

Úthiba-felismerés okostelefonnal, konvolúciós neurális hálózatok segítségével

HORVÁTH Viktor Győző¹, BARSÍ Árpád¹

¹Budapesti Műszaki és Gazdaságtudományi Egyetem, Építőmérnöki Kar, Fotogrammetria és Térinformatika Tanszék

E-mail: horvath.viktor.gyozo@emk.bme.hu

DOI: [10.30921/GK.77.2025.4.3](https://doi.org/10.30921/GK.77.2025.4.3)

Absztrakt

A mély neurális hálózatok alkalmazása napjainkban jelentős fejlődést mutat, számos tudományos és gyakorlati problémakörben kínál hatékony megoldási lehetőséget, a mérnöki jellegű feladatoktól egészen a társadalmi jelenségek elemzéséig. Különösen kiemelt szerepet kapnak a képalapú adatok feldolgozásában, amelyek rögzítésére a modern okostelefonok költséghatékony és széles körben hozzáférhető eszközként szolgálnak. Az úthálózatok állapotának megfigyelése és monitorozása olyan terület, amely számottevően profitálhat az említett technológiák integrációjából, hiszen a releváns vizuális adatok gyűjtése és feldolgozása révén részletes és megbízható információk nyerhetők az aktuális állapotokról. A kutatás keretében olyan kísérleti technológiai leírás készült, amely bemutatja a terepen történő adatgyűjtés folyamatát okostelefon alkalmazásával, majd az így létrejövő adathalmaz feldolgozását mély neurális hálózat segítségével. A feldolgozás eredményei megfelelő vizualizációs módszerekkel kerülnek bemutatásra, továbbá kísérlet történik azok érvényességének vizsgálatára és validálására.

Abstract

The application of deep neural networks has undergone significant advancement in recent years, offering effective solutions across a wide spectrum of domains, ranging from engineering tasks to the analysis of social phenomena. Image-based inputs hold a particularly prominent role in these applications, with modern smartphones serving as cost-effective and widely accessible tools for capturing visual data. Road network monitoring and condition assessment represent domains that can considerably benefit from the integration of such technologies, as the collection and processing of relevant visual information can yield detailed and reliable insights into current conditions. Within the scope of the research, a pilot technological description is developed, outlining the process of field data acquisition using smartphones, followed by the processing of the resulting dataset through a deep neural network. The outcomes of this processing are presented using appropriate visualization methods, accompanied by an attempt to examine and validate their accuracy.

1. Bevezetés

Napjainkban a gépi látás (*computer vision*, CV) területe rendkívül dinamikus fejlődést mutat, amelynek háttérében a mesterséges intelligencia (*artificial intelligence*, AI), valamint a gépi tanulás (*machine learning*, ML) és különösen a mélytanulás (*deep learning*, DL) terén bekövetkezett áttörések állnak. A neurális hálózatok mára a közúti közlekedés számos területén alkalmazásra kerülnek: az önvezető járművek érzékelőrendszereiben, az intelligens forgalomirányító rendszerekben, valamint az úthálózat állapotának monitorozásában. A mesterséges intelligencia és a számítógépes látás így fokozatosan a közlekedésbiztonság és a közlekedésszervezés mindennapi eszközeivé válnak. A gyakorlati alkalmazások közül kiemelkedik az úthibák automatikus felismerése, amely lehetővé teszi a gyors karbantartási beavatkozásokat, valamint a járművek fedélzeti rendszereiben működő vezetéstámogató megoldások, amelyek valós időben érzékelik az útviszonyokat és a forgalmi helyzetet. A közeljövőben várható, hogy ezen technológiák integrációja

hozzájárul a közlekedésbiztonság növeléséhez és az úthálózat fenntarthatóbb kezeléséhez

A számítógépes látás kialakulása az 1970-es évek környékére, egy Stanford Egyetemen megvalósított projekthez köthető. A projekt keretében kifejlesztették az első olyan önálló működésre képes robotot, amely bizonyos mértékben érzékelni tudta a környezetét. Bár azóta a technológia jelentős fejlődésen ment keresztül, e korai kezdeményezés lefektette a számítógépes látás alapjait. (Dobson, J. E., 2023).

