Csatlakozóházak objektumdetektálása, annak alkalmazhatósága az iparban

Object detection of connectors and its application in industry

BÖCZ Gábor, WENESZ Dominik

ACSG Elektronikai és Kereskedelmi Kft., 9024 Győr, Práter u. 5. 00 36 70 628 9355, info@acsg.hu, acsg.hu

Abstract

Object detection and positioning are of particular importance in industrial environments (e.g., robotization, quality assurance). Deep learning has achieved outstanding results in this field in recent years, and we investigated the applicability of these methods for small and medium-sized companies through the detection and pose estimation of connector housing. Emphasis was placed on robustness, liquidity and investment costs for small-scale production.

Keywords: deep learning, computer vision, automation, connectors, object detection

Kivonat

Az objektumdetektálás és pozíciómeghatározás az ipari környezetben kiemelt fontossággal bír (pl.: robotizálás, minőségbiztosítás). A deep learning (mélytanulás) ilyen téren az elmúlt években kiemelkedő eredményeket ért el, ezen módszerek kis- és középvállalkozások számára való alkalmazhatóságát vizsgáltuk kutatásunkban csatlakozóházak detektálásán és pozíciómeghatározásán keresztül. Kiemelt fontosságot tulajdonítva a robosztusságnak, likviditásnak és beruházási költségeknek kisszériás gyártások esetén.

Kulcsszavak: mélytanulás, gépi látás, automatizálás, csatlakozóházak, objektumdetektálás

1. Bevezetés

Az iparban már a 20. század második felétől kezdve meghatározó szerepet tölt be a robotizálás, jellemzően a szerelési, objektum mozgatási folyamatokban jelentős előnyökkel bír az emberi munkaerővel szemben, példaként említve a termelékenységet, ismétlőképességet, precizitást, terhelhetőséget. [1] Azonban a humán munkaerővel járó adaptív problémamegoldókészség, környezethez való alkalmazkodás az esetek többségében nem megtalálható a robotok ipari alkalmazásában. Az Ipar 4.0 hozadéka többek közt a gyártók felé való gyors, likvid igazodás a vevői igényekhez, azaz ha úgy tetszik az egyedi gyártás tömeggyártásban való kivitelezése, végrehajtása. Ebben a mesterséges intelligenciával támogatott kollaboratív robotokkal ellátott gyártócellák nagy segítséget nyújtanak, sőt egyes alkalmazásokban elengedhetetlenek. Az ilyen cellák több szenzorral is kell rendelkezzenek, ezek közül jelen esettanulmányban az RGB kamerák szerepelnek mindösszesen. Az ilyen kamerák által kapott jel igen hasonló az emberi szem által az agyban leképzettel, így a humán problémamegoldás látáson alapuló szegmenseit képes helyettesíteni egy ezen bemeneten alapuló rendszer, azaz tipikusan a minőségbiztosítási szemrevételezést, pozícionálást, szerelést. [1] [4] Kutatásunk során a mélytanuláson alapuló objektumdetektálási módszereket vizsgáltuk alkalmazhatósági szempontból csatlakozóházak kollaboratív robottal való pozícionálása, valamint minőségbiztosítása esetén.

2. Szintetikus adatgenerálás

2.1. Mélytanulás adatigénye

A mélytanulás (deep learning) a mesterséges intelligencián belül a gépi tanulás egy tudományterülete, az alkalmazott modelleket jellemzően neurális hálózatok építik fel, melyek úgynevezett rétegekből állnak. [1] [2] Egy-egy ilyen háló a bemenetére hattatott jelnek megfelelően nemlineáris leképzéseket követően valamely kimeneti jelet szolgáltat. [1] Fontos kiemelni, hogy a nemlinearitást nem algoritmikusan határozzuk meg

predefiniáltan, hanem a tanítási folyamat során az optimalizációs algoritmus(ok) és a veszteségi függvény(ek) következtében alakul ki. [1]

Többféle hálóarchitektúrát különböztethetünk meg, képfeldolgozásban jellemzően valamilyen konvolúciós vagy transzformer hálóról beszélhetünk. Közös jellemzője minden egyes deep learningen alapuló modellnek azonban, hogy a többi mesterséges intelligencián alapuló algoritmushoz, modellhez képest nagyságrendekkel nagyobb a tanítóadatigényük, mely egy kamerakép alapján működő háló esetén a több ezrestől a milliós nagyságrendig terjedhet (db. kép), ez belátható ipari alkalmazásban gátat szabhat, hiszen idő- és tőkeigénye igen magas lehet, továbbá az adatcímkézés szintén hosszadalmas és humán hibával terhelt, ha valódi kameraképekkel szeretnénk dolgozni.

