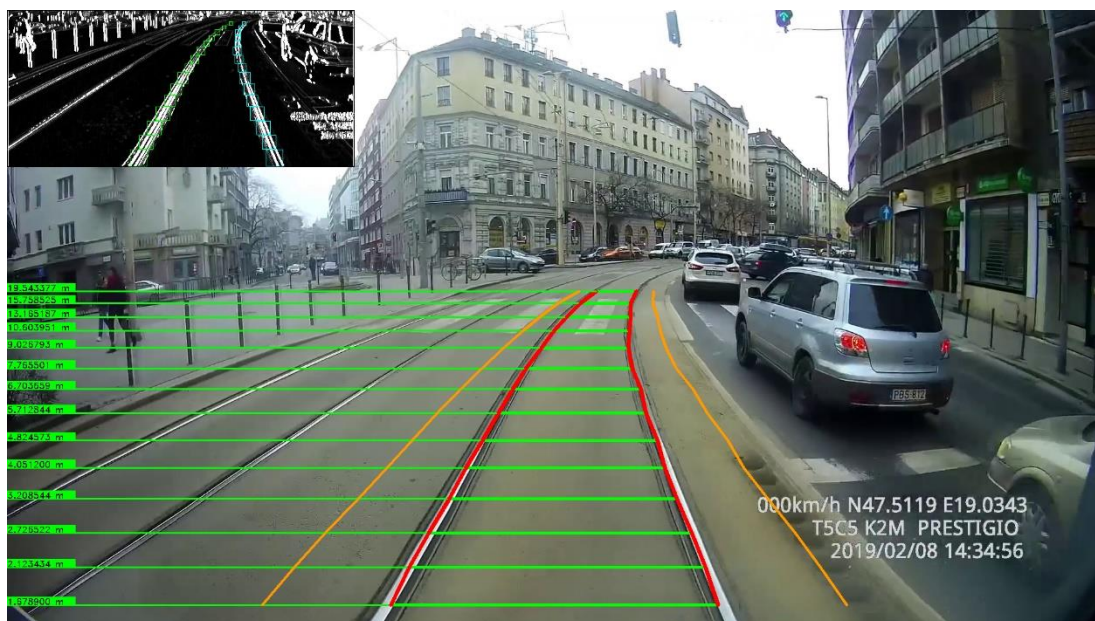
A kitűzött feladattal egy olyan problémára kerestem megoldást, mely megfelelő képfeldolgozási technikák használatával és neurális hálózat segítségével képes elvárt módon azonosítani valós időben fellépő akadályokat. A fejlesztés során sikerült megvalósítanom egy olyan rendszert, mely a kitűzött céloknak funkcionalitásban és sebességben is leginkább eleget tesz. Az elért eredmények bemutatása során ismertetem azokat a problémákat, melyek negatívan befolyásolják a rendszer adott területen nyújtott teljesítményét.

Az eredmények bemutatása során ki kell emelnem, hogy a teszteléseket nem élő kameraképen végeztem, hanem rögzített videófolyamokon. Így a felvételeket biztosító kamerák elhelyezése és az alkalmazott optikák adottak voltak, nem a feladathoz igazítottak. Hasznosabb lett volna, ha a kamerákat magasabbra helyezik a vezetőtérben és nagyobb távolságra fókuszáltak volna. A videók tömörítése is részben korlátozhatta a még pontosabb eredményeket.

Az elért eredményeket három pontba foglalva ismertetem. Az első, a sínparók azonosításának pontossága, mely befolyásolja a detektálható objektumokat és elhelyezésüket. A második az akadályt képző objektumok azonosításának pontossága, melyek elhelyezésének a sínszálak nyújtanak alapot. Az utolsó pontban a rendszer végrehajtási idejét értékelem egy képkockára vetítve. Az eredményeket a fejlesztés során felhasznált felvételek közül egy megadott felvétel [27] klipjein futtatva rögzítettem és értékeltem ki az általam erre a célra fejlesztett egyszerű szoftverek segítségével.

4.1. PÁLYADETEKTÁLÁSI EREDMÉNYEK

Kiértékelés során csak a sínszalakat azonosító modult futtattam számos videóklipen. A klipek megválasztása során leginkább a sínszalak környezetében lévő, különböző felépítményeket vettem alapul. Az értékelés során aszfalttal, bazaltkőkockával és betonlemez borítással rendelkező pályaszakaszok mellett egy zúzottköves magasvágány pályakialakítást és egy alagúton történő áthaladást vettem figyelembe. A különböző útburkolatok különböző méretű és típusú illesztési hibákkal rendelkeznek a felvételeken, melyek néhol erősebb intenzitással bíró körvonalakkal rendelkeznek a sínszalakhoz képest. Ez a jelenség elég erős negatív hatást tud gyakorolni a pályaaazonosítás pontosságára. A magasvágányos pályaszakasz jellegzetes karakterisztikával rendelkezik, melyre a hasonló rendszerek 2.2 fejezetben ismertetett kutatások és fejlesztések építettek. Az alagúton történő áthaladás során azt vizsgáltam, hogy milyen hatást gyakorol a detektálásra a világos környezetből sötétbe és sötétből világosba történő átmenet. A 17. ábra szemlélteti a pályadetektálás eredményét.



17. ábra, Pályadetektálás betonlemez borítással rendelkező pályaszakaszokon

Minden klipet, minden képkockát egyesével elemeztem, hogy eldöntsem, a pályát helyesen sikerült-e azonosítani vagy sem. A kereső algoritmus már eleve egy kellően szűrt képnek a régiók szerint felbontott sorait teljes mértékben feldolgozva kapja, ahol szinte csak maximumokat kell helyesen meghatározni. Ebből a tényből kiindulva egy nagyon egyszerű értékelési rendszerrel álltam elő. Minden olyan képkockát helyesnek tekintek, ahol a csúszó ablakok teljesen fedik a pályát. Mivel a csúszó ablakokon belül a sínszalak leginkább átlósan helyezkednek el, ezért az algoritmus keresés során találkozhatott olyan összegekkel is, melyek

nem csak a pálya élei által generált intenzitást tartalmazzák, hanem a környezeti zajt is, mely nem pontosan tükrözi a sínszál valódi helyzetét. Az említettek miatt egy kisebb tűréshatárt vezettem be, melyen belül még helyesnek mondható a detektált pálya. Minden más esetben helytelennek véltem az eredményt. A vizsgálható pályaszakasz hosszát a felvételeket készítő kamera elhelyezése és beállítása határozta meg, és a tesztek során határoztam meg, mely 18 méteres távolságot jelent. A 3. táblázaton a 18 méteres kereső szakaszon rögzített eredmények láthatók. A feltüntetett felvételekhez egy-egy feldolgozott kép és eredeti nyers képre vetített eredményét szemlélteti a 18. ábra. A fejlesztés során rögzítésre került egy körülbelül 22 méter hosszú szakasz is, melynek eredményei a 4. táblázatban szerepelnek.

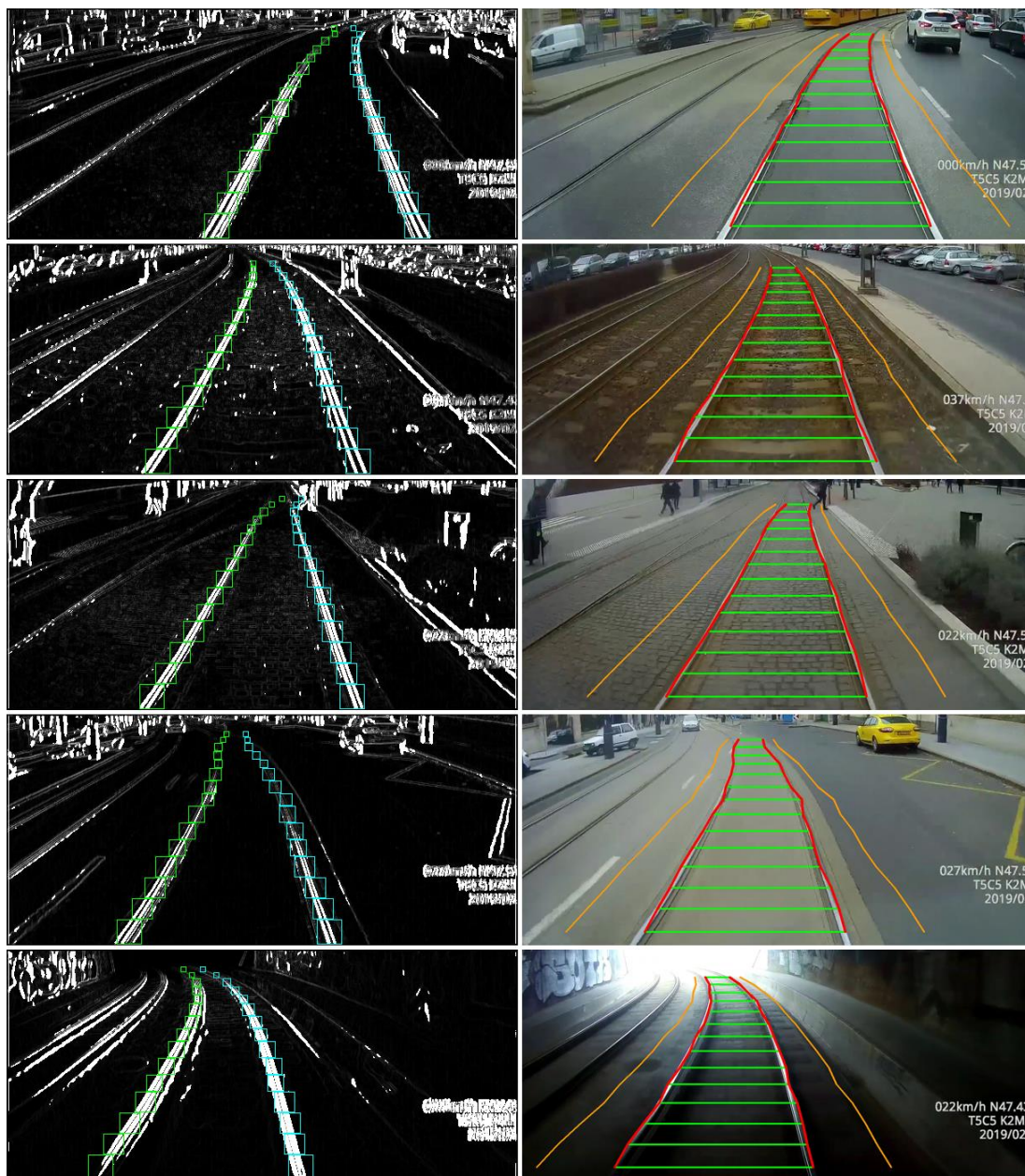
Felvétel	Összes képkocka	Helyes	Helytelen	Pontosság
betonlemez_14_18	480	480	0	100%
magasvagany_32_44	385	385	0	100%
bazaltkocka_17_44	441	434	7	98,41%
aszfalt_11_09	450	433	16	96,44%
alagut_53_06	510	477	33	93,52%
Összegezve:	2266	2210	56	97,67%