A számítógépes látás eredeti célja az volt, hogy a rendszer környezetéről olyan módon szolgáltatson információt, amely a rendszer számára feldolgozható. Lényegében a feladat a valós világ absztrakciója, amelyet legtöbbször az elektromágneses sugárzás mérésével és rögzítésével valósítanak meg. Az ehhez használt érzékelők közé tartoznak a hagyományos kamerák, a LiDAR és a RADAR rendszerek. A számítógépes látás teljes területe arra az alapelvre épül, hogy megpróbálja utánózni az emberi rendszerek működését — ahogyan sok más technológia is (például a neurális hálózatok).

A számítógépes látás folyamata négy fő lépésből áll az irodalom szerint: (Elgendy, 2020)

1. Bemeneti adatok (*Input data*): A rendszer elsődleges forrásai a képek és videók (az utóbbi esetében egyes képkockák), amelyek a környezet vizuális információit közvetítik a feldolgozó rendszer számára.
2. Előfeldolgozás (*Preprocessing*): Az adatok előkészítése annak érdekében, hogy a későbbi feldolgozási lépések számára megfelelő formátumban álljanak rendelkezésre. Ide tartozik például a képek normalizálása, átméretezése, színtér-átalakítása, zajsűrés és egyéb standardizáló műveletek.
3. Jellemzők kinyerése (*Feature extraction*): A képekből olyan releváns tulajdonságok kinyerése történik, amelyek alapján a rendszer különbséget tud tenni különböző objektumok vagy mintázatok között. Ez a lépés kulcsfontosságú a további tanulás és döntéshozatal szempontjából.
4. Gépi tanulási modell (*Machine learning model*): A korábban kinyert jellemzők alapján történik a modell betanítása, amely képessé válik osztályozásra, felismerésre vagy előrejelzésre. Ez lehet például egy konvolúciós neurális hálózat (*convolutional neural network, CNN*), amely azonosítja az objektumokat a képen belül. (Zaidi et al. 2025)

Ez a struktúra jól párhuzamba állítható az emberi látórendszer működésével, ahol az érzékelés, feldolgozás, felismerés és reakció szintén hasonló, egymásra épülő lépésekből áll.

2. Konvolúciós neurális hálózatok algoritmusai objektumok felismeréséhez

Az objektumfelismerés a számítógépes látás egyik feladata, amely két fő célt foglal magában: egy vagy több objektum lokalizálása egy képen, és az egyes objektumok osztályozása. Ezt úgy valósítják meg, hogy a megtalált objektum köré egy befoglaló téglalapot (*bounding box*) rajzolnak az előre jelzett osztállyal együtt. Ez azt jelenti, hogy a rendszer nem csak a kép osztályát jósolja meg, mint az osztályozási feladatok esetén, hanem az adott objektumot körülvevő befoglaló téglalap koordinátáit is. (Liu, L. et al, 2020)

Az objektumfelismerést osztályozási problémaként modellezzük úgy, hogy a bemeneti képből minden lehetséges helyen, rögzített méretű ablakokat veszünk, majd ezeket az ablakokat egy képosztályozó hálózatnak adjuk be. A legtöbb CNN algoritmus abban különbözik, hogy hogyan határozza meg a jellemzőket. Mivel az objektumfelismerést osztályozási problémaként kezeljük, a sikeresség nagymértékben függ az osztályozási pontosságától.

