



IPAR 4.0 INFORMATIKAI VONATKOZÁSAI

Egyetemi doktori (PhD) értekezés

Erdei Timotei István

Témavezető: Prof. Dr. habil Husi Géza

DEBRECENI EGYETEM

Természettudományi és Műszaki Tudományi Doktori Tanács

Informatikai Tudományok Doktori Iskola

Debrecen, 2025

Ezen értekezést a Debreceni Egyetem Természettudományi és Műszaki Tudományi Doktori Tanács, Informatikai Tudományok Doktori Iskola, Informatikai rendszerek és hálózatok ipari alkalmazásokkal programja keretében készítettem a Debreceni Egyetem műszaki doktori (PhD) fokozatának elnyerése céljából.

Nyilatkozom arról, hogy a tézisekben leírt eredmények nem képezik más PhD disszertáció részét.

Debrecen, 2025

.....

Erdei Timotei István

Tanúsítom, hogy Erdei Timotei István doktorjelölt 2019.- 2025. között a fent megnevezett Doktori Iskola, Informatikai rendszerek és hálózatok ipari alkalmazásokkal programjának keretében irányításommal végezte munkáját. Az értekezésben foglalt eredményekhez a jelölt önálló alkotó tevékenységével meghatározóan hozzájárult. Nyilatkozom továbbá arról, hogy a tézisekben leírt eredmények nem képezik más PhD disszertáció részét.

Az értekezés elfogadását javaslom.

Debrecen, 2025

.....

Prof. Dr. habil Husi Géza

AZ ÉRTEKEZÉS CÍME

Értekezés a doktori (Ph.D.) fokozat megszerzése érdekében
az informatika tudományágban

Írta: Erdei Timotei István okleveles mechatronikai mérnök

Készült a Debreceni Egyetem Informatikai Tudományok Doktori Iskola
Informatikai rendszerek és hálózatok ipari alkalmazásokkal programja keretében

Témavezető: Prof. Dr. habil Husi Géza

Az értekezés bírálói:

.....

.....

A bírálóbizottság:

elnök:

tagok:

.....

.....

.....

Az értekezés védésének időpontja: 2025.

Tartalomjegyzék

AZ ÉRTEKEZÉS CÍME	3
Tartalomjegyzék	4
Rövidítések angol nyelven	6
Rövidítések magyar nyelven	9
Ábrajegyzék.....	12
1. Tématerület bemutatás.....	14
1.1. Bevezetés	15
1.2. Probléma konkretizálása	16
1.3. Szakirodalmi áttekintés	18
2. Ipari robotkar kiberfizikai változatának létrehozása.....	25
2.1. KUKA KR5 6 szabadságfokú ipari robotegység	26
2.2. 3D CAD modellezés és importálás	27
2.3. Stabil diffúziós generatív text-to-image mesterséges intelligencia 31	
2.4. Dreambooth alapú pályatérkép és textúra generálás	33
2.5. Inverz és direkt kinematika alkalmazása	38
2.6. Tengely specifikus mozgások PTP, LIN és CIRC	41
2.7. Virtuális mérés elvégzése és elemzés	43
2.8. Fejezet konklúziója	45
3. Digitális-ikerpár tanulóközpont tervezése a robotkar oktatáshoz	46
3.1. A koncepció kialakításhoz vezető események	46
3.2. CPS LAB elemek 3D CAD modellezése	50
3.3. Kiberfizikai és intelligens rendszerek robotlaboratóriuma	52
3.4. Szimulációs keretrendszer és HMI implementáció	55
3.5. KUKA KR5 Digital-ikerpár vezérlés	61
3.6. A Digitális-ikerpár program oktatásra gyakorolt hatásának vizsgálata.....	69
3.7. Fejezet konklúziója	73

4. Kiterjesztett valóság alkalmazás a KUKA KR5 ipari robotegység vizualizációjához	74
4.1. Robot fájl formátum és Unity fejlesztői környezet	74
4.2. A munkatér és grafikus felület vizualizáció.....	77
4.3. KUKA KR5 beágyazása kiterjesztett valóság alapú markerbe ...	78
4.4. Fejezet konklúziója	82
5. Mélytanulás alkalmazása valós időben objektum felismerése céljából SCARA gépegységen.....	85
5.1. SONY SCARA SRX-611 robotegység	85
5.2. Robot kontrol és programozási nyelv.....	89
5.3. SONY SCARA SRX-611 3D CAD modellezése.....	93
5.4. Adatállomány előállítása renderelési technikákkal	95
5.5. Alkalmazott mélytanulási architektúra	99
5.6. Tanítás és kiértékelési folyamat.....	102
5.7. Mélytanulás alapú objektum felismerés és kísérleti eredményei	
104	
5.8. Fejezet konklúziója	110
6. Irodalomjegyzék.....	111

Rövidítések angol nyelven

Jelölés	Megnevezés, megjegyzés
DT	D igital T win
CPS	C yber- P hysical S ystem
AR	A ugmented R eality
DL	D eep L earning
2D	2 D imension
CAD	C omputer A ided D esign
CAM	C omputer A ided M anufacturing
3D	3 D imension
MIT	M assachusetts I nstitute of T echnology
PDP-6	P rogrammed D ata P rocessor model 6
BD	B ig D ata
ISO	I nternational O rganization for S tandardization
IP	I nternet P rotocol
API	A pplication P rogramming I nterface
REST	R epresentational S tate T ransfer
IoT	I nternet o f T hings
AGV	A utomated G uided V ehicle
AI	A rtificial I ntelligence
ML	M achine L earning
FFNs	F eed F orward N eural N etworks
RNNs	R ecurrent N eural N etworks
SL	S upervised L earning
UL	U nsupervised L earning
MDPs	M arkov d ecision p rocess
CGI	C omputer- G enerated I magery
.STL	S tandard T riangle L anguage
.OBJ	W avefront O bject
.FBX	F ilmbox
.ASM	A ssembly M odel L ist
.3DS	3D S tudio M esh F ormat
.SLDPRT	S L W orks P art
CMU	C arnegie M ellon U niversity
RT	R eal- T ime
DOF	D egree o f F reedom
PC	P ersonal C omputer
FDM	F used D eposition M odeling

Voip	Voice over Internet Protocol
.CKPT	Checkpoint file
DNNs	Deep Neural Networks
GUI	Graphical User Interface
Web	World Wide Web
URL	Uniform Resource Locator
CFG	Classifier-Free Guidance Scale
HSV	Hue, Saturation, Value
U-Net	Convolutional Neural Network for Image Segmentation
TCP	Tool Center Point
Vs	Versus
HMI	Human Machine Interface
SMEs	Small and Medium-sized Enterprises
PTP	Point-to-Point
LIN	Linear
CIRC	Circular
VR	Virtual Reality
FPS	First Person Camera
UMG	Unreal Motion Graphics
CMD	Command
SUS	System Usability Scale
URDF	Unified Robotics Description Format
UE	Unreal Engine
ROS	Robot Operating System
OS	Operating System
SCARA	Selective Compliance Assembly Robot Arm
GNU	General Public
LTS	Long Term Support
CPS-LAB	Cyber-Physical and Intelligent Systems Robot Laboratory
E-U	End-User
TETS	Text Encoder Training Steps
REPO	Repository
CLIP-TE	CLIP Text-Encoder
ITC	Image-Text-Compare
EOS	Embedded Operating System
IFR	International of Robotics
TP	Teach Pendant
CtS	Convert to Smart
MS-DOS	Microsoft Disk Operating System
SSP	Synchronous Serial Port

LED	L ight E mitting D iode
Oc-BVH	O ptimal c ompact BVH
BVH	B ounding V olume H ierarchy
SSE	S teaming SIMD E xtensions
IDE	I ntegrated D evelopment E nvironment
RL	R einforcement L earning
ROS	R obotic O perating S ystem
.APK	A pplication P ackage file

Rövidítések magyar nyelven

Jelölés	Megnevezés, megjegyzés
DT	Digitális-Ikerpár
CPS	Kiberfizikai Rendszerek
AR	Kiterjesztett Valóság
DL	Mélyreható Tanulás
2D	Két Dimenzió
CAD	Számítógéppel Támogatott Tervezés
CAM	Számítógéppel Támogatott Gyártás
3D	Három Dimenzió
MIT	Massachusettsi Technológiai Intézet
PDP-6	Programozott Adatfeldolgozó modell 6
BD	Nagy Adathalmaz
ISO	Nemzetközi Szabványügyi Szervezet
IP	Internet Protokoll
API	Alkalmazás Programozási Interfész
REST	Reprezentációs Állapotátvitel
IoT	Tárgyak Internete
AGV	Automatizált Irányítás Jármű
AI	Mesterséges Intelligencia
ML	Gépi Tanulás
FFNs	FeedForward Neurális Hálózatok
RNNs	Rekurrens Neurális Hálózatok
SL	Felügyelt Tanulás
UL	Felügyelt nélküli Tanulás
MDPs	Markov Döntési Folyamat
CGI	Számítógéppel Generált Képek
.STL	Szabványos Háromszög Nyelv
.OBJ	Hullámfront Objektum
.FBX	Filmdoboz
.ASM	Összeszerelési Modell Lista
.3DS	3D Stúdió Hálós Formátum
.SLDPRT	SL Munka Elemek
CMU	Carnegie Mellon Egyetem
RT	Valós-Idejű
DOF	Szabadság fok
PC	Személyi Számítógép
FDM	Szálolvastásos Módszer

Voip	I nternetes H angátviteli P rotokoll
.CKPT	E llenőrző fájl
DNNs	M ély N eurális H álózatok
GUI	G rafikus F elhasználói F elület
Web	V ilágháló
URL	E gységes E rőforrás K ereső
CFG	O sztályozómentes I ránymutatási S kála
HSV	Á rnyalat, T elítettség, F ényesség
U-Net	K onvulúciós N eurális H álózat K épszegmentálásra
TCP	S zerszám K özéppont
Vs	V erzus
HMI	E mber G ép I nterfész
SMEs	K is- és K özépvállalkozások
PTP	P ontról- P ontra
LIN	L ineáris
CIRC	K örkörös
VR	V irtuális V alóság
FPS	B elsőnézetű K amera N ézet
UMG	U nreal M ozgás G rafika
CMD	P arancs
SUS	R endszer H asználhatósági S kála
URDF	E gységesített R obotika L éírási F ormátum
UE	U nreal M otor
ROS	R obot O perációs R endszer
OS	O perációs R endszer
SCARA	S zelektív M egfelelőségi Ö sszeszerelő R obotkar
GNU	Á ltalános N yilvános L icenc
LTS	H osszú T ávú T ámogatás
CPS-LAB	K iberfizikai és I ntelligens R endszerek R obotlaboratóriuma
E-U	V égfelhasználó
TETS	S zövegkódoló B etanítási L épések
REPO	T ároló
CLIP-TE	CLIP S zövegkódoló
EOS	B eágyazott R endszerek
IFR	R obotika N emzetközi S zövetsége
TP	T anító E szköz
CtS	F elokosítás
MS-DOS	M icrosoft L emezes O perációs R endszer
SSP	S zinkron S oros P ort
LED	F énykibocsátó D ióda

Oc-BVH	Optimális Kompakt BVH
BVH	Határoló Térfogat Hierarchia
SSE	Streaming SIMD Kiterjesztések
IDE	Integrált Fejlesztői Környezet
RL	Megerősítéssel Tanulás
ROS	Robotikai Operációs Rendszer
.APK	Android alkalmazás csomag

Ábrajegyzék

1. ábra. A számítástechnika és a mesterséges intelligencia kapcsolata [40]	23
2. ábra. 3D nyomtatott KUKA KR5 adapter.....	27
3. ábra. 3D CAD formátumok kezelése	28
4. ábra. Ipar Napjai alkalmazás – 2021	29
5. ábra. KUKA KR5 csoportosított 3D CAD modell	30
6. ábra. Importált robotcella modell.....	31
7. ábra. Stabil diffúzió kép generálás leírással	32
8. ábra. Adatkészlet előállítása rendereléssel.....	34
9. ábra. A kép generálás folyamata.....	36
10. ábra. Pályatérkép textúra generálása saját adatkészlet alapján.....	36
11. ábra. Az eredeti és a generált pályatérképek összevetése	37
12. ábra. Robot modell generált textúrája.....	38
13. ábra. KUKA KR5 csuklók és tengelyek	39
14. ábra. IK script meghívása	41
15. ábra. LIN mozgás végrehajtása	42
16. ábra. CIRC mozgás végrehajtása	43
17. ábra. TCP elmozdulás és csuklók elfordulásának mérése.....	44
18. ábra. Kiberfizikai és intelligens rendszerek robotlaboratóriuma	48
19. ábra. Kiberfizikai rendszer architektúra	49
20. ábra. KUKA KR5 3D CAD modell.....	52
21. ábra. Kutató létesítmény 3D modellje	53
22. ábra. CPS labor Digitális-iker	54
23. ábra. CPS labor robotcella elemei	54
24. ábra. KUKA KR5 importált robotcella	56
25. ábra. Robotcella kamerák közötti váltásának programrésze	57
26. ábra. Virtuális ember-gép interfész	58
27. ábra. KUKA KR5 virtuális felhasználói felület	59
28. ábra. A virtuális felhasználói felület program rész.....	59
29. ábra. A programozó eszköz virtuális felhasználói felülete	60
30. ábra. Animációs gráf a vizuális szkriptben	62
31. ábra. Tengely rotációs érték változtatása az animáció vezérlőben	63
32. ábra. PTP, LINC és CIRC tengely specifikus mozgások.....	64
33. ábra. Kör középpont számítás Unreal Engine tervezőben	67
34. ábra. CIRC mozgás végrehajtása a kiber-térben	68
35. ábra. Robot kontrol a kibertérben	69

36. ábra. Alapszintű pályabejárási lap.....	70
37. ábra. Feladat végrehajtása a valóságban és Digitális-ikerben	71
38. ábra. KUKA KR5 CAD modell Unity fejlesztői környezetben	76
39. ábra. KUKA KR5 URDF fájl	76
40. ábra. Munkatér és grafikus felület a KUKA KR5 robot kontrolhoz.....	77
41. ábra. Kiterjesztett valóság alkalmazása és jelölő detektálás [94]	79
42. ábra. AR koordináta rendszer [96]	80
43. ábra. KUKA KR5 pozíció digitális jelölőkártya.....	80
44. ábra. Android alkalmazás kiterjesztett valóság alapú KUKA robot vezérléshez.....	81
45. ábra. Fizikai és virtuális robot pozíció felvétele	82
46. ábra. Telefonhasználati szokások és a kiterjesztett valóság applikáció kezelése.....	83
47. ábra. Sony SCARA SRX-611 CPS-LAB robotegység	86
48. ábra. SCARA SRX-611 munkatér és tengely paraméterek	88
49. ábra. Szabványosított CH341 USB-to-RS232 soros interfész	90
50. ábra. SRX PLATFORM emulált Windows XP környezetben.....	92
51. ábra. Neurális háló adatkészlet generálásának metodikája	94
52. ábra. Sony SCARA SRX-611 3D CAD modell.....	95
53. ábra. Virtuális objektum szkennelés	97
54. ábra. Renderelt virtuális objektumok gépi tanuláshoz	98
55. ábra. Címkézés és az alkalmazott szín értékkészlet.....	98
56. ábra. A rögzített dobozok számított értékei k-means++ cluster algoritmust követően	100
57. ábra. A detektor architektúra rétegekkel	101
58. ábra. Robonet-RT detektor veszteség- és hitelesítő görbéje (mAP).....	106
59. ábra. Robonet-RT detektor által felismert és osztályozott objektumok....	108
60. ábra. Robonet-RT detektor által felismert és osztályozott objektumok (gyengébb képminőség).....	109
61. ábra. Robonet-RT detektor – objektum pozíció csere esetén	109

1. Tématerület bemutatás

A disszertáció témaköre több az informatikával és a robotikával kapcsolatos területet is magába foglal. Ezek közül kiemelten foglalkozik ipar 4.0 területtel, amelynek általános jellemzői közé tartozik a teljes és feltétel nélküli digitalizáció és a magas szintű automatizálás. Az intelligens rendszerek kialakítása azonban számos kompromisszumot követel meg a felhasználórészéről az elérhető szabványok miatt.

Az ipar 4.0 informatikai vonatkozásai című disszertáció innovatív labor fejlesztéseket mutat be az oktatás hatékonyságának növelésének érdekében. A vizsgált tématerület magába foglalja a fizikai formában megvalósított robotika labor fejlesztéseit, és a virtuális terek megtervezését és azok alkalmazását. A labor fejlesztések tekintetében egyfajta önálló ökoszisztéma kialakításra helyeztük a hangsúlyt, hogy a hallgatók naprakész tudást szerezhessenek az informatika és a robotika területén. A megvalósított fejlesztések közül az egyik meghatározó technika a digitális ikerpár volt, ami az informatika és a mérnöki tudományok metszéspontjában helyezkedik el. Az általunk végrehajtott labor digitalizáció során a több tengelyes robotkar és az ember-gép interfész viselkedésének megvalósítására fókuszáltunk.

A labor digitalizált elemkészlete lehetőséget szolgáltatott arra, hogy kiterjesztett valóság alapú informatikai megoldásokat alkalmazzunk. Az eljárás segítségével alapszintű robot mozgásokat hajtottunk végre, amihez saját jelölő adatkártya is tervezve lett. Így a felhasználó egy adott jelölőn tudja elérni a szükséges robotmodellt és az adott gépegység általános paramétereit.

Informatikai témakörben, a régi gépegységek felokosítása céljából egy módszert is kidolgoztunk, hogy képesek legyenek mély tanulás alkalmazására. Az adatkészlet előállításához az általunk megtervezett 3D referencia kockák modelljei

szolgáltatták az alapot, amelyeket különböző minőségben és szögekben elforgatva rendereltünk és címkéztünk a neurális háló tanításhoz.

A labor fejlesztések mindegyike informatikai vonatkozásokban kerültek megvalósításra az ipar 4.0 alkalmazásában.

1.1. Bevezetés

Az iparban bekövetkezett változások átértelmezték a gyártással kapcsolatos fogalmakat. Az előtérbe kerülő vevői igények kiszolgálása révén, a méretgazdaságosság helyét fokozatosan elkezdte felváltani a választékgazdaságosság. Ez egyben azt is jelenti, hogy az adott meglévő gyártási folyamatoknak flexibilisen kell reagálniuk annak érdekében, hogy a folyamatosan változó vevői igényeket ki tudja elégíteni, ami egy teljesen más, a korábbiaktól eltérő gondolkodásmódot tesz szükségessé. Az ipar fejlődése jól szegmentálható és az egyes elért fejlettségi szintek egymáshoz képest vizsgálhatók annak függvényében, hogy az adott korban mi indukált radikális újra szerveződést a gyártástechnológiában és a társadalomban. Történelmi vonatkozásban kijelenthető, hogy a technológia rohamos fejlődése megköveteli a folyamatos innovációt az ipart meghatározó szereplőktől, így fenntartva annak versenyképességét.

Az ipar 4.0 kihívásokat támaszt a kor végzett mérnökeivel szemben, mivel a technológiai fejlődés exponenciális görbe mentén halad. A jelenkor számára ez előnyt jelent, hiszen egyre több és egyre kifinomultabb eszközök és algoritmusok állnak rendelkezésünkre ahhoz, hogy minél hatékonyabban, minél több feladatot tudjunk sikeresen elvégezni, ezzel szolgálva a társadalom egészét a fejlődés útján. Ugyanakkor az oktatási részről is magasabb szintű adaptációt tesz szükségessé, hiszen az oktatási módszertant is képesek újra értelmezni az új technológiai vívmányok.

A bemutatásra kerülő disszertációban egy a Debreceni Egyetem, Műszaki Kar, Járműmérnöki Tanszékén kialakított „minta” ipar 4.0 robotlabor kialakításának és

fejlesztéseinek metódusai kerülnek tárgyalásra. A Kiberfizikai és intelligens rendszerek robotlaboratóriuma (Cyber-Physical System & Intelligent Robot Systems Laboratory) kialakításának célja, hogy olyan új technológiák alkalmazásának teremthessünk stabil alapot, amelyek új utakat nyithatnak meg a mesterséges intelligencia és a robotika területén. Továbbá, hogy ipari környezetet „emulálva/szimulálva” lehessen új optimalizálási eljárásokat kidolgozni, amelyek eredményes tesztelésüket követően integrálhatók-átültethetők lesznek más gyártási folyamatokba.

A XXI. században bekövetkezett változások próbára teszik az adaptációs képességünket, mely változásokat, ha képesek vagyunk időben lekövetni, sikerrel tudjuk megoldani a felmerülő akadályokat.

A fentiek tükrében új és régi technológiák kerültek alkalmazásra, kombinálásra innovatív módon. Fontos megemlíteni, hogy a Debreceni Egyetem, mint intézmény elsősorban nem terméket állít elő, hanem olyan gondolkodó értelmiséget, akiknek feladata, hogy a megszerzett és elsajátított tudásukkal a jövőt formálják. A kialakított labor az oktatásban is szerepet vállal, ezért egy kiberfizikai rendszer is létrehozásra került. A labor Digitális-ikerpár mása a kiterjesztett valóságot és a mélytanulást is magában foglalja.

1.2. Probléma konkretizálása

Az általam végzett vizsgálatokhoz a Debreceni Egyetem, Műszaki Kar, Járműmérnöki Tanszékhez tartozó labor került esettanulmányként kiválasztásra. Kijelenthető, hogy a technológiai fejlődés kihívás elé állítja az oktatási rendszert, mivel felveti a rendszeres és folyamatos „frissítés” igényét. A fejlődési ütem egyben lerövidíti az oktatási segédletek életciklusát is. Az 50-100 évvel ezelőtt íródott, és az oktatásban alkalmazott anyagok elavulási ideje régebben jóval hosszabb volt, ma már akár fél évente is frissítésre szorulnak, specifikus tudományterülettől függően. Elképzelhető, hogy az írás alapú betűvetésen túl, mélyebb szintű frissítés válik szükségessé az adott oktatási témakörökben, amelyek között különbséget kell tenni úgy, hogy hardveres vagy szoftveres

változásokat foglal e magában, vagy esetleg mindkettőt. Ezek a változtatások-frissítések azonban nem minden esetben kivitelezhetőek, egyéb, rajtunk kívül álló technikai korlátok miatt.

A problémák sorát gyarapítja, hogy a labor helység mérete mindössze 28,6 [m²]. Alapesetben ez az érték a teljes rendelkezésre állási területet jelöli, azonban a laborban elhelyezett ipari robotok munkaterülete kivonódik, hiszen a mindenkori operátor-felhasználó nem tartózkodhat a gépek munkaterében a balesetvédelmi szabályok miatt. A fenti szempontok figyelembevételével a hallgatói létszám mindösszesen 12 főre korlátozódott.

Szintén komoly kihívás, hogy a régi ipari gépegységek hogyan és milyen módszerekkel alakíthatók át ipar 4.0 kompatibilissé. A labor számos olyan géppel rendelkezik, amely mechanikailag teljesen ép, azonban a specifikációja nem teszi lehetővé a hálózaton történő mérést és kommunikációt. Az iparban is sok olyan gép található, amelyek szerves részét képezik a gyártásnak, és egyszerűen nem lehetséges ezek teljes lecserélése vagy kiváltása a hivatalos támogatás megszűnését követően.

Az elmúlt évek tapasztalatai bebizonyították, hogy történhetnek olyan változások, amelyek globális kihatással lehetnek az egész világ működésére, ezekre direkt módon, illetve indirekten készülni, reagálni szükséges. A COVID-19 -es világjárvány teljesen átírta azon törekvéseinket, amelyek csak lokálisan tették volna elérhetővé a kutató labort a hallgatók számára.

Egyrészt a labor befogadó képessége maximum 12 fős csoport lehetett, másrészt, mivel az üzembe állított gépek hőterhelése működésük során magas, 30-40 percenként szünet elrendelése vált szükségessé. Az utazási korlátozások tovább akadályozták a laborban folyó munkát, valamint belső fejlesztések is hiányt szenvedtek, mivel egyes időszakokban az intézmény területére való belépés is tiltás hatálya alatt volt.

A laborban végzett fejlesztések és átalakítások esettanulmányként rámutattak és egyben szorgalmazták, hogy megoldást találjunk fentebb vázolt problémákra, konkrétan:

- A helység méretéből adódó alacsony létszám és eszköz-gépkezelési korlátok kiküszöbölése
- A COVID-19 -es világjárvány okozta korlátozások ellenére az oktatás színvonalának megőrzése a laborban történő munkát érintő tantárgyak esetében.
- A labor Digitális-ikerpár másolatának létrehozása teszteléshez.
- Ipar 4.0 technikák alkalmazása régebbi gépegységeknél.
- Adatállományok előállítása. képelemzési és objektum detektálás céljából mélytanulás alapon.
- Kiterjesztett valóság alkalmazása specifikus robotkar esetén és saját marker előállítása.

1.3. Szakirodalmi áttekintés

Minden esetben, mikor a tervezésre gondolok, Mózes teremtés könyvének I. A Teremtés fejezete jut eszembe [1]. A teremtés iránti vágy egyidős az emberiséggel, és a világ többi vallásában is megjelenik valamilyen formában, hogy az ember szeretne teremtőjével egyenrangúvá válni az alkotás tekintetében, amit tökéletesíteni kíván a végletekig. Itt jelenik meg az innováció, amely összefonódik az emberi találékonysággal, és az ebből fakadó technológiai fejlődéssel.

Ez a fajta folyamatos fejlődésre való törekvés érhető tetten az iparban, kezdve a kézi manufaktúráktól egészen az autonóm gyártásig. Minden végbement ipari „generáció” váltásnak megvoltak a maga vívmányai, amivel az utókor ma azonosítja [2].

Az ipar 1.0, 1784-től vette kezdetét amikor a gőz-víz, mint segédenergia került kiaknázásra egyszerű gépek működtetése érdekében. A XVIII. század vége felé megindult gépesítési folyamat fokozatosan elkezdte felváltani a tradicionális manufaktúrákat [3].

A textil iparban pedig megjelent gőzüzemű szövőgép, ami a kelmeképző gépek közül a kor egyik legfontosabb vívmánya volt [4]. Richard Trevithick 1804-ben megalkotta az első gőzmozdonyt, amit „Pen-y-Darren Locomotive” néven ismert meg a világ [5], aminek maximális sebessége 3.9 [km/h] volt [6].

Az 1870-es évektől számítják az ipar 2.0 -át [7], aminek egyik fő mozgatórugója az autóipar volt. Az 1903-ban megalapított, és a Henry Ford [8] által vezetett, Ford Motor Company [9], egy sor új eljárást dolgozott ki, a termelés hatékonyságának növelésének érdekében. A tömegtermeléshez futószalagos rendszert vezettek be, a Ford T-model [10] gyártásához [11]. A méretgazdaságoságnak meg voltak az előnyei, a termelési költségek csökkentek, később pedig más ipari ágazatok is átvették a futószalagos rendszert.

Az ipar 3.0 [12] korszakában jelentek meg a félvezetők, és a CAD/CAM rendszerek [13] (Computer Aided Design, Computer Aided Manufacturing) [14], ami számítógéppel segített tervezést és gyártást jelentett. A szoftveres és hardveres korlátok miatt, ekkor még csak jellemzően 2D tervek készültek, azonban a 60-as évektől már 3D CAD terveket alkalmaztak [15].

A robotika és a programozható memóriával ellátott vezérlések terén szintén ezen időszakban indultak meg fejlesztések kutatási intézményekben. Ekkor jelentek meg a ma ismert robotkarok előfutárai, amelyeket később a gyártásban, valamint élelmiszeriparban és a hadiparban is alkalmaztak.

1963-ban a Kaliforniában található Los Amigos Hospital intézmény kutatói hozták létre orvosi feladatok ellátására a „Rancho Arm” -ot. A robotkar feladata az volt, hogy hat ízülete segítségével minél pontosabban imitálja az emberi kar mozgását [16].

1968-ban, az MIT egyik kutatója Marvin Minsky kifejlesztett egy „Tentacle arm” nevet viselő robotegységet, hogy az 12 csuklójával akadályokat tudjon kikerülni a munkaterén belül. A kar maga hidraulikus volt és egy PDP-6 (Programmed Data Processor model-6) [17] látta el a vezérlési feladatokat [18].

David Silver professzor 1974-ben megalkotta a „Silver Arm” nevet viselő robot szervo kart a Massachusetts Institute of Technology kutatói részlegen. A

robotegység képes volt a szerelési feladatokhoz szükséges összetett finom mozgások végrehajtására [19].

A negyedik ipari forradalom (Industry 4.0) koncepcióját 2011-től kezdték el kidolgozni [20]. Az ipar 4.0 egy magasabb szintű, tudatos gyártástervezést és optimalizálást irányoz elő, ezzel indukálva további fejlesztéseket és kutatásokat, amelyek jelenleg is a megvalósítás tárgyát képezik. A XXI. században az egyes iparágak gyorsan változnak, ennek nyomán követése pedig összekapcsolt digitalizációt követel meg.

Az egyik legfontosabb megvalósítandó kihívás a nagy adathalmazok (Big Data) kezelése. A megnevezés hatalmas mennyiségű strukturálatlan adatot jelent. Az iparban a gyártósorokon, a termelési ciklus során rengeteg adat keletkezik, amik továbbítása és feldolgozása komoly hardveres kihívásokat támaszt a szolgáltatókkal-gyártókkal szemben. A gazdaságban is megjelenik ez a fogalom, mivel adatok birtokában a piaci trendek is lekövethetőek [21].

Fontos kiemelni, hogy a Big Data egyfajta szintézise az új technológiáknak. A képződött adathalmazok lehetőséget adnak, hogy elemezve őket, a rendszerek „gyenge” pontjait lokalizálva, optimalizálási eljárásokat dolgozhassunk ki. Amennyiben az adathalmazok nagyobb idő intervallumot fednek le, előre jelezhetővé válhatnak specifikus események, meghibásodások, gépi viselkedések. A nagy adathalmazokat (Big Data) három egymástól jól különíthető jellemző határozza meg, amelyre a szakirodalom 3V -ként hivatkozik, ezek: Volume (mennyiség), Velocity (sebesség) és a Variety (variáció) [22].

A fogalom először 1998-ban jelent meg John Masley által [23]. Jellemzően a folyamatosan bővülő adatállomány miatt szervereket alkalmaznak, ezen adatok kezelésére-feldolgozására. Ezekben az esetekben a kommunikációs réteg a meghatározó, mivel az adatbázisokból az értékes adatok kinyerése és elérése internet protokollon keresztül (IP) keresztül történik. Az általános API-k (Alkalmazás Programozási Interfész) mellett a REST (Reprezentációs Állapotátvitel) is alkalmazzák, ami egy API csomagnak tekinthető és kapcsolatot biztosít a szerverek-proxy-k között.

A gyűjtött adatok feldolgozhatósága és értelmezhetősége fontos szempont, ezért is alkalmaznak vizualizációs technikákat, amelyekkel az adatok grafikonokba, diagramokba és gráfokba rendezhetők.

Szintén a kor vívmányai az IoT, más néven a dolgok internete [24]. A megnevezés a fizikai rendszerek köré épített és üzemeltett hálózatot jelenti, ahol egymástól független IP-MAC cím azonosítóval rendelkező elektronikai eszközök képesek kommunikációt végrehajtani interneten vagy belső hálózaton. Az intelligens eszközök a kommunikáció során információ cserét hajtanak végre.

Az IoT megnevezést 1999-ben Kevon Ashton használta először [25], azonban már az 1980-as években is voltak törekvések eszközök hálózatba történő csatlakozására [26].

Az IoT technológia segítségével maguk a termékek is elláthatók digitális azonosítókkal, amiket nyomon lehet követni az adott gyártósoron. A címkéhez leggyakrabban vonalkód, QR-Code [27] vagy RFID [28] azonosítót alkalmaznak.

A dolgok internete lehetőséget szolgáltat arra, hogy gyorsan reagáljunk a rendszert érintő változásokra, pl. meghibásodásra, hirtelen változó a hőmérséklet tartományra vagy ha egy adott autonóm mobil robot (AMR) útját keresztezi egy másik objektum.

A XXI. meghatározó technológiai vívmánya, amely szintén az ipar 4.0 részét képezi, az a mesterséges intelligencia (AI). Fontos, hogy különbséget tegyünk a tényleges mesterséges intelligencia használatáról és egy adatbázis kezelése között. Az AI rendszerek fejlesztésének meg vannak az alapvető nehézségei. A legfontosabb a megfelelő minőségű és nagyságú adatbázis, hiszen a mindenkori mesterséges intelligenciának meg kell felelnie olyan kihívásoknak, ami még az emberi agy számára is túl komplex.

Funkcionalitás szempontjából nem elhanyagolható tény, hogy pl. egy sakk játszmában a gépi ellenfél valóban „gondolkodási” feladatot hajt e végre vagy pusztán egy betáplált, a nyelvére átfordított, digitalizált sakk-könyv szabályait követve lépésről, lépésre reagál.