2.2. Adatgenerálás

A fentebb említett probléma egy áthidalása az adatdúsítás, mely a valós fizikai képek transzformációjával, konvolúciós műveletekkel érhető el, ennek hátránya, hogy alkalmazásspecifikusan nem minden esetben alkalmazható, illetőleg egyes esetekben nem vezet a modell teljesítményének javulásához.

Egy másik megoldás az RGB képek szintetikus generálása, azaz valamely render algoritmus segítségével tetszőleges képet generálhatunk. Ezen technika nagy előnye, hogy tetszőlegesen módosíthatjuk a kép minden részletét, azaz példaként a megvilágítást, felbontást, zajterheltséget stb. Ennek köszönhetően létre tudunk hozni olyan adatbázist, melyben a valóságot nagymértékben le tudjuk fedni, közelítjük, vagy adott esetben olyan bő szintetikus halmazt hozunk létre, mely biztosan tartalmazza a valóságot is. Alkalmazásfüggő, hogy a fizikai világ minél precízebb vagy bővebb modellezése az inkább célravezető.

Jelen cikk keretein belül egy PBR (physics based rendering) szoftvert, a Blender-t használtuk a mesterségesen generált képek előállításához, melyet Python nyelven írt scriptekkel egészítettünk ki, így elegendő csupán egy 3D CAD modell a csatlakozóházról, hogy tetszőleges mennyiségű és minőségű képeket kapjunk. Fontos megemlíteni, hogy a script az adatcímkézést is elvégzi, továbbá a hagyományos képfeldolgozásban szokásos augmentációs technikák segítségével egy-egy renderelés után adatdúsítást is tud végezni.

A használt hardver adatai: processzor:11th Gen Intel core i7-11700x16, videókártya: NVIDIA GeForce3060 Ti.

Egy-egy 1920x1080 pixel felbontású képet posztprocesszálástól és objektumoktól függően jellemzően 3-8 másodperc alatt tudtunk generálni, mely egy tapasztalt humán munkaerőnek átlagosan 1-3 percet venne igénybe.

1. ábra. *Szintetikus kép (bal) és maszk (jobb)*

3. Csatlakozóházak és a gépi látás

3.1. Csatlakozóházak, hagyományos képfeldolgozás

A csatlakozóházak jellemzően kis objektumok a munkatérben, melyek gyakran egymáson is elhelyezkedhetnek, azaz orientációjuk előre nem megjósolható, továbbá az egyes szériákhoz tartozó csatlakozóházak csupán minimális eltérésekkel rendelkeznek egymáshoz képest, így belátható, hogy a hagyományos képfeldolgozási algoritmusok segítségével (pl.: template matching, feature matching) nehezen megoldható robosztus, könnyen adaptálható, módosítható gépi látáson alapuló rendszer létrehozása.

Ennélfogva a modernebbnek mondható mély neurális hálók használata indokolt ezen problémakörben, hiszen azokkal generalizálható a megoldás.

Megjegyzendő, hogy értékelésünkben a fókusz a minél gyorsabb termékváltásra és precizitásra esik, hiszen a kisszériás és egyéni gyártásban ezek bírnak kiemelkedő fontossággal.

3.2. YOLO

A You Only Look Once (YOLO) egy korszerű, valós idejű objektumészlelési algoritmus, amelyet 2015-ben Joseph Redmon, Santosh Divvala, Ross Girshick és Ali Farhadi vezettek be a „You Only Look Once: Egységes, valós idejű objektumészlelés”. [2] Az objektum-észlelési problémát regressziós problémaként kezeli osztályozási feladat helyett azáltal, hogy térben elválasztja a határolókereteket (boundary box), és egyetlen konvolúciós neurális hálózat (CNN) segítségével minden észlelt képhez valószínűségeket rendel. [1] [5] Előnye, hogy egylépésben megtörténik az objektumdetektálás, így az egyik leggyorsabb az objektumdetektáló algoritmusok közül. További előnye, hogy az algoritmusból adódóan robosztus.