Ahhoz, hogy megértsük a mélytanulás-alapú rendszerek munkafolyamatát az objektumok felismerésében, meg kell

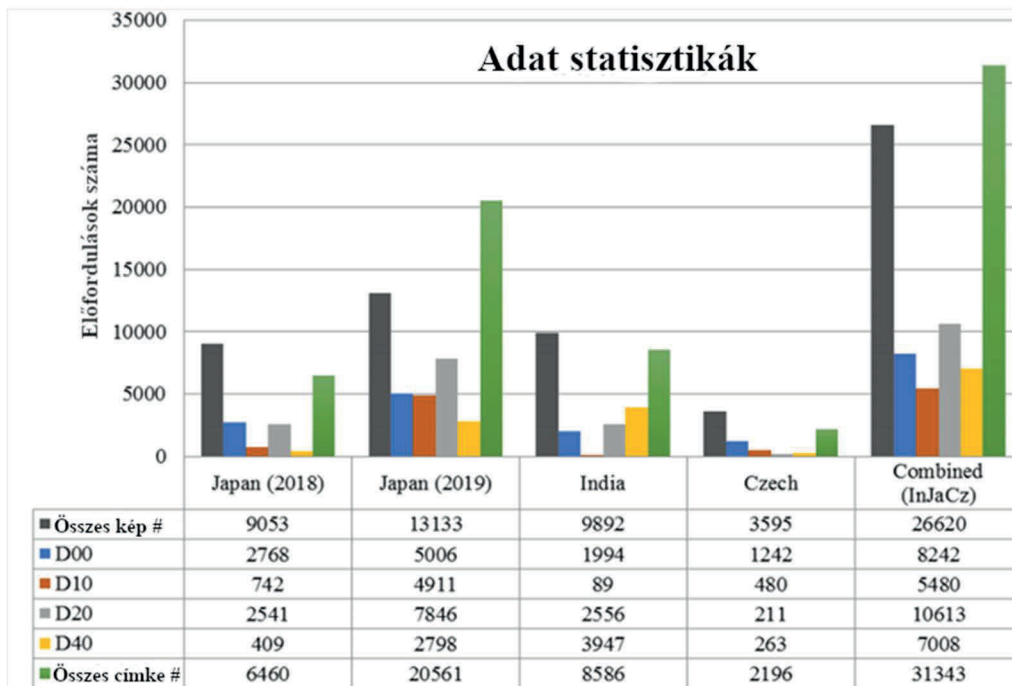
vizsgálunk, hogy milyen általános keretrendszert követnek. Az objektumfelismerő rendszerek működése több, egymásra épülő komponensből áll, amelyek közül az első a régiójavaslat (*region proposal*) generálása. Ebben a lépésben egy algoritmus vagy mélytanulási modell segítségével kerülnek meghatározásra azok a képrészletek (*region of interest, ROI*), amelyek a rendszer feltételezése szerint objektumot tartalmazhatnak. A folyamat nagyszámú befoglaló téglalapot állít elő, amelyekhez objektumtartalom-pontszám (*objectness score*) társul; a magasabb értékkel rendelkező dobozok kerülnek további feldolgozásra a hálózati rétegeken keresztül. Ezt követi a jellemzőkinyerés és a hálózati előrejelzés szakasza, amelynek során minden egyes befoglaló téglalaphoz vizuális jellemzőket rendel a rendszer, majd ezek alapján értékeli, hogy az adott régió valóban tartalmaz-e objektumot, illetve meghatározza annak osztályát. A detektálási folyamat sajátossága, hogy gyakran több téglalap is azonos objektumra vonatkozik, ezért szükséges a nem maximum szűrés (*non-maximum suppression, NMS*) alkalmazása, amely az átfedő detekciókat egyetlen reprezentatív dobozzá egyesíti, ezáltal elkerülve az ismételt felismerést. A modell teljesítményének értékelésére különböző mérőszámok használatosak, amelyek analógok a klasszifikációs feladatoknál alkalmazott pontosság, precizitás és visszahívás mutatóival. Az objektumfelismerő rendszerek esetében a leggyakrabban alkalmazott metrikák közé tartozik a közepes átlagos pontosság (*mean average precision, mAP*), a precizitás–visszahívás görbe (*PR curve*), valamint az átfedési hányad (*intersection over union, IoU*), amelyek együttesen átfogó képet nyújtanak a rendszer hatékonyságáról és megbízhatóságáról. (Gandhi, R. 2018)