A mesterséges intelligencia egy komplex fogalom, ezért számos definíciója ismert. A kifejezés maga az 1957-es években John McCarthy által került meghatározásra:

„*The science and engineering of making intelligent machines*” [29].

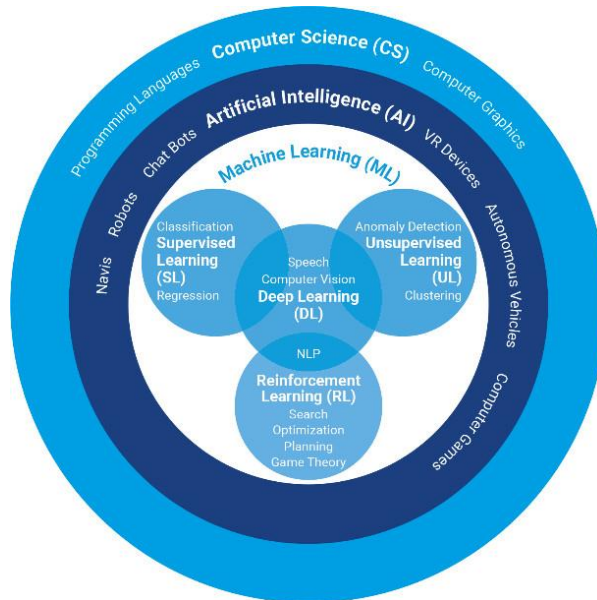
Az egyik fő ága a mesterséges intelligenciának a gépi tanulás (Machine Learning) [30,31]. A gépi tanulás statisztikai algoritmusok felhasználásával tesz általánosíthatóvá komplex számítási feladatokat, explicit utasítások nélkül. Képes adathalmazokból azonosítani keresett tipizált mintákat. Az ML elsőként 1959-ben került definiálásra Arthur Samuel által [32].

A másik részterülete a mesterséges intelligenciának a DL azaz mélytanulás (Deep Learning) [33]. A speciális tudomány terület neurális hálózatok alkalmazásán alapul, amelyek az emberi gondolkodás során használt neuronhálózatok modelljét hivatott lemodellezni. A mélytanulási technika egészen 1943-ig vezethető vissza, amikor két kutató Warren McCulloch és Walter Pitts az emberi agyáról mintázva megalkottak számítógépen egy neurális modell hálót [34]. A neurális hálózatok képesek azonosítani bonyolultabb mintákat és azokat később kategorizálni. Az alkalmazott neurális hálózatok közül megkülönböztetünk FFNs (FeedForward neurális hálót) és RNNs-t (Rekurrens neurális hálót). Az RNNs felépítéséből adódóan tartalmaz ciklusokat, míg a FFNs ezzel nem rendelkezik [35].

Fő különbség a gépi tanulás és a mélytanulás között, hogy az ML algoritmusok numerikus adatokon taníthatók, ezzel szemben a DL algoritmusok azok összetettsége miatt, képesek képi és video adathalmazból is tanulni-elemezni. A nyelvi modellek (chatbotok) kezeléséhez is a neurális hálókat alkalmazzák, azok végrehajtási sebessége miatt. A taníttatásnál, amennyiben indokolt, alkalmazható a felügyelt tanulás (Supervised Learning) [36]. A gépi tanulás ezen formája lehetővé teszi, hogy ha ismert bemeneti és kimeneti adatokat használunk a taníttatás során, úgy a felállított modell a későbbiekben képes lesz számunkra az addig nem ismert bemeneti értékekhez megfelelő kimeneti értéket társítani. Az UL vagyis a

felügyelet nélküli tanítás (Unsupervised Learning) [37] ennek az ellenkezője, vagyis ekkor a kimeneti értékek nem ismertek.

Egy másik technika a megerősítéses tanítás (Reinforcement Learning) [38]. Ez egy interdiszciplináris része az AI-nak, amelyet az optimális vezérlés (path-finding) területénél alkalmaznak. Ebben az esetben nincs szükség előre definiált bement-kimenetre, kizárólag a feladat/cél kerül meghatározásra. Az RL modell felállítása Markov-döntési folyamat (MDPs) szerint történik [39].



1. ábra. A számítástechnika és a mesterséges intelligencia kapcsolata [40]

Fentiekből és az 1. ábrán is látható, hogy szoros kapcsolat áll fent a dolgok internete (IoT), a nagyadathalmazok (Big Data) és a gépi tanulás eljárásai között. A hálózatba csatolt eszközök a szenzorokon keresztül gyűjtik az adatokat, amelyek tárolásra és elemzésre kerülnek, majd később ezen adathalmazok szolgáltatják az alapját a különféle tanuló algoritmusok számára.

Az ipar 5.0 egyelőre koncepcionális szinten létezik, de a jelenlegi generációban vázolt technológiák tovább fejlesztett verziói fognak dominálni, vagyis az okos eszközök, a nagy adathalmazok és az Ember-AI-gép közti kommunikáció tudományos területei.

A tématerület aktualitását jól példázza, Klaus Schwab német közgazdász által 1971-ben alapított Világgazdasági Fórum is (World Economic Forum). A Davos-ban évente meredezésre kerülő nemzetközi fórum központi témája a mesterséges intelligencia és a robotizáció kapcsolata volt, ami az elkövetkezendő évek megkerülhetetlen gazdasági tényezője lesz [41].

2. Ipari robotkar kiberfizikai változatának létrehozása

A kialakított kiberfizikai és intelligens rendszerek robotlaboratóriuma az alapja a megvalósított Digitális-ikerpárnak. Elsőként a KUKA KR5 ipari robotegység került modellezésre, amely a gyakorlati oktatás során is kiemelt szerepet kap. A hat szabadságfokkal rendelkező robotegység elsősorban hegesztési műveleteket lát el az iparban. Fontos kiemelni, hogy a gép nem nyílt forráskódokon alapszik, ami plusz megoldandó problémaként merült fel a robot virtuális implementációja során.

A célkitűzések között szerepel, hogy a gépegység átültetésre kerüljön digitális környezetbe, ahol elvégezhetők az alapszintű műveletek, és lehetőség legyen az utángyártott egyedi 3D nyomtatott alkatrészek tesztelésére virtuális keretek között. Valamint, hogy az oktatás részét is képező 2D geometriai alakzatok, amelyeket a pályabejáráshoz alkalmazunk, saját adatkészlet (dataset) alapján kerüljenek generálásra, 1.5-ös verziójú stabil diffúziós mélytanulási modellel. Így minden egyes végfelhasználó egyedi, személyre szabott feladatot tud megoldani a képzése során.

Az általunk előállított saját adatkészlet modell tanításához a Google Colab felhő alapú szolgáltatást használtuk. A Digitális-ikerpárnak előtesztelési verziója pedig a V-REP szimulációs keretrendszerben lett elkészítve. Az ehhez használt Asztali-PC erőforrásait tekintve egy Intel Core i5 7600K processzorral, Nvidia GTX1070 VGA-val 8 GB DDR5 VRAM-mal és 64 GB DDR4 memóriamodullal rendelkezett.

A tanulmány soron következő fejezete bemutatja a fizikai KUKA KR5 robot „telepítését” a virtuális térbe, és a saját adatkészlet alapján, mélytanulási technikával megalkotott pályabejáráshoz használt geometriai alakzatok létrehozását.

2.1. KUKA KR5 6 szabadságfokú ipari robotegység

A Debreceni Egyetem Műszaki Karának Járműmérnöki Tanszéke új, innovatív technológiákra helyezte a hangsúlyt az oktatási színvonal növelése érdekében. A COVID-19-es világjárvány kitörésével számos ponton átértékelésre kerültek a korábban általánosnak vélt rutinfeladatok. A járványügyi intézkedések nem tették lehetővé, hogy nagy létszámban történjen az ipari gépegységek programozása és az új rendszerek tesztelése.

Az elmúlt években megkezdődött egy modern labor alapköveinek a letétele az ipar 4.0 szellemiségében. A kiberfizikai és intelligens rendszerek robotlaboratóriuma (CPS-LAB), mint izolált környezet lehetőséget biztosít komplex anyagmozgatási feladatok ellátására képelemzéssel.

A labor számos egyedi saját tervezésű elemmel rendelkezik, legyen szó megfogókról, vezérlőkről vagy saját felvételekkel betanított képelemzési technikákról.

A fő robotegység egy 6 szabadságfokkal rendelkező KUKA KR5, melyet az autópárhuzamban is széles körben alkalmaznak. A robot konstrukciót elsősorban ívhegesztési feladatok ellátására alkalmazzák. A munkaterület nagysága $8.4 \text{ [m}^3\text{]}$, ami alkalmassá teszi összetett feladatok elvégzésére, mivel az ismétlési pontossága 0.04 [mm] . A robotra vonatkozó paramétereket a táblázat 1. foglalja össze.

Táblázat 1. KUKA KR5 robot paraméterei [42].

Terhelhetőség	5kg
Maximum terhelés	37kg
Tengelyek száma	6
Vezérlő típus	KR C2
Tömeg	127kg
Üzemi hőmérséklet	+10C – +55C

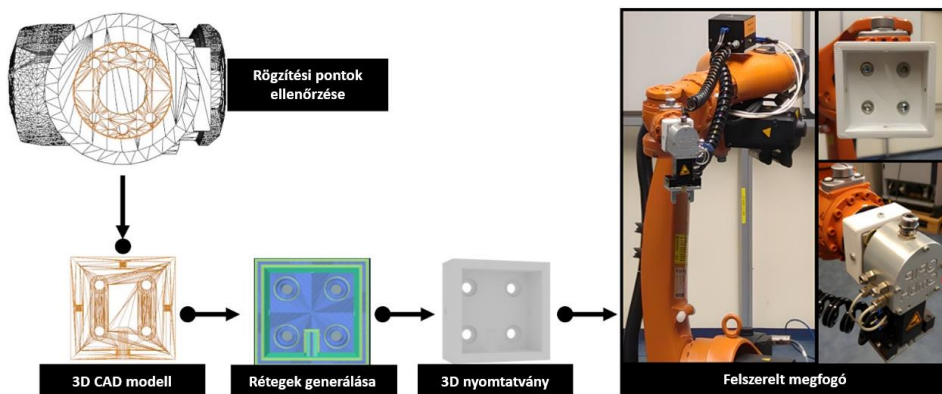
Maga a robotegység eredetileg nem rendelkezett megfogóval, ezért az ehhez szükséges adatper a laborban került megtervezésre és legyártásra egy szálolvasztásos módszert alkalmazó FDM 3D nyomtató segítségével.

Az ipari környezet miatt az adapter, akrilnitril-butadién-sztirol (ABS+) alapanyag lett kiválasztva, amit az autóiparban is alkalmaznak.

Az adatper nyomtatási statisztika:

- **Becsült idő:** 02:01:08
- **Rétegek száma:** 127
- **Vonalak száma:** 64119
- **Szükséges szálhossz [mm]:** 4283

A legyártott adapter segítségével rögzíthetővé vált a KUKA KR5 robotegységre az elektro-pneumatikus megfogó, így az elvégezhető feladatok száma is megnőtt [43]. Az adatper a 2. ábrán tekinthető meg.



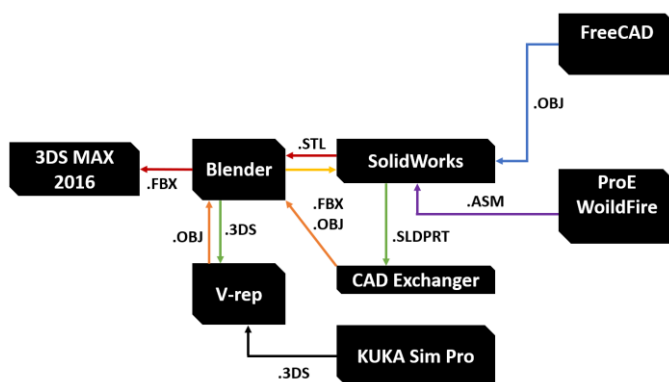
2. ábra. 3D nyomtatott KUKA KR5 adapter

2.2. 3D CAD modellezés és importálás

Az egyes 3D CAD tervező programok között nehéz átjárást találni a speciális fájlformátumok miatt. Vannak olyan fájl kiterjesztések, amelyek nem alkalmasak 3D nyomtatásra, ugyanakkor vannak olyan formátumok, amelyek az exportálást követően az előre letárolt animációkat és textúrákat is tartalmazzák. A modellezésben van olyan eset, amikor egy adott 3D CAD programban csak a 3D modell megalkotása történik meg. A textúrázást vagy az egyes 3D elemek csoportosítását pedig egy másik programban végezzük el. Ehhez viszont ismerni kell a fájlformátumokat, valamint, hogy melyik 3D CAD szoftver képes őket lekezelni exportálást követően.

A leggyakoribb 3D CAD fájlformátumok közé tartozik az. STL (szabványos háromszög nyelv), az .OBJ (hullámfönt objektum), a .FBX (filmbox), és az .ASM (összeszerelési modell lista) [44].

A 3. ábra illusztrálja, hogy az egyes 3D CAD modellező szoftverek között milyen fájlformátumok kezelése lehetséges. Amennyiben a megtervezni kívánt 3D modellnek nincs különösebb funkciója, vagyis statikus modell, akkor. STL formátumot alkalmazzuk. Amennyiben textúrával is el kívánjuk látni az adott 3D modellt, akkor .OBJ formátumot használunk.



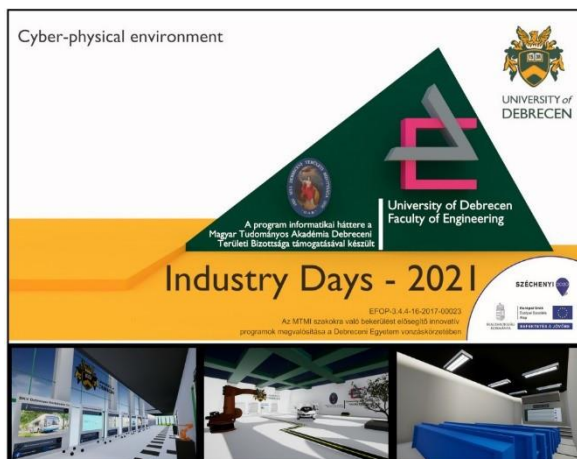
3. ábra. 3D CAD formátumok kezelése

A Digitális-ikerpár technika alkalmazására 2021-ben sor került az intézményben. A Debreceni Egyetem, Ipar Napjai esemény lebonyolítása a COVID-19 okozta járványügyi korlátozások miatt nem volt kivitelezhető jelenléti formában.

Ennek keretei között készült el a kiberfizikai alapú „Industry Days” alkalmazás, ami a Műszaki Kar épületének 3D modelljét tartalmazta, ahol a felhasználók kapcsolatba tudtak lépni bizonyos előre meghatározott elemekkel.

Az Industry Days alkalmazás és a mostani projekt között számos különbség van. Az előbbi egyfajta virtuális tárlat vezetésként szolgált, ahol nagy tereket kellett létrehozni és importálni, valamint előzetesen felvett és letöltött videó anyagokat kellett konvertálást követően hozzárendelni a különböző modellegységekhez. A vezérlés tekintetében a kamera mozgatása, a hangátviteli

és a linkek közötti váltás volt a meghatározó. Fontos megjegyezni, hogy az esemény prezentációs jellege miatt programozási és egyéb vezérlési feladatok nem képezték részét a programnak. A program nyitóképe a 4. ábrán tekinthető meg.



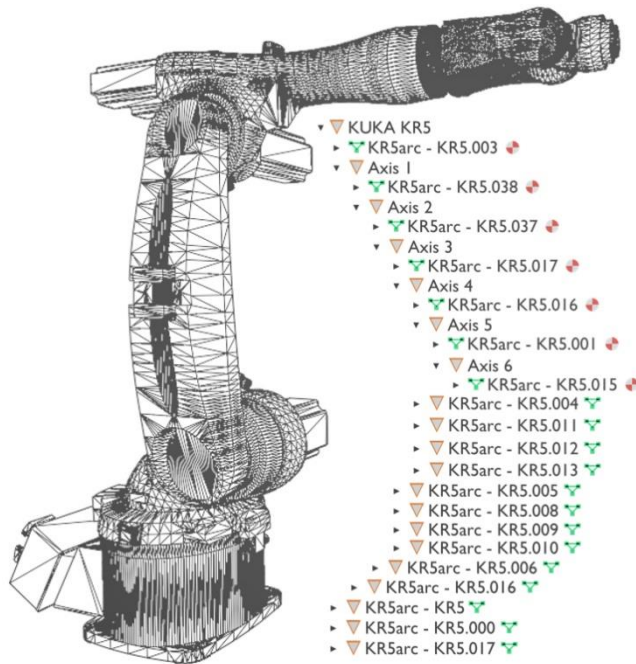
4. ábra. Ipar Napjai alkalmazás – 2021

A projekt aktuális része során a Blender 3D CAD modellező került alkalmazásra a robotkar megtervezésénél. A KUKA KR5 ipari egység modellezésénél figyelembevételre kerültek a korábban felsorolt robotra jellemző paraméterek.

- Tengely 1 (A1) 154°/s
- Tengely 2 (A2) 154°/s
- Tengely 3 (A3) 228°/s
- Tengely 4 (A4) 343°/s
- Tengely 5 (A5) 384°/s
- Tengely 6 (A6) 721°/s [42].

A tengelyek meghatározása során az adott elemek középpontosításra és megnevezésre kerültek A1 tengelytől egészen A6 -os tengelyig. A megfogó és a szerszám beméréshez használt szerszám is külön névvel lett ellátva.

Az al-modellek csoportosítását szükséges volt elvégezni a Blender-ben, hogy az egyes modellek képesek legyenek a helyváltoztatásra. A csoportosított KUKA KR5 modellt az 5. ábra szemlélteti.



5. ábra. KUKA KR5 csoportosított 3D CAD modell

A KUKA KR5 robotkarral együtt, ezen elv mentén került lemodellezésre a robotegység köré épített robotcella is.

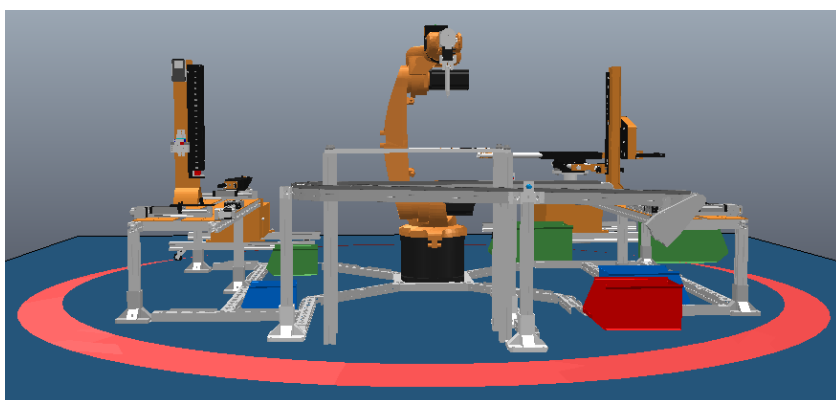
A korábban tárgyaltakat figyelembe véve, szükség volt egy szimulációs keretrendszerre, ami alapját adja a tervezendő kiberfizikai rendszerünknek. A kritériumok között szerepel, a DAE, .OBJ, vagy .STL fájl kiterjesztések kezelése, egyedi program-scriptek írása, illetve kapcsolatba kell tudni lépnie a virtuális modellel a különféle mozgástípusok kivitelezésének érdekében. A kiberfizikai rendszerünk egyfajta előteszterként funkcionálna, ezért a virtuális mérések elvégzése is szükséges, hogy a későbbiekben valós értékekkel is összehasonlítható legyen.

Az általunk meghatározott kritériumoknak a V-REP (Virtuális robot-kísérleti platform) tesz eleget. A szimulációs rendszert ipari és kutatási célokra is használják, platform független és ingyenes minősítéssel rendelkezik. Osztott vezérlési architektúrája révén, lehetséges a Lua, C/C++ és Python scriptek írása is.

Továbbá a fizikai szimulációk esetében a Bullet, ODE és a Vortex motort is kezeli [45].

Az importálás maga többlépcsős folyamat, amit nagyban meghatároz a tervezett 3D CAD modell elemszáma. A modelleket egységenként kell exportálni meghatározott fájl formátumban, így csökken annak valószínűsége, hogy az exportálás során hibák jönnek létre a modellek struktúrájában. Esetünkben az .OBJ fájl kiterjesztés került előtérbe helyezésre.

A fájl formátum kizárólag a 3D geometriát prezentálja, vagyis a modellek koordinátáját, az azokat alkotó poligonokat és a textúrákat. Az importált robotcella modell a 6. ábrán tekinthető meg.

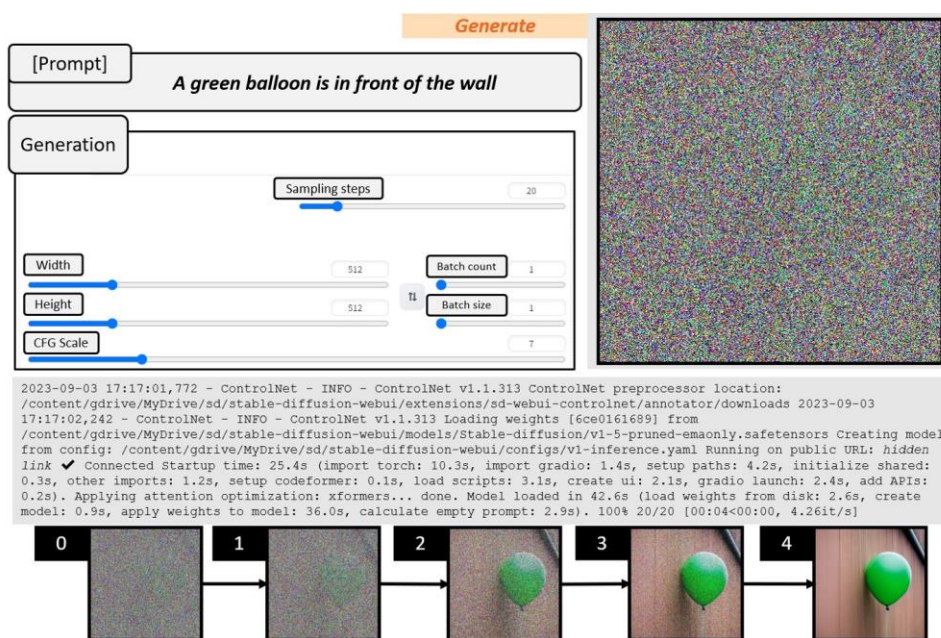


6. ábra. Importált robotcella modell

2.3. Stabil diffúziós generatív text-to-image mesterséges intelligencia

A stabil diffúzió, egy új generációs modell alapú kép generátor, amelynek első verziója 2022-ben került bemutatásra a Google kutató csoportja által [46]. A stabil diffúzió nyílt-forráskódú minősítéssel rendelkezik és képes szöveges leírás alapján (prompt) képek generálására. Annak érdekében, hogy a folyamat sikeres legyen, számtalan, az interneten található képpel kellett taníttatni a nyilvánosan közzétett checkpoint ellenőrző fájlt (.ckpt) mélytanulási (DNNs) neurálishálózattal [47]. A taníttatási folyamat rendkívül erőforrás igényes, emiatt a generált képek felbontása mindössze 512x512. Az image képi fájlknál utólagos felskálázás lehetséges, így

elérhető a magasabb felbontás és a nagyobb pixelsűrűség. A stabil diffúzió használatához először le kell tölteni a v1.5.ckpt ellenőrző modellt. A letöltést követően feltöltjük a Drive felhő tárhely fiókba, majd onnan átcsatoljuk a Google Colab felhő szolgáltatásba. A tárolók mellett telepítjük az AUTOMATIC1111 -et [48] és futtattjuk a webes grafikus felületet a nyilvános URL-en (egységes erőforrás azonosító) keresztül. A felületen keresztül lehetőségünk van számos beállítás elvégzésére, többek között itt adható meg a Classifier-Free Guidance Scale (CFG), vagyis az osztályozó nélküli irányítási skála paraméter megadása is. Ezen érték meghatározó abból a szempontból, hogy az általunk megadott leíró mennyire vegye figyelembe az image generálás során.



7. ábra. Stabil diffúzió kép generálás leírással

A kép generálási folyamat során először körül írjuk, hogy mit szeretnénk látni az 512x512 felbontással rendelkező képen. Ezt követően a modell futtatásakor véletlenszerűen zajt fog előidézni, amit lépésenként elkezd zajtalanítani (denoise), és ekkor körvonalazódik az általunk kért generált kép. Esetünkben, ahogy a 7. ábrán látható, a leíró „A green balloon is in front of the wall” (zöld léggömb, egy

fal előtt), amivel egyben az is tesztelésre került, hogy a modell meghívható és futtatható Google Colab szolgáltatáson keresztül.

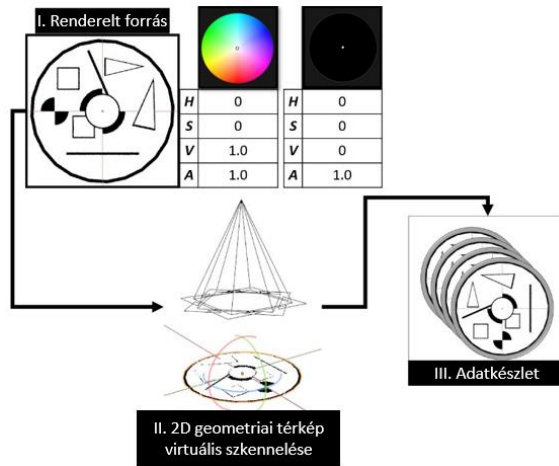
2.4. Dreambooth alapú pályatérkép és textúra generálás

Az oktatásban résztvevők részéről igény mutatkozott, hogy egyedi módszerek alkalmazásával színesítsük a tantárgyak palettáját. Virtuális környezetben PTP (pontról-pontra), LIN (lineáris) és CIRC (körkörös) mozgások végrehajtása mellett szükséges azon probléma megoldása is, amely az elvégezhető feladatok alacsony számából ered. A bázis beméréshez használt „trajectory map”, vagyis pályatérkép egy fixen felszerelt fémlemezre lett rögzítve. A lapon található geometriai alakzatok száma azonban véges és nem felcserélhető.

Annak érdekében, hogy egyedi pályatérképek kerüljenek létrehozásra, mélytanulási technika került alkalmazásra nyílt forrás kódú alapon, hogy saját modellt hozhassunk létre [49].

Az adatkészlet előállításához a Blender-t használtuk, amely lehetővé tette a testre szabott geometriai alakzatok megalkotását, és egyfajta virtuális szkennelő szerepét töltötte be.

A megtervezett 2D geometriai „minta” térkép, köröket, négyzeteket, háromszögeket és egyenes vonalakat tartalmazott. A színek esetén a Blender-ben a HSV (színárnyalat, telítettség, fényesség) palettát használtuk, ahol a fehér és fekete árnyalatok domináltak. Ezt követően Cycle renderelő motor beállítással [50], a 3D CAD tervező programba egy virtuális kamerát helyeztünk. A kamera felülnézetben volt és az alatta pozícionált 2D geometriai térképet renderelte. Minden egyes renderelt képkockát követően, a Z tengely körül 36 fokkal lett elforgatva a virtuális kamera. Mivel a modell tanításhoz alapszinten elvárt a legalább 10 darab képkocka szám, így összesen 10 darab képkockát készítettünk el a vázolt módszer szerint, amelyek felbontása 512x512 volt. A metódus a 8. ábrán tekinthető meg.



8. ábra. Adatkészlet előállítás rendereléssel

A referencia modellünk az általunk létrehozott adatkészletből került tanításra, aminek alapja a stabil diffúzió volt [51]. Önállóan a tanított modell nem lenne alkalmas az általunk előírányzott képek előállítására, ezért szükséges a tanítatás során úgynevezett „trigger word”, vagyis aktiváló szó megadása. Ezek figyelembevételével a generált kimeneti kép már konzisztensen a tanított elemeken fog alapulni.

A tanítatás során a Dreambooth [52] metódus lett alkalmazva, ami rendelkezik két komoly korláttal. Az egyik, hogy tanítást végző feldolgozóegységnek, magas hardver követelménynek kell megfelelnie. Ez a gyakorlatban minimum 10 GB VRAM-al rendelkező grafikus feldolgozót jelent. Illetve a képfelbontás nem haladhatja meg, az 512x512 értéket. A tanítási folyamat véghezvitelére a Google Colab [53] lett kiválasztva, ami korlátozott ideig biztosít virtuális felhőalapú számításra alkalmazható gépet, ami Nvidia Tesla architektúrájú T4 GPU -t 16 GB VRAM-al [54], Intel Xeon CPU-t [55] és 12 GB RAM -ot jelent.

A Dreambooth Colab Notebook-nál, a *fast-DreamBooth.ipynb* lett futtatva [56], maga a Colab összekapcsoltan működik a Google Drive-al, ami azért fontos, mert program szkript végrehajtása során a tanításhoz használandó fájlok és a.ckpt ellenőrző ideiglenesen egy erre a célra kijelölt Drive fiókban kerül tárolásra.

Esetünkben a Hugging Face-en [57] keresztül létrehozunk egy „token”-t, vagyis egy azonosítót, ami szükséges a stabil diffúzió v1.5.ckpt (checkpoint) [58] alap modell letöltéséhez. A letöltést követően a munkamenet névnek (geometricmap-et) adjuk meg, ami egyben aktiváló „trigger word”-ként fog funkcionálni.

A tanítás megkezdése előtt a bemenetként szolgáló, általunk előállított képek, új névvel kerülnek ellátásra numerikus azonosítóval együtt, hogy sorrendben történjen a fájlok betöltése.

A képi fájlok tanításánál két érték lesz meghatározó, amelyek közül az egyik az U-Net (konvolúciós neurális hálózat képszegmentálás) [59] paraméter. Esetünkben a feldolgozandó adatállomány 10 darab képi fájl, amihez 650 step érték került megadásra. Kis elemszám esetén, ennél alacsonyabb értéknél a hálótanítása nem lesz megfelelő, magasabb értéknél viszont túl tanulás következhet be.

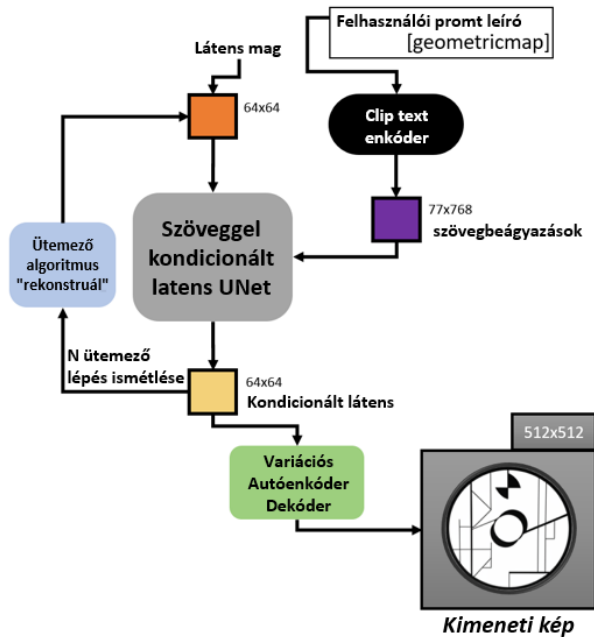
A másik fontos érték a szövegkódoló betanítási lépésekre vonatkozik (Text Encoder Training Steps) [60], aminek paramétere 250–450 közé kell essen korlátozott elemszámú adatkészlet esetén (400-as érték került megadásra).

A Dreambooth tanítási fázisa 1.2 órán tartott felhő alapú számítással. A futtatást követően a fájl automatikusan Drive fiókba került feltöltésre .ckpt fájl kiterjesztéssel, amelynek mérete 1.98 gigabyte volt.

Az elő-tanított (pre-trained) .ckpt modellek használatakor a képek generálásához három fő komponens szükséges, az automatikus enkóder (auto encoder), az U-Net és a CLIP szövegkódoló (CLIP Text-Encoder). Az auto encoder egy kódolóból és egy dekódolóból áll. A kódoló rész alakítja át a megalkotandó képet az U-Net számára, majd a dekódoló alakítja vissza azt képpé. Az U-Net szerepe, hogy az alacsonyabb felbontású képprezentációt nagyobb felbontású képprezentációba alakítsa át. A Text-Encoder pedig értelmezi a bemenetként szolgáló leíró, ami a generálandó kép tartalmaként szolgál [61].