Az elmúlt években a YOLO algoritmus alapján több neurális háló is született, melyek mindmáig a legkorszerűbb detektáló hálóarchitektúrák. [2] [3] Ennélfogva a legutóbbi (a v8-at nem számítva) három verziót használtuk a csatlakozóházak ipari környezetben történő detektálására, ezek közül is a legkiseb paraméterszámmal rendelkező modelleket, hiszen a gyártásban lévő céleszközön limitált a számítási kapacitás. Gyorsaság szempontjából, azaz FPS-t (frame per second) nem vizsgáltuk külön a hálókat, hiszen mindegyik a célnak megfelelő valós időben képes detektálni.

A YOLO előnye, hogy az objektum méretétől függetlenül egylépésben képes detektálni objektumokat, hiszen az eredeti képet több különböző méretű felosztásban konvolúciós rétegeken futtatja át, így precíz és valós idejű működést tesz lehetővé. [3] [4]

A validációnál a mAP (mean avarage precision) értéket alkalmaztuk, mely standard az objektumdetektáló hálók jóságának jellemzésére. [1]

3.3. Objektumdetektálás (bounding box)

Az objektumdetektálás alatt azon folyamatot értjük, melyben egy digitális képen minden egyes általunk detektálni kívánt objektumhoz rendelünk egy befoglaló téglalapot, melynek oldalai párhuzamosak a kép széleivel (tengelyeivel), illetőleg az objektumhoz tartozó osztályt is meg tudjuk mondani (pl.: kutya, macska stb.). [1] [2] [4] [7]

A csatlakozóházak ipari környezetben való detektálását tekintve az elhelyezkedésről információt kapunk, viszont az orientációról nem, így számolási, minőségellenőrzési feladatoknál megfelelő csupán a bounding box-ra hagyatkozni, azonban manipulátorral való mozgatás esetén önmagában ez kevés információt tartalmaz.

Az alábbi táblázatban láthatók az általunk a problémakörben tesztelt hálóarchitektúrák, fontos megjegyezni, hogy csak azon modellek kerültek ezek közé, melyek valós időben képesek működni.

Különböző osztályszámossághoz, osztályonkénti tanítóadatmennyiséghez, vizsgáltuk valós képeken a modellekhez tartozó mAP az osztályok számossága alapján az alábbi módon: [5 | 10 | 20 | 50].

Az ide tartozó táblázatok a tanító adat minőségét tekintve az alábbiak: 1.: adatdúsítás nélkül, egyszínű háttér, véletlen kameraállás, minimálisan változó megvilágítás, egyszínű objektumok, 2.: véletlen háttér (COCO adatbázisból), sok különböző kameraállás és megvilágítás, egyszínű textúra a csatlakozóházakon, 3.: háttér, kamerapózok, megvilágítás a 2.-kal egyező, véletlen textúra a csatlakozóházakon, zajjal terhelés, posztprocesszálási adatdúsító eljárásokkal.

Bounding box detection – 1 — 1. táblázat

Img/class \ Modell	Yv5t-640				Yv6n-640				Yv7t-640				Yv5t-1280				Yv6n-1280				Yv7t-1280			
100	22	17	12	9	23	16	13	10	26	18	14	11	23	19	17	12	24	19	18	14	26	21	20	18
500	25	19	17	13	26	20	17	11	29	24	19	17	30	25	21	19	30	26	22	20	33	27	22	21
1000	28	25	20	19	28	26	21	19	29	27	24	22	31	28	25	22	32	28	25	22	36	28	25	24
2000	32	29	25	23	30	29	25	21	31	30	29	24	33	31	27	23	34	31	27	25	36	30	29	27
5000	32	31	28	24	32	32	27	23	33	32	28	24	34	33	29	25	35	32	29	27	37	31	29	28