3. Saját alkalmazás fejlesztése

Az alkalmazásunk ötlete két korábbi japán tanulmányból származik. Ebben H. Maeda és munkatársai a Tokiói Egyetem kutatóiként egy okostelevonos alkalmazás fejlesztéséről írnak, amely saját tanított neurális hálózataikat használja az útkárosodások felismerésére és osztályozására (Maeda et al, 2016, Maeda et al, 2018).

Publikációjukban megosztották a tanító adatokat, a betanított modelleket és magát az alkalmazást is, bár a forráskódot nem tették közzé. Kutatásuk során 6 különböző alaphálózatot (*base networks*) próbáltak ki. Ezeket az alaphálózatokat átviteli tanulás (*transfer learning*) alapjaként használták, nyilvánosan elérhető, előre betanított (*pretrained*) hálózatokat alkalmazva a jellemzők kinyerésére. Később további modelleket publikáltak különböző alaphálózatokkal és eltérő tanulóadatokkal.

Az eredeti japán adatállomány 9 053 útkárosodásról készült képet tartalmazott, amelyek összesen 15 435 károsodási esetet foglaltak magukban. Ezt az adatállományt további képekkel bővítették, melyek közül sok más országból származott. A 2019-ben közzétett végleges adatállomány 26 620 képet tartalmazott, többek között Indiából és a Cseh Köztársaságból származó képeket is. A képek eloszlása az 1. ábrán látható (Arya et al. 2020a).



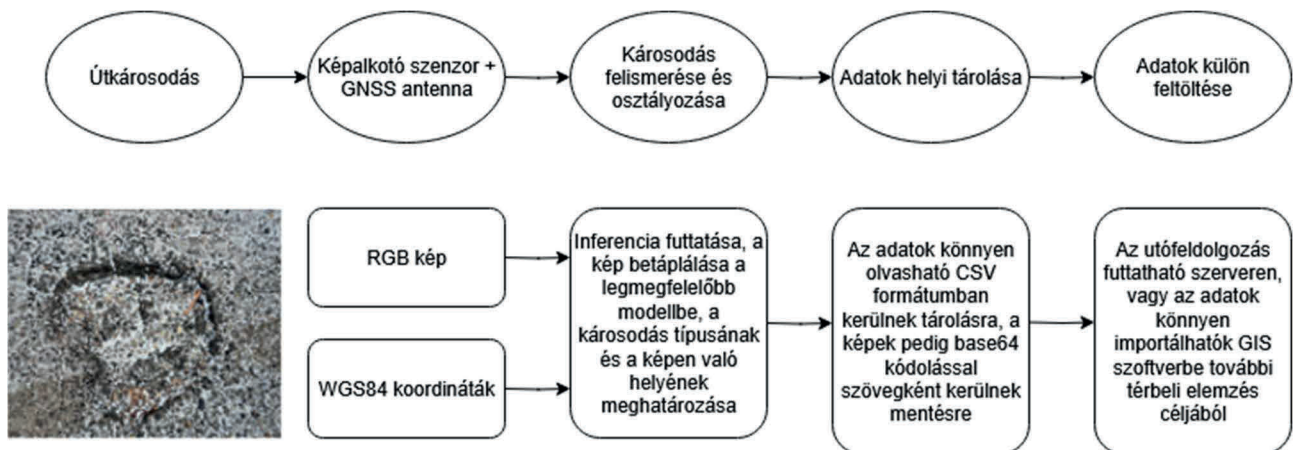
1. ábra Tanító adatok statisztikái (Arya et al. 2020a)