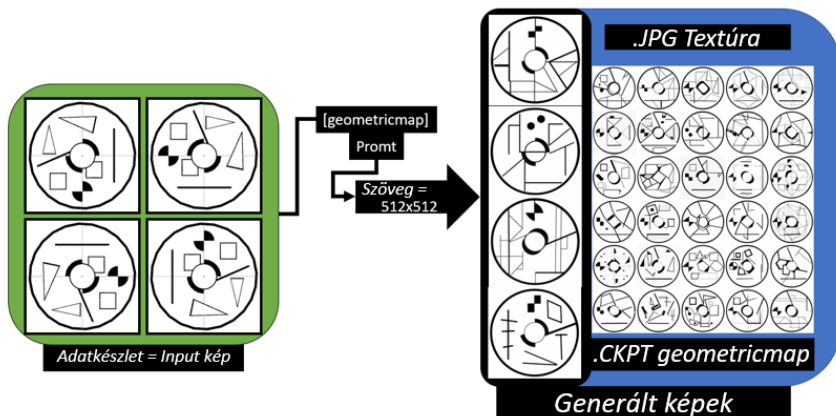
A hardveres teljesítmény minimalizálása érdekében alacsony felbontású képek generálódónak (64x64), amik később felskálázásra kerülnek (512x512).

A kép generálás folyamatának lépései a 9. ábrán tekinthetők meg.



9. ábra. A kép generálás folyamata

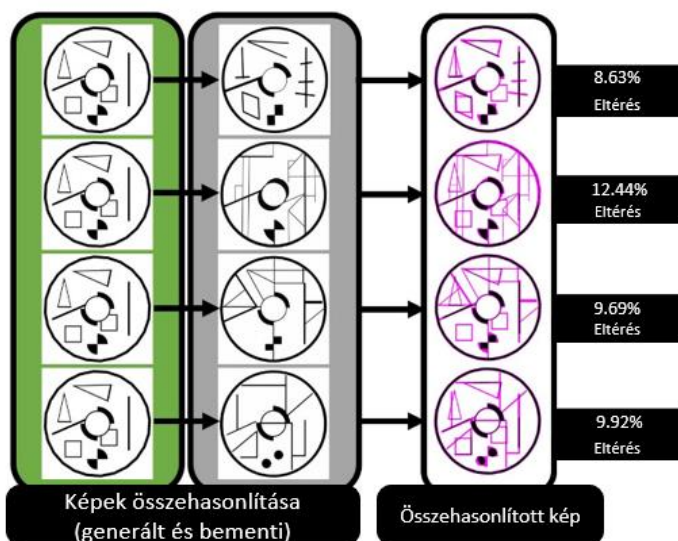
A saját adatkészletből előállított .ckpt modell használatához az AUTOMATIC1111 webes grafikus felhasználói felülettel rendelkező Colab -ot használtuk [62]. A modellünkből generált egyedi geometriai alakzatokat tartalmazó képek a 10. ábrán tekinthetők meg.



10. ábra. Pályatérkép textúra generálása saját adatkészlet alapján

A futtatás során .ckpt kiterjesztésű fájlt töltöttük fel az AUTOMATIC1111 webes felületén. A generálás során Euler algoritmust adtunk meg, a mintavételnél (sample step) pedig 20-as értéket. A CFG, más néven az osztályozómentes iránymutatási skála (Classifier-Free Guidance Scale) az általunk megadott prompt leírók figyelembevételének mértékét jelenti.

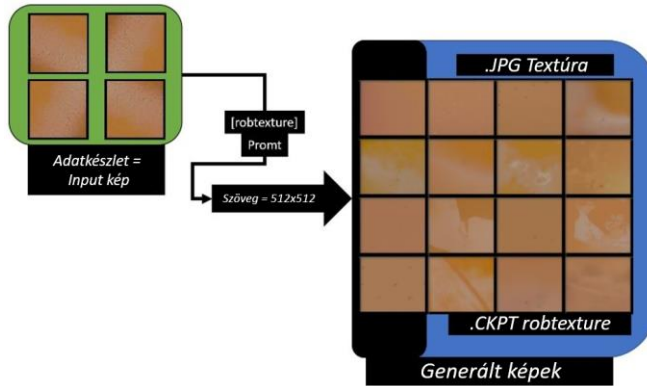
Az adathalmazból előállított képeket ezután összehasonlítottuk egymással. Ehhez szükség volt az eredeti bemeneti képre, amelyből az adatkészletet generáltuk, és négy általunk véletlenszerűen kiválasztott generált képre. Ezután feltöltöttük a kiválasztott anyagokat az ingyenes image-text-compare [63] online webes felületre, hogy összevessük a képfájlokat.



11. ábra. Az eredeti és a generált pályatérképek összevetése

Az ábrán látható, hogy a képek egymáshoz képest történő eltérése %-os értékben is kiírásra került, és pixel szinten történt meg az image fájlok összehasonlítása. Az azonostól eltérő részek rózsaszínnel lettek kijelölve. Az eltérések a 11. ábrán tekinthetők meg.

Ezen elv mentén néhány főbb 3D CAD modell esetén is alkalmazásra került a technika, azonban ott az autentikusság jegyében tisztán robot textúra előállítás céljából (lásd 12. ábra).



12. ábra. Robot modell generált textúrája

2.5. Inverz és direkt kinematika alkalmazása

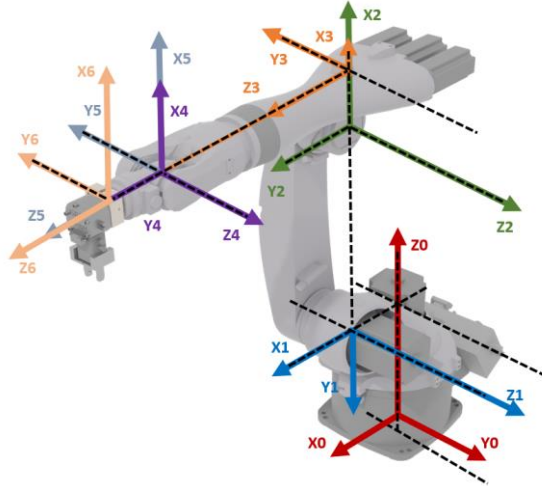
A direkt és inverz kinematika esetünkben kulcs fontosságú, hiszen általa meghatározható a test helyzete tetszőleges időpontban, továbbá a TCP (szerszám középpont) koordináta rendszer pozíciója is kiszámíthatóvá válik, tehát a robotkar tengelyeinek és a szöghelyzeteinek az ismeretében meghatározható annak a térbeli pozíciója.

A direkt kinematika szolgáltat eredményt arra vonatkozóan, hogy az adott robotkar pillanatnyi konfigurációban milyen TCP pozíciót vesz fel. Ennek meghatározásához szükségesek a KUKA KR5 fizikai paraméterei (tengelyek száma, munkatere, robotkar dimenziója).

A direkt kinematikai feladatok megoldhatók Denavit-Hartenberg mátrix segítségével. Maga a mátrix egy egyszerű tag és csukló leírási módszer, ami minden robot konfigurációhoz alkalmazható, annak komplexitásától függetlenül [64].

Minden egyes csuklót egymás után határozunk meg az előző csuklóból a szükséges transzformációt alapul véve. Kijelölünk egy csuklót, ami az „ n ” lesz. Ezt követően az „ n ” és a következő csukló is kap egy lokális koordináta-rendszert [65].

A lokális keret felvételére jellemző, hogy minden csuklón egy „z” tengely megy át, ami lehet rotációs vagy translációs. Az importált és pozícionált KUKA KR5 DH paramétere a V-REP -ben a 13. ábrán és 2. táblázatban látható.



13. ábra. KUKA KR5 csuklók és tengelyek

Táblázat 2. KUKA KR5 robotkar – Denavit-Hartenberg paraméterek

Joints	Joint Angle (θ) [changeable]	Joint Offset (d) [mm]	Link Length (a) [mm]	Twist Angle (α) [degree]
1	θ_1	400.014	180.004	90
2	θ_2	0	600.005	0
3	θ_3	0.003	120.002	90
4	θ_4	620.044	0	90
5	θ_5	0.002	0	90
6	θ_6	0	0	0

A DH táblázatban alkalmazott szimbólumoknak meghatározott jelentésük van, amelyek a következők:

θ : A „théta” szög az X_{n-1} és az X_n tengelyek között mérhető a Z_{n-1} normális síkon.

d : A „d” paraméter a merőleges távolság az O_{n-1} origótól mérve az X_n és a Z_{n-1} tengelyek metszéspontjáig a Z_0 tengely mentén.

a: Az „a” paraméter a Z_{n-1} és a Z_n tengelyek közötti távolság, amelyet az X_n tengely mentén mérünk.

α : Az „alfa” szög a Z_{n-1} és Z_n tengelyek között mérhető, az X_1 normális síkban [66].

Az inverz kinematika szemben a direkt kinematikával, egy sokkal komplexebb számítási metodikát igénylő módszer. A vizsgált KR5 robotra az inverz kinematikai feladatban adott a robotegység pozíciója, és meghatározandók a robot csuklók „theta” szögei.

Fontos kiemelni, hogy a robot geometriai modellje annak térbeli pozícióját írja le. A kinematikai modell a robotegység sebességgel kapcsolatos problémákat válaszolja fel. Ehhez szükség van a Jacobi mátrix alkalmazására, ahol megkülönböztetésre kerülnek a csuklósebességek és a végberendezés (End-Effector) sebessége egyaránt.

A csuklósebesség:

- Szögsebesség a forgástengelyhez képest (rotational joint).
- Lineáris sebesség a mozgás tengely mentén (joint translation).

Végeffektor sebessége:

- Szögsebesség x, y, z körül
- Sebességek x, y, z mentén [66].

$$\mathbf{y} = \begin{pmatrix} v \\ \omega \end{pmatrix}, \text{ ahol } \mathbf{v} = \begin{pmatrix} v_x \\ v_y \\ v_z \end{pmatrix} \quad \boldsymbol{\omega} = \begin{pmatrix} \omega_x \\ \omega_y \\ \omega_z \end{pmatrix} \quad (1)$$

Direkt geometriai feladatból kell kiindulni az összefüggés meghatározásánál.

Az végeffektor pozíciója és annak orientációja:

$$\mathbf{x}(\mathbf{q}) = \begin{bmatrix} p_x(q) \\ p_y(q) \\ p_z(q) \\ \alpha_x(q) \\ \alpha_y(q) \\ \alpha_z(q) \end{bmatrix} \quad (2)$$

A D-H ${}^0_n[T]$ utolsó oszlop 1-3 eleme.

$$\dot{x} = \frac{\partial x}{\partial q} \cdot \dot{q}, \text{ ahol } \frac{\partial x}{\partial q} = J = \begin{bmatrix} \frac{\partial x_1}{\partial q_1} & \frac{\partial x_1}{\partial q_2} & \dots & \frac{\partial x_1}{\partial q_m} \\ \vdots & \vdots & \ddots & \vdots \\ \frac{\partial x_n}{\partial q_1} & \frac{\partial x_n}{\partial q_2} & \dots & \frac{\partial x_n}{\partial q_m} \end{bmatrix} \quad (3)$$

ahol:

m – A robot szabadságfoka

n – Az adott robot munkatér dimenziója

J – Jacobi mátrixa a robotnak

A Jacobi mátrix inverz szükséges a kinematika feladat megoldásához:

$$\dot{q} = J^{-1}(q)\dot{x} \quad (4)$$

Végeffektor és a csuklógyorsulás közötti összefüggés:

$$\ddot{x} = J(q)\ddot{q} + \dot{J}(q)\dot{q} \quad (5)$$

2.6. Tengely specifikus mozgások PTP, LIN és CIRC

A fentieket figyelembe véve meghívásra kerül az IK plugin a V-REP -ben, hogy a PTP, LIN, és CIRC mozgástípusokat alkalmazni tudjuk a későbbiekben. Mivel az egyes robotok eltérők lehetnek, így megadásra kell kerüljön a „Base”, a „TCP” és a „Target” (lásd 14. ábra).

```

1 function sysCall_init()
2     --way:
3     local simBase=sim.getObjectHandle('Base')
4     local simTip=sim.getObjectHandle('TCP')
5     local simTarget=sim.getObjectHandle('Target')
6     -- create an IK environment:
7     ikEnv=simIK.createEnvironment()
8     -- create an IK group:
9     ikGroup_undamped=simIK.createIkGroup(ikEnv)
10    -- set its resolution method to undamped:
11    simIK.setIkGroupCalculation(ikEnv,ikGroup_undamped,simIK.method_pseudo_inverse,0,6)
12    -- create an IK element based on the scene content:
13    simIK.addIkElementFromScene(ikEnv,ikGroup_undamped,simBase,simTip,simTarget,simIK.constraint_pose)
14    -- create another IK group:
15    ikGroup_damped=simIK.createIkGroup(ikEnv)
16    -- set its resolution method to damped:
17    simIK.setIkGroupCalculation(ikEnv,ikGroup_damped,simIK.method_damped_least_squares,1,99)
18    -- create an IK element based on the scene content:
19    simIK.addIkElementFromScene(ikEnv,ikGroup_damped,simBase,simTip,simTarget,simIK.constraint_pose)

```

14. ábra. IK script meghívása

Mivel a cél, hogy pontosan azon robot folyamatok kerüljenek reprodukálásra, amiket fizikai oktatás során is végzünk, ezért a bázis beméréshez használt elemet, mint textúrát is importáljuk. A nehézséget az okozza, hogy a fizikai robotkarhoz tartozik egy KRC2 vezérlőszekrény és egy programozó eszköz (Teach Pendant) [67], amelyek viszont nem nyílt-forráskódúak.

A PTP (Pontról-Pontra) mozgás esetében direkt kinematika került alkalmazásra. A szerszám ilyenkor a fizikai robotegység konfigurációjához mérten a leggyorsabb pályán mozog a célpont elérésének az érdekében. A mi esetünkben ennek a mozgástípusnak a reprodukciója során, miután a virtuális robotegység elérte az általunk megadott pontot, a csukló koordináták letárolásra kerülnek a *Poscollect=function* parancs meghívásával.

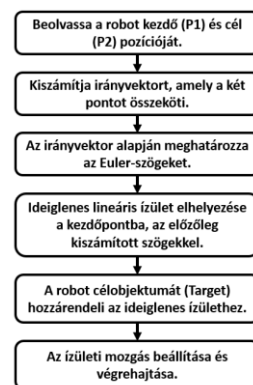
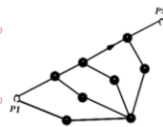
A LIN, vagyis a lineáris mozgásra konkrét parancs nincs. Ehhez egy célpont, vagyis egy „Target” objektum megadásra szükséges a robot munkaterében. Az objektum határozza meg magát az elérendő célpontot és biztosítja, hogy a robot elmozdulása egyenes vonalon történjen.

Ugyanakkor ehhez szükséges úgynevezett prizmatikus csukló (prismatic joint) létrehozására is, aminek meg tudjuk adni a pillanatnyi helyzetét a 3D térben. A megadott „Target” objektum pedig prizmatikus csukló cél pozícióját jelöli (lásd 15. ábra).

```

1 LIN=function(TO)
2 FROM=sim.createDummy(0)
3 sim.setObjectPosition(FROM,Target,(0,0,0))
4 Distance=sim.getObjectPosition(TO,FROM)
5 Dis=math.sqrt((Distance[1])^2+(Distance[2])^2+(Distance[3])^2)
6 Z1=(Distance[1])/Dis
7 Z2=(Distance[2])/Dis
8 Z3=(Distance[3])/Dis
9 if Z3 < 0 then
10   sim.setObjectOrientation(FROM,world,(0,1,0,1))
11   Distance=sim.getObjectPosition(TO,FROM)
12   Dis=math.sqrt((Distance[1])^2+(Distance[2])^2+(Distance[3])^2)
13   Z1=(Distance[1])/Dis
14   Z2=(Distance[2])/Dis
15   Z3=(Distance[3])/Dis
16 end
17 local JO=sim.createJoint(sim.joint_prismatic_subtype,sim.jointmode_passive,0,(0.0001,0.0001),nil)
18 sim.setObjectPosition(JO,FROM,(0,0,0))
19 sim.setJointInterval(JO,false,(-100,200))
20 if Z3 < 0 then
21   sim.setObjectOrientation(JO,FROM,(math.pi-math.atan(-(Z2)/(Z3)),(math.asin(Z1)),0))
22 else
23   sim.setObjectOrientation(JO,FROM,(math.atan(-(Z2)/(Z3)),(math.asin(Z1)),0))
24 end
25 sim.setObjectParent(Target,JO,true)
26 sim.moveToConfig(-1,(0),nil,(0.05),(1),(1),(Dis),nil,callback_ik_JO)
27 sim.setObjectParent(Target,TO,true)
28 sim.removeObject(JO)
29 sim.removeObject(FROM)
30 end

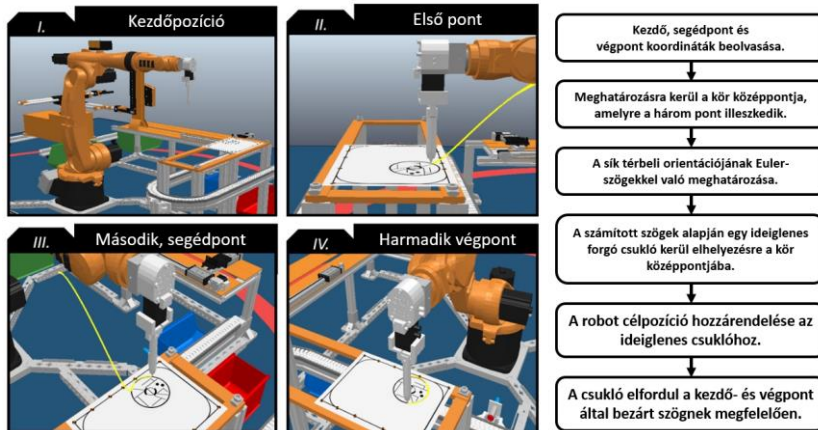
```



15. ábra. LIN mozgás végrehajtása

A tengelyek menti elfordulás esetén Euler szögek kerülnek megadásra, amelyet a program kódunk ki is számol a robot konfiguráció végrehajtása során.

A CIRC mozgástípusnál három pontot adunk meg. Egy kezdő pontot, egy segédpontot és egy végpontot (lásd 16. ábra).



16. ábra. CIRC mozgás végrehajtása

A megírt kódunk a bemenet CIRC mozgás során, ahol szintén három pont lesz, viszont nem adjunk meg külön a virtuális KUKA KR5 alap pozícióját (ettől még ő lesz a kezdőpont). Esetünkben a segédpont és végpont lesz megadva (a scriptben Through és To -ként szerepel).

A három pontból kerül kiszámításra a körnek a középpontja. A kódsor tartalmazza a pontok bemért távolságát, az „a” a kezdeti, a „b” Through, „c” pedig a To pont.

A könnyebb láthatóság érdekében a virtuális robot a pálya megtétele során egy, a háttértől eltérő sárga csíkot húz.

2.7. Virtuális mérés elvégzése és elemzés

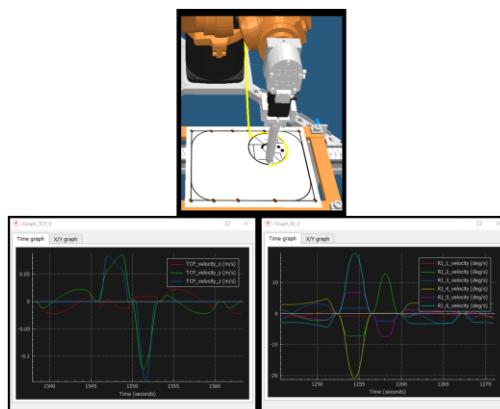
Lehetőségünk van virtuális mérések elvégzésére az általunk létrehozott kiberfizikai környezetben. A mérések a jövőbeni analízisek elvégzéséhez nyújtanak támpontokat a számunkra. A valós fizikai KUKA KR5 és a kibertérbe helyezett KUKA KR5 között lehetnek eltérések (pl. KRC2 vezérlőszekrény nem nyíltforráskódú). A mérések birtokában további módosítások is kivitelezhetők

lehetnek, annak céljából, hogy a gép minél pontosabb viselkedése legyen lemodellezve.

A TCP_V és az RJ_V kerül mérésre virtuális módon. Előbbi a Tool Center Point -nak a sebességét méri és grafikus módon rajzolja ki, utóbbi az elfordulási sebességét méri (Rotation Joint), mind a 6 tengely esetében.

A megvalósítás során két „graph” elem lett létrehozva, amiket a *function sysCall_init()*-ben meghívkoztunk, majd megadtuk a program kódban az értékeket (x, y, és z irányú sebességek jelölése), illetve a mértékegységeket, valamint egymástól eltérő színeket.

A TCP esetén [m/s] mértékegységben lettek az értékek kiíratva. Az RJ -nél pedig [deg/s] mértékegységben (17. ábra). A kialakított rendszer nemcsak egy adott robotkar mozgásra alkalmazható. Bármilyen más egyéb, egyedileg felvett pont a térben mérhető grafikus formában.



17. ábra. TCP elmozdulás és csuklók elfordulásának mérése

Fontos megjegyezni, hogy a KUKA KR5 robotegység szerszám mérőegysége gyártás alatt van, így a fizikai mérés és annak összehasonlítása későbbi tanulmány részét képezi.

A kibertérbe helyezett labor alkalmazásának egyik előnye, hogy a virtuális robotkarhoz és robotcellához bármilyen plusz 3D modell hozzáadható és eltávolítható. Valamint a fizikai robotegységhez képest költséghatékonyabb az üzemeltetése, mivel nincs villamosenergia fogyasztása és az anyagkopás sem

jelentkezik a használat során. Kiemelendő, hogy a robot vezérlőrendszere nem nyílt-forráskódú, vagyis az egyes parancsok végrehajtása során is eltérések lehetnek.

2.8. Fejezet konklúziója

Elkészült a virtuális modellje a labor robotcellájának, vagyis egyfajta elő-Digitális-ikerpár. A KUKA KR5 és a robotcella a V-REP-ben lehetőséget szolgáltat arra, hogy egyes előzetesen legyártandó 3D CAD modellt virtuálisan is teszteljünk, annak tényleges legyártása előtt. Az alapszintű PTP, LIN és CIRC mozgások is elvégezhetők a szimulációs környezetben belül, ezzel prezentálva az oktatásban résztvevők számára az elvégzendő feladatokat.

Mélytanulási technika alkalmazásával, saját adatállomány alapján tanításra került egy .ckpt modell, amelynek lényege, hogy a pályatérképek ne csak a meglévő fix elemekre hagyatkozzon, hanem egyénre szabott képek is létrehozhatók legyenek textúrák formájában, így minden hallgató, ezáltal egyedi feladatokat oldhat meg az oktatás során. A textúrák robotcella részére szintén ugyanezen elv mentén lettek tanítva-generálva.

3. Digitális-ikerpár tanulóiközpont tervezése a robotkar oktatáshoz

Az esettanulmány következő része, a Kiberfizikai és intelligens rendszerek robotlaboratórium digitalizált másának megalkotása volt, amely tartalmazza a laborban történő gyakorlati képzés során alkalmazott szükséges virtuális elemkészletet. Szemben az előzőekben tárgyaltakkal, a Digitális-ikerpár egy önálló applikációként futtatható felhasználóbarát alkalmazás, aminek fő prioritása az oktatás segítése volt.

A 6 szabadságfokú ipari robotkar és a hozzá tartozó saját robotcella itt központi szerepet tölt be. A virtuális robotkarral pedig végrehajthatók általános PTP, LIN és CIRC helyváltoztatási műveletek. A végfelhasználó képes új programok létrehozására és annak mentésére, amik később felhasználhatók-lekérdezhetők lesznek. A virtuális labor környezetben a gépegység HMI rendszere is integrálásra került, hogy segítse a mindenkori felhasználót.

A kialakított szoftveres környezet segíthet a kis és középvállalkozásoknak (SMEs) saját munkatársaik képzése során, mivel a program nem teszi kötelezővé a valós-fizikai gépegység meglétét, ezért telepítési és karbantartási költségek sem terheli őket.

Digitalizált formában érhető el a labor, így kapacitási probléma sem lép fel a terem hallgatói létszáma miatt. A felhasználók egymástól elszigetelve, párhuzamosan vehetnek részt a robotprogramozási gyakorlatokban. A hallgatók-operátorok könnyebben sajátíthatják el a gépkezeléshez szükséges készségeket.

A következő szakasz tárgyalandó témája a létrehozott kiberfizikai rendszer jelenlegi állapotát tükrözi, kibővítve azt az elvégzett feladatokkal.

3.1. A koncepció kialakításhoz vezető események

A probléma felvetés, amely életre hívta a Digitális-ikerpár koncepciót esetünkben az, hogy növelni kívántuk az oktatás színvonalát és annak hatékonyságát. A labor

digitális infrastruktúrája előnyt jelenthet a szemeszterek során, hiszen az adott oktatónak korlátozott erőforrások állnak rendelkezésre a képzés ideje alatt. A hallgatói létszám magas, viszont a labor mérete a telepített gépegységek munkaterülete miatt erősen korlátozott, mindez az oktatásban résztvevő valamennyi személy számára komoly kihívást jelent.

A másik fő ok, ami életre hívta a Digitális-ikerpárt, hogy a tömegtermelés önmagában nem elégséges feltétel a termelékenység növeléséhez. Fontos, hogy az adott gyártósor flexibilisen tudjon reagálni a vevői igényekre. Ez a fajta gondolkodásmód paradigma váltást von maga után az oktatásban és az iparban. A hatékonyság növelésének érdekében, a gyártás újra tervezése-hangolása szükségessé teszi az adott részleg virtuális modelljeinek meglétét.

Esetünkben az általunk alkalmazott innovatív módszer fő prioritása, hogy az oktatás résztvevői magas szintű digitális kompetenciákkal gazdagodjanak a képzésük során és képesek legyenek a későbbiekben ipari környezetben, az ipar által megszabott feladatok megoldására. A labort érintő egyes fejlesztési ciklusok nem szorítkoztak szimplán egy félévre, hanem félévtől függetlenül, folyamatosan zajlanak. A labor ismert az általunk kialakított fejlesztési ciklusáról, amely során azon személyek, akik a Kiberfizikai és intelligens rendszerek robot laboratóriumában aktívan végeztek feladatokat, részletes dokumentációkat írtak, amik átadásra kerültek a következő évfolyamok számára, így biztosítva folyamatosan megújuló tudásanyagot.

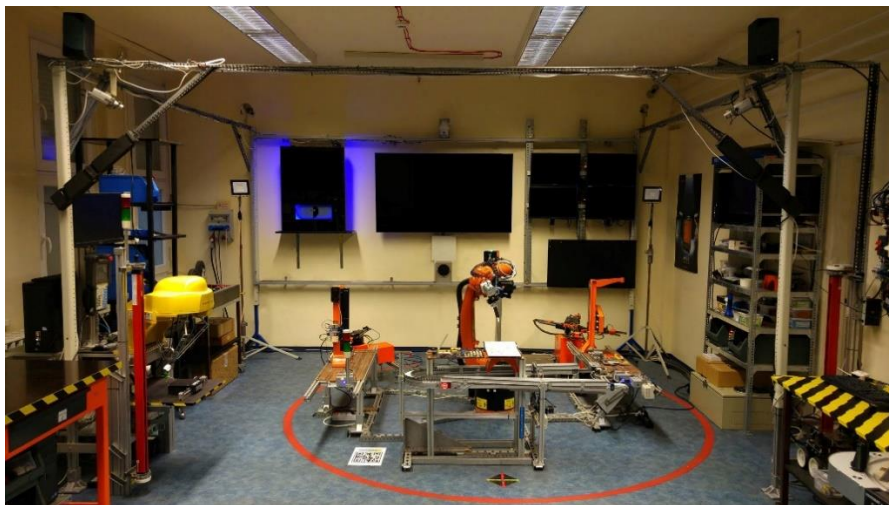
A labor fő alapelve, hogy izolált „minta” Ipar 4.0 -ként funkcionáljon, ahol lehetőség van előzetesen tesztelni vezérléseket és ipari gépeket, mielőtt azok átültetésre kerülnek a céges partnerek által üzemeltetett gyártórészlegekre [68].

Tény, hogy egy effektíven funkcionáló-működő rendszer létrehozása azért is ütközik akadályokba, mert a gépek nem nyílt-forráskódúak sem hardver sem szoftvere tekintetében, így mindig az adott rendszerre jellemző viselkedési modellt kell felállítani, hogy reprodukálásra kerülhessen oktatási-kutatási céllal.

A COVID-19-es világjárvány [69] és a chiphiány egységesen negatívan befolyásolták a labor működését, ezért annak érdekében, hogy továbbra is

folytatni lehessen a korábban megkezdett kutatásokat és az oktatást, új alapokra kellett helyezni annak funkcionálását. Ehhez a 18. ábrán szereplő labor teljes digitalizációjára-integrálására volt szükség a virtuális kibertérbe.

A világjárvány egyértelműen rávilágított azon törekvéseink létjogosultságára, hogy a fejlesztési és kutatási anyagjainkat hatékonyabban kell digitalizálni. Bebizonyosodott továbbá, hogy önmagában a szerverek használata nem elégséges ennek a problémának az eliminálása céljából.



18. ábra. Kiberfizikai és intelligens rendszerek robotlaboratóriuma

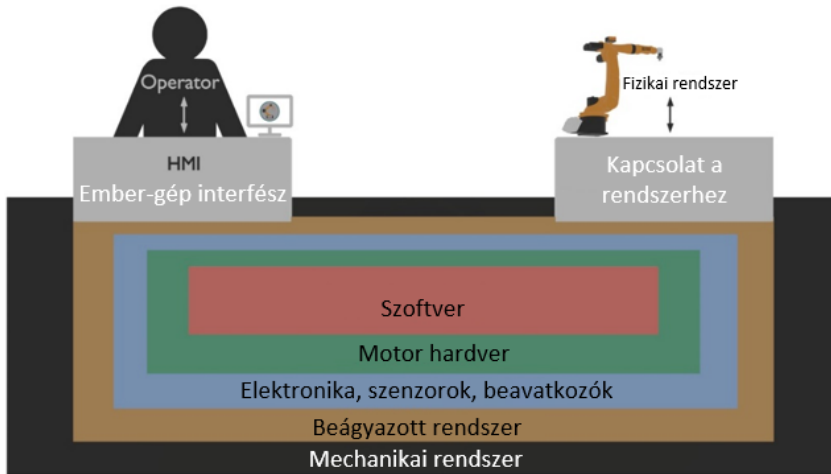
Az új kihívások tükrében, a létesítmény teljes digitalizációját kellett kivitelezni a rendelkezésünkre álló adatok alapján.

A kiberfizikai rendszer, mint technika az ipar 4.0 szerves részét képezi. A problémaszituáció az, hogy számos definíció létezik magának a fogalomnak [70]. Bizonyos elvek szerint elsősorban az ipari gépegységek hálózatra csatolása az elsődleges [71], más perspektívából nézve, szenzor adatok lekérdezése és azok elemzése kell, hogy hangsúlyos legyen, valamint a folyamatos kétirányú-interaktív kommunikáció [72].

Egy adott termelési részleg, vagy nem elsősorban termeléssel foglalkozó terület digitális replikáns mása esetében, nem kötelező érvényű elvárás a

folyamatos kommunikáció az elemek között. Mivel önmagában is lehetőséget szolgáltat az adott részlet virtuális úton történő használatára-újra tervezésére [73].

Ugyanakkor maga a labor integrálása a virtuális kibertérbe komplex feladat részeként áll elő, mivel ahogy szimulációk osztályozása is különféle csoportokra bontható, úgy a kiber-fizikai rendszerek is (lásd 19. ábra).



19. ábra. Kiberfizikai rendszer architektúra

Az iparban a szimulációkat széleskörben alkalmazzák, ugyanis általuk elkerülhetők a tervezési hibák, könnyebb az összetett folyamatok ellenőrzése, határteljesítmények és vezérlések összehasonlítása és a minősítő paraméterek vizsgálata [74].

Az imént felsoroltak esetünkben is lényegesek, hiszen a labor folyamatos fejlődése során számos egyedi, saját tervezésű elemmel egészült ki, ami mind meghatározó a rendszer egészére nézve. Így ezek integrálása is meghatározó a tervezendő virtuális környezetbe. Fontos megjegyezni, hogy a laborban mindössze 100 Mbit/s hálózati sebesség érhető el, ami nem teszi lehetővé az üzemszerű dinamikus két irányú kommunikációt a fizikai és virtuális elemek között, így a virtuális feladatok megoldására szorítkozunk.