3. táblázat. 18 méteres távolságot bezáró vizsgálat eredményei

Az eredményeken jól látszik, hogy a pályaaazonosítás az esetek több mint 97%-ban helyesnek mondható. Ez az érték kifejezetten jónak mondható, mivel a felvételek minőségén sokat rontott a tömörítés. Két felvételen is 100% pontosságot értünk el, mely annak a ténynek tudható be, hogy ezeken a sínszálak jellegzetes körvonalait igen jól ki lehet venni. A sínszálakat körülvevő beton felülete nagyon kis mértékű intenzitásváltozással rendelkezik, kevés zaj van körülötte, mely egyértelműen meghatározza a közre zárt sínszálakat. A magasvágány esetében a sínszálak környezete nagyobb zajjal bír a zúzottköveknek és aljaknak köszönhetően, viszont a nem csak a sínkorona látszódik, hanem a sínszál teljes egésze, mely több erős élben nyilvánul meg, ezzel egyértelműen behatárolva a pályát.

A bazaltkockás pályaszakasznál fellelhető egy kevés pontatlanság, melyet nem a sínszálakat körbevevő bazaltkocka intenzitása okozza, hanem a felvételen átsétáló gyalogos. A gyalogos lábai erősebb éllel rendelkeznek a tompa, egyszínű háttértől elkülönülve, melyeket az algoritmus mindaddig lekövetett, amíg a gyalogos a sínszál közvetlen közelében tartózkodott. Érdekes megemlíteni, hogy a probléma csak addig állt fenn, amíg a gyalogos a tényleges úrszelvényen belül tartózkodott. Az akadályészlelés eredményein látható, hogy az

algoritmus a kamerához legközelebbi akadálytól tovább már nem számítja a pályát keresztező többi objektumot, ezért a pályadetektálás az első akadályt követően már nem szükséges figyelembe venni.



18. ábra. Felvételek feldolgozott és nyers képei, fentről lefele haladva: betonlemezés útburkolat, magaszvágány, bazaltkőkockás burkolat, aszfalt és alagút

Az aszfalttal borított szakaszon szereplő 16 helytelen detektálás a felvétel rossz minőségének tudható be. A tömörített felvételen a pályaszakasz távolabbi része nem volt látható se az ember, se az algoritmus számára. Az útburkolat színe teljesen megegyezett a sínkoronával, ezért az algoritmus teljesen jogosan egy másik erős élen folytatta a keresést. Ez a pontatlanság

a pályaszakasz végén jelentkezik, melyre a legegyszerűbb megoldás nem tömörített felvételek használata, ami a fejlesztés során számomra nem állt rendelkezésre.

Az alagúton való áthaladás során meglepő módon egész jó eredményeket értem el. Az algoritmus nagyon jól reagált a fényviszonyváltozásokra. A világosból sötétbe történő átmenetnél a távolabbra eső csúszó ablakok is nagyon pontosan tudták azonosítani a gyenge éleket. Ez betudható annak, hogy a keresett pályaszakasz vége sose esett bele a teljesen sötét alagútba, a kamera folyamatos alkalmazkodása az alagúthoz lehetővé tette, hogy valamennyi gyenge él megmaradjon. A sötétből világosba történő átmenetnél is hasonlókat lehet tapasztalni. Az egyedüli pontatlanságot a keresett pályaszakasz végén lévő ív jelentette. A sínszálak környezetében lévő nagyobb zaj eltérítette a csúszó ablakokat. Ez a jelenség leginkább 15-16 méteres távolságban jelentkezett, mely a keresett pályaszakasz végén található. Ezen a távolságon a csúszó ablakok nagyon kis területeket vizsgálnak, melyekben a zaj sokkal nagyobb negatív hatással bír, mint a kisebb távolságok esetében. A legjobb megoldást az jelentené, ha megnövelném a gradiens kép felbontását, hogy több részlettel tudjon dolgozni a rendszer.

Az elért eredmények ismételten csak arra utalnak, hogy a rendszer zajos környezetben is nagy pontossággal képes helyesen azonosítani a villamos pályáját, viszont ez csak a korábban említett 18 méteres távolságig igaz. Továbbá fontos, hogy a zaj ebben az esetben a fényviszonyok okozta színeltérések, árnyékok tekintetében értendők, melyek nem okoznak kirívóan hatalmas intenzitás változást. 18 méter felett a gradiens kép kis felbontása miatt a rendszer nagyon érzékeny a zajra, melyen túl nem célszerű további vizsgálatokat folytatni. Az így elért eredmények a 4. táblázatban láthatóak. A korábban ismertetett hibák minden felvételre ugyan úgy igazak 18 méter felett.

Felvétel	Összes képkocka	Helyes	Helytelen	Pontosság
magasvagany_32_44	385	385	0	100%
bazaltkokocka_17_44	441	421	20	95,46%
aszfalt_11_09	450	410	40	91,11%
betonlemez_14_18	480	410	70	85,42%
alagut_53_06	510	410	100	80,39%
Összegezve:	2266	2036	230	90,48%

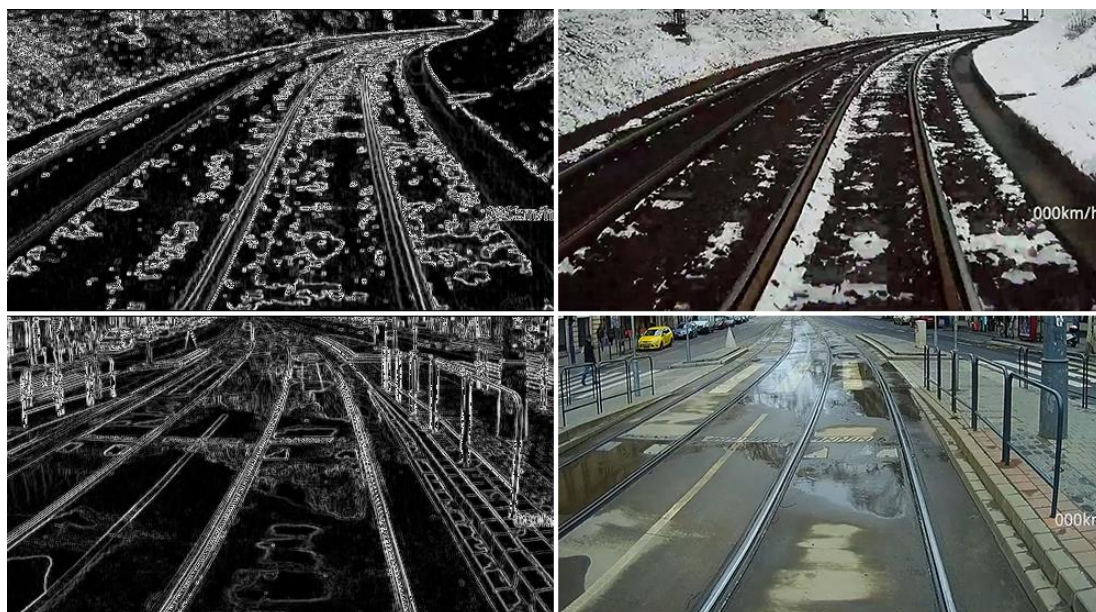
4. táblázat. 22 méteres távolságot bezáró vizsgálat eredményei

Ahogy azt látni lehet, szinte minden felvétel pontossága csökkent 5-10%-kal. Minden esetben az utolsó, legtávolabbi csúszó ablak vétette a legtöbb hibát. Érdeemes kiemelni a betonlemez felvételt, ahol az ívelt pályaszakaszon való haladáskor a szemből érkező villamos oldallemezei nagyobb intenzitással rendelkeztek, mint a pályaszakasz, ezért a bazaltkőkockás gyalogos esetéhez hasonló módon, a megnövelt távolsággal vizsgáló algoritmus hibásan követte le a tényleges pályát.