Úthiba típusa		Részletesen	Osztály neve	
Repedés	Lineáris repedés	Hosszirányú	Keréknyom rész	D00
		Keresztirányú	Építési illesztési rész	D01
			Egyenlő osztásköz	D10
	Krokodilrepedés	Építési illesztési rész	Építési illesztési rész	D11
			Részleges burkolat, teljes burkolat	D20
		Egyéb károsodás	Nyomvályú, egyenetlenség, kátyú, felválás	D40
	Fehér vonal elmosódás	D43		
	Gyalogátkelő elmosódás	D44		

2. ábra Útkárosodás-típusok a Japán Útkarbantartási és Javítási Útmutató alapján

Az egyes útkárosodás-típusokat a 2013-as Japán Útkarbantartási és Javítási Útmutató (*Japanese Road Maintenance and Repair Guidebook*) alapján kategorizálták (2. ábra). Ez az útmutató 8 kategóriába sorolja az útkárosodásokat. Ezek a kategóriák

szolgálták a CNN modellek kimeneti osztályaként. Ezekből a modellekből négy különböző változatot publikáltak, amelyek elnevezése a használt alaphálózat nevét és az évszámot tartalmazta.



3. ábra Az úthiba-felismerő rendszer folyamatábrája

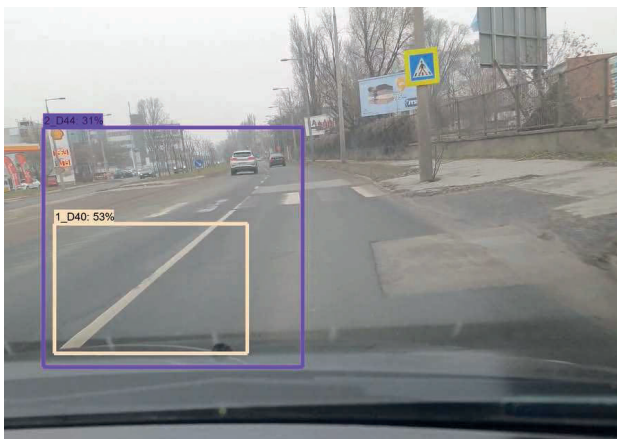
3.1. Az alkalmazás munkafolyamata

A kidolgozott alkalmazásunk áttekintő folyamata a 3. ábrán látható. A begyűjtött adatokat két külön fájlban tárolják. Az adatbázis séma két táblából áll: a **collected_data** a detektált útkárosodások jellemzőit (típus, biztonsági szint, befoglaló téglalap koordináták) rögzíti, míg az **images** a képek Base64-kódolt tartalmát és földrajzi koordinátáit (WGS84) tárolja. A két tábla azonosítókon keresztül kapcsolódik, ezáltal biztosítva a képi adatok és a detektált károsodások strukturált összekapcsolását.

3.2. Megfelelő modell kiválasztása

A modellt egy általunk Jupyter notebookban írt Python szkript segítségével választottuk ki. Ez a szkript egyszerre két modellt képes összehasonlítani: egyszerre tud betölteni két modellt, és képet vagy videót fogad bemenetként. Amennyiben videót kap, annak minden képkockáját kinyeri, majd mindkét modellen lefuttatja az adatokat (inferencia). A kimenet egy olyan kép vagy videó, amelynek minden képkockáján téglalapok és megbízhatósági százalékok jelennek meg.

A japán modellek mindegyike képtenzorokat fogad bemenetként, és visszaadja a károsodás típusát, a károsodás helyét a képen a körülvevő téglalap koordinátaival, valamint egy megbízhatósági százalékot.