Egy digitalizált robotlabornak számos előnye van, az oktatást illetően. Az adott személynek nem szükségszerűen kötelező fizikai formában a laborban lenni

a gép alapszintű vezérléséhez. A gép virtuálisan tetszőleges irányból megközelíthető-körbejárható, szemben a valódi fizikai géppel, aminek felépített robotcella munkaterében nem lehet tartózkodni, a balesetveszély elkerülése érdekében [75]. A robotkar pontos mása kerül létrehozásra az interfésszel együtt, aminek használatával elsajátítható annak kezelése [76]. Lehetséges a virtuális térben való mozgás és az egyes objektumokkal a kapcsolatba lépés VR (Virtual Reality) technológia felhasználásával, abban a formában ahogy az iparban is több esetben is használják [77]. Jelen esetben, a térben való mozgás egér és billentyűzet segítségével történik, ezzel elkerülve a VR alkalmazása miatt jelentkező „motion sickness” -t.

Fontos megjegyezni, hogy a hallgatók nem érzékelik a virtuális gyakorlat során az egyes elemek súlyát, ami a módszer egyik fő hátrányát jelenti [78]. Az oktatás során előfordulhat az is, hogy a rendszer felhasználói felé hamis biztonságérzetet közvetít, amennyiben nem vesznek részt valamilyen formában a fizikai gyakorlaton is [79].

Az általunk létrehozott Digitális-ikerpár több szempontból is különbözik a többi oktatási módszertől. A legfőbb különbség, hogy az általunk az oktatásban használt robotegységek egyedi hardveres fejlesztéseket kaptak az elmúlt években (robotcella, ipari megfogó, futószalag). A digitalizált laborban történő tevékenység folyamatosan monitorozásra került a virtuális kamerákkal, így internetes hangátviteli protokollon (VOIP) keresztül streamelhető mások számára „online” formában.

3.2. CPS LAB elemek 3D CAD modellezése

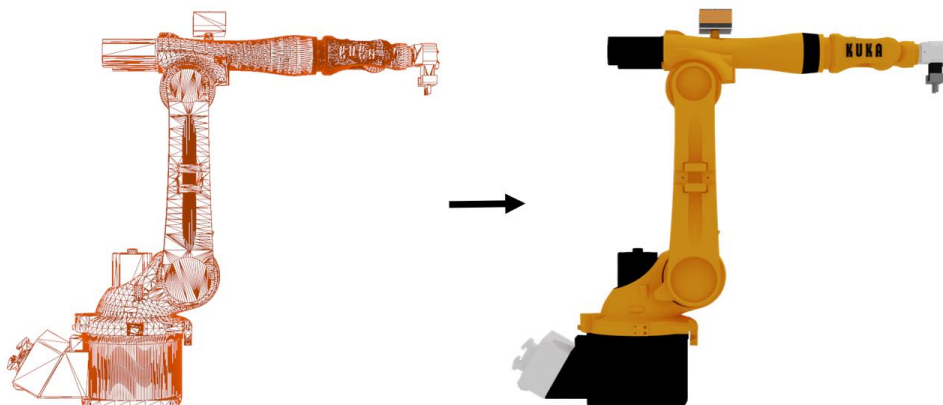
A labor kiberfizikai rendszerszerbe való átültetésének alapkövetelménye a 3D modellek megléte és azok optimalizációja. A Kiberfizikai és intelligens rendszerek robotlaboratóriumában megkezdődött az ipari eszközpark állománynak a felmérése. Az ipari laborban a gépegységek paramétereinek begyűjtése révén lehetőség nyílik, hogy a rendszerek átültethetők legyenek a virtuális kibertérbe szimulációs és vezérlési feladatok ellátása céljából „offline” formában.

Ezeket felhasználva közelebb kerülhetünk a viselkedésük megértéséhez, ami kulcsfontosságú tényező a virtuális térbe történő integrálásuk során. A különböző paraméterek, mint a kinematikai felépítés, munkatér, alkalmazott koordináta rendszerek, hasznos teher értéke szintén összegyűjtésre kerültek. Ezek alapján a kategorizálásuk is megtörtént. A birtokunkban lévő adatok alapján a 3D modellek tervezése megkezdődhetett a rendelkezésünkre álló számítási kapacitás figyelembevételével.

A Kiberfizikai és intelligens rendszerek robotlaboratóriumban számos egyedi fejlesztés zajlik. A 3D modellalkotásnál a labor minden egyes robotegysége modellezésre került. A rendelkezésünkre álló CUDA alapú számítási kapacitás korlátozott, ezért ezen elemek egyenként kerültek 3D CAD modellezésre. A modell alkotáshoz most is a Blender [80] lett kiválasztva.

A labort érintő fejlesztéseket általánosan 3D CAD modellezés előzte meg, így az elemkészlet pontos és részletgazdag Digitális-párt eredményezett.

A korlátozott erőforrások miatt a modellek önmagukban nem voltak alkalmasak arra, hogy minden egyes elem egy adott szimulációs környezetbe kerülhessen integrálásra, ennek elsődleges oka a modellek száma és azok textúra felbontása volt, ami megnövekedett számítási kapacitást vont maga után. A 3D modelleket utólagosan optimalizálni kellett, elsősorban a poligon számok kerültek csökkentésre, illetve a felhasznált elem számok. A textúrák esetében egyszerű csere nem segített volna, ezért azok törlésre kerültek és csak sima RGB színskálát használtunk [81]. Az egyik központi modell a 20. ábrán tekinthető meg.



20. ábra. KUKA KR5 3D CAD modell

A különböző színskálák alkalmazásának elsősorban nem esztétikai okai vannak. Célja a különböző objektumok egymástól elhatárolása, illetve az egyes modellek jobban jelölhetők általa. Minden egyes modell a fentiek szerint került optimalizálásra, ami így már lehetővé tette a nagyobb elemszámú modellek kezelését.

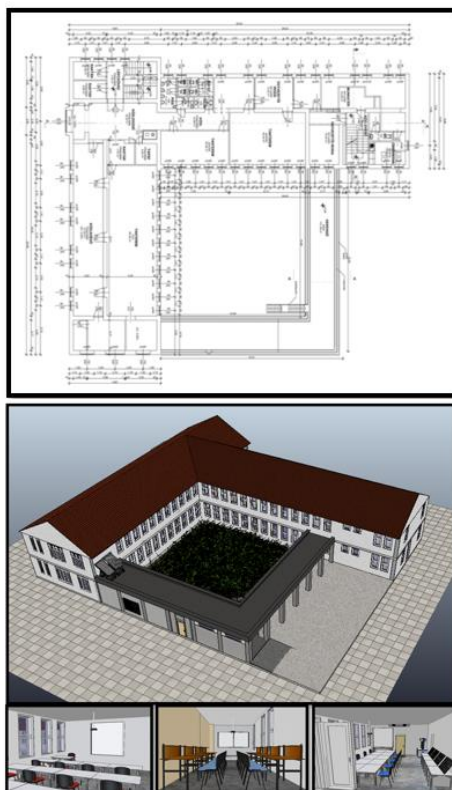
3.3. Kiberfizikai és intelligens rendszerek robotlaboratóriuma

A COVID-19 okozta járványügyi korlátozások egyértelműen rámutattak a mérnöki területen az oktatás során megoldandó problémákra. Az adott hardveres eszközökhöz való hozzáférés nem volt lehetséges minden esetben a hallgatók számára. A Digitális-ikerpár a CPS laborra fókuszál, ez azonban egy nagyobb volumenű projekt részeként jelenik meg.

A főbb modelleken kívül más elemek 3D CAD modellezése is megtörtént. A járványhelyzetre való tekintettel és az oktatás színvonalának növelése érdekében nem volt elengedő mindössze néhány főbb robotegység modellezése. Ennek eredményeképpen a teljes kutató létesítmény modellezése megtörtént.

A fejlesztéseknek és kutatásoknak otthont adó létesítmény az épület hivatalos tervrajzai alapján került megtervezésre. A modell maga szintén átment az

optimalizációs eljáráson annak érdekében, hogy a hardveres erőforrásokat csökkenteni lehessen. A kutatólétesítmény modellje a 21. ábrán látható.



21. ábra. Kutató létesítmény 3D modellje

A modellen belül minden egyes labor virtuális mása megtalálható, annak minden fontosabb eszközével együtt, legyen szó gépegységekről vagy vezérlőkről.

A modellek birtokában lehetőségünk nyílt arra, hogy teljesen virtuális keretek között építsük fel a labor pontos mását. A Kiberfizikai és intelligens rendszerek robot laboratóriumának Digitális-ikerpárja a 22. ábrán tekinthető meg.

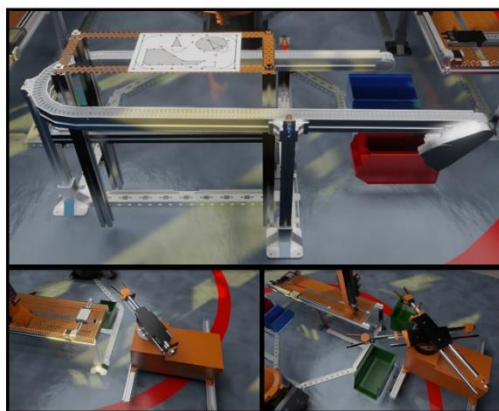


22. ábra. CPS labor Digitális-iker

A robotcella ebben külön kiemelt szerepet kapott, mivel az egy teljesen egyedi belső fejlesztés részét képezi.

A cellamodell fő elemei közé sorolandók az MP-9 és MP-11-es ipari robot manipulátorok, amelyek korábbi projektek mentén kerültek ismét üzembehelyezésre egy futószalaggal egyetemben, ami alkalmas anyagmozgatási feladatok ellátására a cellán belül [82].

A labort meghatározó folyamatos fejlődésre való igény szintén tetten érhető a digitalizált replikánsban, ugyanis lehetőséget biztosít a későbbi bővítésre. A robotcella további főbb elemei a 23. ábrán láthatók.



23. ábra. CPS labor robotcella elemei

Főbb 3D modellek, a CPS labor robotcellában:

- KUKA KR5 robotkar
- Megfogó egység
- Tárgyasztal
- Futószalag platform
- Bal és Jobb oldali platform
- Programozó eszköz.

3.4. Szimulációs keretrendszer és HMI implementáció

Az Unreal Engine 4 [83] egy fejlett 3D alkotó eszköz, amelyet az építészet, a járműipari és filmipar területén használnak. Mint szimulációs keretrendszerre azért van szükség, mert a Blender nem alkalmazásfejlesztő, ezért a programon belül csak a modellezés és azok összecsatolása történhet meg.

A projekt kivitelezése az alábbi gépkonfigurációval valósult meg:

- CPU: AMD Ryzen 5 3600X AM4 (6 core, 12 threat, 3.8Ghz-4.4Ghz)
- VGA: nVidia GeForce RTX 2060 SUPER
- RAM: 16 GB DDR4 3200Mhz
- SSD Samsung 860 QVO 1TB [84].

A modellek kezeléséhez és azok importálásához a fentebb ismertetett Asztali-PC ált rendelkezésre. Az alkalmazás elkészítése során kiemelten fontos volt, hogy alacsony gépigényű és kisméretű végfelhasználói szoftver kerüljön létrehozásra. Mivel az egyes CAD szoftverek más mértékegységgel, illetve orientációval kezelik a modelleket, így figyelembe kellett venni a Blender-ből való exportálásnál.

Az .FBX exportálás előtt a Blender-ben a *Scene Unit Length* beállítását centiméterre szükséges állítani az importálni kívánt modelleknél. A *scale*-t pedig százzal szorozni, hogy Unreal Engine-ben is megfelelő méretű legyen a modell. Az exportálásnál azt is figyelembe kell venni, hogy az Unreal Engine balsodrású

koordinátarendszert használt, a Blender pedig jobbsodrású koordináta rendszerrel dolgozik.

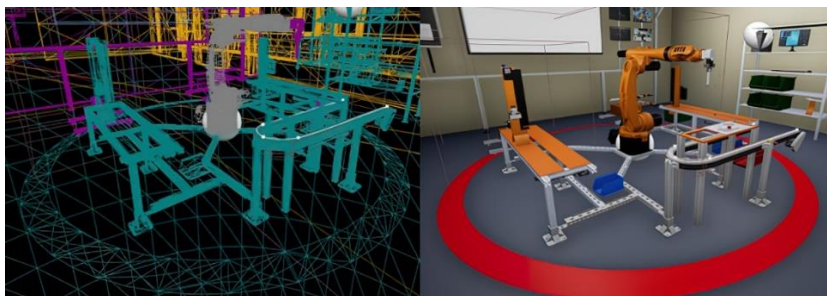
Az Unreal Engine 4 importálási beállításánál az „*Import Normals and Tangents*” szükséges, mivel a modelleket ennek hiányában darabosan importálja.

A KUKA KR5 modelljéhez Blender-ben készült egy egyszerű csontvázlánc (Bone chain), amely a későbbi vezérlést segíti. Emiatt a robot modellt csontvázhálóként (Skeletal mesh) kellett importálni.

Az Unreal Engine-ben lehetőség van úgynevezett ütközők, vagyis Collision hozzáadására, viszont a modellre nem megfelelően helyezkedő Collision erőteljesen ronthatja a teljesítményt, ezért sok esetben „*Box Simplified Collision*” lett alkalmazva a modellekhez.

A futószalag esetében, a modell bonyolultsága miatt több *box collision* elhelyezése volt indokoltá.

Az importált robotcella a 24. ábrán tekinthető meg.



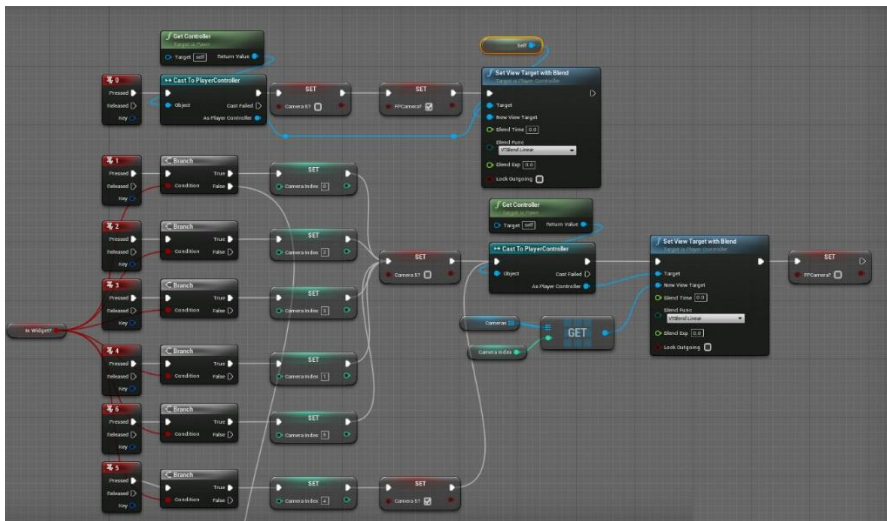
24. ábra. KUKA KR5 importált robotcella

Az Unreal Engine 4 projekt létrehozásakor lehetőséget biztosít különböző alap sablonok, vagyis template-ek felhasználására. Az *FPS template* tartalmazza a térbeli navigációt, illetve minimális elemkezelést. Ezen template-be került hozzáadásra egy irányító (controller).

A sablon alapvetően tartalmazza a W, A, S, D-vel való mozgást az adott térben továbbá az egérrel történő elfordulás lehetőségét is. A KUKA KR5 több szemszögből való megfigyelésére 5 különböző virtuális kamera került telepítésre,

amikhez a billentyűzet egy-egy gombja lett rendelve 1-től 5-ig. A 0-s gomb megnyomásával lehet visszaváltani a belső nézetű kamera állásba (FPS).

A felszerelt kamerákkal, illetve a belsőnézethez tartozó kamerával is van lehetőség ráközelíteni az egyes tárgyakra az egér görgőjének használatával. A KR5 mögé felszerelt kamera 2 tengelyen képes forgó mozgást végezni. A kamerák közötti váltáshoz a programrész a 25. ábrán látható.



25. ábra. Robotcella kamerák közötti váltásának programrésze

A kamerák elhelyezését követően a KUKA HMI digitalizált másának létrehozása következett. Ahogy korábban említésre került, a virtuális és a fizikai elemek között nincs két oldalú kommunikáció, így közvetlenül a fizikai KUKA KR5 robotra sem ültethető át a megalkotott program.

Mivel a KUKA programozó eszköz (Teach Pendant) 3D digitalizált mása is elkészült, így a fizikai hardverre jellemzők is implementálhatók. A programozó eszköz rendelkezik színes gombokkal, amelyek specifikáltan csak egy adott funkció végrehajtására határoztak meg. Vannak továbbá alap billentyűk is, amik szintén a sorváltást teszik lehetővé az adott program részen belül. A 3D CAD modellezett programozó eszköz a 26. ábrán tekinthető meg.



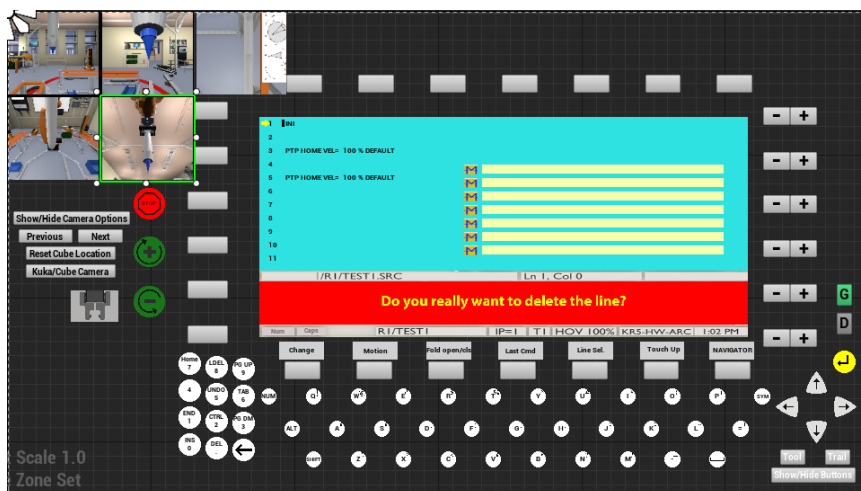
26. ábra. Virtuális ember-gép interfész

Az Unreal Engine 4 *Visual Scripting* lehetőséget biztosít „*Custom Event*”-ek létrehozására, amik segítenek a program átláthatóságán, mivel több részre bontható egy hosszabb programrész. A függvényekhez és makrókhoz hasonlóan hozzáadhatók bemenetek is, viszont ellentétben az első kettővel egy *Custom Event*-nek nincs *return node*-ja vagy *Completed output*-ja. Előnye a függvényekkel és makrókkal szemben, hogy ugyanakkor használhatók késleltető csomópontok (*Delay node*), időzítő csomópontok (*Timer node*) és idővonal csomópontok (*Timeline node*) is.

A felhasználói felület tervezéséhez az Unreal Engine beépített mozgógrafikai tervező lett alkalmazva (*Unreal Motion Graphics UI Designer*).

Az Unreal Engine jellemzően, vizuális szkriptelési módszert alkalmaz (widget blueprint) amely rendelkezik egy *Designer* és egy *Construction tab*-al. Az előbbin elhelyezhetők és testre szabhatók a különböző gombok, képek és szövegek, míg az utóbbin adhatók hozzá a különböző funkciók.

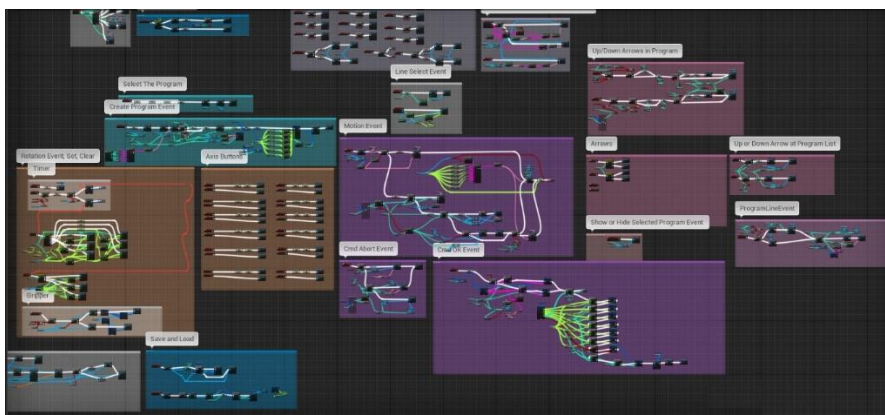
A 27. ábrán látható a KUKA KR5 virtuális felhasználói felülete.



27. ábra. KUKA KR5 virtuális felhasználói felület

A konstrukciós fül tartalmaz alapvető „On Clicked event”-et, amik egyszer futnak le. Az *Event Tick* speciálisabb, ugyanis általa frissíthetők az egyes objektumok pozíciói és viselkedésük.

A felsoroltak használatával adhatók hozzá azok a további funkciók, amelyek a valóságban a „fizikai” programozó eszközön érhetők el. Lehetőségünk van további kommentek hozzáadására is, ezzel is átláthatóbbá tehető a vizuális szkript. Így a csomópontok (node) sokasága között tudunk címszavak alapján keresni, illetve színeket is tudunk csatolni az egyes, egymással összefüggő programrészekhez (lásd 28. ábra).



28. ábra. A virtuális felhasználói felület program rész

A virtuális robotvezérlés átláthatóságát szolgálják az olyan funkciók, amik a valóságban nincsenek jelen. A mutatás/elrejtés (*Show/Hide Buttons*) gomb megnyomásával a gép mozgatásához nem szükséges elemek eltüntethetők vagy megjeleníthetők tetszés szerint (29. ábra).



29. ábra. A programozó eszköz virtuális felhasználói felülete

A szerszámközéppont (TCP) köré felszerelt virtuális kamera a valóságban nincs jelen, mivel a mozgás végrehajtása során, a robotot vezérlő személy megkapja a szükséges vizuális információkat.

A vizuális szkripten belül történik meg a felhasználó számára az adatok tárolása és mentése is.

A Unreal Engine 4-en belül az egyes részek dinamikus tömbökbe lettek átrakva (*Array*). Így könnyebb az alap adatok tárolása (KUKA program neve, dátum, sorok száma).

A mentési funkció implementálásához a vizuális szkripten belül létre lett hozva egy „*SaveGame*” rész, amihez két változó hozzáadása volt szükséges. Egy integer típusú változó, amely az utolsó létrehozott program indexét tárolja el. Illetve egy programstruktúra tömb (*Program Structure Array*), ami lehetővé teszi, hogy tömbben struktúrát tároljunk. A mentéskor a rendszer egyes változó értékei kerülnek letárolásra a „*SaveGame*” program részben.

Egy összetett programhoz képest kevés adat tárolására van szükség, amely nem terheli a számítógépet. Amikor a felhasználó létrehoz egy új programot, akkor a felhasználói felületen belül a programstruktúra tömbhöz (*Program Structure*

Array) hozzáadódik egy új elem a beírt névvel, a létrehozás dátumával, és az előre definiált sorral. Ebben 2 PTP mozgástípus szerepel (mindkettő a Home pozíciót tartalmazza). Egy új sor hozzáadásakor vagy egy már meglévő törlése után is végbemegy „*Save program event*”, ezzel biztosítva, hogy minden változtatás mentésre kerüljön.

3.5. KUKA KR5 Digital-ikerpár vezérlés

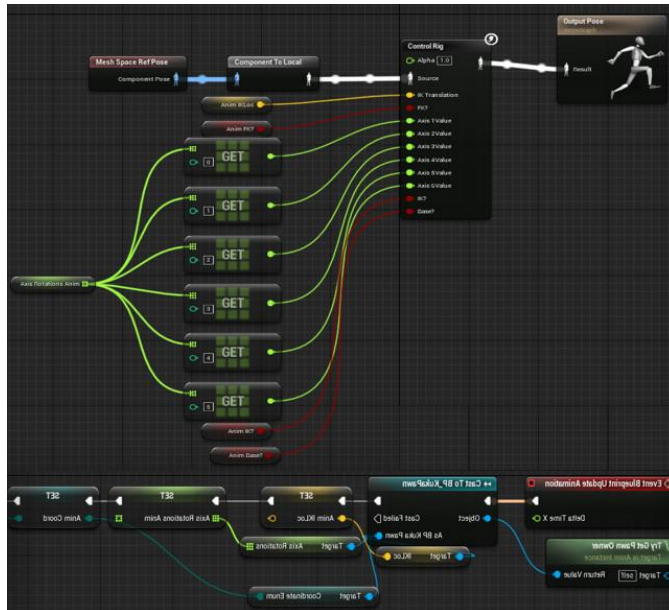
Fontos megjegyezni, hogy az Unreal Engine egy teljesen más környezet a V-REP-hez képest, a robotvezérlés is más logika mentén kerül végrehajtásra. A virtuális felületen a gombokat a felhasználó az egér segítségével aktiválja. Nincs lehetőség egyszerre több gombot alkalmazni, ezért a biztonsági kapcsolót (Deadman Switch) itt egy billentyű helyettesít. Az egyes gombok középhelyzetben tartás érzékelésére sincs mód.

KUKA KR5 mozgására két lehetőség van: vagy a csuklók forgatásával juttatjuk el a szerszámközéppontot (TCP) a kívánt pozícióba (csukló koordináta rendszer), vagy inverz kinematika segítségével (világi koordináta rendszer).

Az Unreal Engine-ben való mozgáshoz szükség van egy úgynevezett animációs vizuális szkriptre (Animation Blueprint). Ez egy speciális program rész, amelynek segítségével kontrollálható a hozzá tartozó csontvázháló (Skeletal mesh). Az animáció vizuális szkript gráfban található meg, amely egy animációs frissítő csomópont (Event Blueprint Update Animation node), és minden esetben lefut. Optimális esetben azonnal továbbíthatók az információk a KUKA KR5 robothoz tartozó programrészből.

Az animációs gráf (AnimGraph) segítségével érhető el az úgynevezett *Control Rig*, vagyis vezérlő rendszer. Fontos, hogy ez alatt a 3D modellek részletes vezérlését értjük. Ez szolgáltat lehetőséget többek között az inverz kinematika virtuális megalkotására.

A vizuális szkriptben az animációs gráf a 30. ábrán tekinthető meg.



A mozgásért felelős gombok lenyomásakor az időzítőnek köszönhetően (Set Timer by Event) meghatározható az egyes parancsok ideje. Az adott program rész, amely a robot mozgásért felel egészen addig fut, ameddig fel nem engedjük az irányító gombot. Ekkor lefut az időzítő törlése és az érvénytelenítési parancs (Clear and Invalidate Timer by Handle).

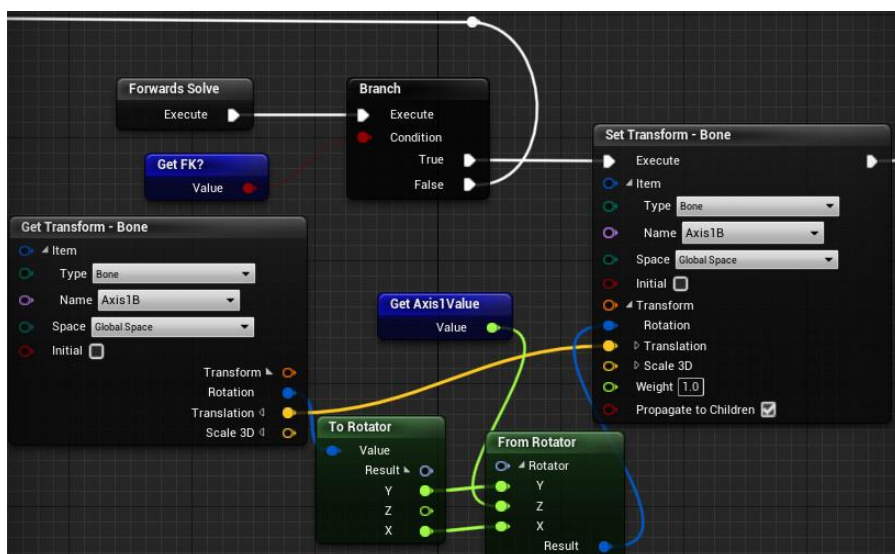
A szabad kinematika (Free kinematics) a KUKA KR5 vizuális program részénél a tengely elfordulást (Axis Rotations) végzi, és az egydimenziós tömb értékét változtatja (1D array). Az irányításért felelős gombok megnyomásakor egy integer változóban kerül elmentésre, ami egy 0 és 5 közötti szám. Továbbá az irányért felelős lebegőpontos szám (float) kerül elmentésre +1 vagy -1 értékkel. Az FK szabad kinematika, az integer változóhoz tartozó tengely értékét változtatja meg az előre meghatározott (*FKAdding Value*) plusz vagy mínusz egyszeresével. A robot tengely végállásoknak a kezelésére az úgynevezett Clamp float node”, vagyis lebegőpontos korlátozás csomópont lett alkalmazva.

A KUKA KR5 vizuális szkriptjében lévő tengely elfordulásának értékei az animációs gráfon keresztül változók segítségével jutnak el az animáció vezérlőhöz

(Control rig), ahol az addigi pozíciós értékek felülírásra kerülnek az új elfordulással.

A világi koordinátának a használatához szükség van az inverz kinematikára, amely szintén az animáció vezérlőn keresztül építhető fel (Control Rig). Az Unreal Engine 4-ben a 4.26-os verziótól kezdve érhető el a teljes test inverz kinematika csomópont (Full-Body IK node). A funkció több lehetőséget biztosít a futtatás közbeni változtatásokra. Létrehozhatók egyedi vezérlők, amelyek változók segítségével biztosítják a mozgást [85].

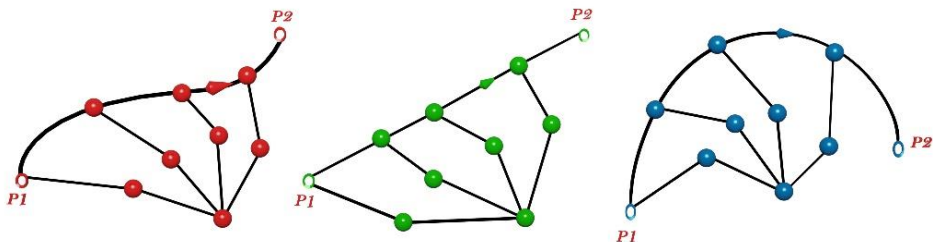
A 31. ábrán tekinthető meg a tengely értékének változtatása az animáció vezérlőben.



31. ábra. Tengely rotációs érték változtatása az animáció vezérlőben

Az IK pozíciója, az animáció animációs gráfon keresztül jut el itt is animáció vezérlőhöz (Control Rig). Az A1, A2, A3 csuklók esetében elegendő egy olyan felülírása a teljes test inverz kinematikának (Full-Body IK node), amely csak az egyik tengely körüli forgatást engedélyezi a limitek megadásával. Az A4, A5, A6 csuklók esetében szükség van egy úgynevezett egy specifikált csomópont, ami a matematikai műveletet látja el (Aim Math node), ennek függvényében lehetséges a kívánt orientáció elérése.

A KUKA KR5 vezérlése után a létrehozott programok bővíthetők a HMI, vagyis virtuális interfész segítségével. Minden alkalommal amikor a programban a „Motion” mozgás gomb megnyomásra kerül a következő sorban megjelenik egy szöveg, amely tartalmazza az adott mozgástípust és a megfogó állapotát (nyitott vagy zárt). Ilyenkor a háttérben ideiglenes változók tárolják a programsort, a csukló pozíciókkal együtt. A parancs rendben (Command Okay) parancs kiadása után az ideiglenes változókban lévő adatok hozzáadódnak a kiválasztott elemhez, majd lefut a „Save Event” parancs. A „Command Okay” parancs kiadása előtt lehetőség van változtatni a mozgástípust, amely egy ENUM (enumeráció) nevű változóban tárolódik. Ennek elsősorban program lejátszásakor van szerepe, ugyanis „CIRC motion” kiválasztásakor szükséges hozzáadni egy további segédpontot (Auxiliar) és egy végpontot (End). A mozgások könnyebb lejátszásának érdekében a kiválasztott mozgástípustól függetlenül minden hozzáadáskor mentésre kerül a csuklópont forgási érték és a TCP vektorkoordinátája. A különböző mozgás típusok ábrái a 32. ábrán láthatóak.



32. ábra. PTP, LINC és CIRC tengely specifikus mozgások

A program futtatásához a billentyűzet G gombját kell lenyomni, amely elindítja a „Play Program” szkript részt. Ennek van egy integer bemenete, ami mozgáshoz tartozó kezdeti indexet adja meg. Ezt az értéket megkapja a KUKA KR5 vizuális szkriptjében „Motion Index” integer változó.

PTP mozgás esetében egy enumeráció szerinti választási csomópont lesz meghívva (Switch on Enum node) és egy FK mozgás végrehajtás (FKMovement). A LIN mozgás esetében egy egyszerű mozgás (Simple Movement) kerül meghívásra a programon belül, míg a CIRC végrehajtásakor a körkörös mozgás

(Circular Movement). Mindhárom mozgás típusban közös, hogy végrehajtás során lefut a mozgás vége parancs (EndOfMotion Custom Event). A megfogó vezérlésének érdekében, aminek két állapota van (nyitott és zárt) a programon belül foglal helyet egy „Gripper Event”, aminek az index értéke határozza meg, hogy nyitott vagy zárt állapotban van.