A 2. fejezet során bemutatott hasonló rendszerek közül kiemeltem kettőt ([4] és [6]), melyek hasonló formában hajtják végre a pályadetektálást. A [6] megvalósítás közel 100 méteres távolságban képes azonosítani a pályát, mely egy jelentősen nagyobb távolság a saját rendszeremhez képest. Ennek oka, hogy a vasúti szerelvényen magasabban helyezték el a kamerát, mint a villamoson, így nagyobb rálátással rendelkezik a pályára. Továbbá nem érdemes elhanyagolni a tényt, hogy a nagyvasúti sínszálak élei sokkal erősebbek, mint az útburkolattal egy szinten lévő villamos pálya, ezért nagyobb távolságban jobban kivehetőek lesznek. A rendszerem tesztelése során mind 18 méter felett és alatt is nagyvasúti magasságágon 100% pontosságot ért el az általam megvalósított pályadetektálás, mely közel azonos étékkel bír a [6] rendszerrel szemben. A [4] megvalósítás során kiemelték, hogy a rendszer jól kezeli az árnyékos környezetet, ami a gradiens képen való élkeresésnek köszönhető. Rendszerem hasonló megvalósítással bír. A teszteredményeknél, az alagúton történő áthaladás során, különböző fényviszonyokban 93,5% pontossággal teljesített a rendszerem, melyből levonható, hogy véleményem szerint közel azonos képességgel bír a két rendszer.

A rendszert teszteltem különböző időjárási jelenségek tekintetében, nevezetesen havazást és esőzést követő pályaszakaszokon. A rendszer eredménye teljes mértékben függ a sínszálak környezetében fellépő zaj mértékétől. Az eső okozta kis méretű állóvíz rossz pályaburkolaton nagyon könnyen eltéríti a csúszó ablakokat, mert a foltos felületen gyakran jóval erősebb élek formájában jelenik meg az állóvíz, mint a sínszálak, ezért ezen jelenségeket nagyon intenzív környezeti zajnak tekintem. A 19. ábra szemlélteti ezen intenzív környezeti zajokban előfeldolgozott képet. Az ábrán csak a 3×3-as gradiens kép eredményét szemléltetem, melyen látni, hogy nagyobb távolságban a pályaszakasz már szabad szemmel sem kivehető. A 3×3 és 5×5 gradiens kép kombinálása értelmezhetetlen. A pályadetektálás nem végezhető el, ha a víz teljes mértékben elfedi a pályát vagy az állóvíz kellő távolságban fedi a sínszálakat. Ugyan ez elmondható a havas vagy kevés hóval, esetleg hófoltokkal körbevett pálya kapcsán. A 19. ábra

felső képen látható, hogy a hófoltok kevésbé értelmezhető képet eredményeznek, mint az alatta lévő vizes pályaszakasz.



19. ábra. Gradiens és eredeti kép, fentről lefele; elszórtan hóval és vízzel fedett pályaszakasz

A pályadetektálási eredmények zárása előtt érdemes említést tenni a folyamatot megvalósító modul egy korai fejlesztési változatát, mely más szemszögből igyekezett megoldani a problémát. A pályadetektálási folyamat lépései nem sokban különböztek az aktuális megvalósítástól, viszont egy nagy különbséget jelentett, hogy milyen szemszögből volt feldolgozva a képkocka. A jelenlegi perspektív szemlélettől eltérve korábban a [6] cikkhez hasonlóan fentről tekintett lefele a pályára¹⁶. Hasonló módon első lépésben létre lett hozva a kép szürkeárnyaltos konverziója. Ezután egy perspektív transzformációt hajtottam végre, mely során előállt a pálya felülnézeti képe. Erre azért volt szükség, hogy a pálya sínszáalai ne a horizont egy pontjába tartsanak, hanem párhuzamosokként jelenjenek meg. Következően előállítottam a gradiens képet és *Median* szűrő alkalmazásával csökkentettem a zajt. Mivel a gradiens képen a sínszáalak élei párhuzamosok formájában voltak jelen, ezért a képet hasonló módon régió sávokra bontottam és minden régiónak előállítottam az élerősség tömbjét. Az élerősség tömb minden eleme megadja, hogy az adott régió magasságától függően a vertikális élek milyen erősséggel vannak jelen a megfelelő indexű helyen. Ez az intenzitásösszegző módszernél egy precízebb sínszálkeresést eredményez. Végezetül az élerősségtömbökön

¹⁶ Általános kifejezéssel élve „madártávlatból” (bird’s eye view)

régióként lentről felfele haladva maximumkereséssel meghatároztam a sínszálakat (20. ábra, jobb oldali kép).



20. ábra. Felülnézeti pályadetektálás, balról jobbra, eredeti kép és detektált pálya

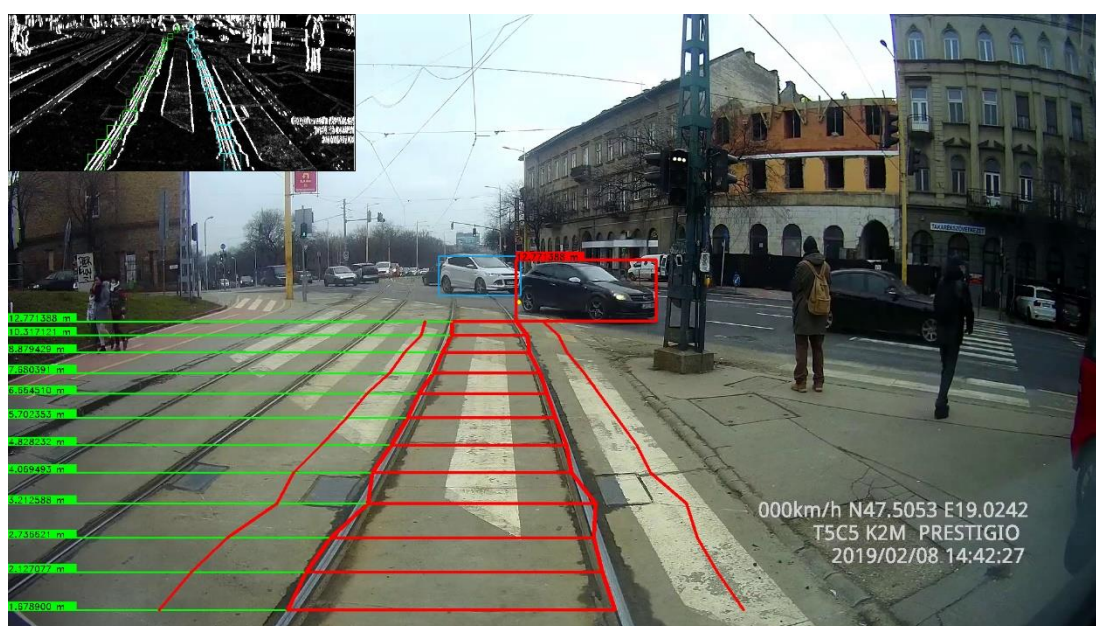
Felmerül a kérdés, hogy ha precízebb sínszálkeresést tett lehetővé az élerősségtömb használata, akkor miért nem lett alkalmazva a jelenlegi pályadetektálási eljárás során? A válasz teljesítményproblémákra és adatvesztésre vezethető vissza. A teljesítmény problémát a 2.3 fejezet [6] rendszer technikájának értékelése során ismertettem, melynek eredményét a pályadetektálási modul ezen korai változata nyújtotta. Az adatvesztés a teljesítményproblémát jelentő a transzformáláshoz vezethető vissza, mivel lényegében a pálya kamerához közelebb eső részét „összenyomjuk”, a távolabbit pedig „széthúzzuk”. Az összenyomás során adatvesztés történik, a széthúzásakor pedig újak létrehozása, ezért a folyamat nem nyújt konzisztens pályareprezentációt. Végezetül élerősségtömböket csak horizontális, vagy csak vertikális élekből lehet előállítani. Amennyiben egy olyan él szerepel ívelt pályán, amely erősen vertikális irányban van jelen, úgy az helytelen pályadetektálást eredményezhet a gyenge sínszálak miatt. Az említett problémák miatt maradtam a kamera szemszögéből történő pályafeldolgozásnál.

Összességében elmondható, hogy 18 méter alatt a pályadetektáló modul különböző fényviszonyokban és különböző útburkolattal rendelkező villamos pályát közel 98%-os pontossággal képes helyesen azonosítani. A hibatartomány leginkább a pályaszakasz végére tehető, ahol a kis felbontás miatt, leginkább zajos környezetben nagyon nehéz helyesen kiemelni a sínszálakat. Az előfeldolgozás során végrehajtott átméretezéskor célszerű egy optimális méretre csökkenteni a képet, mely még nem okozna jelentős teljesítmény veszteséget.

4.2. AKADÁLYÉSZLELÉSI EREDMÉNYEK

Az eredmények rögzítése olyan videóklipen történt, melyeken valamelyest különböző forgalmi helyzetek vannak jelen. Mivel a legtöbb villamos pálya városon belül található, ezért a felvételek városi környezetet mutatnak be és így a rendszert csak útkereszteződéseken és gyalogosátkelőhelyeken tudtam érdemben tesztelni.

A járművet keresztező akadályok szempontjából próbáltam többféle olyan helyzetet keresni, melyben különböző irányban és távolságban haladnak el a járművek és gyalogosok a kamera előtt. A rendszert így három felvételen teszteltem. Az első¹⁷ felvételen a gyalogosok és járművek körülbelül 7 és 18 méter közötti távolságban keresztezik a járművet, a villamospályához képest merőlegesen haladva. A keresztező objektumok távolságai ideálisnak mondhatók a pályadetektálás során meghatározott maximális detektálási távolság szerint, mely 18 méter volt. A második¹⁸ felvételen körülbelül 15 méteres távolság felett járművek keresztezik a villamos pályát, srégen haladva az észlelési távolság peremén. A harmadik¹⁹ felvételen a kamerához közel, 2 és 7 méter távolság között haladnak el gyalogosok a kamera előtt, szintén srégen.