Bounding box detection – 2 2. táblázat

Img/class \ Modell	Yv5t-640				Yv6n-640				Yv7t-640				Yv5t-1280				Yv6n-1280				Yv7t-1280			
100	44	39	34	31	45	38	35	32	48	40	36	33	45	41	39	34	46	41	40	36	48	43	42	40
500	47	41	39	35	48	42	39	33	51	46	41	39	52	47	43	41	52	48	44	42	55	49	44	43
1000	50	47	42	41	50	48	43	41	51	49	46	44	53	50	47	44	54	50	47	44	58	50	47	46
2000	54	51	47	45	52	51	47	43	53	52	51	46	55	53	49	45	56	53	49	47	58	52	51	49
5000	54	53	50	46	54	54	49	45	55	54	50	46	56	55	51	47	57	54	51	49	59	53	51	50

Bounding box detection – 3 3. táblázat

Img/class \ Modell	Yv5t-640				Yv6n-640				Yv7t-640				Yv5t-1280				Yv6n-1280				Yv7t-1280			
100	60	55	50	47	57	50	47	44	60	52	48	45	66	62	60	55	63	58	57	53	77	72	71	69
500	63	57	55	51	60	54	51	45	63	58	53	51	73	68	64	62	69	65	61	59	84	78	73	72
1000	66	63	58	57	62	60	55	53	63	61	58	56	74	71	68	65	71	67	64	61	87	79	76	75
2000	70	67	63	61	64	63	59	55	65	64	63	58	76	74	70	66	73	70	66	64	87	81	80	78
5000	70	69	66	62	66	66	61	57	67	66	62	58	77	76	72	68	74	71	68	66	88	82	80	79

Egyértelműen látható, hogy a kis osztályonkénti szintetikus tanítóadat nem elegendő, hogy valós képeken is jól performáljon a modell, továbbá az is megfigyelhető, hogy 1000-2000-es nagyságrend a legtöbb modellnél már elfogadható eredményt ad. Egyértelműen látható a YOLO családban verziónkénti fejlődés, kiemelten a v7 esetén, melynél szignifikánsan kevesebb training adat is elegendő ugyanazon mAP pontszámhoz.

A generál adatokat tekintve a mennyiségen túl az augmentáció mennyisége és minősége is kimutatható. A csupán valamely egyszínű háttér esetén minimálisan változtatott megvilágítási viszonyok mellett és egyéb adatdúsítási technikákat mellőzve a tesztelés során nagyon gyengén teljesítő modellt kapunk. Ellenben minél több augmentációs technikát használunk, különböző zajjal terheljük a képet, nagy mértékben jobban teljesítő modellekhez jutunk, ez betudható annak, hogy minél bővebbre sikerül a szintetikus adathalmazt növelni (tulajdonságokat tekintve), annál nagyobb részét képes lefedni a valóságnak, robosztusabb, generikusabb lesz a neurális háló.

A bemeneti képfelbontást tekintve megállapítható, hogy a nagyobb felbontás pontosabb modellt eredményez, persze természetesen ennélfogva lassabb is lesz.

3.4. Objektum szegmentálás (segmentation)

Az objektum szegmentálás [1] esetében (instance segmentation) a digitális képen szintén minden egyes általunk detektálni kívánt objektumhoz az annak megfelelő osztályt rendeljük, viszont ellentétben az objektumdetektálással, ebben az esetben nem befoglaló téglalapot rendelünk egy-egy objektumhoz, hanem pixeleket, azaz maszkoljuk a képet. Sokkal precízebb helymeghatározás érhető el így, viszont általánosságban komplexebb modell szükséges hozzá, mely nagyobb számítási igényt is jelent, továbbá tanítási erőforrás igényét tekintve (tanítási idő, memóriaigény, adatigény) is meghaladja az egyszerűbb bounding box objektumdetektálást.

Az alábbi táblázatokban a különböző mennyiségű és minőségű szintetikus adatok, osztályszámosságok és modellek függvényében a mAP értékeket.