A mozgások lejátszásának alapját egy idővonal (timeline) és a lineáris interpoláció (lerp) kombinációja adja.

Annak érdekében, hogy a mozgás ne egy másodpercen belül történjen meg az idővonal indítása előtt megváltoztatjuk a lejátszási sebesség (Play rate) értéket. Ez az érték PTP és CIRC mozgás esetén az elfordulási szög nagyságától, LIN mozgáskor pedig a kezdeti és végső pontok közötti távolságtól függ.

A KUKA KR5 PTP mozgás megkezdésekor a lineáris interpoláció (lerp) „A” bemenetéhez hat program rész kerül meghívásra (Custom event), amelyek tartalmazzák a csuklók forgási értékeit. A „B” bemenetre pedig a program által mentett forgatási értékek kerülnek (Program structures). Így mind a hat tengely egyszerre kezd el forogni a mozgás végrehajtása során és amikor mindegyik idővonal lefut (timeline), meghívásra kerül a mozgás vége parancs (EndOfMotion).

LIN mozgástípus esetén a lineáris interpoláció vektorkoordinátákkal dolgozik. A mozgás megkezdésének pillanatában bekérésre kerül a robot IK (Actor) pozíciója, amely a lineáris interpoláció „A” bemenetéhez tartozik. Az inverz kinematikai vektor tömb (IK vektor-array) adott mozgáshoz tartozó eleme lesz a „B” bemenet.

CIRC mozgás végrehajtáskor először meg kell határozni, a kezdő, a segéd és végpontokat (start, aux, end), ygy meghatározható a kör középpont.

Legyen a kezdőpont, segédpont, és végpont a , b , c kisbetűkkel jelölve. P_1 az $[ab]$ szakasz felezőpontja, P_2 a $[bc]$ szakasz felezőpontja, d a kör középpontja. A b pontból P_1 és P_2 felé mutató vektorok keresztszorzata megad egy a , b , c pontok által meghatározott síkra merőleges vektort [86].

$$\vec{S} = \frac{\overrightarrow{bP_1}}{2} \times \frac{\overrightarrow{bP_2}}{2} \quad (6)$$

Felírhatók az O1 és O2 vektorok, amelyek a P1, P2 pontokból mutatnak a kör középpontja felé.

$$\overrightarrow{O_1} = \vec{S} \times \frac{\overrightarrow{bP_1}}{2} \quad (7)$$

$$\overrightarrow{O_2} = \frac{\overrightarrow{bP_1}}{2} \times \vec{S} \quad (8)$$

p és q szerint:

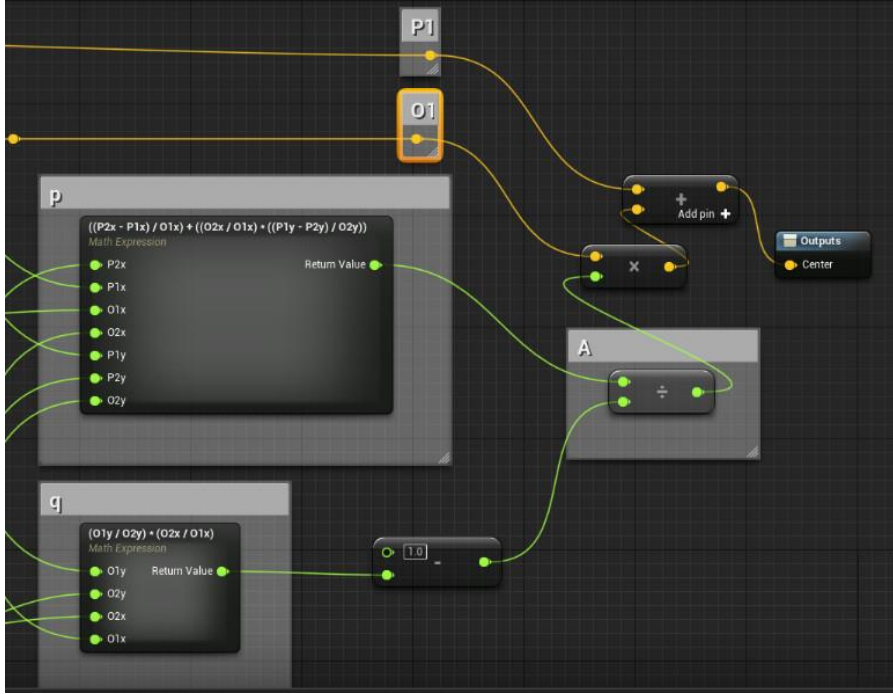
$$p = \frac{P_{2x} - P_{1x}}{O_{1x}} + \frac{O_{2x}}{O_{1x}} * \frac{P_{1y} - P_{2y}}{O_{2x}} \quad (9)$$

$$q = \frac{O_{1y}}{O_{2y}} * \frac{O_{2x}}{O_{1x}} \quad (10)$$

Ekkor a kör középpontja felírható:

$$d = P_1 + O_1 * \frac{p}{1 - q} \quad (11)$$

Az egyenletek felvezethetők az Unreal Engine-be, úgynevezett matematikai kifejezés csomópontok (Math expression node) segítségével. A csomópontok alkalmazása a 33. ábrán látható.



33. ábra. Kör középpont számítás Unreal Engine tervezőben

A kör középpontjának koordinátáinak kiszámolása után három szögeérték kiszámítása szükséges.

$$\alpha = \frac{\vec{da} * \vec{dc}}{|\vec{da}| * |\vec{dc}|}, \beta = \frac{\vec{da} * \vec{db}}{|\vec{da}| * |\vec{db}|}, \gamma = \frac{\vec{db} * \vec{dc}}{|\vec{db}| * |\vec{dc}|} \quad (12)$$

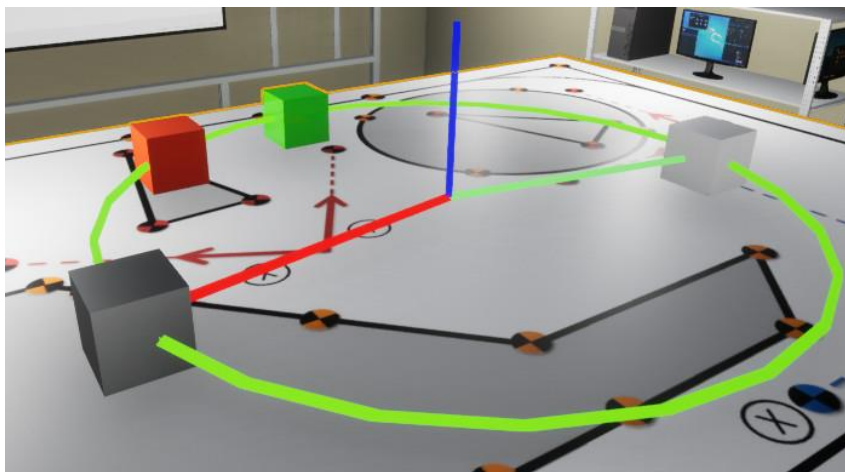
A körpályán való mozgás egy forgatással történik. A kör középpontjából a kezdeti pontba mutató vektor kerül forgatásra α -val, egy a kör síkjára a kör középpontjában merőleges vektort használva forgástengelynek. Ez a vektor legyen $\vec{D}$.

$$\vec{D} = \vec{daxdc} \quad (13)$$

Amennyiben $\alpha = \beta + \gamma$, akkor α , ellenkező esetben $360 - \alpha$ lesz a forgatási fok.

A forgatásért az Unreal Engine 4-ben megtalálható „vektor tengely körüli elforgatása” (RotateVectorAroundAxis) csomópont felel, amelynek bemeneti értékei: Vektor bemenet, szög, tengely (In Vect, Angle Deg, Axis).

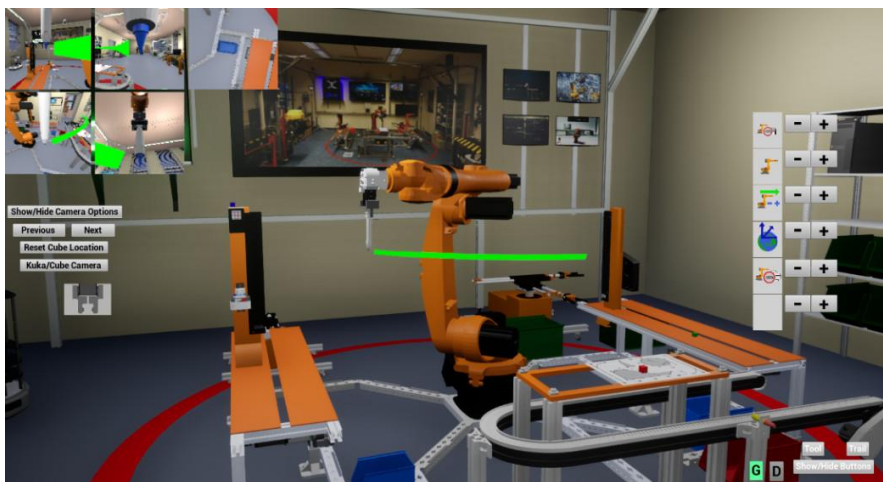
A „Debug Circle & Line” parancssal kirajzoltatható a kiszámított középpont alapján létrehozott kör. A 34. ábrán a szürke kocka jelképezi a kezdeti helyzetet, a narancssárga az segédpontot, a fehér a végpontot és a zöld kocka végzi a körmozgást.



34. ábra. CIRC mozgás végrehajtása a kiber-térben

A vörös vonal által jelképezett $\vec{da}$ vektor kerül megforgatásra a kék vonallal kirajzolt $\vec{D}$ vektor körül. A vektor tengely körüli elforgatás (RotateVectorAroundAxis) kimenete egy vektor, amelyhez hozzáadva a kör középpontjának koordinátáit megkapjuk a végpontot. Az idővonal (timeline) és a lineáris interpoláció (lerp) kombinációját használva körmozgás hozható létre, amennyiben a lerp „A” bemenet 0, a „B” bemenet pedig az elforgatási fok. A lineáris interpoláció kimenete csatlakozik a forgatás „Angle Deg” bemenetéhez. Egy egységnyi sugarú, origó középpontú, XY síkon elhelyezett kör esetében, ha $a = (1,0,0)$, $b = (0,1,0)$, $c = (-1,0,0)$, akkor $\alpha = 180^\circ$, $\beta = \gamma = 90^\circ$. Ugyanezeket a szögértékeket, ha $b = (0, -1, 0)$, a, c változatlan.

A digitális-ikerpár a 35. ábrán látható.



35. ábra. Robot kontrol a kibertérben

3.6. A Digitális-ikerpár program oktatásra gyakorolt hatásának vizsgálata

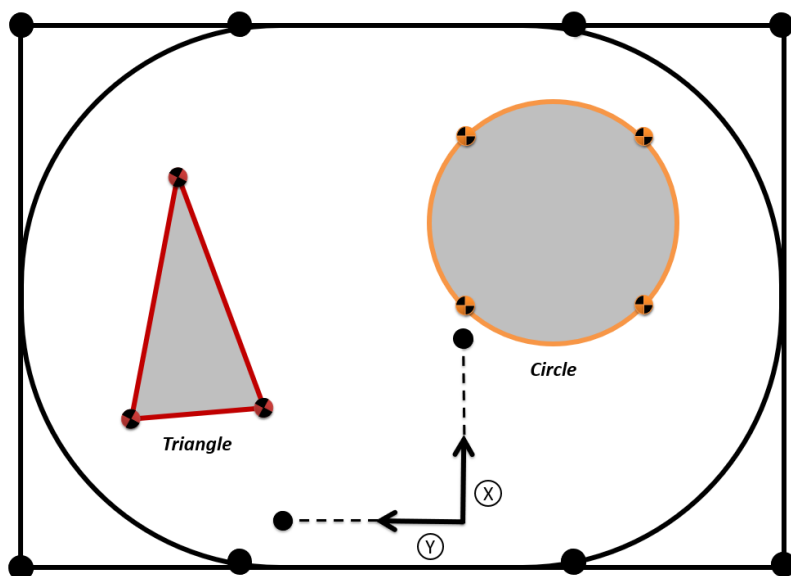
Hogy megvizsgálhassuk a rendszer oktatásra gyakorolt hatásait, önként jelentkező diákok bevonására került sor. Olyan személyek lettek kiválasztva, akik korábbi tanulmányaik alapján, sosem vettek részt robot programozási feladat végrehajtásában. A bevont személyek 40% nő volt, 60% -a pedig férfi. Az életkor szerinti megoszlás pedig 19 – 25 év között volt. A hallgatók közül gépészmérnök, mechatronikai és műszaki manager képzésen résztvevő személyek voltak. A résztvevők 50%-át magyar hallgatók alkották, míg a másik 50%-ot nemzetközi hallgatók, akik angol nyelvű kurzusokon tanultak.

A tíz bevont önkéntes diák két 5 fős csoportra lett osztva (Team-A, Team-B). Az „A” csapatnak lehetősége volt a Digitális-ikerpár program használatára. A „B” csoport tagjai ezzel szemben a hagyományos, írott dokumentum csomagot kapták meg (.PPT, .PDF, .Prezi). Ezek után egy előre meghatározott időpontban jelent meg a két csoport a laborban és a KUKA KR5 robottal meghatározott feladatokat végeztek el.

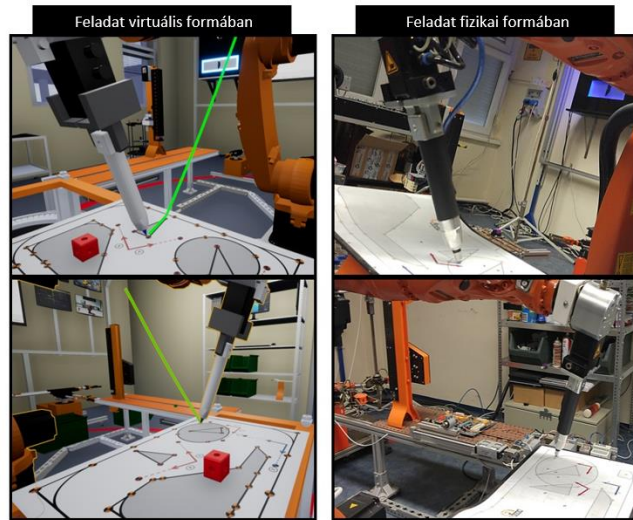
A KUKA KR5 fizikai programozása során összetett pályabejárási feladatot kellett elvégezniük, LIN és CIRC mozgások végrehajtásával. Az időt pedig stopperrel mértük, hogy lássuk mennyi idő alatt teljesítik a feladatokat.

A feladat elvégzése során csak az adott hallgató és az oktató volt jelen, aki segítséget nem nyújtott a hallgató számára, és mindössze a fellépő veszélyhelyzet elkerülése esetén avatkozott be.

A 36. ábrán látható a geometriai alakzatokat tartalmazó lap.



36. ábra. Alapszintű pályabejárási lap



37. ábra. Feladat végrehajtása a valóságban és Digitális-ikerben

A két feladat LIN mozgás alkalmazásával egy háromszög, és egy CIRC körív mozgás végrehajtása volt, amit az ábra 37. reprezentál.

Az eredmények a 3. és 4. táblázatban kerültek összegzésre.

Táblázat 3. „A” csapat – Kiberfizikai alapú program használatával elért eredmény

„A” kék csapat			
	LIN – Háromszög	CIRC - Kör	Idő (perc)
1	6:18	12:27	18:45
2	8:45	10:16	19:01
3	4:35	9:56	14:31
4	9:27	15:37	25:04
5	7:03	8:46	15:49

Táblázat 4. „B” csapat – Hagyományos dokumentumok használatával elért eredmény

„B” piros csapat			
	LIN – Háromszög	CIRC - Kör	Idő (perc)
1	14:24	16:37	31:01
2	8:36	18:47	27:47
3	10:22	15:36	25:36
4	17:01	11:18	28:19
5	9:34	21:58	31:32

A feladat végeztével mindkét csapat tagjainak ki kellett tölteni egy Google Forms alapú kérdőívet [87]. A kérdésekre 1-5 terjedő skálán válaszoltak.

A 10 feltett kérdés, a SUS azaz a rendszerhasználhatósági skála (System usability scale) elvét követték [88]. A visszajelzések azt mutatták, hogy az értékek 3.6-tól egészen az 5.0 értékig terjedtek, amelyek értelmében kijelenthető, hogy elégedettek voltak a programmal. Természetesen a résztvevő személyek anyanyelvén kerültek kiküldésre a kérdések.

Konklúzióként levonható, hogy a rendszeres használatra igény mutatkozik hallgatói oldalról (lásd táblázat 5.).

Táblázat 5. A SUS értékek és a megválaszolt kérdések

Sorszám	Válaszok	Átlag ± Eltérés	Terjedelem
K1	Felhasználóbarát a digitális-ikerpár program.	4.8±0.4	4-5
K2	A digitális-ikerpár program alkalmas a robotvezérlés megértéshez.	4.4±0.49	4-5
K3	Kényelmes használni a programot.	4.4±0.8	3-5
K4	A program hasznos a robot mozgások elsajátításához.	5±0	5
K5	A virtuális labor segít a logisztikai kérdések megválaszolásában.	3.6±0.49	3-4
K6	Az ember-gép interfész robotfunkciói jól integráltak.	4.6±0.49	4-5
K7	A gyakorlati képzés hatékonyabb a digitális-ikerpár programmal.	4.2±0.4	4-5
K8	A virtuális ikerpár más műszaki területen is alkalmazható.	4±0.632	3-5
K9	Egyetértek, hogy hatékonyabban gépvezérlés elsajátítása.	4.8±0.4	4-5
K10	A digitális-ikerpár növelte az érdeklődési körömet a robotika iránt.	5±0	5

Megállapítást nyert, hogy azon csoport tagjai, akiknek lehetőségük volt a program használatára, képesek voltak gyorsabban befejezni a kiadott feladatot szemben azokkal, akik csak a hagyományos oktatási írott dokumentumokból készültek fel a feladatra.

3.7. Fejezet konklúziója

A diákok oktatásának színvonalának növelése érdekében elkészült a labor Digitális-ikerpár típusú változata. Az új innovatív megoldással lehetőségünk van a labor kapacitást érintő probléma áthidalására. A meghibásodási és fenntartási költségekkel sem kell számolnunk, mivel a labor virtuális formában érhető el.

A hallgatóknak lehetőségük van az előzetes fizikai gyakorlati órák megkezdése előtt gyakorolni a robot kezelést-programozást.

A későbbiekben, további belső útmutató anyagok is hozzáadásra kerülnek a rendszerhez a könnyebb kezelhetőség érdekében.

4. Kiterjesztett valóság alkalmazás a KUKA KR5 ipari robotegység vizualizációjához

Az ipar 4.0 technológia magába foglalja a virtuális terek alkalmazását, ami hozzájárul a termelékenység növeléséhez. A kiterjesztett és a virtuális valóság (Augmented and Virtual reality) technológiák közvetlenül és közvetetten is pozitívan tudják befolyásolni a gyártási folyamatokat alkalmazásuk esetén.

Miután a labor Digitális-ikerpár változata megalkotásra került, rendelkezünk annak minden elemkészletével is. Az önálló applikáció formában való használata megköveteli az erősebb hardveres gépigényt, valamint az egyéb informatikai perifériákat (billentyűzet, monitort, egér stb.). Amennyiben viszont a KUKA KR5 önállóan kerül kiemelésre kiterjesztett valóság technika alkalmazásával, úgy egy merőben másabb oktatási formációnak is megalapozhatunk.

A fő témaköre ennek az alfejezetnek, hogy interaktív alapokon kiterjesztett valóság technikát alkalmazva (AR) elérhetővé tegyük az oktatás fókuszában lévő KUKA KR5 robotegységet.

A fejlesztés révén a hallgatóknak egyéb járulékos költség nélkül, mobil feldolgozó egységüket használva lesz lehetőségük AR alapon a gép irányítására, amivel közvetlenül kapcsolatba tudnak lépni. A tengelyenként vezérlés, illetve a munkatér és a szerszámközéppont által megtett útvonal is jelölésre kerül. Amennyiben az okostelefon applikációjaként eljuttatásra kerül a végfelhasználóhoz a marker megadásával együtt, úgy az adott felhasználónak lehetősége lesz az adott robotegység pontos megismerésére.

4.1. Robot fájl formátum és Unity fejlesztői környezet

Az AR és VR technológiákat széles körben alkalmazzák az ipari szektorban. Elsősorban a tervezésben és a karbantartási feladatok ellátásában kapnak jelentős szerepet [89].

A megjelenítő motorok között számos ingyenes vagy nyílt-forráskódú érhető el. Ezek alkalmazását elsődlegesen az adott feladat határozza meg. Vannak olyan motorok, amelyeket a nagy nyílt terek kezelésére terveztek (Unreal Engine) és vannak, amik mindössze 2D-s megjelenítést teszik lehetővé.

Esetünkben az AR/VR alapú megjelenítésre alkalmas motor volt a fő preferencia, ezért esett a választás a Unity -ra. Maga motor ingyenesen elérhető program tervezési célokra. Előnyei közé tartozik, hogy képes a nagy adatbázisok kezelésére valós idejű fény- árnyék képzésére, valamint számos harmadik-féltől származó modul (third party plugin) érhető el. A Unity támogatja az ismertebb 3D CAD (.FBX, .OBJ) fájl formátumokat [90].

A Blender és a Unity motor között működésbiztos az átjárás, így a korábban beállított és központosított csoport modellek a KUKA KR5 robotnak, az RGB színekkel együtt átvitetők .FBX fájl exportálással.

Azért, hogy viszonyítási alapunk legyen, egy korábbi projektem kapcsán a Robotikai operációs rendszer (ROS OS) Gazebo Linux környezetben alkalmazott URDF (Unified Robotics Description Format) fájl, mintaként szolgált a KR5 tekintetében. Fontos megjegyezni, hogy a Gazebo elsősorban szimulációs tesztelési célokat szolgál, nem pedig alkalmazások fejlesztését végfelhasználók részére [91].

Az alkalmazandó robot kinematikai modelljét meg kell határozni annak mozgatásának érdekében a virtuális térben. Ezért az egyes elemeket úgy nevezett „*empty game objects*”-ekhez rendeltünk.

A Unity-be importált KUKA KR5 a 38. ábrán tekinthető meg.



38. ábra. KUKA KR5 CAD modell Unity fejlesztői környezetben

A robot maga egy kinematikai lánc, amit csuklók alkotnak. Van tömegük, inerciájuk és végállásuk.

Ebben nyújt egyfajta keretrendszert számunkra a ROS Gazebo-nál alkalmazott URDF, amit a különböző robotoknál alkalmaznak.

A formátum maga a robotkarokra jellemző felépítést taglalja szkriptek formájában (39. ábra). Maguk az URDF fájlok a ROS nyílt adatbázisából kérdezhetők le a fájl konvertálását követően [92].

```
<robot name="kuka_kr5_arc">
  <!-- LINKS -->
  <!-- base link -->
  <link name="base_link">
    <inertial>
      <origin rpy="0 0 0" xyz="-0.017519394031846 8.11394768450848E-
07 0.0975924044854325"/>
      <mass value="7.58585707666653"/>
      <inertia ixx="0.0441803207364147" ixy="5.37433593375545E-
07" ixz="0.004045564920652" iyy="0.0584309411563095" iyz="-4.8990117845347E-
07" izz="0.0514875586171667"/>
    </inertial>
    <visual>
      <origin rpy="0 0 0" xyz="0 0 0"/>
      <geometry>
        <mesh filename="package://kuka_kr5_support/meshes/kr5_arc/vis-
ual/base_link.dae"/>
      </geometry>
    </visual>
    <collision>
      <origin rpy="0 0 0" xyz="0 0 0"/>
      <geometry>
        <mesh filename="package://kuka_kr5_support/meshes/kr5_arc/colli-
sion/base_link.stl"/>
      </geometry>
    </collision>
  </link>
```

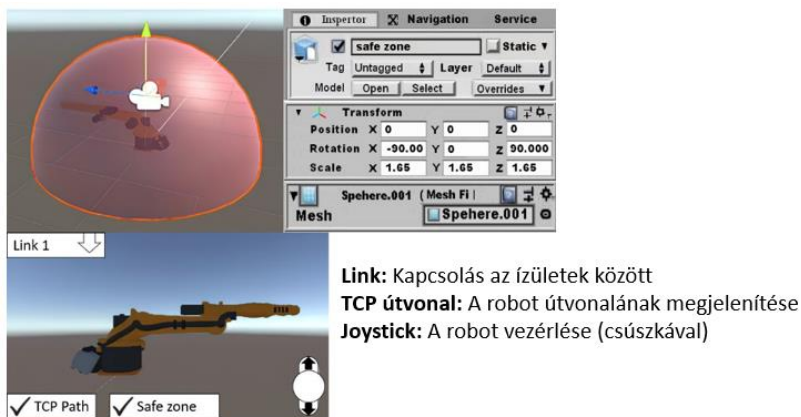
39. ábra. KUKA KR5 URDF fájl

Annak érdekében, hogy a robot ne tudjon átmenni a saját 3D modelljén, „*child element collision*” -okat kell hozzárendelni az egyes tagokhoz, ekkor pedig a (mesh) tömeg elemek nem fognak tudni áthaladni egymáson. A „*child element*” az inercia helyett az adott link (csukló) tömegét írja le, így meghatározható a tömegközéppont és a tehetetlenségi tenzor.

A csukló lehet prizmatikus, rotációs vagy fix rögzített, és tartalmazhatja a fizikai és dinamikai paramétereket. A „*limit element*” tárolja a rotációs és transzlációs értékeket, valamint az elérhető maximális sebességet is.

4.2. A munkatér és grafikus felület vizualizáció

A vizualizáció fontos része a robotikának, mivel a gépek munkaterében való tartózkodás a feladat-munkavégzése során szigorúan tilos. A balesetvédelmi szabályok betartása pedig kötelező érvényű. Az adott gépegységhez tartozó munkatér tartományának jelölése fontos szempont, hiszen ez az a terület, amelyet a gép el tud érni. A robot munkaterének jelölését, a félgömb (primitive shape) 3D CAD modell hivatott jelölni (40. ábra).



40. ábra. Munkatér és grafikus felület a KUKA KR5 robot kontrolhoz

Az adott csukló kontrolhoz külön legördülő menü lett hozzáadva. Alatta egy függőleges joystick kapott helyett. Egyszerre csak egy csukló vezérlése lehetséges direkt kinematikának megfelelően.

A joystick tartomány értéke -1 - +1 terjed, és minden ízülete rendelkezik végállással, amelyek a fizikai gépegységre is érvényesek.

Ahogy az a Unity C# fejlesztői környezetében is látható, a „*Controller GameObject*” képes a megadott értéknek megfelelően beállítani a csuklókat. A kontroller script meghívja az összes csuklót, és a joystick értékének megfelelően az elfordulás értékét veszi fel. Továbbá a szerszámközéppont útja is jelölésre kerül a munkatérben.

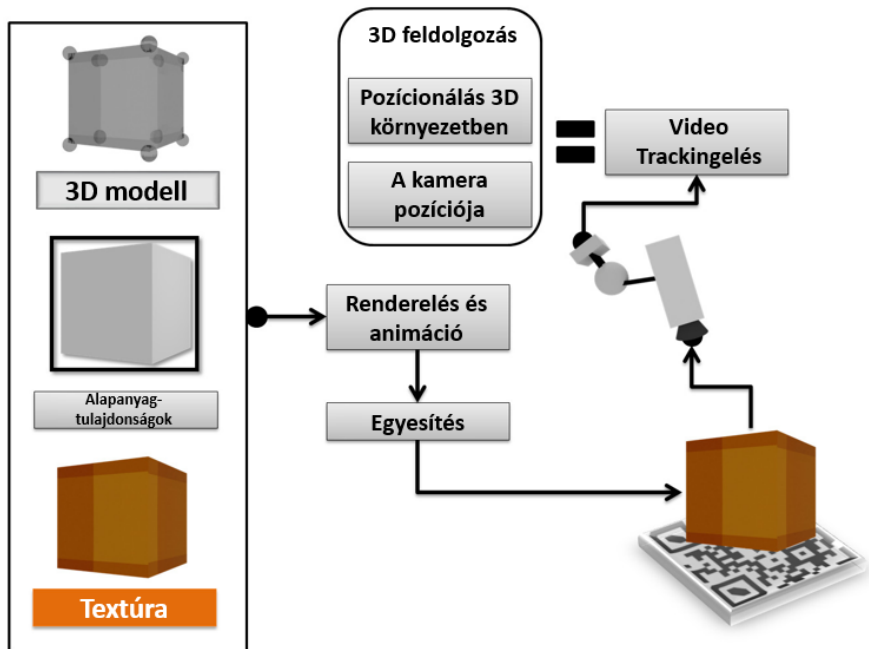
4.3. KUKA KR5 beágyazása kiterjesztett valóság alapú markerbe

A kiterjesztett valóság egy fajta kibővítése a valóságunknak, plusz térbeli elemekkel bővítve. Ezen elemek lehetnek passzívak és aktívak, amelyekkel kapcsolatba tudunk lépni. Az első kiterjesztett valóság (AR) technológiát alkalmazó program az 1960-as években készült [93].

A technológia három fontos elem köré épül: a 3D CAD modellre, a kamera egységre, valamint a jelölő (marker) elemre.

A jelölő központú kiterjesztett valóság rendszerek esetén egy virtuális objektum kerül elhelyezésre a mi valóságunkban. A kameraegység látóterébe kerülő jelölő segítségével pozicionálja és rá helyezi a 3D CAD modellt.

A kamera ebben az esetben egyfajta szkennelést hajt végre, hogy detektálja a jelölő kódot, és ezután vetíti rá a modellt (41. ábra). Általánosan elmondható, hogy a marker és a kameraegység egymáshoz viszonyított pozíciójából pontosan meghatározható az adott 3D alakzat minden egyes pontja.



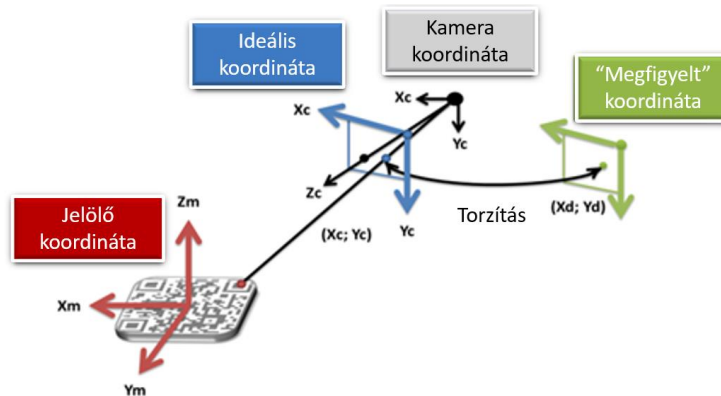
41. ábra. Kiterjesztett valóság alkalmazása és jelölő detektálás [94]

A kameraegység meghatározó tényező a detektálás szempontjából. A feladat elvégzése során, egy LG G4 H815 okostelefon került kiválasztásra, amely az alábbi paraméterekkel rendelkezik:

- CPU: Qualcomm Snapdragon 808 MSM8992
- GPU: Adreno 418
- RAM: 3 GB
- Hátlapi kamera: 16 MPx [95].

Fontos megjegyezni, hogy a mozgásmező nem „ideális”. Ennek oka, hogy a 3D-s tér nem minden elemének és pontjának a mozgását tudjuk meghatározni.

Az technika alkalmazásakor nem elégséges csak a mozgást detektálni, hanem a marker-kód dekódolása mellett, annak pontos orientációját és helyzetét is meg kell tudni határozni (42. ábra).

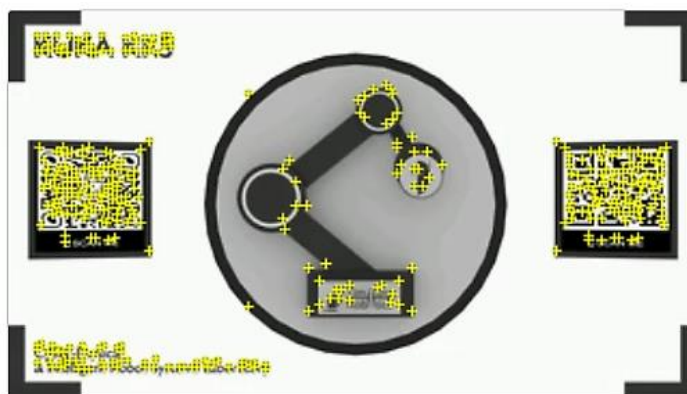


42. ábra. AR koordináta rendszer [96]

Esetünkben a marker több egyszerű jelölő elemnél, ugyanis egyfajta digitális kártyaként funkcionál az operátorok számára.