21. ábra. Akadályészlelés a második felvételen, 263. képkocka

¹⁷ obstackle_num2_07_38

¹⁸ obstackle_num0_21_18

¹⁹ obstackle_num1_17_12

A kiértékelés során képkockáról képkockára meghatároztam, melyik objektumot lehetett akadálnak tekinteni. A 21. ábra szemlélteti, hogyan is néz ki a képkocka feldolgozása során azonosított akadály. Akadálnak kell venni minden olyan objektumot, mely a villamospálya úrszelvényében tartózkodik. Mivel a rendszer lentől felfele vizsgálja az objektumok elhelyezkedését és a legelső akadálydetektálás után nem hajt végre további vizsgálatot, ezért legfeljebb egyszerre csak egy objektum képezhet ténylegesen akadályt, mely mindig a kamerához legközelebb eső lesz. Mivel minden akadály akár több képkockán keresztül is jelen tud lenni, ezért azt vizsgáltam, hogy a tényleges akadályokat hány képkockán keresztül tudta helyesen azonosítani a rendszer. Az objektumok alakját egy durva becsléssel határoztam meg, ezért a közelítésből fakadó helytelen azonosításokat is rögzítettem. Emellett, mivel a YOLO nem biztos, hogy helyesen vagy egyáltalán detektálta az objektumot, ezért az észlelésnek hiányát is rögzítettem. Az akadályészlelés pontosságát ezért az összes képkocka és helyes detektálás százalékos arányában mértem. Az eredményeket az 5. táblázat ismerteti.

Elsősorban érdemes megemlíteni, hogy a közel ezer képkockán keresztül fennálló akadályokat körülbelül 84,5%-os pontossággal sikerült meghatározni, mely véleményem szerint egy kifejezetten jó értéknek mondható. Azonban, ha megnézzük a hibák számát, akkor látható, hogy a helytelen azonosításokhoz képest közel négyszer annyi alkalommal nem sikerült az akadály észlelése. A problémát két szempont szerint célszerű megvizsgálni. Megkülönböztetem azokat az eseteket, ahol az észlelt objektum már a sínszalakon és az általuk közre zárt területen tartózkodik, illetve azt az esetet, amikor még csak az úrszelvény szélén, de a pályán még nem tartózkodik.

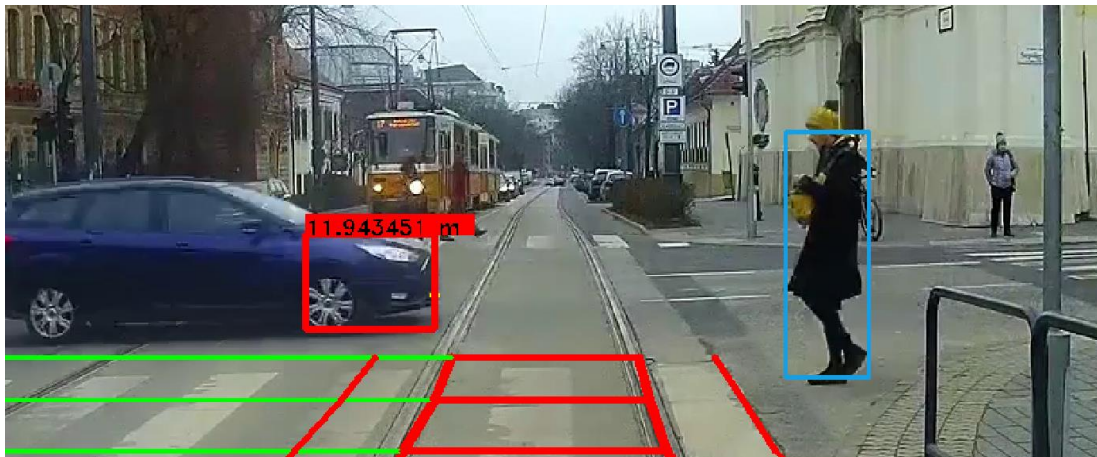
Az első esetben elmondható, hogy szinte mindig kellő mennyiségű jellemző áll rendelkezésre az objektumok észlelésére. A modul egyik problémája, hogy a több, egymást fedő objektumokat sokszor nagyon nehezen tudja megkülönböztetni. Ez a probléma jellemzően akkor szembetűnő, ha egy többsávos kereszteződésben számos autó halad keresztül a pálya előtt. Sokszor egy csoportba vonva kezeli a különböző objektumokat és van, hogy egy esetleg elhaladó gyalogost pár képkockán keresztül emiatt észre sem vesz, mert a gyalogos jellemzői túl hasonlóak a járművekéhez²⁰. Ezek az esetek a felvételeken jellemzően elszórva nyilvánulnak meg.

²⁰ Hasonló jellemzők lehetnek olyan körvonalakkal behatárolt alakzatok, melyek például közel azonos színű kitöltéssel rendelkeznek.

Felvétel	Akadály sorszám	Átlagos távolság (m)	Első és utolsó képkocka	Összes képkocka	Helyes	Hibás	Nem észlelt	Pontosság
obstacle_num2_07_38	1.	13,00	4 - 19	16	15	0	1	93,75%
	2.	6,40	20 - 92	72	62	0	10	86,11%
	3.	14,50	92 - 98	6	3	3	0	50,00%
	4.	17,20	99 - 107	8	7	0	1	87,50%
	5.	7,20	108 - 175	67	60	3	4	89,55%
	6.	7,50	176 - 200	25	23	0	2	92,00%
	7.	14,50	214 - 219	6	6	0	0	100,00%
	8.	12,00	220 - 242	23	18	3	2	78,26%
	9.	14,80	243 - 246	4	1	0	3	25,00%
	10.	11,90	282 - 286	5	3	0	2	60,00%
	11.	8,90	287 - 349	63	49	0	14	77,78%
	12.	17,50	379 - 411	32	29	2	1	90,63%
	13.	12,50	422 - 442	21	19	0	2	90,48%
	14.	17,50	443 - 467	24	23	1	0	95,83%
	15.	12,00	481 - 495	15	12	0	3	80,00%
	16.	6,50	496 - 566	71	66	0	5	92,96%
	17.	17,00	567 - 590	24	21	1	2	87,50%
	18.	16,50	613 - 618	6	5	1	0	83,33%
	19.	12,50	619 - 641	23	20	0	3	86,96%
obstacle_num0_21_18	1.	21,50	61 - 69	9	9	0	0	100,00%
	2.	18,00	83 - 120	38	33	0	5	86,84%
	3.	18,00	135 - 168	34	26	2	6	76,47%
	4.	16,60	171 - 200	30	29	1	0	96,67%
	5.	21,40	201 - 202	2	2	0	0	100,00%
	6.	19,45	203 - 234	32	28	2	2	87,50%
	7.	15,20	241 - 265	25	19	2	4	76,00%
obstacle_num1_17_12	1.	5,45	18 - 43	26	26	0	0	100,00%
	2.	3,00	44 - 102	59	47	3	9	79,66%
	3.	7,00	103 - 115	13	10	3	0	76,92%
	4.	5,90	116 - 184	69	61	5	3	88,41%
	5.	5,85	185 - 213	29	21	2	6	72,41%
	6.	7,00	214 - 235	22	21	1	0	95,45%
	7.	2,00	236 - 291	56	36	0	20	64,29%
	8.	8,50	292 - 310	19	12	0	7	63,16%
Eredmények összegezve:				974	822	35	117	84,39%

5. táblázat. Akadályészlelés eredmények

Ha az űrszelvény határait nézzük, akkor az objektumdetektálást mindig egy, az űrszelvény környezetét körülvagott képen kell végrehajtani. A széleken jellemző módon az objektumok kis részletei látszódnak, általában srégen ketté vágva. Ha detektálás során a modul kellő mennyiségű összefüggő jellemző birtokába kerül, akkor a csonka objektumokat a vágás szélén meglepő módon egész jól képes azonosítani. Jellemzően a járművek elejét már egy fényszóróról és egy kerékről is helyesen meg tudja határozni, melyet a 22. ábra szemléltet.



22. ábra. Akadályészlelés az első felvételen, 281. képkocka

A vágás szélein történő helyes azonosításhoz kellő mennyiségű jellemzőre van szükség, melyek nem biztos, hogy az az objektum űrszelvénybe történő lépése előtt rendelkezésre fognak állni. A probléma kiküszöböléséhez a vágási határokon kell módosítani oly módon, hogy az űrszelvény környékéből minél több részlet látszódjon. Ez felvet egy új problémát, miszerint a jellemzően hosszabb járművek, esetleg szerelvényt befoglaló téglalapok sarkai oly mértékben metszik az űrszelvényt, melyek nem tükrözik a jármű tényleges elhelyezkedését és helytelen detektálást eredményeznek. A probléma kiküszöbölésére lett bevezetve az objektum alakjának geometriai úton történő becslése, mely kisebb méretű befoglaló téglalapok esetén jól működik, de egy bizonyos méret felett már nem. Az űrszelvény környezetének körül vágása erre a problémára nyújt egy nagyjából jól működő megoldást. Amennyiben túl sok részletet mutatunk a környezetből, akkor a probléma ismét előjön.