A 3.3-as ponthoz hasonlóan az eredmények az alábbi táblázatokban láthatók:

Instance segmentation – 1 4. táblázat

Img/class \ Modell	Yv5t-640				Yv6n-640				Yv7t-640				Yv5t-1280				Yv6n-1280				Yv7t-1280			
100	17	12	7	4	18	11	8	5	21	13	9	6	18	14	12	7	19	14	13	9	21	16	15	13
500	20	14	12	8	21	15	12	6	24	19	14	12	25	20	16	14	25	21	17	15	28	22	17	16
1000	23	20	15	14	23	21	16	14	24	22	19	17	26	23	20	17	27	23	20	17	31	23	20	19
2000	27	24	20	18	25	24	20	16	26	25	24	19	28	26	22	18	29	26	22	20	31	25	24	22
5000	27	26	23	19	27	27	22	18	28	27	23	19	29	28	24	20	30	27	24	22	32	26	24	23

Instance segmentation – 2 5. táblázat

Img/class \ Modell	Yv5t-640				Yv6n-640				Yv7t-640				Yv5t-1280				Yv6n-1280				Yv7t-1280			
100	32	27	22	19	31	24	21	18	37	29	25	22	45	41	39	34	46	41	40	36	47	42	41	39

500	35	29	27	23	34	28	25	19	40	35	30	28	52	47	43	41	52	48	44	42	54	48	43	42
1000	38	35	30	29	36	34	29	27	40	38	35	33	53	50	47	44	54	50	47	44	57	49	46	45
2000	42	39	35	33	38	37	33	29	42	41	40	35	55	53	49	45	56	53	49	47	57	51	50	48
5000	42	41	38	34	40	40	35	31	44	43	39	35	56	55	51	47	57	54	51	49	58	52	50	49

Instance segmentation – 3 6. táblázat

Img/class \ Modell	Yv5t-640				Yv6n-640				Yv7t-640				Yv5t-1280				Yv6n-1280				Yv7t-1280			
100	50	45	40	37	51	44	41	38	57	49	45	42	60	56	54	49	59	54	53	49	70	65	64	62
500	53	47	45	41	54	48	45	39	60	55	50	48	67	62	58	56	65	61	57	55	77	71	66	65
1000	56	53	48	47	56	54	49	47	60	58	55	53	68	65	62	59	67	63	60	57	78	72	69	68
2000	60	57	53	51	58	57	53	49	62	61	60	55	70	68	64	60	69	66	62	60	79	74	73	71
5000	60	59	56	52	60	60	55	51	64	63	59	55	71	70	66	62	70	67	64	62	81	75	73	72

Általánosságban megállapítható, hogy a mAP értékek a bounding box detektálással szemben kisebb, hiszen a feladat is komplexebb, mivel minden egyes pixelhez osztályt rendelnek a modellek. A 3.3-as pontban tett megállapítások egy az egyben igazak itt is a tanító adatokra vonatkozólag, illetve a YOLO verziófejlődésére is.

3.5. Kulcspont detektálás (keypoint detection)

A keypoint detection [1] egy olyan objektumdetektáló megközelítés, amikor is a képen való elhelyezkedésen és osztályon kívül az objektum egyes pontjai pontos helyét is meg szeretnénk állapítani. Főként a videó alapján történő humán pózmeghatározásban használják.

Esetünkben csatlakozóház helyének és helyzetének meghatározását teszi lehetővé igen precíz szinten, hiszen az egyes kitüntetett pontok alapján a további folyamatok irányítása leegyszerűsödik.

A keypoint detection először a YOLOv7 [6] modell architektúrában lett publikálva a YOLO családból, így ennek tesztelése csak a YOLOv7-tiny-keypoint modellen történt nagymértékben variált és dúsított tanítóadatokkal (3.3 és 3.4 esetén a 3. táblázathoz hasonlóan), változó osztály-, és osztályonkénti keypoint számossággal, melyek az alábbi módon kerülnek ábrázolásra egy cellán belül: [2 keypoint/osztály | 5 keypoint/osztály | 10 keypoint/osztály].

YOLOv7-tiny keypoint detection 7. táblázat

Tanító adat \ Osztályszám	5			10			20			50		
100	57	45	34	52	42	28	46	36	21	36	27	19
500	67	60	52	60	48	35	54	46	31	45	34	27
1000	73	67	60	68	62	54	65	57	48	64	56	45
2000	82	77	69	78	72	67	76	70	66	71	65	59
5000	87	85	84	83	81	79	80	76	71	75	72	67

Konklúzióként hasonló megállapításokat tehetünk, mint a 3.3 és 3.4-es pontban, kiegészítve azzal, hogy a keypointok számának növelésével szignifikánsan megnő a szükséges tanítóadat mennyisége.