Ez azt jelenti, hogy nem csak 3D CAD KUKA KR5 modell pozicionálását segíti és vetíti rá a modellt a későbbi vezérlés céljából, hanem rendelkezik a géptípus dokumentációjával és a Kiberfizikai és intelligens rendszerek robotlaboratórium elérhetőségével is. A marker, ami a labor logóját is tartalmazza a Blender-ben lett kirenderelve .PNG fájl formátumban. Ezt követően a Vuforia AR SDK plugin-t telepítettük a Unity-ba, illetve feltöltöttük az általunk renderelt jelölőt konvertálási célból, ami ezután egy 1-5 terjedő skálán került osztályozásra, hogy mennyire alkalmas a tényleges markerkénti használatra [97].

A digitális jelölő a 43. ábrán tekinthető meg.



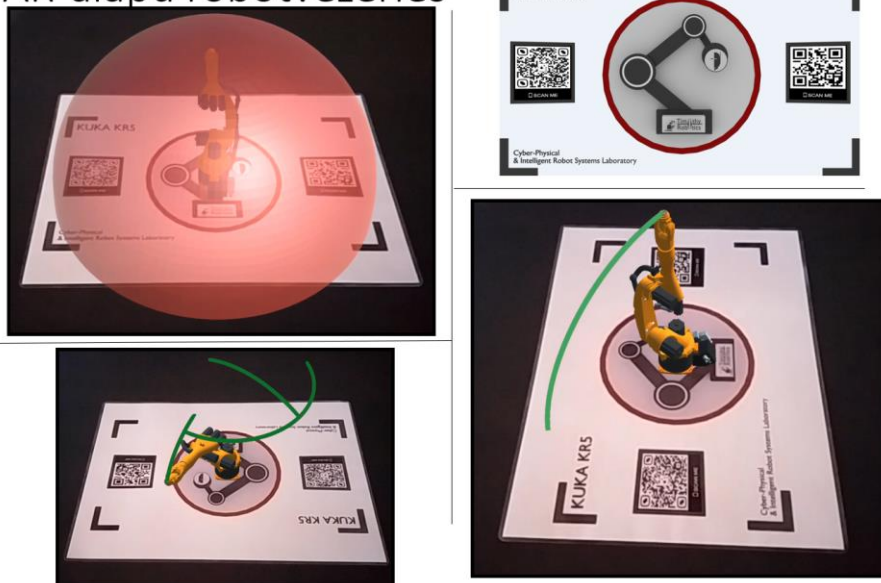
43. ábra. KUKA KR5 pozíció digitális jelölőkártya

A marker készítés végeztével exportáltuk a Unity jelenetet (scene), mint szabványos .apk alkalmazás kiterjesztést, így az Android Os okostelefonokra is telepíthetővé vált, mint önálló alkalmazás [98].

Az A4-es méretű marker és a telepített alkalmazás futtatását követően a KUKA KR5 tengelyeként vezérelhető, illetve a munkatere is vizuálisan megjelenik. Illetve ezen túl a szerszámközéppont (TCP) által megtett út is látható. Az A4-es marker lap méret elegendő részlettel rendelkezik az AR modell számára. Az eredeti modellhez képest arányosan lettek csökkentve robot méretei.

A futatott alkalmazást a digitális marker segítségével a 44. ábra prezentálja.

AR-alapú robotvezérlés



44. ábra. Android alkalmazás kiterjesztett valóság alapú KUKA robot vezérléshez

Ellenőrzésképpen a fizikai robotegységgel került párhuzamban állítva a virtuális változat (45. ábra). A KUKA KR5 fizikai robotegységgel felvettünk egy tetszőleges pozíciót, amelyet lekérdeztünk a KRC2 vezérlőn keresztül. A grafikus felhasználói felületen kiíratott értékeket felhasználva felvettük a pozíciót a virtuális kiterjesztett valóság alapú KUKA KR5 robottal, hogy lássuk azonos robotkonfiguráció reprodukálható-e.

Robot Position		
Name	Value	Unit
Tool/Base		
-	□NONE	Tool
-	□NONE	Base
Position		
X	396.160	mm
Y	40.539	mm
Z	1255.304	mm
Orientation		
A	4.30	deg
B	26.93	deg
C	10.59	deg



45. ábra. Fizikai és virtuális robot pozíció felvétele

4.4. Fejezet konklúziója

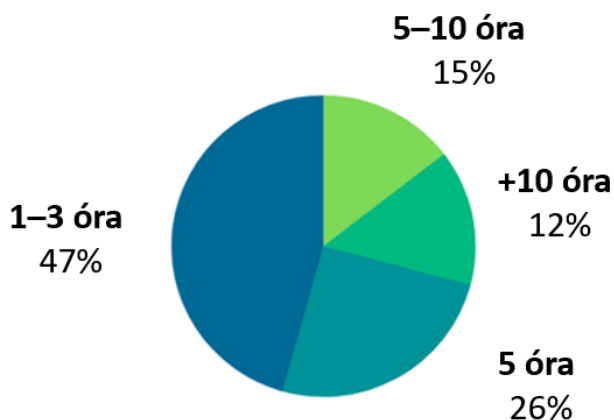
A COVID19-es járvány által életre hívott koncepciót önkéntes személyekkel teszteltük, annak céljából, hogy alkalmas lehet-e az applikáció oktatási formában való használatra. A technika előnye, hogy okostelefon segítségével dekódolható a marker kód, és vezérlési feladatokat is el lehet látni a robotegységgel.

Hallgatói véleményezéssel kapcsolatban összesen 30 hallgató került bevonásra, hogy választ adjanak a

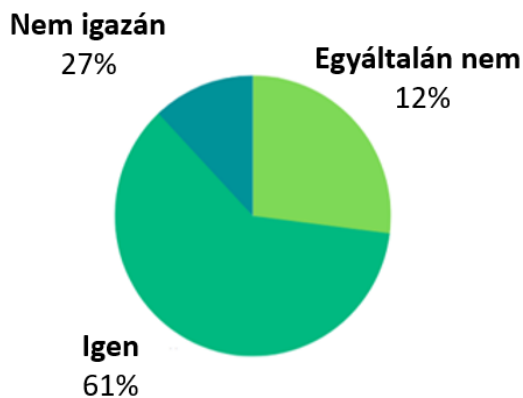
telefonhasználati szokásaikra, az applikáció használhatóságára és annak kezelhetőségére.

A válaszok a 46. ábrán láthatók.

Telefonhasználati szokás



Hasznos volt az AR alkalmazás a robot működésének megértéséhez?



46. ábra. Telefonhasználati szokások és a kiterjesztett valóság applikáció kezelése

A kérdéseket a Google forms szolgáltatással állítottuk össze és a válaszok Excel táblázatban kerültek feldolgozásra [99].

A növekvő mobil eszközök száma miatt lehetőség nyílt a kiterjesztett valóság alapú megoldások alkalmazására. Az okos eszközök növekvő teljesítménye lehetővé teszi, hogy kvázi valós időben történjen detektálása a markereknek és a 3D CAD modell illesztése. A technológia segít a felhasználónak az adott robotmodell működésének megértésében.

A megvalósítás során a KUKA KR5 robotkar AR alapú applikációja elkészítésre került, aminek alapja egy saját marker, amely egyben a robotra vonatkozó dokumentációkat is taglalja-részletezi.

A kiterjesztett valóság alkalmas komplex és-vagy absztrakt információk valóságghű reprezentálására. A módszer alkalmas lehet gyártási, kutatási és oktatási feladatok kiegészítésére.

5. Mélytanulás alkalmazása valós időben objektum felismerése céljából SCARA gépegységen

A Digitális-ikerpár 3D CAD modell készlet nemcsak arra szolgáltat lehetőséget, hogy elemeket mozgassunk a virtuális térben oktatási célból, hanem akár egyes előre meghatározott modellek szerepet játszhatnak neurális hálók tanításában, ha kiemeljük őket.

A tanulmány egyik kiemelt célja volt egy intelligens robotrendszer alapjainak lefektetése, amely az ipar 4.0 alapvető törekvéseit-elveit is magába foglalja. Neurális hálók alkalmazásával egy erre a célra kialakított izolált környezetben a módszer tesztelésére is sor került egy ipari robotegységen. A mélytanulás alkalmazásához szükséges feltétel kielégítéshez egy saját képalapú adatkészlet került létrehozásra adatszintézis által. A technikával lehetőséget teremtettünk arra, hogy az iparban alkalmazott elavultnak tekinthető gépek hatékonysági rátáját növeljük, ezáltal a termelékenységet is, plusz anyagi ráfordítás nélkül. A módszer végrehajtásához bemutatásra került a választott robot, amely üzembe állítása pillanatában nem volt alkalmas objektum detektálásra a munkaterében. Áttekintettük a témakörhöz vonatkozó kutatási területeket. Valamint részletesen dokumentálásra került a tetszőleges adatszintézis létrehozásához szükséges módszer, valamint a robotegység munkaterülete, annak 3D CAD modellezése, továbbá a számítógéppel renderelt (CGI) képi anyagok.

5.1. SONY SCARA SRX-611 robotegység

A globális alkatrész-chiphiány és a COVID-19 -es világjárvány egységesen felvetette az igényt arra, hogy régebbi gépegységek is rendelkezzenek a mélytanulás valamilyen formájával, ami az ipar 4.0 egyik fő kritériuma.

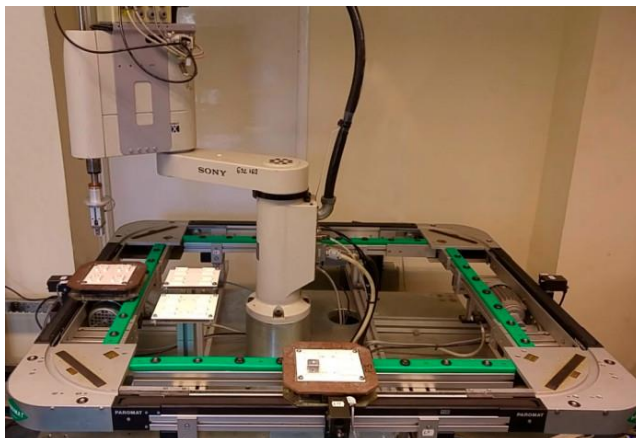
A labor több modern és régebbi gépegységgel rendelkezik, ami kutatásaink egy meghatározó részét képezi annak céljából, hogy módot találjunk ezen telepített

gépek alkalmazhatóságának kibővítésére. Kijelenthető, hogy a korábban telepített gépek több szempontból elavultnak tekinthetők az iparban. Ennek oka, hogy az ipari gépek kirívóan magasabb életciklussal rendelkeznek, ezért gyakran 50-60 éves gépekkel is találkozni lehet a gyártásban. Többségüket nem úgy tervezték, hogy lehetőség legyen hardver és szoftveres bővítésre-frissítésre. Az üzembeállításuk pillanatában nem rendelkeztek azon modern ipari szabványokkal, amik ma már szükségszerűek a jól strukturált, flexibilis gyártósorok kialakításához [100].

Az elavult gépek átalakítása önmagában is komoly kihívást jelent, mivel a vezérlést szolgáltató hardver beágyazott operációs rendszeren alapul (Embedded Operating System), ami nem teszi lehetővé kifinomult képelemző algoritmusok futtatását.

A fentieket figyelembe véve, kidolgozásra került egy olyan módszer, ami lehetőséget szolgáltat arra, hogy régebbi gépegységek is sikerrel tudjanak mélytanulás alapú algoritmusokat alkalmazni, hozzáadott anyagi ráfordítás nélkül. Az általunk választott és vizsgált robot, amelyen funkció bővítést hajtottunk végre egy SCARA modell. A Japánban gyártott Sony SCARA SRX-611 robotot széles körben alkalmazták az iparban (lásd 47.ábra).

A gyártósori robotok különféle szempontok szerint kategorizálhatók: felépítés, vezérlés, alkalmazott segédenergia, integrálhatóság stb.



47. ábra. Sony SCARA SRX-611 CPS-LAB robotegység

Az ipari robotok használatának fő okai gazdasági eredetűek, ahogyan az a Robotika Nemzetközi Szövetsége (International of Robotics) jelentésében is szerepel [101]:

- Kiszámíthatóság a gyártásban
- Termelékenység növelése
- Hibás termékek számának csökkentése
- Üzemeltetési költségek csökkentése
- Magasabb termékminőség.

Következtetésképpen, a fentiek nem következnek szükségszerűen az ipari robotok integrálásából a gyártásba, viszont meghatározzák-konkretizálják a fejlődés irányát.

Az ipari rendszerek alkalmazása során figyelembe kell venni az esetlegesen fellépő baleseti-vészhelyzeteket is.

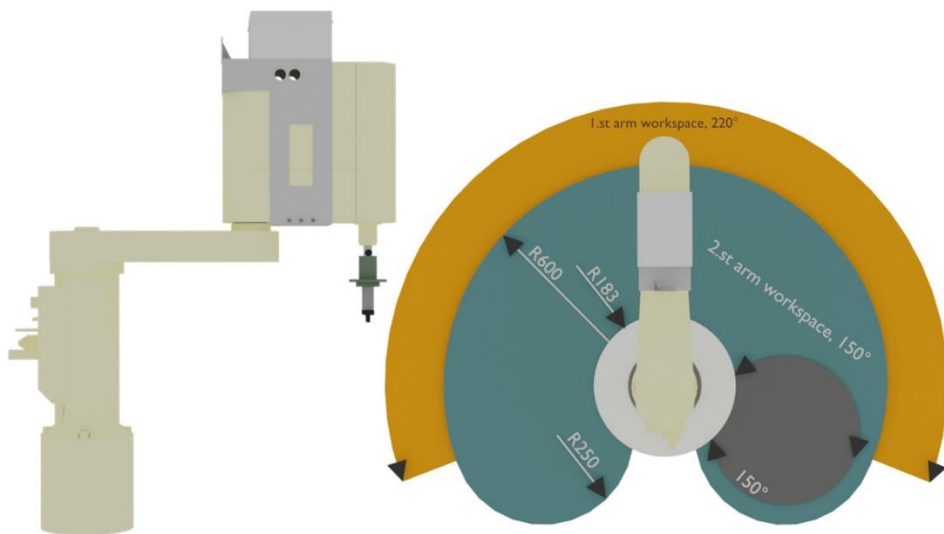
Esetünkben fontos megemlíteni, hogy a mélytanulás algoritmus, ami arra szolgál, hogy a későbbiekben a robot munkatérében lévő objektumokat tudjuk detektálni és azonosítani, nem közvetlenül a Sony SCARA vezérlőjén kerül futtatásra.

A detektált objektumtól függően pedig nem fog végrehajtani elkerülő manővert a munkatérében.

A SCARA gépegység (szelektív megfelelőségi összeszerelő robotkar), Hiroshi Makino Japán mérnök 1978-as tervén alapszik [102]. A SCARA típusú robotok alkalmasak nagy sebességű tengelymozgások végrehajtására. A gépeknél az egyik fő meghatározó paraméter a tengelyek száma.

A Sony SCARA esetén három tengely került kialakításra, amelyből kettő rotációs és egy transzlációs mozgást végez.

A gép RRT kialakítása meghatározza a robot munkatérét, ahol elvégezhetők a beültetési és összeszerelési feladatok. A gépre vonatkozó paraméterek a 48. ábrán és a 6. táblázatban láthatók [103].



48. ábra. SCARA SRX-611 munkatér és tengely paraméterek

Táblázat 6. Sony SCARA SRX paraméterek [103].

Kar hossza		Munkatér		Ciklus idő	Maximális sebesség		Ismétlési pontosság	
1.	350 mm	1.	220°	0,6 [s]				
Tengely		Tengely						
2.	250 mm	2.	+ - 150°					
Tengely		Tengely						
		Z.	150		Z	770	Z	+ - 0,02 mm
		Tengely	mm		Tengely	mm/sec	Tengely	
		R.	+ -		R	1150/sec	R a	+ - 0,03 mm
		Tengely	360°		Tengely		Tengely	

A gépegység munkaterét tekintve gömbjellegű (spherical). A karok terhelhetősége 2 [Kg], a gép össztömege pedig 35 [Kg]. A robotkarokra jellemzően ez a modell is háromfázisú AC szervomotorokat alkalmaz a tengelyek mozgására. A motorok inkrementális enkóderrel és fékekkel is el vannak látva, így a pozíciók lekérdezése és azok tartása is lehetséges. Az összeszerelés/beültetési feladatok ellátásához kétujjú elektro-pneumatikus megfogó áll rendelkezésre. A robot munkaterében található, egy PARO QE 01 31–

6000 futószalag rendszer, amely az alapanyagok helyváltoztatását teszi lehetővé palettákkal. A közlekedő alapanyagokat induktív szenzorok érzékelik, és szükség szerint képesek megállítani a palettákat (stop mechanizmus) [104].

5.2. Robot kontrol és programozási nyelv

Az ipari robotegységek hardver komponensei különböznek a kereskedelmi forgalomban kapható (commercial) besorolású asztali gépektől. Ennek oka, hogy ipari szabványoknak kell megfelelniük és ipari környezetben kell üzembiztosan működniük.

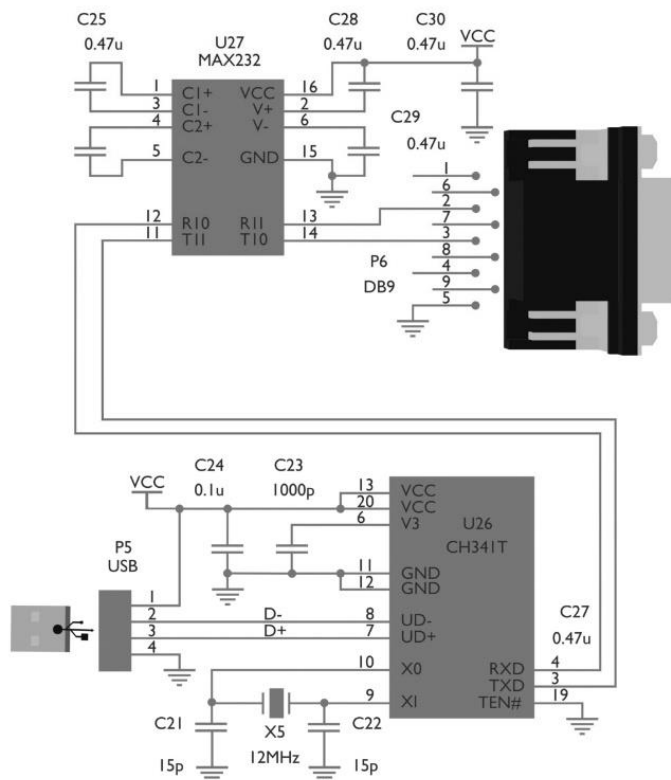
A Sony SCARA vezérlési rendszer három jól elkülöníthető részre bontható. Az első maga az SRX-611 robotkar elem és a PARO QE 01 31–6000 futószalag rendszer. A második az SRX-Robot kontroller, ami LED fénykibocsátó dióda indikátorokkal van ellátva. A harmadik elem pedig a programozáshoz, koordináta váltáshoz és sebesség megadásához szükséges úgynevezett tanító-programozó eszköz (Tech Pendant). Esetünkben azonban a SCARA robot nem rendelkezett programozó eszközzel, így más alternatív módszert kellett kieszközölni.

A régebbi robot modellek már nem rendelkeznek semmilyen, a gyártói oldal felől támogatással. Gazdasági okokra hivatkozva, az újabb termék paletta beszerzését preferálják. Emiatt egy adott gyártórészlegen mindig döntést kell hozni, hogy a meglévő gépparkot használják e tovább azok meglévő korlátozott funkcióival, vagy lecserélés-kiváltása mellett döntenek, ami komoly költségvetési vonzattal jár.

A fentieket figyelembe véve, esetünkben az a döntés született, hogy a laborban található SRX-611 nem kerül lecserélésre, hanem a meglévő, rendelkezésre álló hardvert felhasználva tesszük lehetővé mélytanulás alkalmazását (convert to smart). Az SRX robotkar vezérlője, egy Intel i486DX2 (50 MHz) CPU [103] köré épül, amely önmagában is egyfajta korlátot jelent, mivel a hardver erőforrás nem teszi lehetővé képelemzési feladatok ellátását. Rontja továbbá a lehetőségeket, hogy MS-DOS (Microsoft Lemezes Operációs Rendszer) alapú operációs

rendszert használ. A kommunikáció soros porton (serial porton) keresztül lehetséges a robottal.

Megoldás lehet a jövőre nézve, ha egy számunkra elérhető Asztali-PC hardver erőforrásainak egy része leválasztásra kerül, és a robot vezérlési feladatokra allokáljuk. Ez mindenképp szükségessé teszi, a soros porton keresztül történő kommunikációt. Esetünkben ehhez ez GB-USBR232 USB 2.0- to-RS232 konverter került felhasználásra. A busz konverter egy CH341 chipet tartalmaz (lásd ábra 49.), amely biztosítja a szinkron és párhuzamos port (Synchronous Serial Port - SSP) kommunikációját [105]. Amennyiben valamilyen formában emulálni tudjuk az említett operációs rendszert a rendelkezésünkre álló hardver erőforrás mellett, akkor a SRX PLATFORM futtatása lehetséges.



49. ábra. Szabványosított CH341 USB-to-RS232 soros interfész

Az SRX PLATFORM, ami az „Offline” programozási feladatok ellátását hivatott végrehajtani, csak DOS/WindowsXP platformot támogat. A DOS, vagyis a lemezes operációs rendszer semmilyen formában sem támogatja a jelenleg használatban lévő modern szabványokat. 2024-ben pedig, a Microsoft a Windows XP hivatalos támogatását is megszüntette. Továbbá a mostani alaplapok és a driverek sem támogatják már a két OS-t [106].

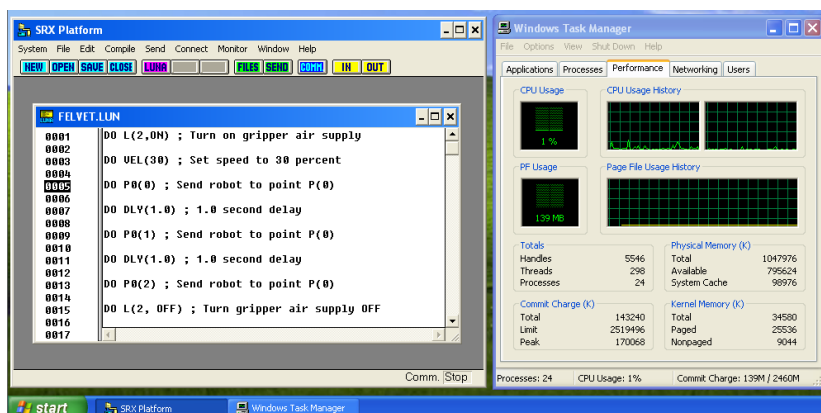
A két operációs rendszer közül a Windows XP lett kiválasztva, amelyhez VMware Player-t használtunk.

A VMware [107] rendelkezik azon funkcióval, hogy teljes rendszerek emulációját végrehajtsa, így képes rendszer specifikus illesztő programokat kezelni. Fontos kiemelni, hogy az emulált-szimulált operációs rendszer performanciája függőségi viszonyban áll a mindenkori konkrét operációs rendszer, vagyis a Core OS teljesítményt illetően, aminek hardveres teljesítménye kerül allokalásra a Windows XP futtatásához.

Asztali-PC paraméterei:

- CPU: Corei3 2021 (4-mag)
- RAM Memória: 4GB
- HDD: 500 GB
- Grafikus megjelenítő: Intel HD [108].

A VMware-t, egy Windows 10-et futtató Asztali-PC-re telepítettük, amely elérhető összes performanciájának felét allokaláltuk a Windows XP OS emulációjára. A virtuális gép az Intel Corei3 2021-ből két magot kapott, 2 GB RAM memóriát és 250 GB merevlemeznyi területet. A Windows XP rendszer, a Microsoft által leírt módon került telepítésre a VMware-en belül. Ezt követően a kötelező alapszintű illesztőprogramok lettek telepítve, a GB-USBRS232 USB 2.0-to-RS232 konverter driverrel együtt. A telepítését és konfigurálást követően lehetségessé vált az SRX PLATFORM futtatása (lásd 50. ábra).



50. ábra. SRX PLATFORM emulált Windows XP környezetben

A robotgyártó vállalatok saját programozási nyelvet preferálnak, ami az adott rendszer szerves részét képezi. A Sony SCARA esetén a LUNA programnyelv az, amit használ [103]. A LUNA programozási nyelvre jellemzően az egyes pontokat külön kell deklarálni, valamint az X, Y, Z koordináták mellett meg kell adni a forgástengely R értéket is. Az „offline” programozást lehetővé tevő, SRX PLATFORM grafikus felhasználói felületen definiálhatjuk a bemeneteket/kimeneteket, az elérendő pontokat, és a szövegrészeket, amelyek később hivatkozhatók lesznek a programban.

Azon sorok, amik DO-val kezdődnek végrehajtásra kerülnek, vagyis, ha a DO után megadásra kerül általunk egy koordináta a gép munkaterében, akkor a SCARA robotkar felveszi a kívánt pozíciót. A DO L-hoz (ON/OFF) más jelentés társul, ugyanis ezzel a programrésszel az elektro-pneumatikus megfogót levegő ellátását lehet vezérelni. A megfogó nyitása pedig a DO L8-al (ON/OFF) történik. A paletták deklarálására is van lehetőség, így társíthatunk hozzájuk nevet, számot és pozíciót is. Pl. D0 L5 (ON) L5 PALLET: PALL1. A D0 L5 (ON), jelentése, hogy aktiváljuk az L5 futószalagrészt, ahol a PALLET: PALL1 helyezkedik el. Az SRX PLATFORM-on belül megírt program, kiküldhető a SCARA robot részére a fordító (compile) menün keresztül. Ekkor generálunk egy .dat kiterjesztésű fájlt, a .pon kiterjesztésből. Az átfordítást követően, a „Send” vagyis

a küldés segítségével ez feltöltésre kerül a SCARA robotra. A Start gomb megnyomásával pedig lefut az általunk írt program.

A fentieket végrehajtva lehetségessé vált a Sony SCARA robot programozása (Tech Pendant) tanítóeszköz nélkül, emulált operációs rendszer környezetben, egy olyan Core Os segítségével, amely modern Windows 10 -et futtat, így a korábbi kompatibilitási problémák megszűntek.

5.3. SONY SCARA SRX-611 3D CAD modellezése

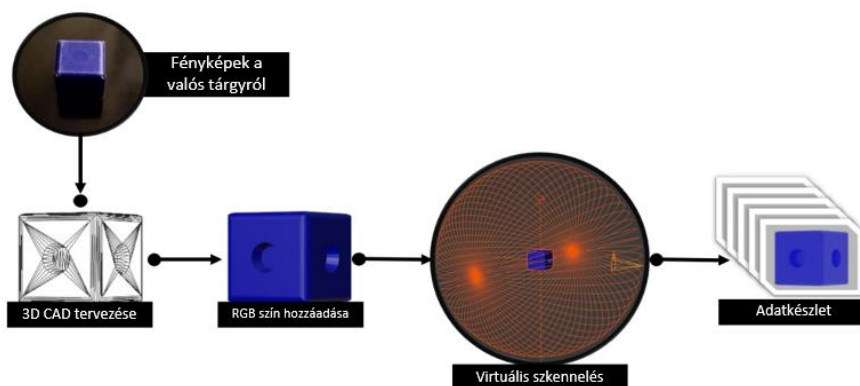
A korábban részletezett indokok egyértelművé tették, hogy a gépi tanulás direktben nem futtatható a SCARA vezérlőn, és az emulált Windows XP rendszeren sem. Új innovatív megoldásra kellett helyoznünk a hangsúlyt, amely alkalmazható anélkül, hogy figyelembe vennénk a hardveres komponenseket.

Fontos megjegyezni, hogy az ipari gépek önmagukban nem képesek, plusz érzékelők hiányában, detektálni a nyersanyagok pozícióját a térben. Gépi tanuláson alapuló módszer kidolgozása elégséges feltétel lehet a probléma leküzdésére, ha elegendő számú adatkészlet kerül létrehozásra a neurális háló tanításához, így a SCARA robot összeszerelő munkaterében detektálhatóvá és azonosíthatóvá válnak az egyes objektumok optikai szenzor segítségével.

Általánosan kijelenthető, hogy a többlépcsős folyamat egyik fő kritériuma a neurális háló tanításához a nagyszámú képi fájl, ami akár több tízezret is jelenthet. A nagy adathalmazok specifikáltan nem állnak rendelkezésre, így ezek előállítása rendkívül időigényes feladat, amennyiben valós képekből kerül összeállításra. Abban az esetben viszont, ha rendelkezünk az adott gépegység virtuális 3D CAD modelljével és annak felhasználni kívánt objektumaival (SONY SCARA), akkor a képzéshez szükséges adatállomány generálható lehet.

A koncepció lényege, hogy fényképek készülnek a detektálandó objektumokról, amelyet később használunk majd a mesterséges intelligencia alapú neurális háló tanításához. A képekből megalkotásra kerül a mindenkor 3D CAD modell, ami

tartalmazza az RGB szín paletta kódját, vagy a levett-másolt textúrát a fényképről. Ezt követően tetszőleges számban generálható CGI (számítógéppel generált) kép virtuális kamerával, amely szolgáltatja majd az adatkészlet alapját (lásd 51. ábra).



51. ábra. Neurális háló adatkészlet generálásának metodikája

Az elemek 3D CAD modelljének elkészítése a Trimble SketchUp-ban történt [109], amely a modell előállításánál alacsony poligon számmal hozza létre a modellt alkotó felületeket. SCARA SRX-611 CAD modellje során elvárt kritérium volt, hogy részletgazdag legyen, így több mint 1500 almodell alkotja a robotot. A megtervezett objektum ezután .OBJ file formátumban lett exportálva, amelynek a kiterjesztésére jellemzően tartalmazza a szükséges pozíciókat és modelleket. A CGI rendereléshez a Blender applikáció került kiválasztásra.

A modell importálását követően végrehajtásra kerültek az al-modelleket érintő optimalizációs műveletek, a renderelési folyamat elősegítése érdekében. Először is poligon szám került lecsökkentésre, ezzel is minimalizálva a mindenkori fájl méretet. Az egyes modellekhez, HSV (árnyalat, telítettség, fényesség) palettának megfelelő szín értéket rendeltünk, amely az eredeti képről származó fényképről került átültetésre. Ezzel is biztosítva a realisztikus ábrázolást, és a későbbi jobb objektum felismerést, hiszen csökkenthetjük a különbséget a virtuális 3D CAD modell és valós, fizikai objektumok között.

A specifikus textúrák nem kerültek átemelésre a fényképekről egy az egyben, ezzel is alacsonyan tartva a fájlméretet, ami meghatározó a renderelési sebességre nézve. A robot 3D CAD modellje az 52. ábrán tekinthető meg.



52. ábra. Sony SCRA SRX-611 3D CAD modell

5.4. Adatállomány előállítása renderelési technikákkal

Az előzőekben kifejtésre került, hogy a mélytanuláshoz nagy adathalmazra van szükség amiatt, hogy pontos eredményeket kaphassunk az objektum detektálását illetően. Ehhez valós képeket alkalmaznak, ami nemcsak magas hibafaktorról jár, de időigényes is, illetve, mint ismeretes, nincs két egyforma fénykép.

Ezt szem előtt tartva, egy olyan metódus lett kidolgozva, mellyel tetszőleges számban állíthatók elő képi anyagok, ha rendelkezésre áll az adott 3D modell. Így mind a kép mennyiséget, mind pedig a felbontást mi határozzuk meg.

A kiválasztott Blender program több renderelési lehetőséget is kínál, amelyek alkalmasak a CGI képek előállítására (textúrák, geometriák, fény-árnyék, kameraállások).