Az eredményeket érdemes összevetni két korábban ismertetett hasonló rendszerrel. A [4] rendszer a pályadetektáláson túl akadályészlelési feladatokat is ellát, mely kiterjed a sínszálak közvetlen környezetére és a sínszálak közti területre. A cikkben bemutatott eredményeket aszerint értékelték, hogy a rendszer sikeresen azonosította-e a felvételen elhelyezett akadályt vagy sem. 10 esetből 10 alkalommal sikeresen azonosítottak akadályt, mellyel 100%-os pontosságot jelentettek. Rendszeremet nagyobb részletességgel értékeltém ki, ezért az

összehasonlítás érdekében az eredményeket leegyszerűsíttem sikeres és sikertelen akadály azonosítások arányára. Az 5. táblázaton láthatjuk, hogy minden akadályt sikeresen azonosítottunk, ezzel elérve a [4] rendszer pontosságát. Továbbá fontos kiemelni, hogy rendszerem nem csak a sínszálak és köztes területén, hanem a teljes úrszelvényre kiterjesztve hajtja végre az akadályészlelést.

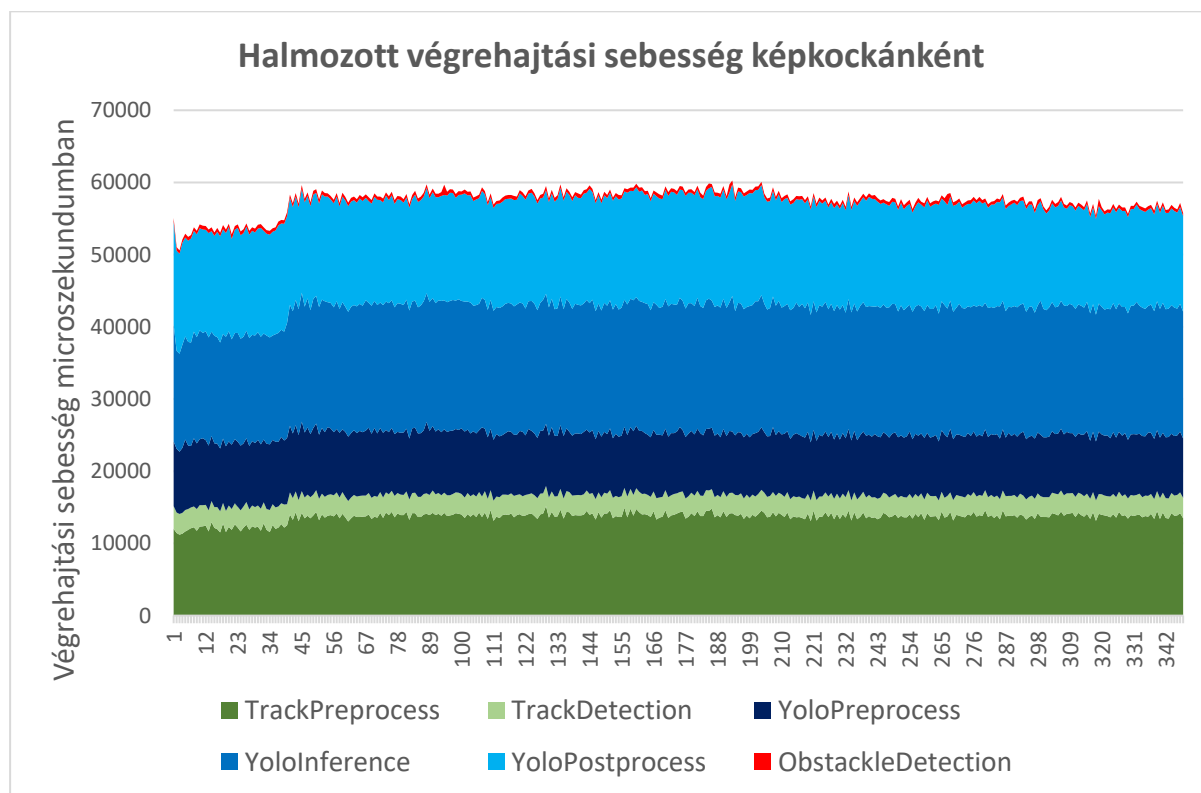
A [9] cikkben nem egy akadályészlelő rendszert mutatnak be, hanem egy objektumdetektáló megoldást, mely egy nagyvasúti környezetben hajt végre egyéni fejlesztésű neurális hálózattal vonat, gyalogos és bőrönd észlelést. Rendszerük átlagosan 91,61%-os pontossággal bír, melyet értékelés során összehasonlítottak más objektumdetektáló megoldásokkal is, köztük a YOLO első verziójával, mely átlagosan 75,67%-os pontosságot tanúsított. Rendszeremben a YOLO ötödik verzióját használom, mely jelentős teljesítmény ugrást jelent mind pontosságban, mind pedig sebességben az első verzióhoz képest. Az objektumdetektálás precizitásának vizsgálatára az 5. táblázaton feltüntetett „nem észlelt” oszlop összesítését kell vizsgálni az összes képkocka tükrében, mely 87,98%-os pontosságot eredményez. Rendszeremben használt neurális hálózat általános célú, nem kifejezetten vasúti környezetre lett fejlesztve, ezért ilyen szempontból nagyon is jónak tekinthetők az eredmények, mely során kevesebb mint 4%-os differencia mérhető a [9] megvalósításhoz képest.

Az akadályészlelés eredményeit összefoglalva elmondható, hogy a rendszer közel 85%-os pontossága nem elhanyagolható érték, ám figyelembe kell venni azokat a szélsőséges eseteket, melyek során az észlelés nagy valószínűséggel nem fog megvalósulni. Ilyen esetek lehetnek az egy csoportot képző, hasonló jellemzőkkel bíró objektumok, melyek entitásait külön-külön nem, vagy csak nehezen tudja behatárolni az objektumdetektáló modul, illetve az úrszelvény környezetének körül vágása során, az úrszelvény szélén fellépő objektumok kevés jellemzői nem mindig elegendő az észleléshez.

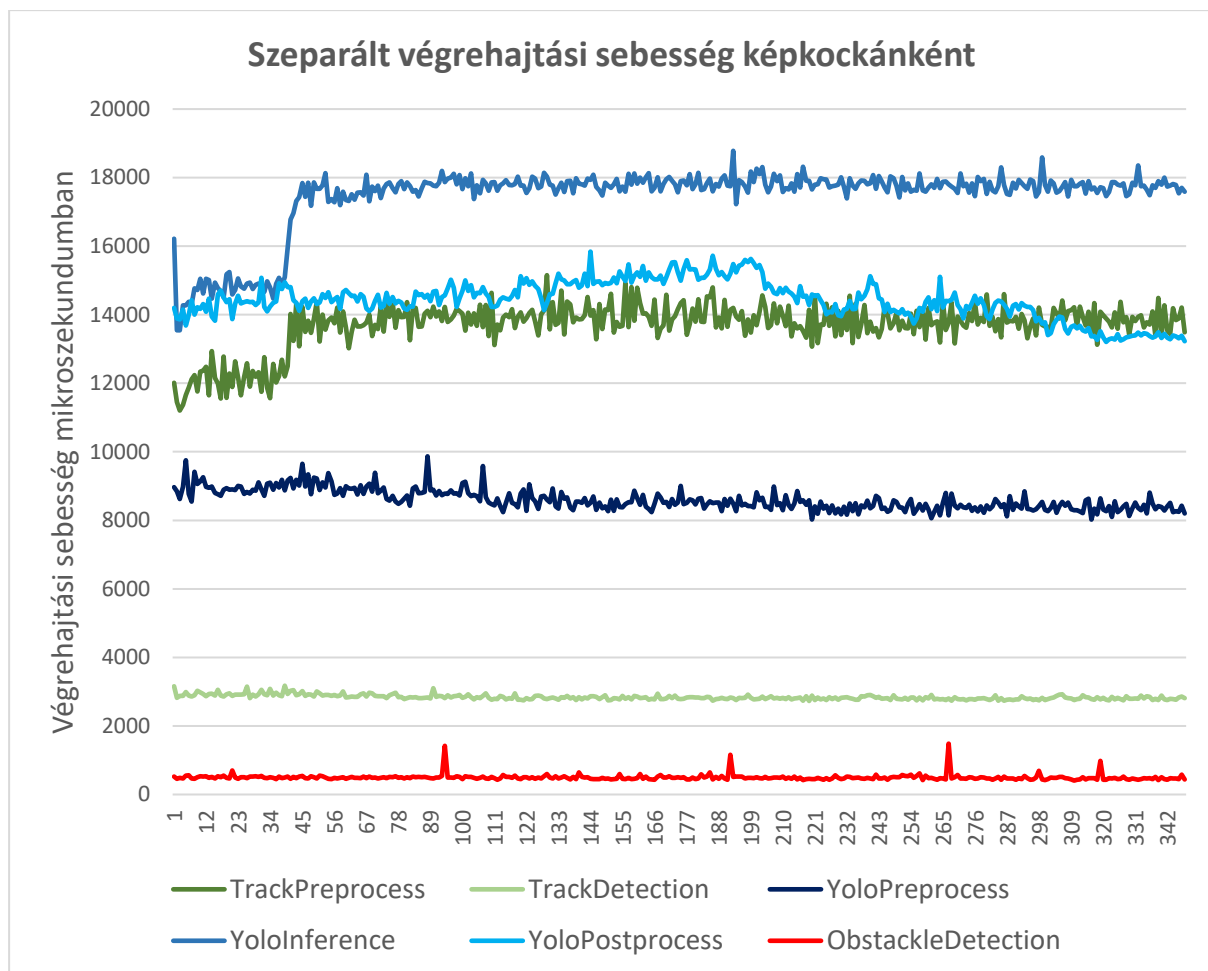
4.3. VÉGREHAJTÁSI SEBESSÉG EREDMÉNYEK

A rendszer teljesítményének vizsgálata során a képkockák feldolgozásához szükséges időt vettem alapul. A végrehajtási idő kritikus pontja az akadályészlelésnek, ezért arra törekedtem, hogy a valós idejű feldolgozásnak eleget tudjon tenni a rendszer. Az akadályészlelés során felhasznált felvételek közül kiemeltem egyet, melyen elvégeztem a futási idők rögzítését.