4. ábra Felismert fehér vonal elmosódás és zebra festés elmosódás Dél-Budán

A 4. ábra egy összehasonlító videó egy képkockája, amelyen az első modell a fehér vonalat az aszfaltrétegek közötti elválasztásként detektálja 53%-os megbízhatósággal, ezért jelenik meg a „1_D40: 53%” felirat. Az „1” jelzi, hogy a befoglaló téglalap az első, a programnak megadott modelltől származik, míg a „D40” az útkárosodási típusok táblázatára utal. A második modell helyesen ismeri fel az elmosódott gyalogátkelőt, de csak 31%-os megbízhatósággal. A programnak megadható egy küszöbérték, amely alatt a megbízhatóság nem elegendő, így nem rajzolja ki a téglalapokat és a feliratokat.

A legjobb modellt úgy választottuk ki, hogy az összes videót minden modellpárral lefuttattuk, majd vizuális ellenőrzéssel értékeltük az eredményeket.

Fontos szempont volt az inferencia ideje is, vagyis az az idő, amely alatt a modell képes feldolgozni egy képkockát – másképp fogalmazva: alkalmas-e valós idejű feldolgozásra.

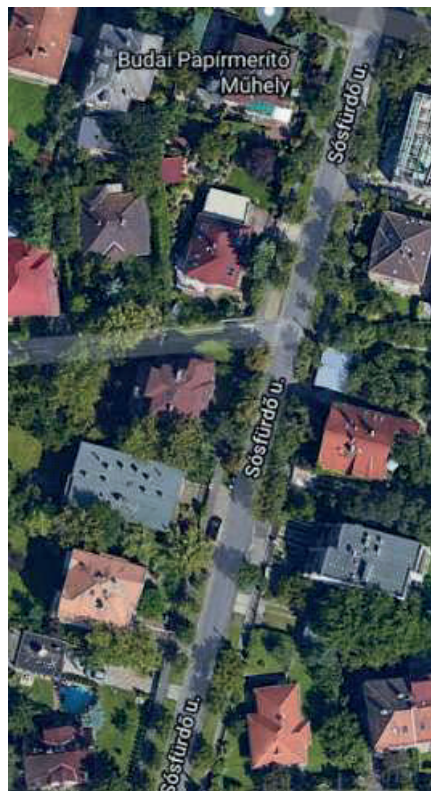
A modellek kiválasztásakor több, előre betanított objektumdetekciós hálózat közül lehetett választani. Ezek eltérnek egymástól felépítésükben (pl. ResNet, Inception, MobileNet alapú modellek), illetve abban, hogy milyen pontosságot és milyen feldolgozási sebességet biztosítanak. A választás során fontos szempont volt, hogy a modell valós időben képes legyen feldolgozni a képkockákat, ezért megmértem az egyes hálózatok képfeldolgozási idejét. (Mittal, P. 2024)

Az eredmények alapján a frozen_inference_graph_resnet_2019 modell körülbelül 4 másodpercet igényelt egyetlen kép feldolgozásához, ami valós idejű alkalmazásra alkalmatlan. A MobileNet-alapú modellek viszont lényegesen gyorsabbak (150–200 ms tartomány), miközben megfelelő pontosságot is biztosítanak.

Ezért a továbbiakban a MobileNet-en alapuló modellt alkalmaztam, mivel ez nyújtotta a legjobb egyensúlyt a sebesség és a detekciós teljesítmény között.

4. Tesztelés

A teszteléshez Budapest XI. kerületében jelöltünk ki egy rövid útszakaszt, amely kis területen belül nagy sűrűségben tartalmaz úthibákat, így alkalmas a detektálási és osztályozási eljárások értékelésére. A kiválasztott utca a Sósfürdő utca, melynek műholdfelvétele (Google Maps) az 5. ábrán látható.