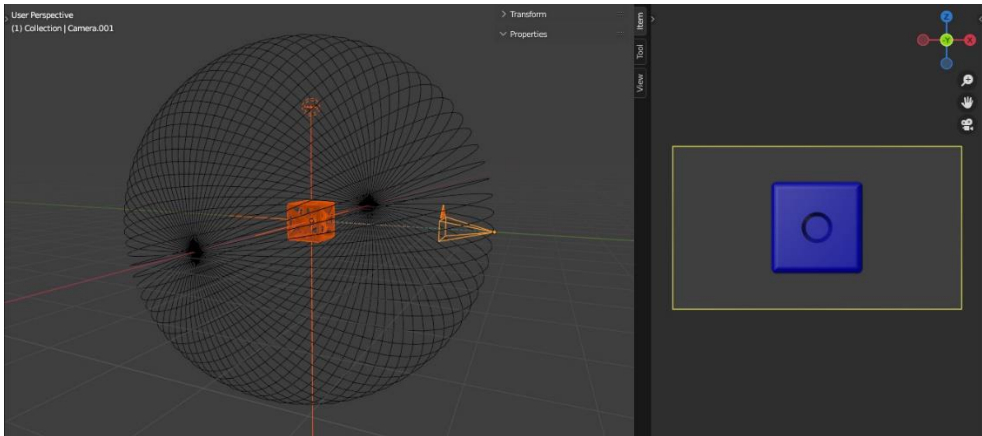
Az nyílt forráskódú rendszerek, szemben a zárt forráskódú programokkal, gyakran kapnak frissítéseket, ami hozzájárul azok fragmentációjához. A Blender esetén is jelentkezik a probléma, hogy az egyes LTS (hosszú távú támogatás) verziók kapnak javító csomagokat, és képesek az újabb verzióval készült fájlok megnyitására, de az esetek többségében nem maradéktalanul. A kompatibilitási problémák pedig a renderelési folyamatot lassíthatják vagy megszakíthatják.

Hibaforrásként léphet fel, hogy egyes verziók nem támogatják az adott rendereléshez használt GPU -t, ami GUI szinten is megjelenítési hibákat eredményezhet.

Az adatkészlet előállításához, a Blender 3.1 került felhasználásra, amely többek között tartalmazza az OptiX zajcsökkentő funkciót. A verzió kiválasztása azért volt előnyös, mert a bevezetett zajszűrő funkció lehetővé teszi alacsonyabb „mintavételezés” mellett is a tisztább-élesebb képet, így rövidebb idő alatt állítható elő egy képkocka.

Fontos kiemelni, hogy a CPU-k közül nem mindegyik alkalmas ezen funkció alkalmazására, mert az SSE 4.1 (Streaming SIMD kiterjesztések) az elvárt követelmények között szerepel.

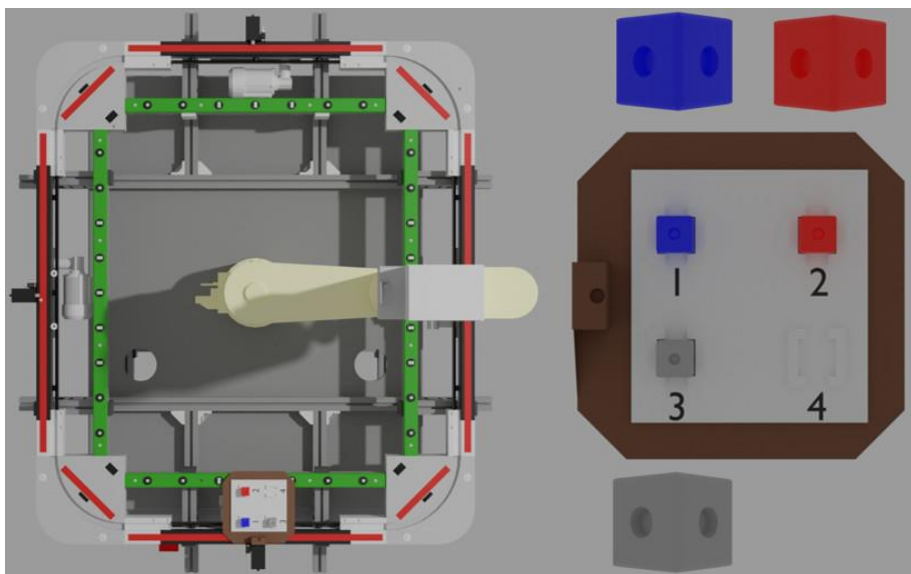
A képszintézishez elsőként egy jelenetet hozunk létre, ami egy fajta virtuális szkennerként fog funkcionálni (lásd 53. ábra) kulcsképkocka segítségével, oly módon, hogy a 3D-s virtuális tér origójába helyezett tetszőleges objektum körül egy szabályos kör helyezkedik el. Ez lesz a körpályája a virtuális szkennelés végző kamera egységnek. Egy kör megtétele a középpontba helyezett objektum körül 360 képkockát eredményez, amelyet az összes forgási tengelyen megismételtünk fokenként léptetve.



53. ábra. Virtuális objektum szkennelés

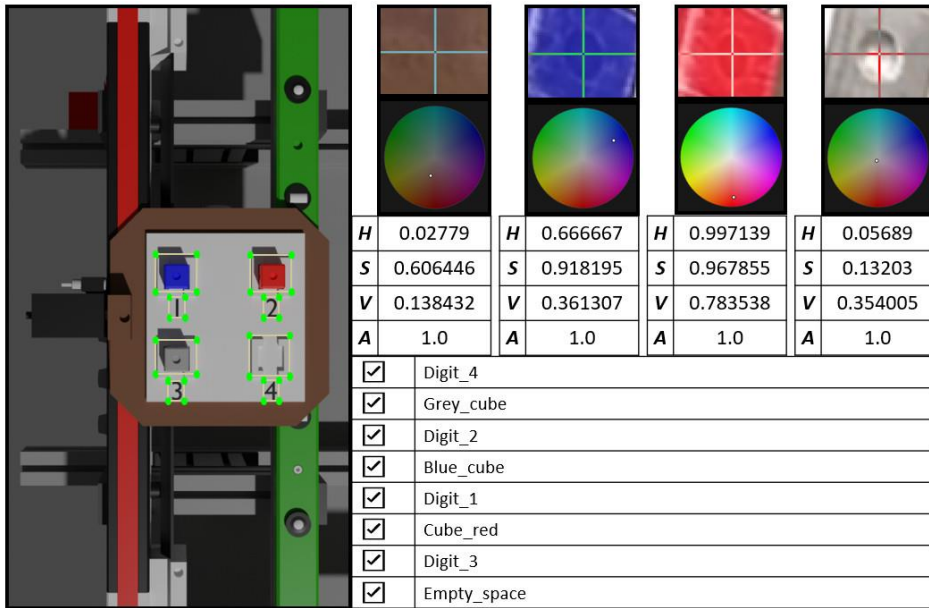
A következő lépésként hozzáadtuk a virtuális térhez a korábban megtervezett teljes Sony SCARA SRX-611 robot 3D CAD modelljét, és végig futtattuk a szkennert különböző felállásokban (lásd 54. ábra). Az így létrejött a neurális hálótanításához használt adathalmaz összesen 36.000 képet tartalmazott.

A renderelési beállításnál Cycles motort került alkalmazásra, amely mintavételének 32 érték lett megadva. A képkockánkénti felbontás pedig a Full-HD standard szabványt követve 1920x1080 pixel volt. A kimeneti fájl kiterjesztésnek pedig .PNG lett beállítva.



54. ábra. Renderelt virtuális objektumok gépi tanuláshoz

Az azonosíthatóság érdekében, az egyes elemek, amiknek detektálása kulcsfontosságú lesz, a valós fényképekről képekről rögzített színnel lettek renderelve. Vagyis a szín koordináták a valós-fizikai palettáról származnak, ahogy referencia kockáké is (lásd az 55. ábrát).



55. ábra. Címkezés és az alkalmazott szín értékkészlet

A munkafolyamat során nagy mennyiségű adatot kellett renderelni, ezért a Kiberfizikai és intelligens rendszerek robotlaboratórium gépparkja került bevonásra. A képek között egyenletesen kerültek szétosztásra a részfeladatok hálózaton keresztül.

A renderelést végző gépek alsó-közép kategóriájuk voltak:

- CPU: Corei7 3770
- RAM memória: 16 GB DDR3
- HDD: 500 GB
- Grafikus megjelenítő: Nvidia GTX1050 Ti [110].

Az adatkészlet előállítás folyamatban összesen három Asztali-PC vett részt és a renderelési ciklust 4 nap 5.16 óra alatt fejezték be.

Az annotációhoz a LabelImg [111] software került kiválasztásra, amely alkalmas több neurális háló által támogatott fájl formátum létrehozására [112]. A programban definiáltuk az objektumokra vonatkozó címkéket, majd kijelöltük az annotációkat és mentettük a képet a YOLO formátum szerint (lásd 55. ábra).

5.5. Alkalmazott mélytanulási architektúra

Ebben az alfejezetben kerül ismeretesre a Robotnet-RT detektorunk felépítése, amely a YOLO neurális háló implementációja (You Only Look Once) [113].

Esetünkben a tanításhoz használandó adathalmaz renderelését követően, az alábbi kritériumokat fogalmaztuk meg: valós idejű működés, megfelelő detektálási pontosság az adott objektumokra vonatkozóan, erőforrás igény minimalizálása, hogy alsó-középosztályú Asztali-PC-ken is alkalmazható legyen.

Az elvárásokra hivatkozva, a YOLO Neural Network architektúra került kiválasztásra [114], aminek implementációja C nyelvben íródott. Az architektúra futtathatósági követelményei is másabbak szemben a modernebb detektorokkal (SSD [115], DSSD [116], MobileNetV2 [117], R-FCN [118] és a Mask R-CNN [119]).

A neurális háló architektúra kiválasztása után, a saját DL detektor tervezése volt a következő lépés. Az elérhető és szabadon használható struktúra alapján kifejlesztésre került egy saját minimalizált verzió, hogy kielégítse a fentebb elvart kritériumokat.

Kiindulási pontnak a YOLOV3-tiny [120] lett választva, ami alkalmas minimalizált erőforrások mellett is kvázi valós idejű detektálásra.

Mivel nagy felbontású Full-HD szabványnak megfelelő képméreteink voltak, így elsőként csökkentettük a konvolúciós rétegek számát háromra a „Backbone” szakaszban, ezzel csökkentve a számítási igényt és növelve a fókusz a képen szereplő és detektálandó tartalomra. A filterek (szűrők), száma is csökkentésre került az optimális értékre, javítva ezzel a detektálási arányt.

Minden egyes módosítást követően újból tesztelésre került a detektor, amit finomra hangoltunk.

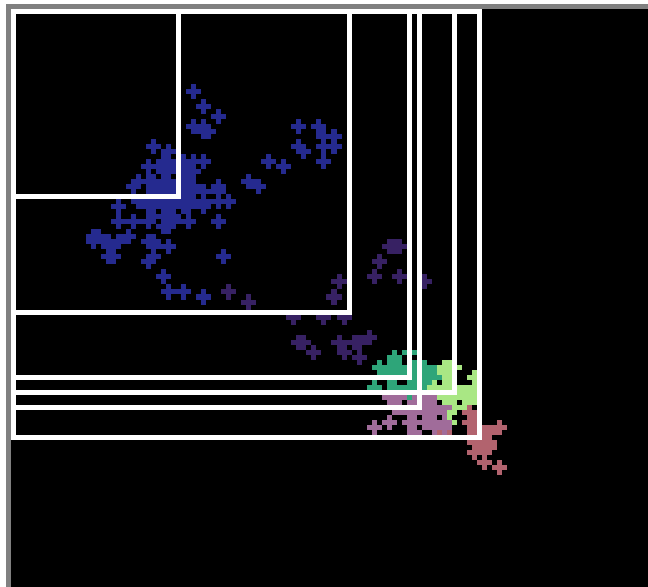
A következő elvégzendő lépés során módosításra kerültek a „shortcut layer”-ek, annak érdekében, hogy konvolúciós rétegek megkapják a megfelelő kimeneteket a „Backbone” szakaszban.

A befejező lépés részeként, a detektor rétegek véglegesítése történt meg, valamint a konvolúciós réteg F filter száma is meghatározása került egyenlet segítségével [121].

$$F = 3 \cdot (C + 5) \quad (14)$$

ahol C a detektálandó osztályok számát jelöli.

Ezt követően újra számoltuk a YOLO rétegekben megtalálható anchor box-okat, azaz a rögzítési keretek értékeit, amelyet a k -means ++ klaszterező algoritmus alkalmazásával valósítottunk meg. Ez egy meghatározó lépése a tervezésnek, mivel a választott rögzítési (anchor) értékek jelentősen befolyásolják a detektálás pontosságát. Az algoritmus kimenetét, és a hozzá tartozó metszet és unio arányának értéket (Intersection over Union) az 56. ábra szemlélteti.



56. ábra. A rögzített dobozok számított értékei k -means++ cluster algoritmust követően

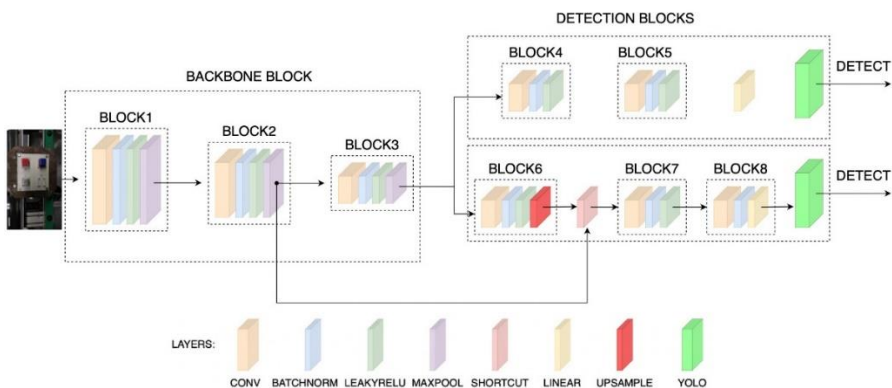
A rögzített dobozokra számított értékeknél a megadott paraméter IoU 0.8652 volt. A kialakított „design” két fő részre bontható, a Backbone-ra és a Detektor-ra. A Backbone három nagy blokkot tartalmaz, amely 3x3 konvolúciós réteget tartalmaz (4x4-es Batchnormalization layer, 2x2-es Maxpool layer, Activation layer). Az aktivációs függvény pedig a LeakyRELU [122], amellyel az alábbi egyenlet segítségével fejezhető ki:

$$y_i = \begin{cases} x_i & \text{if } x_i \geq 0, \\ \frac{x_i}{a_i} & \text{if } x_i < 0, \end{cases} \quad (15)$$

Ahol a_i egy fix paraméter lesz az s $(1, +\infty)$ értéktartományon.

A detektor maga további két nagyobb alblokkot tartalmaz (3x3 konvolúciós réteg). Ehhez is tartoznak Batchnormalization és Shortcut layer-ek. Minden subblock végén YOLO réteg található, az előzőleg kiszámolt 6db „anchor box” értékével. És minden konvolúciós réteghez tartozik aktivációs függvény is, amely ezekben az esetekben is LeakyRELU lesz, kivéve a YOLO réteget közvetlen megelőző konvolúciós layerben, ami lineáris aktivációs függvényt tartalmaz, ez a valószínűsítést fogja végezni.

A vázolt architektúra szerkezete az 57. ábrán tekinthető meg.



57. ábra. A detektor architektúra rétegetekkel

Részletesebben ismertetve az detektor architektúrát, a Backbone rész 3 fő blokkból áll, és az ezekben megtalálható 3x3-as konvolúciós réteg filter száma sorrendben 32, 128 és 256. A szegélyek és lépésköz (padding és stride) paraméterek értékei pedig egy lesznek. Az első MaxPool réteg 4x4-es kernelméretű a Block1-ben. A detektor részben a konvolúciós réteg filter száma sorrendben 256 és 512 lesz, kivéve a Block6-ban, ahol 128 értéket fog felvenni. Az felskálázó (Upsample) réteg 2x2-es, ahol a szegélyek és lépésköz paraméterek szintén egy értéket fognak felvenni.

5.6. Tanítás és kiértékelési folyamat

A soron következő fejezet részletezi a javasolt detektor tanítását, a választott adat augmentációs (data-augmentation) módszerrel.

A tanuló és megerősítő adathalmazt az előző fejezetekben taglaltak szerint építettük fel. A képgenerálás és renderelés után kapott képi halmazt az alábbi nagyobb alcsoportokra bontottuk fel a következő választott arányokkal: a tanuló adathalmazba a képek 80%-a, a megerősítő adathalmazba a képek 20%-a került bele, ezek számszerűsítve 28.800 és 7.200 képet jelentenek az egyes képi adathalmazok esetében.

Következő lépésként a főbb adathalmazok létrehozását követően további adat augmentációs eljárásokat alkalmaztunk [123], ezzel biztosítva a képek változatosságát. Ennek oka, hogy el akartuk kerülni a neurális háló „túltanítási” problémáját. Az alkalmazott algoritmusok és paramétereik a 7. táblázatban tekinthetők meg.

Táblázat 7. Adataugmentáció és paramétereik.

Metódus neve	Paraméter
Telítettség	1.2
Expozíció (fényerő beállítása)	1.2
Átméretezés	1.3
Szín telítettség	0.1

Az adat augmentációt követően kiválasztásra került a megfelelő optimalizáló (optimizer) algoritmus, amely a mi esetünkben az Adam [124] volt, aminek főbb paraméterei a momentum (pillanat) és a decay (csökkenő) értékeit tekintve 0.9 és 5×10^{-4} lettek.

Az optimáló megválasztását követően további tanulási paraméterek konfigurálása következett. Kiválasztottuk a megfelelő tanulási ráta (learning rate) ütemezőt, amellyel még pontosabb és stabilabb betanítás érhető el, esetünkben ez az exponenciális ütemező volt (exponential scheduler) volt, amelyet az alábbi képlet fejez ki [125]:

$$LR = LR_{initial} \cdot e^{-\frac{k}{epochs}} \quad (16)$$

ahol $LR_{initial}$ a kezdeti tanulási rátát jelöli, a k egy hiper paraméter, az „epochs” pedig az iterációk számát jelöli a tanítás során.

A tanulási ráta és a k értékét többszöri tanítási futtatása során határoztuk meg, felosztva az értéktartományt logaritmikusan. Ennek eredményeként, az első paraméter legjobb értékének az 5×10^{-4} , a második paraméternek pedig a 5×10^{-1} kaptuk.

További fontos paraméter még, a burn-in (bemelegítési szakasz) és a maximális epoch, amelyek szintén megadására kerültek. A burn-in alkalmazásának célja, hogy a tanulási ráta értékét fokozatosan növelje egy kezdeti igen alacsony értékről egész a maximális kezdeti választott értékig véges számú lépés alatt, ezáltal stabilan tartja az optimáló konvergenciáját és kiküszöböli a tanítás sikertelenségét, ami több esetben is előfordulhat a tanítási folyamat elején. A mi esetünkben az iteráció szám, amelynél a tanulási ráta (learning rate) eléri a maximális értékét, 160 lett.

Az epoch szám maximális értékét az alábbi képlet fejezi ki.

$$E = 2000 \cdot C \quad (17)$$

ahol E , a maximális értékét jelöli az epoch-nak, míg C a detektált osztályok számát. Esetünkben ezen érték 16.000 volt, mivel 8 darab detektálandó objektum van.

5.7. Mélytanulás alapú objektum felismerés és kísérleti eredményei

A fejezetben összefoglalásra kerülnek az általunk megalkotott Robonet-RT eredményei.

A YOLO architektúrán alapuló detektorok (loss function) veszteség függvénye, amelyet az optimalizáló a betanítási folyamat során minimalizál, a következő négy összetevőből áll [126,127]:

$$loss = clsLoss + locLoss + confLoss_d + confLoss_m \quad (18)$$

név szerint: osztályozási veszteség (classification loss), a lokalizációs veszteség (localization loss) és két esetben megbízhatósági veszteség (confidence loss) lép fel. A klasszifikációs veszteséget az alábbi egyenlet definiálja [126,127]:

$$clsLoss = \sum_{i=0}^{S^2} \iota_i^{obj} \sum_{c \in classes} (p_i(c) - \hat{p}_i(c))^2, \quad (19)$$

ahol S^2 a cellák számát jelöli, ι_i^{obj} pedig az adott detektálandó objektum meglétét az adott i cellában, amely 1-es értéket vesz fel abban az esetben, hogyha az adott objektum ténylegesen előfordul, máskülönben nulla lesz. A 19-es számú egyenletben, a $p_i(c)$ a feltételes osztályvalószínűséget jelöli (conditional class probability), a c osztály esetén minden cell i -ben, $\hat{p}_i(c)$ pedig, a hozzátartozó előre jelzett értéket.

A lokalizációs veszteség (localization loss), amely a valószínűsített határoló doboz (boundary box) pozíciót és méretbeli eltérését adja meg, a következő összefüggésből kapható meg [126,127]:

$$locLoss = \lambda_{coord} \sum_{i=0}^{S^2} \sum_{j=0}^B \iota_i^{obj} [(x_i - \hat{x}_i)^2 + (y_i - \hat{y}_i)^2] + \quad (20)$$

$$\lambda_{coord} \sum_{i=0}^{S^2} \sum_{j=0}^B \iota_i^{obj} \left[(\sqrt{w_i} - \sqrt{\hat{w}_i})^2 + (\sqrt{h_i} - \sqrt{\hat{h}_i})^2 \right], \quad (21)$$

ahol ι_i^{obj} szintén az adott objektum meglétét jelenti az adott i , cellában és értéke 1 lesz, abban az esetben, ha a j -edik határoló doboz az adott i cellában megtalálható (responsible) az objektum a detektálásnál, ellenkező esetben, ha nem történik észlelés, nulla értéket vesz fel.

A λ_{coord} megadja a megfelelő súlyozott értéket a határkoordinátákhoz, hogy a veszteségfüggvény a tanítás során konvergáljon, és a modell stabil legyen. A koordináta (x, y) az egyes cellák középpontjának koordinátaértékeit határozza meg, az S^2 a rácsok számát jelöli, B pedig az egyes cellákban lévő előre becsült dobozkét. A w és h jelölések az előre jelzett dobozok szélességét és magasságát jelölik. Az $\hat{x}, \hat{y}, \hat{w}, \hat{h}$ pedig, a háló által előre jelzett értékeket határozza meg.

Esetünkben, a megbízhatósági veszteség (confidence loss) kétféleképpen adható meg attól függően, hogy az objektumra vonatkoztatva történt-e észlelés. Ha az észlelés megtörtént, a veszteség a következőképpen fejezhető ki [126,127]:

$$confLoss_d = \sum_{i=0}^{S^2} \sum_{j=0}^B \iota_{ij}^{obj} (C_i - \hat{C}_i), \quad (22)$$

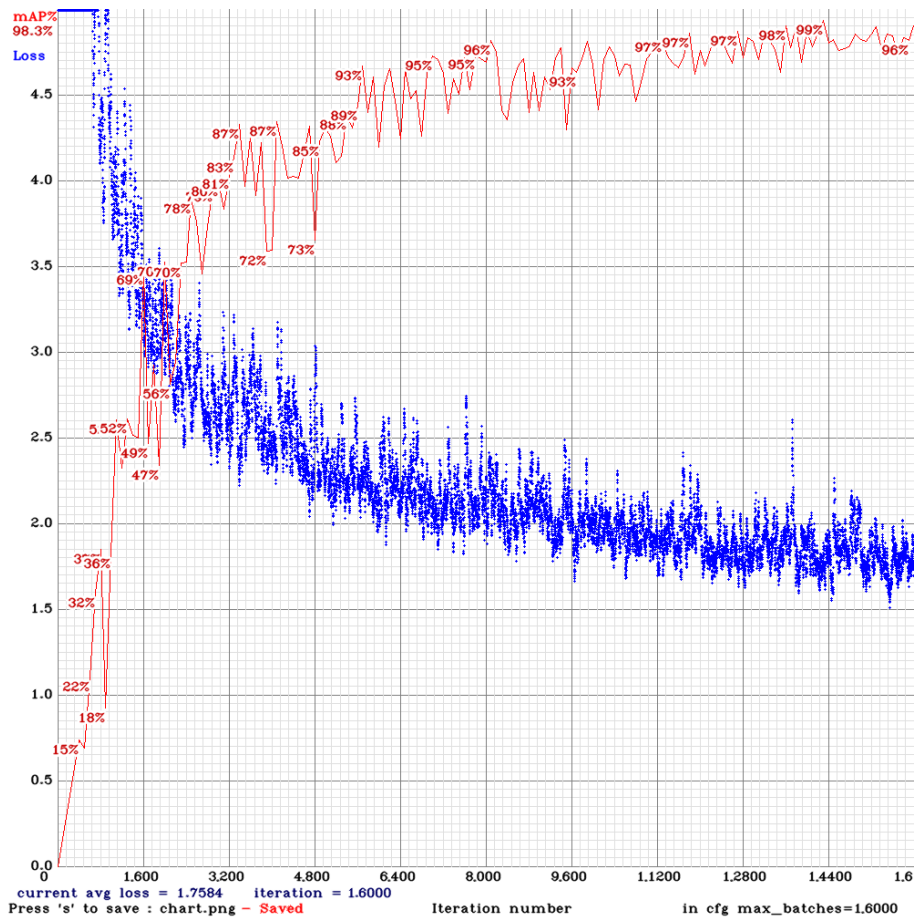
Ahol $\hat{C}_i$ a „box” úgynevezett bizalmi pontszámát jelöli (confidence score), minden i cellában és ι_{ij}^{obj} esetén, ami pedig az objektum meglétét jelöli.

Ha az objektum észlelése nem történt meg akkor a veszteséget a következő képen lehet kifejezni [126,127]:

$$confLoss_m = \lambda_{noobj} \sum_{i=0}^{S^2} \sum_{j=0}^B \iota_{ij}^{noobj} (C_i - \hat{C}_i), \quad (23)$$

ahol λ_{noobj} egy súlyérték lesz, ι_{ij}^{noobj} pedig az objektum hiányát jelzi. A többi jelölés jelentése megegyezik a 22. egyenletben tárgyaltakkal.

A neurális háló tanulási görbéje és a kapott találati ráta grafikus úton is kiíratásra került. A veszteségfüggvény értékeit a kék görbe, míg a hitelesítő adathalmaz értékeit a piros görbe mutatja százalékos pontosságú skálán. A kék görbe meredekségét vizsgálva a tanulási ráta és az ütemező kezdeti értékét megfelelően választottuk meg, és a piros hitelesítő görbe értékét figyelembe véve a mélytanulás detektor képes volt elérni az általunk elvárt pontosságú szintet (58. ábra).



58. ábra. Robonet-RT detektor veszteség- és hitelesítő görbéje (mAP)

A detektorral a videokártya által még kezelhető maximális értékekkel tanítottuk be. Esetünkben ez 16-os tételemméretet (batch size) és 8-as minitételemméretet (minibatch size) jelent. A bemeneti képek felbontása 960x512

volt. A betanított detektor eredményeit a táblázat foglalja össze. A táblázat az egyes osztályok felismerési pontosságát mutatja, a kiválasztott átlagos pontosság (AP) metrikával és a hozzájuk tartozó True Positive (TP) és False Positive (FP) értékekkel. Az értékelést 400 darab 960x512 méretű tesztképen végeztük el, amelyeket a tanulási és megerősítő adathalmazban szereplő képektől teljesen függetlenül generáltunk. A generált képek között létrehoztunk egy olyan képet is, amely nem tartalmazott objektumokat, ezzel vizsgáltuk detektorunk robusztusságát, és mértük a hibás észlelések számát (8. táblázat).

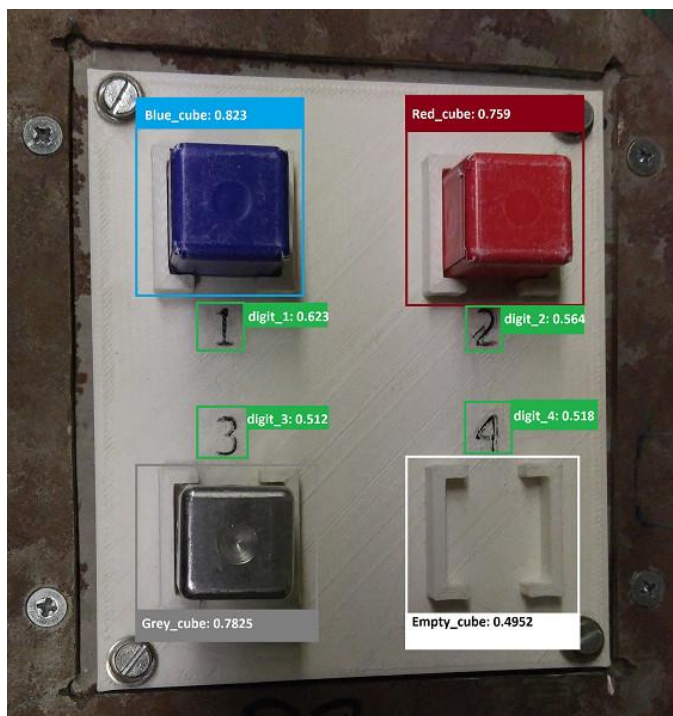
Táblázat 8. A javasolt Robonet-RT detektor kvantitatív eredményei

Osztályazonosító, Név	ÁP (%)	Igaz Pozitív (TP)	Hamis Pozitív (FP)
Osztály 0, számjegy_1	96.02	63	14
Osztály 1, számjegy_2	98.95	73	7
Osztály 2, számjegy_3	98.22	73	8
Osztály 3, számjegy_4	97.81	72	7
Osztály 4, kék_kocka	95.64	75	11
Osztály 5, piros_kocka	98.88	73	4
Osztály 6, üres_kocka	99.02	75	2
Osztály 7, szürke_kocka	97.65	76	3

A javasolt detektor pontosságát az összes osztályra vonatkozóan a következő mérőszámokkal mértük: Precizitás, visszahívás, F1-pontszám és átlagos pontosság (mAP). A kiértékelés során vizsgáltuk a precizitást (precision), a visszahívást (recall), az F1-értéket (F1-score) és az átlagos pontosságot (mAP). Az értékek: 0,93 precizitás, 0,93 visszahívás, 0,91 F1-score és 0,9865 volt az átlagos pontosságra vonatkozóan. Az átlagos észlelési sebesség 12,65 [ms], ami képkockákra átszámítva 79 képkockát jelent másodpercenként, ami megfelel az általunk támasztott kritériumnak.

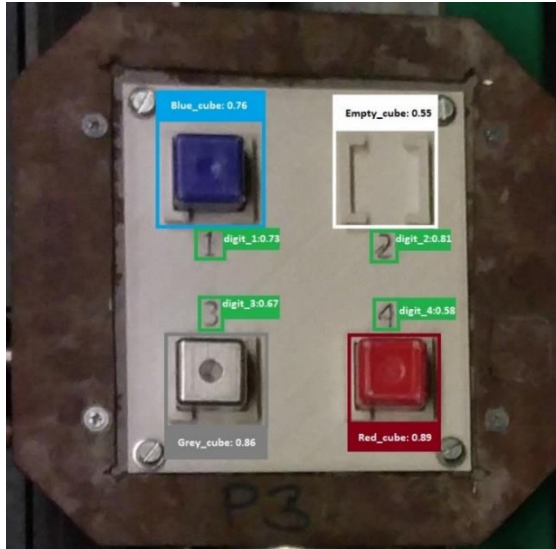
A tanított háló alkalmazásra került a gépegység fix szállítószalagos munkaterében, ahol elhelyeztük a 3D ABS+ nyomtatott palettán a referencia

munkadarabokat. Az általunk tanított háló képes volt felismerni az objektumok, illetve azok helyzetének változtatását is követően (59. ábra).



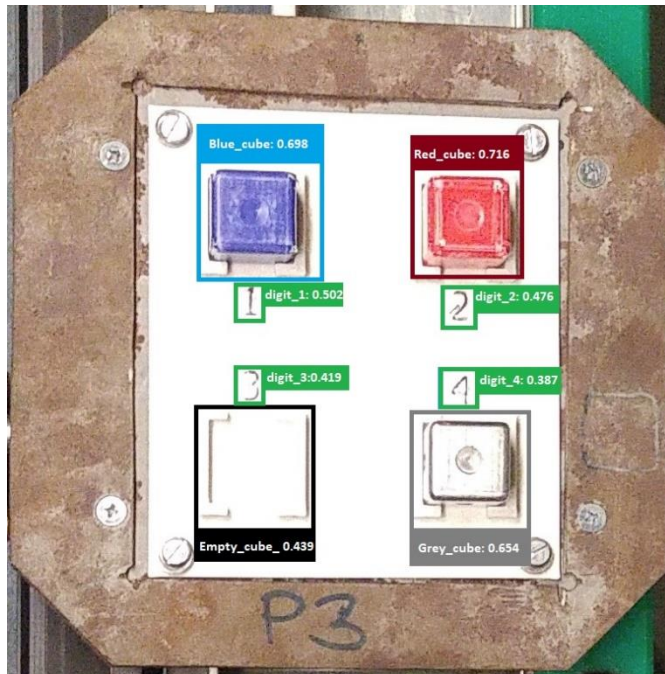
59. ábra. Robonet-RT detektor által felismert és osztályozott objektumok

Minden objektum a saját kategóriájának megfelelően került elhelyezésre a palettán (60. ábra).



60. ábra. Robonet-RT detektor által felismert és osztályozott objektumok (gyengébb képminőség)

A 61. ábrán a pozíciók felcserélésre kerültek az ABS+ palettán.



61. ábra. Robonet-RT detektor – objektum pozíció csere esetén

A Robonet-RT detektor rossz fényviszonyok között is képes volt az objektumok detektálására.