Az eredmények rögzítése során külön tároltam az egyes modulok egyes komponenseinek futásidejét. A pályadetektálást során különválasztottam a GPU-n futtatott előfeldolgozást és a CPU-n implementált szekvenciális pályadetektáló algoritmust. Az objektumdetektálást három fő részre bontottam. Az előfeldolgozásra, mely során az adott képkockából előáll a YOLO input tenzora, a YOLO-t futtató GPU implementációra, illetve az utófeldolgozásra, mely tartalmazza a YOLO output tenzorának szűrését és a nem-maximumok elnyomását. Az akadályészlelő modult nem bontottam tovább. A sebességrögzítést tíz alkalommal ismételttem meg ugyan azon felvételen. Erre azért volt szükség, mert minden futtatással más és más, minimálisan eltérő sebességeket mértem az inkonzisztens ütemezés miatt. Az így rögzített eredmények átlagát véve ténylegesen reprezentatív sebességmutatókat kaptam, melyek ideálisak a rendszer teljesítményének tükrözésére.



23. ábra. Modulok rétegjeinek összesített futási ideje képkockánként



24. ábra. Modulok rétegjeinek futási ideje képkockáknként

Az eredményeket két diagrammon összesítettem. A 23. ábra szemlélteti a különböző modulok részeinek eredményeit halmozottan összesítve képkockáknként. Az ábrán jól látható, hogy az átlagolást követően a különböző képkockák feldolgozási ideje nem konzisztens. A végrehajtási sebesség körülbelül 50 ms és 60 ms között mozog, de a rendszer átlagosan 57,47 milliszekundum sebességgel teljesített. A rendszer megvalósítása során elvetettem a pályadetektálás és akadályészlelés párhuzamosan történő végrehajtását, mert az előzetes eredmények szerint is kis teljesítmény növekedést érnek el. A pályadetektálás előfeldolgozása és a YOLO futtatása egy és ugyan azon GPU-n történik. Ezek a folyamatok teljes mértékben kihasználják a GPU-n belüli multiprocesszorokat, melyeken további párhuzamosítás nem hajtható végre, csak a kernelek késleltetésének rovására. Ahhoz, hogy az említett folyamatokat hatékony módon tudjuk futtatni párhuzamosan, úgy célszerű bevezetni egy második GPU-t.

A 24. ábra szemlélteti, hogy az egyes modulok részei milyen sebességgel rendelkeznek egymáshoz viszonyítva. Érdeemes megemlíteni, hogy a pályadetektálás (világoszöld) és akadályészlelés (vörös) CPU-n futó, nem párhuzamosítható folyamat. Sebességük a többi

folyamathoz képest elhanyagolhatóan kicsi, képkockánként kicsi sebességingadozás tapasztalható. Ellentétben az objektumdetektálás lépéseivel és a pályadetektálás előfeldolgozásával, melyek ki tudják használni a GPU nyújtotta párhuzamosítási lehetőségeket. Ezzel egyidejűleg jelentős sebességingadozásokat mértem képkockánként.

A 2.2. fejezetben ismertetett hasonló rendszerek nem mindegyike valósítja meg a teljes akadályészlelést, leginkább csak részfolyamatait. A [6] rendszer pályadetektálási sebessége eléri a 25 FPS feldolgozási sebességet, mely 40 milliszekundumnak feleltethető meg. Az eredmény diagramon (23. ábra) látható, hogy a pálya előfeldolgozása (világos zöld) és detektálása (sötét zöld) együtt átlagosan 17 milliszekundum időt tesz ki, mely során megvalósul a teljes pályadetektálás. Elmondható, hogy rendszerem több mint kétszer gyorsabban képes ellátni a feladatot a [6] rendszerrel szemben.

A [4] rendszer teljes akadályészlelést valósít meg pályadetektálással együtt, mely során szintén 25 FPS végrehajtási sebességet tudtak elérni. Saját megvalósításomnak a végrehajtási idő közel 75%-át az objektumdetektálás teszi ki, ezért a [4] rendszernél közel 40%-kal lassabbnak mondható a ~17,5 FPS értékkel. További optimalizálással jobb eredményeket érhetők el.

Végezetül kizárólag objektumdetektálási szempontból a [9] rendszernél 26,3 FPS végrehajtási sebességet mértek, mellyel rendszerem objektumdetektáló modulja közel azonos idővel fut, majdnem pontosan 25 FPS-el. Sebesség és pontosság tekintetében megközelítőleg azonos eredményeket értem el objektumdetektálás terén.

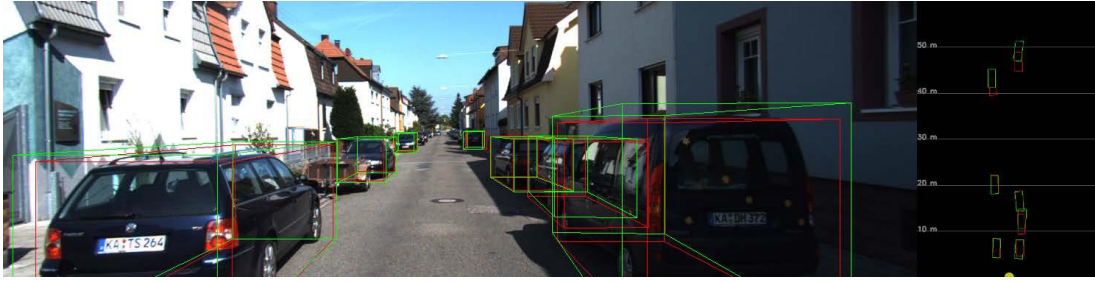
Összességiben elmondható, hogy a rendszer teljesítménye minden modult összesítve jónak tekinthető. A valós idejű végrehajtást korábban a kamera által szállított képek sebességével azonosítottam, mely relatív értelmet nyer az algoritmus teljesítményének meghatározásakor. Ha a kamera körülbelül 20 képkockát szolgáltat másodpercenként, akkor elmondható, hogy a rendszer közel valós idővel tud működni. Ellenben, ha másodpercenként több képkockát szállít a kamera, akkor a teljesítmény egyenesen arányosan csökken.

4.4. TOVÁBBFEJLESZTÉSI LEHETŐSÉGEK

Az eredmények ismertetése során nyilvánvalóvá vált, hogy az 4.2 alfejezetben bemutatott szélsőséges esetek az akadályészlelésre kiemelten negatívan hatnak, mely nem elhanyagolandó probléma. Továbbá, a rendszer teljesítményére számos optimalizálási lehetőséget eszközöltem, melyekkel egy elfogadható feldolgozási sebességet sikerült elérni, viszont további optimalizálással magasabb sebességet érhető el. A rendszer továbbfejlesztési lehetőségére az érzékelés pontosságának növelését és a végrehajtási sebesség növelését találtam optimálisnak, melyek tényleges megoldást nyújtanának a korábban említett problémákra.

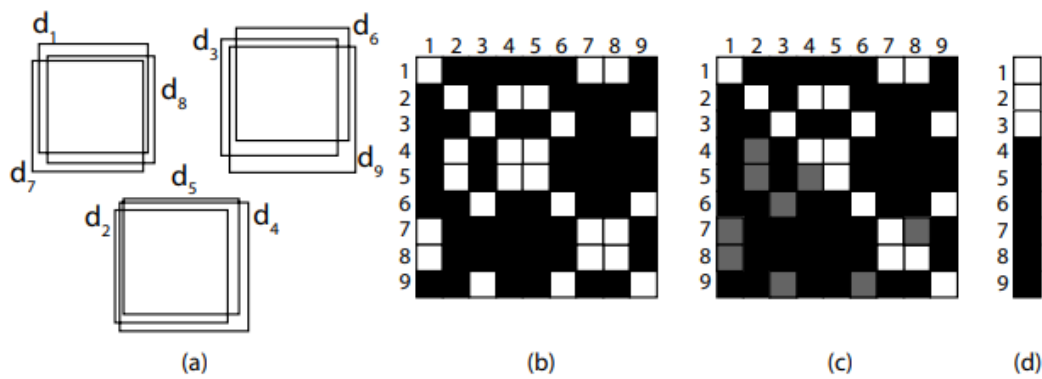
Az akadályészlelés során azokat a helyzeteket tekintem szélsőséges esetnek, ahol az észlelendő objektum az űrszelvény szélén tartózkodik. Problémát az jelenti, hogy az űrszelvény környezete ki van maszkolva és a maszkolás határán szereplő objektumok jellemzőinek egy része le van „nyesve”. Az objektumdetektálás így számos esetben nem történik meg, csak ha rendelkezésre áll kellő mennyiségű jellemző, például egy autó elejének felismeréséhez elegendő az elülső fényszóró és kerék (22. ábra). A maszkolásra azért volt szükség, mert számos esetben a szerelvény mellett elhaladó objektumok alakjának határait a YOLO által szolgáltatott befoglaló téglalapok nem reprezentálják helyesen, mely során fals akadálynak tekinti az elhaladó járművet a rendszer. Az optimális megoldás az volna, ha az objektumokat nem a 2D-s síkon kezelnénk, hanem a 3D-s térben. Ilyenkor nem egy síkbeli befoglaló téglalappal kellene dolgozni, hanem az objektumot magába foglaló téglatest sarokpontjai határoznák meg az objektum térbeli pozícióját. A pályadetektálás során detektált pálya szegmenseit térben egyértelműen el lehet helyezni, melyekhez lehetne viszonyítani a hexaéder koordinátáit. Az eredmény elméletben egy jóval pontosabb, kevesebb becslést igénylő akadályészlelő módszer lenne, melyet az űrszelvény teljes környezetére ki lehet vetíteni. A 3D objektumdetektálás egy módszerét fejlesztették ki 2021-ben, melyet a „*M3DSSD: Monocular 3D Single Stage Object Detector*” [28] cikkben ismertetnek. A rendszer képes 3D téglatestek prediktálására (25. ábra) valós időben, mely az ismertetett eredmények alapján körülbelül 6 ms végrehajtási idővel rendelkezik²¹. Az implementáció egyszerre nyújtana számottevő növekedést mind az akadályészlelés pontosságának növelésében, mind pedig a fejlesztett rendszer végrehajtási sebességében.