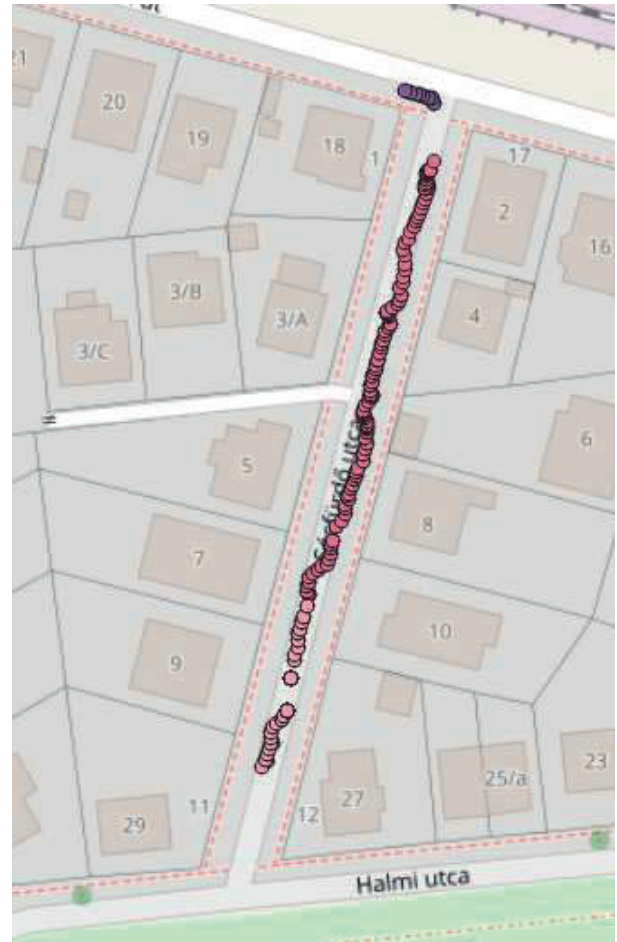
5. ábra A kiválasztott terület műholdképe

A tesztelést gyalogosan végeztük előre néző, azaz a haladási irányban tartott okostelefonnal. Néhány tesztfelvétel készült úgy is, hogy a telefont egy autó szélvédője mögé helyeztük, azonban ezeket a felvételeket főként a legjobb modell kiválasztására használtuk a Python szkript segítségével. A 6. ábrán a futó alkalmazás felhasználói felületét lehet látni a megtalált és osztályozott burkolati hibák befoglaló téglalapjaival és felirataival.

Az alkalmazás által generált eredményfájlok egyszerűen feltölthetők egy szerverre vagy betölthetők a QGIS-be további elemzések céljából. Az adatok ezután vizualizálhatók, és további térbeli elemzések alapjául szolgálhatnak. A 7. ábra térképén minden egyes pont egy olyan fényképi helyet jelöl, ahol a rendszer valamilyen úthibát felismert.

Az egyes képek azonosítóit összekapcsolhatjuk a rajtuk észlelt károsodásokkal, amelyek egy külön fájlban vannak tárolva.

A munkamenetet képernyőfelvevő alkalmazás segítségével rögzítettük. A felvételek vizuális ellenőrzése során a mintaútszakaszon 17 jól elkülöníthető útkárosodást azonosítottunk, amelyek közül a rendszer 9-et detektált sikeresen. A munkamenet során az elfogadható megbízhatósági küszöbértéket 40%-ra állítottuk be. Hat esetben hamis pozitív detektálás történt: négy csatornafedél, egy térkőből kialakított fekvőrendőr, valamint egy térkőburkolatú bejárat esetében. Ezeknél az előfordulásoknál a modell 60% alatti megbízhatósági értéket rendelt a detektáláshoz, azonban a küszöb további csökkentése a valódi pozitív találatok elvesztését eredményezte volna. A modell érzékenysége különösen magas az aszfalttól eltérő burkolati felületek esetében, emellett bizonyos aszfalttípusoknál kifejezetten hajlamos a hamis negatív eredmények előállítására. Előfordult például olyan burkolattípus, amelynek szemcsemérete olyan nagy, hogy a rendszer a teljes felületet repedésként azonosította.