A detektor tanításra egy AMD Ryzen 1700 CPU Asztali-PC lett felhasználva, amely 64 DDR4 RAM -al, és Nvidia GPU RTX2070 rendelkezett.

5.8. Fejezet konklúziója

A Kiber-Fizikai és Intelligens Rendszerek Robotlaboratóriumban található Sony SCARA SRX-611 ipari robotegység szolgált alapjául a tanulmány ezen részének. A gépegység további használatára egy „convert to smart” módszer lett kidolgozva, amely mélytanulás technikát alkalmaz. A tanításhoz szükséges adatállományt saját forrásból állítottuk elő, 3D CAD modellek alapján, amelyet az ipari robotegységről és a felismerendő objektumról mintáztunk. A renderelésre bevonásra kerültek a labor asztali számítógépei, hogy az adatkészlethez szükséges képmennyiség generálható legyen. A YOLO architektúra módosítását követően, saját paraméterek mentén elvégeztük a tanítási folyamatot, amit ezt követően teszteltünk az adott gépegységen változó körülmények között.

A leírtaknak megfelelően, a régi gépegységek újbóli használatára lehetőség van az iparban emulációval, valamint funkciójuk kiegészítésével a mesterséges intelligencia által.

6. Irodalomjegyzék

- [1] Biblia Mózes I. könyve [Online] Available: <http://biblia.biblia.hu/read.php> [Accessed: 2023.12.28].
- [2] A. W. Colombo et al., "A 70-Year Industrial Electronics Society Evolution Through Industrial Revolutions: The Rise and Flourishing of Information and Communication Technologies," in IEEE Industrial Electronics Magazine, vol. 15, no. 1, pp. 115-126, March 2021, DOI: 10.1109/MIE.2020.3028058.
- [3] Mohajan, Haradhan. (2019). The First Industrial Revolution: Creation of a New Global Human Era. 5. 377-387.
- [4] Robert C. Allen; The Industrial Revolution: A Very Short Introduction (Very Short Introductions), ISBN-10 – 9780198706786; April 1, 2017.
- [5] Marsden, G. & Lyons, Glenn & Beecroft, Mark & Chatterjee, Kiron. (2002). Transport visions: vehicles and infrastructure.
- [6] Rolt, L.T.C. "Richard Trevithick". Encyclopedia Britannica, 18 Apr. 2023, [Online] Available: <https://www.britannica.com/biography/Richard-Trevithick>. [Accessed: 2023.12.28].
- [7] Iyer, Anandi. (2018). Moving from Industry 2.0 to Industry 4.0: A case study from India on leapfrogging in smart manufacturing. Procedia Manufacturing. 21. 663-670. 10.1016/j.promfg.2018.02.169.
- [8] Gelderman, Carol W. "Henry Ford". Encyclopedia Britannica, 24 Dec. 2023, [Online] Available: <https://www.britannica.com/biography/Henry-Ford>. [Accessed: December 28, 2023].
- [9] Britannica, The Editors of Encyclopaedia. "Ford Motor Company". Encyclopedia Britannica, 27 Dec. 2023, [Online] Available: <https://www.britannica.com/topic/Ford-Motor-Company>. [Accessed: December 28, 2023].
- [10] "August 1908: First ford model T completed - [Opinion]", in IEEE Spectrum, vol. 55, no. 8, pp. 19-19, Aug. 2018, doi: 10.1109/MSPEC.2018.8423578.
- [11] E. A. Torrero, "Automating the production line: Henry Ford began it all when he designed the first car assembly line in 1914," in IEEE Spectrum, vol. 14, no. 11, pp. 71-72, Nov. 1977, DOI: 10.1109/MSPEC.1977.6501657.
- [12] Zakoldaev, Danil & Shukalov, A & Zharinov, I. (2019). From Industry 3.0 to Industry 4.0: production modernization and creation of innovative digital companies. IOP Conference Series: Materials Science and Engineering. 560. 012206. 10.1088/1757-899X/560/1/012206.

- [13] Ross, Douglas T. (17 March 1961). Computer-Aided Design: A Statement of Objectives, MIT USAF 8436-TM-4.
- [14] Corney, Jonathan & Hayes, C. & Sundararajan, V. & Wright, P. (2005). The CAD/CAM interface: A 25-year retrospective. *Journal of Computing and Information Science in Engineering - JCISE*. 5. 10.1115/1.2033009.
- [15] Norman Sanders. A Possible First Use of CAM/CAD. Arthur Tatnall. *Reflections on the History of Computing: Preserving Memories and Sharing Stories*, AICT-387, Springer, pp.43-56, 2012, IFIP Advances in Information and Communication Technology (SURVEY), ff10.1007/978-3-642-33899-1_3ff. fffhal-01526813f
- [16] Ichbiah D. *Robots: From Science to Technological Revolution*. Harry-N Abrams Publishing unit; 2005. p. 116.
- [17] Programmed data processor-6 Handbook, PDP-6 system; Digital Equipment Corporation; Maynard – Massachusetts; 1964.
- [18] R. P. Harris; THE PDP-6 - The dec pdp-6 time-sharing computer; *Datamation magazine*, November 1964.
- [19] Moran, Michael. (2007). Evolution of robotic arms. *Journal of Robotic Surgery*. 1. 103-111. 10.1007/s11701-006-0002-x.
- [20] K. Zhou, Taigang Liu and Lifeng Zhou, "Industry 4.0: Towards future industrial opportunities and challenges," 2015 12th International Conference on Fuzzy Systems and Knowledge Discovery (FSKD), Zhangjiajie, China, 2015, pp. 2147-2152, doi: 10.1109/FSKD.2015.7382284
- [21] A. Ashabi, S. B. Sahibuddin and M. S. Haghighi, "Big Data: Current Challenges and Future Scope," 2020 IEEE 10th Symposium on Computer Applications & Industrial Electronics (ISCAIE), Malaysia, 2020, pp. 131-134, doi: 10.1109/ISCAIE47305.2020.9108826.
- [22] A. Bhardwaj, R. Singh, V. Deep and P. Sharma, "BDT3V — A Technique for Big Data Testing considering 3V's," 2018 Second International Conference on Green Computing and Internet of Things (ICGCIoT), Bangalore, India, 2018, pp. 222-225, doi: 10.1109/ICGCIoT.2018.8752996.
- [23] Mashey, J. R. (1999). Big data and the next wave of {InfraStress} problems, solutions, opportunities. [Online] Available: <https://www.usenix.org/conference/1999-usenix-annual-technical-conference/big-data-and-next-wave-infrastress-problems> [Accessed: December 29, 2023].
- [24] P. Suresh, J. V. Daniel, V. Parthasarathy and R. H. Aswathy, "A state of the art review on the Internet of Things (IoT) history, technology and fields of deployment," 2014 International Conference on Science

- Engineering and Management Research (ICSEMR), Chennai, India, 2014, pp. 1-8, DOI: 10.1109/ICSEMR.2014.7043637.
- [25] Granell, C., Kamilaris, A., Kotsev, A., Ostermann, F.O., Trilles, S. (2020). Internet of Things. In: Guo, H., Goodchild, M.F., Annoni, A. (eds) *Manual of Digital Earth*. Springer, Singapore. https://doi.org/10.1007/978-981-32-9915-3_11
- [26] I. Priyadarshini, B. Bhola, R. Kumar and C. So-In, "A Novel Cloud Architecture for Internet of Space Things (IoST)," in *IEEE Access*, vol. 10, pp. 15118-15134, 2022, DOI: 10.1109/ACCESS.2022.3144137.
- [27] J. F. C. B. Ramalho, L. D. Carlos, R. A. S. Ferreira and P. S. André, "Coloured QR codes for the Internet of Things," 2019 SBMO/IEEE MTT-S International Microwave and Optoelectronics Conference (IMOC), Aveiro, Portugal, 2019, pp. 1-3, DOI: 10.1109/IMOC43827.2019.9317554.
- [28] Kubo, "The Research of IoT Based on RFID Technology," 2014 7th International Conference on Intelligent Computation Technology and Automation, Changsha, China, 2014, pp. 832-835, doi: 10.1109/ICICTA.2014.199.
- [29] S. L. Andresen, "John McCarthy: father of AI," in *IEEE Intelligent Systems*, vol. 17, no. 5, pp. 84-85, Sept.-Oct. 2002, doi: 10.1109/MIS.2002.1039837.
- [30] Pavel Hamet, Johanne Tremblay, *Artificial intelligence in medicine, Metabolism*, Volume 69, Supplement, 2017, Pages S36-S40, ISSN 0026-0495, <https://doi.org/10.1016/j.metabol.2017.01.011>.
- [31] H. Wang, C. Ma and L. Zhou, "A Brief Review of Machine Learning and Its Application," 2009 International Conference on Information Engineering and Computer Science, Wuhan, China, 2009, pp. 1-4, DOI: 10.1109/ICIECS.2009.5362936.
- [32] Wiederhold, Gio & McCarthy, John. (1992). Arthur Samuel: Pioneer in Machine Learning. *IBM Journal of Research and Development*. 36. 329 - 331. 10.1147/rd.363.0329.
- [33] F. Q. Lauzon, "An introduction to deep learning," 2012 11th International Conference on Information Science, Signal Processing and their Applications (ISSPA), Montreal, QC, Canada, 2012, pp. 1438-1439, DOI: 10.1109/ISSPA.2012.6310529.
- [34] Tara h. Abraham; (Physio)logical circuits: the intellectual origins of the mcculloch – pitts neural networks; <https://doi.org/10.1002/jhbs.1094>; *Journal of the History of the Behavioral Sciences*, Vol. 38(1), 3 – 25 Winter 2002.
- [35] Ravi, Vinayakumar & Kp, Soman & Poornachandran, Prabakaran. (2017). Evaluation of Recurrent Neural Network and its Variants for Intrusion

- Detection System (IDS). International Journal of Information System Modeling and Design. 8. 43-63. 10.4018/IJISMD.2017070103.
- [36] Crisci, Carolina & Ghattas, Badih & Perera, Gonzalo. (2012). A review of supervised machine learning algorithms and their applications to ecological data. Ecological Modelling. 240. 113–122. 10.1016/j.ecolmodel.2012.03.001.
 - [37] M. Usama et al., "Unsupervised Machine Learning for Networking: Techniques, Applications and Research Challenges," in IEEE Access, vol. 7, pp. 65579-65615, 2019, DOI: 10.1109/ACCESS.2019.2916648.
 - [38] W. Qiang and Z. Zhongli, "Reinforcement learning model, algorithms and its application," 2011 International Conference on Mechatronic Science, Electric Engineering and Computer (MEC), Jilin, China, 2011, pp. 1143-1146, DOI: 10.1109/MEC.2011.6025669.
 - [39] M. Abu Alsheikh, D. T. Hoang, D. Niyato, H. -P. Tan and S. Lin, "Markov Decision Processes With Applications in Wireless Sensor Networks: A Survey," in IEEE Communications Surveys & Tutorials, vol. 17, no. 3, pp. 1239-1267, thirdquarter 2015, DOI: 10.1109/COMST.2015.2420686.
 - [40] AI, Advanced Analytics, Big Data und ihre Bedeutung für die BI [Online] Available: <https://qunis.de/data-science-ai-advanced-analytics-und-big-data-so-haengts-zusammen-und-das-bedeutet-es-fuer-die-bi/> [Accessed: December 29, 2023]
 - [41] World Economic Forum; AI - artificial intelligence - at Davos 2024: Here's what to know; Available online: <https://www.weforum.org/stories/2024/01/artificial-intelligence-ai-innovation-technology-davos-2024/> (accessed on 08.02.2025)
 - [42] KUKA Robotics, „Offical documentation of Industrial ARC Welder robot arm”, [Online] Available: https://www.eurobots.net/robot_kuka_kr5_arc-en.html, [Accessed:01.11.2022].
 - [43] Timotei István Erdei, G. Husi, „Hibajavítás és megelőzés 3D printer FDM gyártási folyamat alkalmazása során,” University of Debrecen - Thesis material. 20.12.2017.
 - [44] Mchenry, Kenton & Bajcsy, Peter. An overview of 3D data content, file formats and viewers, National Center for Supercomputing Applications University of Illinois at Urbana-Champaign, Urbana, IL October 31, 2008.
 - [45] L. Pitonakova, M. Giuliani, A. Pipe, Alan F T Winfield „Feature and Performance Comparison of the V-REP, Gazebo and ARGoS Robot Simulators”, Conference: Proceedings of the 19th Towards Autonomous Robotic Systems Conference (TAROS 2018)At: Bristol, United Kingdom.
 - [46] Rombach, Robin and Blattmann, Andreas and Lorenz, Dominik and Esser.; Shimizu, T., "High-Resolution Image Synthesis With Latent Diffusion Models," Proceedings of the IEEE/CVF Conference on

- Computer Vision and Pattern Recognition, June, 2022. <https://doi.org/10.48550/arXiv.2112.10752>.
- [47] Kumari, Nupur & Zhang, Bingliang & Zhang, Richard & Shechtman, Eli & Zhu, Jun-Yan. (2022). Multi-Concept Customization of Text-to-Image Diffusion. 10.48550/arXiv.2212.04488.
- [48] Hidalgo, Rafael & Salah, Nesreen & Jetty, Rajiv Chandra & Jetty, Anupama & Varde, Aparna; Personalizing Text-to-Image Diffusion Models by Fine-Tuning Classification for AI Applications; Conference: Intelligent Systems Conference (IntelliSys) 2023, Springer PublishersAt: Amsterdam, The Netherlands.
- [49] Munir, H.; Vogel, B.; Jacobsson, A. Artificial Intelligence and Machine Learning Approaches in Digital Education: A Systematic Revision. *Information* 2022, 13, 203. <https://doi.org/10.3390/info13040203>.
- [50] I. A. Astuti, I. H. Purwanto, T. Hidayat, D. A. Satria, Haryoko and R. Purnama, "Comparison of Time, Size and Quality of 3D Object Rendering Using Render Engine Eevee and Cycles in Blender," 2022 5th International Conference of Computer and Informatics Engineering (IC2IE), Jakarta, Indonesia, 2022, pp. 54-59, doi: 10.1109/IC2IE56416.2022.9970186.
- [51] Dehouche, Nassim & Dehouche, Kullathida. (2023). What's in a text-to-image prompt? The potential of stable diffusion in visual arts education. *Heliyon*. 9. e16757. 10.1016/j.heliyon.2023.e16757.
- [52] N. Ruiz, Y. Li, V. Jampani, Y. Pritch, M. Rubinstein, and K. Aberman, "DreamBooth: Fine Tuning Text-to-image Diffusion Models for Subject-Driven Generation," arXiv preprint arxiv:2208.12242, 2022.
- [53] M. Canesche, L. Bragança, O. P. V. Neto, J. A. Nacif and R. Ferreira, "Google Colab CAD4U: Hands-On Cloud Laboratories for Digital Design," 2021 IEEE International Symposium on Circuits and Systems (ISCAS), Daegu, Korea, 2021, pp. 1-5, doi: 10.1109/ISCAS51556.2021.9401151.
- [54] Choi, H.; Lee, J. Efficient Use of GPU Memory for Large-Scale Deep Learning Model Training. *Appl. Sci.* 2021, 11, 10377. <https://doi.org/10.3390/app112110377>.
- [55] Akter, S.; Prodhan, R.A.; Pias, T.S.; Eisenberg, D.; Fresneda Fernandez, J. M1M2: Deep-Learning-Based Real-Time Emotion Recognition from Neural Activity. *Sensors* 2022, 22, 8467. <https://doi.org/10.3390/s22218467>.
- [56] Dreambooth implanted in – Google Colab notebook cloud server. Available online: <https://colab.research.google.com/github/TheLastBen/fast-stable-diffusion/blob/main/fast-DreamBooth.ipynb> [Accessed on 18. December 2022].

- [57] HuggingFace "open-source code and technologies" <https://huggingface.co/docs/transformers/index>, [Accessed: 01 January 2023].
- [58] Esser, Patrick & Rombach, Robin & Blattmann, Andreas & Ommer, Björn. (2021). ImageBART: Bidirectional Context with Multinomial Diffusion for Autoregressive Image Synthesis. 10.48550/arXiv.2108.08827.
- [59] Jiao, L.; Huo, L.; Hu, C.; Tang, P. Refined UNet: UNet-Based Refinement Network for Cloud and Shadow Precise Segmentation. Remote Sens. 2020, 12, 2001. <https://doi.org/10.3390/rs12122001>.
- [60] Stöckl Andreas; Evaluating a Synthetic Image Dataset Generated with Stable Diffusion (2022), 10.48550/arXiv.2211.01777.
- [61] Patil Suraj and Cuenca, Pedro, Lambert, Nathan and von Platen, Patrick; Stable Diffusion (<https://huggingface.co/blog/rlhf>), [Accessed date: 05 May 2023].
- [62] AUTOMATIC1111/stable-diffusion-webui (<https://github.com/AUTOMATIC1111/stable-diffusion-webui>, [Accessed date: 05 January 2023].
- [63] Compare & find Differences in two Image Files; <https://www.textcompare.org/image/>, [Accessed date: 10 February 2024].
- [64] S. B. Niku, "Introduction to Robotics Analysis Systems, Applications," Upper Saddle River, NJ: Prentice Hall, 2001, pp 67-71, 92- 94.
- [65] V. Abhishek, A. A. Hayat, A. D. Udai, S. Saha, „ Identification of Dynamic Parameters of an Industrial Manipulator”, (IMSD 2014) and the 7th Asian Conference on Multibody Dynamics (ACMD 2014) At: Busan, Korea.
- [66] M. Lőrinc, F. Áron "Inverse geometry of robots ", Erdélyi Magyar Tudományegyetem, Scientia Kiadó 2019.
- [67] KUKA Roboter, „KUKA System Software V5.x”, KUKA Roboter GmbH Zugspitzstraße 140 D Augsburg Deutschland.
- [68] Erdei, T.I.; Molnár, Z.; Obinna, N.C.; Husi, G. A Novel Design of an Augmented Reality Based Navigation System & Its Industrial Applications. In Proceedings of the 15th IMEKO TC10—Technical Diagnostics in Cyber-Physical Era, Budapest, Hungary, 6–7 June 2017.
- [69] Nieto-Chaupis, H. Theory and Parameterization of Infections and Waves by Covid-19: A 6-Countries Data Analysis. In Proceedings of the 2021 IEEE 21st International Conference on Bioinformatics and Bioengineering (BIBE), Kragujevac, Serbia, 25–27 October 2021; pp. 1–5.
- [70] Gajdzik, B.; Grabowska, S.; Saniuk, S. A Theoretical Framework for Industry 4.0 and Its Implementation with Selected Practical Schedules. Energies 2021, 14, 940.
- [71] Zhang, C.; Wei, B.; Zhang, L. Research on Cyber-Physical Systems Based on Software Definition. In Proceedings of the 2021 IEEE 12th

- International Conference on Software Engineering and Service Science (ICSESS), Beijing, China, 20–22 August 2021; pp. 174–177.
- [72] Jadala, V.C.; Pasupuletti, S.K.; Raju, S.H.; Kavitha, S.; Bhaba, C.M.H.S.; Sreedhar, B. Need of Internet of Things, Industrial IoT, Industry 4.0 and Integration of Cloud for Industrial Revolution. In Proceedings of the 2021 Innovations in Power and Advanced Computing Technologies (i-PACT) Kuala Lumpur, Malaysia, Kuala Lumpur, Malaysia, 1–3 July 2021; pp. 1–5.
 - [73] Ahelerooff, S.; Xu, X.; Zhong, R.Y.; Lu, Y. Digital Twin as a Service (DTaaS) in Industry 4.0: An Architecture Reference Model. *Adv. Eng. Inform.* 2021, 47, 101225.
 - [74] Ahmadi Ahangar, T.; Rosin, A.; Nabavi, A.; Palu, I. A review on real-time simulation and analysis methods of microgrids. *Int. Trans. Electr. Energy Syst.* 2019, 29, e12106.
 - [75] Badilla-Quintana, M.G.; Sandoval-Henríquez, F.J. Students’ Immersive Experience in Initial Teacher Training in a Virtual World to Promote Sustainable Education: Interactivity, Presence, and Flow. *Sustainability* 2021, 13, 12780.
 - [76] Andhare, P.; Rawat, S. Pick and place industrial robot controller with computer vision. In Proceedings of the 2016 International Conference on Computing Communication Control and automation (ICCUBEA), Pune, India, 12–13 August 2016; pp. 1–4.
 - [77] Knoke, B.; Quandt, M.; Freitag, M.; Dieter Thoben, K. Virtual reality training applications in industry—Towards a user-friendly application design. In *Competence Development and Learning Assistance Systems for the Data-Driven Future*; GITO Verlag: Berlin, Germany, 2021.
 - [78] El Kharki, K.; Berrada, K.; Burgos, D. Design and Implementation of a Virtual Laboratory for Physics Subjects in Moroccan Universities. *Sustainability* 2021, 13, 3711.
 - [79] Moustakas, L.; Robrade, D. The Challenges and Realities of E-Learning during COVID-19: The Case of University Sport and Physical Education. *Challenges* 2022, 13, 9.
 - [80] Patil, G.V.; Deshpande, S.L. Distributed rendering system for 3D animations with Blender. In Proceedings of the 2016 IEEE International Conference on Advances in Electronics, Communication and Computer Technology (ICAECCT), Pune, India, 2–3 December 2016; pp. 91–98.
 - [81] Kase, D.; Hamamoto, T.; Hangai, S. An interactive quality evaluation of reduced polygon model. In Proceedings of the Proceedings 2000 International Conference on Image Processing (Cat. No. 00CH37101), Vancouver, BC, Canada, 10–13 September 2000; IEEE: Piscataway, NJ, USA, 2000; Volume 2, pp. 879–882.

- [82] P. P. Horváth, T. I. Erdei, G. Husi „Restoration and PLC control of a material handling MP-9S industrial manipulator platform” ANNALS OF THE ORADEA UNIVERSITY Fascicle of Management and Technological Engineering Volume XXIX (XIX), 2019/2(2).
- [83] Huo, Y.; Yang, A.; Jia, Q.; Chen, Y.; He, B.; Li, J. Efficient Visualization of Large-Scale Oblique Photogrammetry Models in Unreal Engine. ISPRS Int. J. Geo-Inf. 2021, 10, 643.
- [84] Lenovo ThinkCentre M75s. Available online: <https://www.lenovo.com/hu/hu/desktops/thinkcentre/m-series-towers/ThinkCentre-M93P/p/11TC1TMM93P> (accessed on 16 February 2022).
- [85] Diaz, S.M. Motion Matching UE4 Santiago Moreno. In Proceedings of the Motion Matching in Unreal Engine 4, Sheffield, UK, 17 May 2019.
- [86] Brecher, C.; Müller, S.; Kuz, S.; Lohse, W. Towards Anthropomorphic Movements for Industrial Robots. In Proceedings of the 4th International Conference on Digital Human Modeling and Applications in Health, Safety, Ergonomics, and Risk Management: Human Body Modeling and Ergonomics, Las Vegas, NV, USA, 21–26 July 2013; Volume 8026.
- [87] Sari, E.; Usman, U.; Hakim, A. Effectiveness of Using Google Forms in the Problem Based Learning Model to Increase the Critical Thinking Ability of High School Students. In Proceedings of the 2nd Educational Sciences International Conference (ESIC 2019), Samarinda, Indonesia, 15–16 September 2019.
- [88] Brooke, J. SUS: A Quick and Dirty Usability Scale. Usability Eval; Redhatch Consulting Ltd.: Earley, UK, 1995.
- [89] L. -C. Bazavan, H. Roibu, F. B. Petcu, S. I. Cismaru and B. N. George, "Virtual Reality and Augmented Reality in Education," 2021 30th Annual Conference of the European Association for Education in Electrical and Information Engineering (EAEIE), 2021, pp. 1-4, doi: 10.1109/EAEIE50507.2021.9531005.
- [90] A. Hussain, H. Shakeel, F. Hussain, N. Uddin, T. L. Ghouri "Unity Game Development Engine: A Technical Survey", University of Sindh Journal of Information and Communication Technology (USJICT), Volume 4, Issue 2, July 2020.
- [91] Wei Qian, Zeyang Xia, Jing Xiong, Yangzhou Gan, Yangchao Guo, Shaokui Weng, Hao Deng, Ying Hu, Jianwei Zhang, "Manipulation Task Simulation using ROS and Gazebo", 2014 IEEE International Conference on Robotics and Biomimetics, December 2014.
- [92] ROS OS - URDF, „Database”, [Online] Available: <http://wiki.ros.org/urdf>, [Accessed:01.11.2022].
- [93] Thomas P. Caudell, David Mizell, "Augmented reality: An application of heads-up display technology to manual manufacturing processes",

- System Sciences, 1992. Proceedings of the Twenty-Fifth Hawaii International Conference on Volume: II.
- [94] T.I.Erdei, Lean mérnök szakdolgozat; Ipar 4.0 robot labor fejlesztése LEAN módszertan integrálásával & gyártási rendszer folyamatok optimalizációja Mixed-Reality alapon, Debreceni Egyetem, Műszaki Kar, 2018.12.20
 - [95] LG G4 – H815, „Specification”, [Online] Available: <https://www.lg.com/uk/mobile-phones/lg-H815>, [Accessed:01.16.2022].
 - [96] AR Toolkit, „Computer Vision Coordinate System”, [Online] Available: <http://www.hitl.washington.edu/artoolkit/documentation/cs.htm> [Accessed:01.16.2022].
 - [97] P. Kumar, A. Mantri, S. Anireddy "Innovations in Tourism Industry & Development Using Augmented Reality (AR), Virtual Reality (VR)", Conference: Innovations in Tourism Industry & Development Using Augmented Reality (AR), Virtual Reality (VR).
 - [98] M. Haris, B. Jadoon, F. Hassan Khan "Evolution of Android Operating System: A Review", Conference: International Conference on Advanced Research At: Melbourne, Australia Volume: 2nd, November 2017.
 - [99] A. Strzelecki and P. Rutecka, "Direct Answers in Google Search Results," in IEEE Access, vol. 8, pp. 103642-103654, 2020, DOI: 10.1109/ACCESS.2020.2999160.
 - [100] Erdei, T.I.; Molnár, Z.; Obinna, N.C.; Husi, G. Cyber physical systems in mechatronic research centre; MATEC Web Conf. Volume 126, 2017 Annual Session of Scientific Papers IMT ORADEA 2017, <https://doi.org/10.1051/mateconf/201712601006>.
 - [101] Tikhonova, A.V. Modeling the Social Consequences of Industrial Robotization. In Proceedings of the 2nd International Scientific and Practical Conference on Digital Economy (ISCDE 2020), Yekaterinburg, Russia, 5–6 November 2020; Atlantis Press: Yekaterinburg, Russia, 2020.
 - [102] SCARA Robots: Robot Hall of Fame. Available online: <http://www.robothalloffame.org/inductees/06inductees/scara.html> (accessed on 18 February 2022).
 - [103] SONY SCARA SRX—11. High-Speed Assembly Robot, Operation Manual; SONY Corporation: Tokyo, Japan, 1996.
 - [104] PARO QE 01 31-6000. Manual of the Modular Conveyor; PARO AG: Subingen, Switzerland, 2016.
 - [105] DataSheet: USB Bus Convert Chip CH341. Available online: <http://www.anok.ceti.pl/download/ch341ds1.pdf> (accessed on 27 February 2022).

- [106] Feng, X. Towards Real-Time Enabled Microsoft Windows. In Proceedings of the 5th ACM International Conference on Embedded Software, Jersey City, NJ, USA, 18–22 September 2005; pp. 142–146.
- [107] VMware. Available online: <https://www.vmware.com/products/workstation-player.html> (accessed on 20 April 2023).
- [108] Lenovo: ThinkCentre M93p Mini Tower. Available online: <https://www.lenovo.com/hu/hu/desktops/thinkcentre/m-series-towers/ThinkCentre-M93P/p/11TC1TMM93P> (accessed on 02 January 2023).
- [109] Xu, H.; Badawi, R.; Fan, X.; Ren, J.; Zhang, Z. Research for 3D visualization of Digital City based on SketchUp and ArcGIS. In Proceedings of the SPIE—the International Society for Optical Engineering, Wuhan, China, 14 October 2009.
- [110] Lenovo:ThinkCentre M92p Tower. Available online: <https://www.lenovo.com/gb/en/desktops/thinkcentre/m-series-tiny/m92p/> (accessed on 20 April 2022).
- [111] Labellmg. Available online: <https://github.com/tzutalin/labellmg> (accessed on 6 May 2024).
- [112] Casuat, C.D.; Merencilla, N.E.; Reyes, R.C.; Sevilla, R.V.; Pascion, C.G. Deep-Hart: An Inference Deep Learning Approach of Hard Hat Detection for Work Safety and Surveillanc. In Proceedings of the 2020 IEEE 7th International Conference on Engineering Technologies and Applied Sciences (ICETAS), Kuala Lumpur, Malaysia, 18–20 December 2020.
- [113] Juan Terven, Diana Cordova-Esparza; A Comprehensive Review of YOLO Architectures in Computer Vision: From YOLOv1 to YOLOv8 and YOLO-NAS, arXiv:2304.00501.
- [114] Peiyuan Jiang, Daji Ergu, Fangyao Liu, Ying Cai, Bo Ma, A Review of Yolo Algorithm Developments, Procedia Computer Science, Volume 199, 2022, Pages 1066-1073, ISSN 1877-0509, <https://doi.org/10.1016/j.procs.2022.01.135>.
- [115] Liu, W.; Anguelov, D.; Erhan, D.; Szegedy, C.; Reed, S.; Fu, C.Y.; Berg, A.C. SSD: Single Shot MultiBox Detector. In Proceedings of the 2016 European Conference on Computer Vision (ECCV), Amsterdam, The Netherlands, 8–16 October 2016; p. 21.
- [116] Fu, C.Y.; Liu, W.; Ranga, A.; Tyagi, A.; Berg, A.C. DSSD: Deconvolutional Single Shot Detector. 2017. Available online: <https://arxiv.org/abs/1701.06659> (accessed on 12 May 2022).
- [117] Sandler, M.; Howard, A.; Zhu, M.; Zhmoginov, A.; Chen, L.-C. MobileNetV2: Inverted Residuals and Linear Bottlenecks. In Proceedings of the 2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition, Salt Lake City, UT, USA, 18–23 June 2018; pp. 4510–4520.

- [118] Dai, J.; Li, Y.; He, K.; Sun, J. R-FCN: Object Detection via Region-based Fully Convolutional Networks. In *Advances in Neural Information Processing Systems*; Lee, D., Sugiyama, M., Luxburg, U., Guyon, I., Garnett, R., Eds.; Curran Associates, Inc.: Red Hook, NY, USA, 2016; Volume 29.
- [119] He, K.; Gkioxari, G.; Dollár, P.; Girshick, R. Mask R-CNN. In *Proceedings of the 2017 IEEE International Conference on Computer Vision (ICCV)*, Venice, Italy, 22–29 October 2017; pp. 2980–2988.
- [120] Adarsh, P.; Rathi, P.; Kumar, M. YOLO v3-Tiny: Object Detection and Recognition using one stage improved model. In *Proceedings of the 6th Inter-national Conference on Advanced Computing and Communication Systems (ICACCS)*, Coimbatore, India, 6–7 March 2020.
- [121] Bochkovskiy, A.; Wang, C.Y.; Liao, H.Y.M. YOLOv4: Optimal Speed and Accuracy of Object Detection. *arXiv* 2020, arXiv:2004.10934.
- [122] Khalid, M.; Baber, J.; Kasi, M.K.; Bakhtyar, M.; Devi, V.; Sheikh, N. Empirical Evaluation of Activation Functions in Deep Convolution Neural Network for Facial Expression Recognition. In *Proceedings of the 43rd International Conference on Telecommunications and Signal Processing (TSP)*, Milan, Italy, 7–9 July 2020; pp. 204–207.
- [123] Perez, L.; Wang, J. The Effectiveness of Data Augmentation in Image Classification using Deep Learning. *arXiv* 2017, arXiv:1712.04621.
- [124] Kingma, D.P.; Ba, J. A Method for Stochastic Optimization. *arXiv* 2015, arXiv:1412.6980.
- [125] Li, Z.; Arora, S. An Exponential Learning Rate Schedule for Deep Learning. *arXiv* 2019, arXiv:1910.07454.
- [126] Redmon, J.; Divvala, S.; Girshick, R.; Farhadi, A. You Only Look Once: Unified, Real-Time Object Detection. In *Proceedings of the 2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, Las Vegas, NV, USA, 27–30 June 2016; pp. 779–788.
- [127] Redmon, J.; Farhadi, A. YOLO9000: Better, Faster, Stronger. In *Proceedings of the 2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, Honolulu, HI, USA, 21–26 July 2017; pp. 6517–6525.