²¹ A megvalósított objektumdetektáló modul végrehajtási ideje átlagosan 40ms, összehasonlítva a bemutatott [31] rendszerével, mely közel hétszer gyorsabban teljesít.



25. ábra. Az M3DSSD eredményét reprezentáló 3D téglatek

Amennyiben a jelenlegi objektumdetektáló modult kívánjuk használni, úgy további optimalizálási lehetőségekkel lehet élni a sebesség növelésének érdekében. A 4.3 alfejezetben ismertetett sebesség eredményeknél látható volt, hogy a YOLO utófeldolgozása átlagosan 14 ms-re tehető. Ez a folyamat teljes mértékben CPU-n fut, melynek jelentős részét a YOLO eredmény mátrixának küszöbérték szerinti szűrését követően történő nem-maximumok elnyomása (NMS) képezi. A NMS hatékony párhuzamosítására 2016-ban kifejlesztettek egy módszert, mely a „*Work-efficient parallel non-maximum suppression for embedded GPU architectures*” [29] cikkben olvasható. A technika eredetileg beágyazott rendszerekre lett fejlesztve, de az ismertetett eredmények szerint hagyományos, nem beágyazott GPU esetén is körülbelül 2 ms végrehajtási idővel rendelkezik függetlenül a detektált objektumok számától, mely egy hétszeres gyorsulást eredményez YOLO utófeldolgozás során. Az NMS GPU-n történő végrehajtásának lépéseit a 26. ábra szemlélteti.



26. ábra. GPU-NMS [29] lépései²²

Általánosságban elmondható, hogy a képmanipuláló CUDA kernelek teljesítménye a GPU-ban megtalálható SIMD feldolgozó egységek számától függ. A fejlesztés során

²² (a) detektor által meghatározott egyedek; boolean mátrix (b) klaszterezést követően és (c) nem reprezentatív javaslatok szűrése; az (d) eredmény az AND redukciót követően

felhasznált Nvidia GTX 1080-as GPU 20 SM²³ multiprocesszorral rendelkezik. A pályadetektálás előfeldolgozása során alkalmazott szűrés és gradienskép létrehozása 608×275-ös felbontáson történik, melyet 164 darab 1024 szás blokkra tudunk bontani. Az említett blokk számot a GPU közel 9 iteráción keresztül tudja feldolgozni. Könnyen belátható, hogy az SM multiprocesszorok számának növelésével csökken az iterációk száma, ezért célszerű minél nagyobb multiprocesszor (SM) számmal rendelkező GPU-t használni kisebb végrehajtási idő érdekében.

Összességében elmondható, hogy a legtöbb gyorsítási lehetőség mögött a folyamatok GPU-n történő implementálása áll, viszont az optimális megoldást az említett M3DSSD 3D téglatest prediktáló technika implementálása jelentené mind teljesítményben, mind pedig az akadályészlelés pontosságának növelésében.

²³ Streaming multiprocessor, az Nvidia GPU-k SIMD feldolgozó egysége

5. ÖSSZEFOGLALÁS

Dolgozatomban egy olyan komplex biztonsági rendszer kidolgozásával és fejlesztésével foglalkoztam, mely aktív védelmet nyújt a tömegközlekedés kötöttpályás ágazatának. A városi környezetre fejlesztett akadályészlelő rendszerem a lehető legkevesebb szenzor felhasználásával nyújt valós idejű pályameghatározást és akadályészlelést a haladás során a vezetőfülkébe felszerelt RGB kamera képének feldolgozásával. Kutatásom során olyan összetett magas és alacsony szintű képfeldolgozási technikákat vizsgáltam, beleértve kortárs irodalomban fellelhető hasonló rendszerek módszereit, melyek segítettek a hatékony pálya és objektumdetektálás megvalósításában.

A megvalósított rendszert architektúráisan három modulra bontottam. Elsőnek a pályadetektálást végző modul meghatározza a kamera képen szereplő megfelelő pályaszakasz űrszelvényét számos képmanipuláló technikák összefűzésével. Elvégzem a kép szürkeárnyaltos konverzióját, melyet utána átméretezek és kivágok, hogy csak a feltételezett pálya területe szerepeljen benne. Ezután éldetektálással azonosítom az éleket és az egybefüggő élcsoportokat vizsgálva lentől felfele haladva meghatározom a pálya szegmenseket, mely eredményeül előáll a jármű előtt fellelhető pálya. Második lépésben az objektumdetektáló modul elvégzi az akadályt képezhető objektumok azonosítását az űrszelvény szűk környezetében. Amennyiben a képkockán szerepel detektált objektum, úgy a folyamat végén geometriai úton összevetjük az űrszelvénnel az objektum(ok) helyzetét. A vizsgálat során megállapításra kerül, hogy az objektum becsült alakja milyen kapcsolatban áll a megfelelő űrszelvénytérrel. Amennyiben az objektum alakja metszi az űrszelvény szakaszt vagy azon belül helyezkedik el, úgy a feldolgozását követően a rendszer jelzi, hogy a detektált objektum akadályt képez adott pillanatban.

A rendszert különböző fény- és pályaviszonyok között készült több órányi felvételen teszteltem. A pályadetektálási eredményekhez felhasznált, az eredeti felvételekből kivágott klipeket képkockáról képkockára elemeztem és értékeltem ki, eldöntve a pálya helyes vagy helytelen azonosítását. Az akadályészlelés eredményeinél is hasonló módon jártam el, a fennálló akadályokat vizsgáltam és helyes vagy helytelen azonosításukat rögzítettem. Az eredmények igazolják rendszerem hatékonyságát. Megállapításra került, hogy az adott kamerabeállítás mellett a rendszer a pályadetektálást 18 méteres távolságig közel 98%-os pontossággal valósítja meg fényviszonyoktól többnyire függetlenül. Ennek oka a feldolgozás során alkalmazott kép átméretezések és kivágásokból fakadó távoli sínszál jellemzők

elvesztése. Jelentős esőzést vagy havazást követően pályaszakasz detektálás nem, vagy csak korlátozva kivitelezhető. A működés jellemzői tovább javíthatók, ha a vezetőfülkében a kamerát és az alkalmazott optikát a feladat igényeinek megfelelően helyezik el, illetve állítják be. További hátráltató tényező volt a tesztelés során a felhasznált video tömörítettsége.

Hasonló rendszerekkel összevetve nagy hatékonyságot tanúsít saját rendszerem sebesség és pontosság terén. A valós idejű feldolgozást GPU-n működő részbeni megvalósítással közel sikerült kivitelezni, a rendszer 17-18 képkocka/másodperces sebességgel képes feldolgozni a kamera videófolyamát. A továbbfejlesztési javaslatok során ismertettem számos lehetőséget, melyek alkalmassá teszik a rendszert a valós idejű végrehajtás elérésében. Míg az objektumdetektálást egy általános célra fejlesztett neurális hálózat végzi, megoldásom így is megközelíti sebességben és pontosságban a hasonló rendszerek különböző technikáit. Az akadályészlelést bemutatott teszteredményeim alapján közel 84%-os pontossággal kivitelezte rendszerem. Mivel akadályészlelés során a mélységi adatok hiánya miatt egy becslést végzünk az objektum alakjára, ezért szélsőséges helyzetekben némi pontatlanságot tanúsít rendszerem, melyet az eredmények alátámasztanak. Az egyik ilyen szélsőséges eset leginkább az úrszelvény környékén végrehajtott maszkolás szélein tapasztalható. A szélek mentén elhelyezkedő objektumokat számos alkalommal nem- vagy pontatlanul azonosított. A másik problémát az egybefüggő objektumcsoportok okozzák, melyeken belül az egyes objektumok külön-külön történő azonosítása sokszor problémát jelent.

Összességiben elmondható, hogy sikerült megalkotni egy közel valós időben, nagy pontossággal működő akadályészlelő rendszert, mely egy színes kamera felhasználásával viszi végbe a pálya- és akadálydetektálást. Mivel a képfeldolgozást egy szenzor segítségével viszi végbe, ezért a rendszernek nagy számítási kapacitással rendelkező hardverre van szüksége.