3.6. SCARA robot esetén általunk alkalmazott architektúra

A fentebb említett eredményeknél és technikai megfontolások után az általunk alkalmazott architektúra a YOLOV7-t-1280 modell osztály és boundary box kimenettel, kiegészítve három további kimenettel (head), melyek a csatlakozóház térbeli elfordulását állapítják meg. Ezen adatok alapján egyértelműen meghatározható, hogy az adott objektum a kívánt osztály(ok)ba tartozik-e, illetve az aktuális megfogóval felvehető-e vagy sem.

4. Összefoglalás

Összességében elmondható, hogy a deep learningen alapuló YOLO objektumdetektáló modell család a csatlakozóházak esetén az iparban kisszériás gyártás esetében is alkalmazható, nem túl nagy osztályszámossággal, a szintetikus adatok generálásával együtt 0,5-3 óra alatt, mely megfelelő logisztikával frekventált termékpaletta váltást tesz lehetővé tetszőleges gyártócellában.

Az eredmények alapján két opciót javaslunk az ipari alkalmazásra, amennyiben elegendő csupán az objektum osztályát, helyét (középpont és bounding box), térbeli elfordulását és esetlegesen felvehetőségét tudni, akkor bármely újabb YOLO architektúra kimentét bővítve a feladatra alkalmas detektort kapunk. Amennyiben pontosabb meghatározás szükséges, esetlegesen kitüntetett pontok pontos helye is kritikus, akkor a YOLOv7 (vagy a jövőben újabb) architektúrát ajánljuk, ebben az esetben viszont több tanító adatra van szükség. A szegmentálás számításigénye meghaladja az előbb említett két megoldást és a robotokkal történő objektummozgatáshoz nem tud plusz információt szolgáltatni, mely érdemben elősegíteni a manipuláció irányítását, ennélfogva alkalmazását nem gondoljuk előnyösnek ezen problémakörben.

Beruházási költségeket tekintve is igen kedvező, hiszen a jellemzően szükséges hardver összköltsége nem haladja meg a 2500-5000 euro, melyben a tanításhoz és adatgeneráláshoz szükséges eszközök is szerepelnek, ezeken felül egy-egy további egység költsége 100-400 euro közé tehető. Ennélfogva az ipar 4.0 és IoT irányvonalba könnyedén beilleszthető, gyártócellák hatékony fejlesztése lehetséges ilyen módon.

Köszönetnyilvánítás

Szeretnénk köszönetünket kifejezni az ACSG Kft.-nek a hardveres háttér biztosításáért.

Irodalmi hivatkozások

[1] Z.-Q. Zhao, P. Zheng, S.-t. Xu, X. Wu *Object detection with deep learning: a review* IEEE Trans. Neural Netw. Learn. Syst. (2019)

[2] J. Redmon, S. Divvala, R. Girshick, A. Farhadi *You only look once: unified, real-time object detection* 2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR) (2016), pp. 779-788, 10.1109/CVPR.2016.91 (Utolsó letöltés: 2023.01.27.)

[3] J. Redmon, A. Farhadi *YOLO9000: better, faster, stronger* https://arxiv.org/abs/1612.08242v1 (Utolsó letöltés: 2023.01.12.)

[4] J. Redmon, A. Farhadi *YOLOv3: an incremental improvement* https://arxiv.org/abs/1312.2249 (2018) (Utolsó letöltés: 2023.01.11.)

[5] A. Bochkovskiy, C.-Y. Wang, H.-Y.M. Liao *YOLOv4: optimal speed and accuracy of object detection* https://arxiv.org/abs/2004.10934 (2020) (Utolsó letöltés: 2023.01.18.)

[6] C.-Y. Wang, A. Bochkovskiy, H.-Y. M. Liao *YOLOv7: Trainable bag-of-freebies sets new state-of-the-art for real-time object detectors* (2022) https://arxiv.org/abs/2207.02696 (Utolsó letöltés: 2023.01.12.)

[7] D. Thuan Evolution of YOLO algorithm and YOLOv5: the state-of-the-art object detection algorithm (2021)