A dolgozatban említett felvételek és feltüntetett eredmények elérhetők az alábbi Google Drive linken keresztül: https://drive.google.com/drive/folders/15UIFFnt7wIb0TXQVjFpfR-sU4_n2EcNw?usp=share_link

IRODALOMJEGYZÉK

- [1] Központi Statisztikai Hivatal, ksh.hu, „24.1.1.23. A közúti gépjárművek száma járműnemenként, az év végén,” 2021. [Online]. Available: https://www.ksh.hu/stadat_files/sza/hu/sza0023.html. [Hozzáférés dátuma: 5 március 2022].
- [2] Központi Statisztikai Hivatal, ksh.hu, „4.1.1.44. Közlekedési balesetek,” 2021. [Online]. Available: https://www.ksh.hu/stadat_files/ege/hu/ege0061.html. [Hozzáférés dátuma: 5 március 2022].
- [3] D. Ristić-Durrant, M. Franke and K. Michels, "A Review of Vision-Based On-Board Obstacle Detection and Distance Estimation in Railways," *Sensors*, vol. 21, no. 10, p. 3452, 2021.
- [4] H. Wang, X. Zhang, L. Damiani, P. Giribone, R. Revetria and G. Ronchetti, "Video Analysis for Improving Transportation Safety: Obstacles and Collision Detection Applied to Railways and Roads," *International MultiConference of Engineers and Computer Scientists*, pp. 909-915, 15-17. március 2017.
- [5] J. Canny, "A computational approach to edge detection," *IEEE Transactions on pattern analysis and machine intelligence*, p. 679–698, 1986.
- [6] F. Maire and A. Bigdeli, "Obstacle-free range determination for rail track maintenance vehicles," *2010 11th International Conference on Control Automation Robotics & Vision*, pp. 2172-2178, 2010.
- [7] R. O. Duda and P. E. Hart, "Use of the Hough transformation to detect lines and curves in pictures," *Communications of the ACM*, vol. 15, p. 11–15, 1972.
- [8] H. Mukojima, D. Deguchi, Y. Kawanishi, I. Ide, H. Murase, M. Ukai, N. Nagamine and R. Nakasone, "Moving camera background-subtraction for obstacle detection on railway tracks," *2016 IEEE International Conference on Image Processing (ICIP)*, pp. 3967-3971, 2016.
- [9] X. Yuchuan, G. Chunhai, Y. Lei, T. Simon and W. Guodong, "Real-time Obstacle Detection Over Rails Using Deep Convolutional Neural Network," *2019 IEEE Intelligent Transportation Systems Conference (ITSC)*, pp. 1007-1012, 2019.
- [10] W. Liu, D. Anguelov, D. Erhan, C. Szegedy, S. Reed, C.-Y. Fu and A. C. Berg, "SSD: Single Shot MultiBox Detector," in *Computer Vision – ECCV 2016*, Springer International Publishing, 2016, p. 21–37.
- [11] J. Redmon, S. Divvala, R. Girshick and A. Farhadi, *You Only Look Once: Unified, Real-Time Object Detection*, arXiv, 2015.
- [12] N. Kanopoulos, N. Vasanthavada and R. L. Baker, "Design of an image edge detection filter using the Sobel operator," *IEEE Journal of solid-state circuits*, vol. 23, p. 358–367, 1988.
- [13] Budapest Közlekedési Vállalat Zrt., „F.1.-F.2. SZÁMÚ JELZÉSI ÉS FORGALMI UTASÍTÁS A KÖZÚTI VASUTAK SZÁMÁRA,” 2016, p. 10.
- [14] G. M. H. Amer and A. M. Abushaala, "Edge detection methods," *2015 2nd World Symposium on Web Applications and Networking (WSWAN)*, pp. 1-7, 2015.

- [15] G. E. Sotak and K. L. Boyer, "The laplacian-of-gaussian kernel: A formal analysis and design procedure for fast, accurate convolution and full-frame output," *Computer Vision, Graphics, and Image Processing*, vol. 48, pp. 147-189, 1989.
- [16] A. Kumar and S. S. Sodhi, "Comparative Analysis of Gaussian Filter, Median Filter and Denoise Autoencoder," *2020 7th International Conference on Computing for Sustainable Global Development (INDIACom)*, pp. 45-51, 2020.
- [17] A. Agarwal, S. Gupta and D. K. Singh, "Review of optical flow technique for moving object detection," *2016 2nd International Conference on Contemporary Computing and Informatics (IC3I)*, pp. 409-413, 2016.
- [18] S. L. Hijazi, R. Kumar and C. Rowen, "Using Convolutional Neural Networks for Image Recognition," in *Cadence Design Systems, Inc.*, 2015.
- [19] R. Girshick, J. Donahue, T. Darrell and J. Malik, "Rich feature hierarchies for accurate object detection and semantic segmentation," *CoRR*, vol. abs/1311.2524v5, 2014.
- [20] M. Maity, S. Banerjee and S. Sinha Chaudhuri, "Faster R-CNN and YOLO based Vehicle detection: A Survey," *2021 5th International Conference on Computing Methodologies and Communication (ICCMC)*, pp. 1442-1447, 2021.
- [21] O. R. developers, *ONNX Runtime*, 2021.
- [22] G. Jocher, A. Chaurasia, A. Stoken, J. Borovec, NanoCode012, Y. Kwon, TaoXie, K. Michael, J. Fang, imyhxy, Lorna, C. Wong, 曾. Yifu), V. Abhiram, D. Montes, Z. Wang, C. Fati, J. Nadar, Laughing, UnglvKitDe, tkianai, yxNONG, P. Skalski, A. Hogan, M. Strobel, M. Jain, L. Mammana and xylieong, *ultralytics/yolov5: v6.2 - YOLOv5 Classification Models, Apple M1, Reproducibility, ClearML and Deci.ai integrations*, Zenodo, 2022.
- [23] M. Garland, S. Le Grand, J. Nickolls, J. Anderson, J. Hardwick, S. Morton, E. Phillips, Y. Zhang and V. Volkov, "Parallel Computing Experiences with CUDA," *IEEE Micro*, vol. 28, pp. 13-27, July 2008.
- [24] M. A. Haseeb, J. Guan, D. Ristic-Durrant and A. Gräser, "DisNet: a novel method for distance estimation from monocular camera," *10th Planning, Perception and Navigation for Intelligent Vehicles (PPNIV18), IROS*, pp. 139-144, 2018.
- [25] B. Bilgic, B. K. P. Horn and I. Masaki, "Efficient integral image computation on the GPU," *2010 IEEE Intelligent Vehicles Symposium*, pp. 528-533, 2010.
- [26] P. Jaccard, *Etude comparative de la distribution florale dans une portion des Alpes et du Jura*, Impr. Corbaz, 1901.
- [27] Lemezes01, „17-es villamos (Bécsi út VÁ. - Savoya park) / Budapest trams: Line 17,” 2019. [Online]. Available: <https://www.youtube.com/watch?v=tlePRA9EZQ0>.
- [28] S. Luo, H. Dai, L. Shao and Y. Ding, "M3DSSD: Monocular 3D Single Stage Object Detector," *CoRR*, vol. abs/2103.13164, 2021.
- [29] D. Oro, C. Fernández, X. Martorell and J. Hernando, "Work-efficient parallel non-maximum suppression for embedded GPU architectures," *2016 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pp. 1026-1030, 2016.

ÁBRA- ÉS TÁBLÁZATJEGYZÉK

1. ábra. Bazaltkőkockás pályán alkalmazott Canny operátor előtte és utána.....	9
2. ábra. Sínszálak összefutása	14
3. ábra. Ívelt pályán haladó villamos.....	15
4. ábra. Nyomcsatorna, tiszta (fent), homokolt (lent)	15
5. ábra. A vágányra merőleges űrszelvény (zöld)	16
6. ábra. Balról jobbra: Eredeti kép, Sobel 3×3, Sobel 3×3 és 5×5 kombinálva	18
7. ábra. Pályadetektálás lépései előfeldolgozás után.....	21
8. ábra. Detektált pálya.....	22
9. ábra. Objektum predikciók.....	25
10. ábra. Objektumot reprezentáló téglatest (piros) és 2D befoglaló téglalap (kék).....	27
11. ábra. Befoglaló téglalap átmenete ellipszisbe	28
12. ábra. Objektum helyzetének esetei az űrszelvényhez viszonyítva	29
13. ábra. Objektum legközelebbi pontjának meghatározása az űrszelvényhez 1/2	30
14. ábra. Objektum legközelebbi pontjának meghatározása az űrszelvényhez 2/2	30
15. ábra. Rendszer működésének folyamatábrája	32
16. ábra. Objektumdetektáló modul folyamatábra	33
17. ábra, Pályadetektálás betonlemezes pályaszakaszon	35
18. ábra. Felvételek feldolgozott és nyers képei, fentről lefele haladva: betonlemezes útburkolat, magasvágány, bazaltkőkockás burkolat, aszfalt és alagút	37
19. ábra. Gradiens és eredeti kép, fentről lefele; elszórtan hóval és vízzel fedett pályaszakasz	40
20. ábra. Felülnézeti pályadetektálás, balról jobbra, eredeti kép és detektált pálya.....	41
21. ábra. Akadályészlelés a második felvételen, 263. képkocka.....	42
22. ábra. Akadályészlelés az első felvételen, 281. képkocka	45
23. ábra. Modulok rétegjeinek összesített futási ideje képkockánként	47
24. ábra. Modulok rétegjeinek futási ideje képkockánként.....	48
25. ábra. Az M3DSSD eredményét reprezentáló 3D téglatestek	51
26. ábra. GPU-NMS [29] lépései	51
1. táblázat. Hasonló rendszerek összegzése pályaaazonosítás módja szerint	10
2. táblázat. Hasonló rendszerek összegzése objektumdetektálás alapján.....	11
3. táblázat. 18 méteres távolságot bezáró vizsgálat eredményei	36
4. táblázat. 22 méteres távolságot bezáró vizsgálat eredményei	38
5. táblázat. Akadályészlelés eredmények	44