7. ábra A rendszer által feldolgozott képek pozíciói, amelyeken úthibákat észlelt



6. ábra Néhány kép a rendszerről működés közben

5. Összefoglalás és kitekintés

Összességében elmondható, hogy az alkalmazás alkalmas az útkárosodások detektálására és lokalizálására, bár megbízhatósága további vizsgálatokat igényel. A japán tanulmányban szereplő rendszer már hivatalos szervek által is használt olcsó adatgyűjtési megoldásként. A jövőben érdekes lenne megvizsgálni, hogy a hálózat teljesítménye javítható lenne-e magyarországi adatokkal történő finomhangolással. Erre azonban jelenleg nincs lehetőség, mivel csak a lefagyasztott (frozen) modellek érhetők el, nem pedig a tanítható, nyers hálózatok vagy a tanítási kód. Emiatt a modell továbbképzése vagy domain adaptációja technikailag nem kivitelezhető.

Az általunk fejlesztett rendszer képes az okostelefon kamerájának használatára, és a kameraképkockákat bármilyen, útkárosodás-detektálásra vagy más célra tervezett neurális hálózat számára átadni. Az alkalmazással képek valós idejű feldolgozása lehetséges megközelítőleg 9-10 képkocka/másodperc sebességgel. A begyűjtött adatok könnyen feltölthetők egy SQL adatbázisba, így utófeldolgozás tárgyát képezhetik. A Python szkript módosítható úgy, hogy az adatbázisból vegye a képeket, és minden képen lefuttassa valamennyi modellt, továbbá ha egy útkárosodást több modell is észleli, a megbízhatósági pontszám módosítható.

A modell továbbfejlesztésével olyan költséghatékony és megbízható útkárosodási adatgyűjtési módszer alapjai teremthetők meg, amely alkalmas a gyakorlatban történő alkalmazásra. Az alkalmazás által előállított adatok jól strukturáltak és könnyen utófeldolgozhatók, a kimeneti fájlok felhasználása egyszerű. Ugyanakkor a képek hatékony tárolása még kihívást jelent.

Irodalomjegyzék

Arya, D. – Maeda, H. – Ghosh, S. K. – Toshniwal, D. – Mraz, A. – Kashiyama, T. – Sekimoto, Y. 2020. Transfer learning-based road damage detection for multiple countries. arXiv:preprint 2008.13101v1. <https://doi.org/10.48550/arXiv.2008.13101>

Arya, D. – Singh, J. – Sharma, R. – Khandelwal, S. – Singh, N. 2020. Global Road Damage Detection: State-of-the-art Solutions. arXiv preprint arXiv:2011.08740. doi: 10.48550/arXiv.2011.08740

Dobson, J. E. 2023. The Birth of Computer Vision. *Routledge*, London and New York, p. 224. ISBN 9781517914219.

Elgendy, M. 2020. Deep Learning for Vision Systems. *Manning*, Shelter Island, NY, p. 480. ISBN-13: 978-1617296192

Gandhi, R. 2018. R-CNN, Fast R-CNN, Faster R-CNN, YOLO — Object Detection Algorithms. Understanding object detection algorithms. link: <https://towardsdatascience.com/r-cnn-fast-r-cnn-faster-r-cnn-yolo-object-detection-algorithms-36d53571365e> Utolsó elérés: 2025. november 20.

Liu, L. – Ouyang, W. – Wang, X. – Fieguth, P. – Chen, J. – Liu X. – Pietikäinen, M. 2020. Deep Learning for Generic Object Detection: A Survey. *International Journal of Computer Vision* 128, 261–318. <https://doi.org/10.1007/s11263-019-01247-4>

Maeda, H. – Sekimoto, Y. – Seto, T. 2016. Lightweight road manager: smartphone-based automatic determination of road damage status by deep neural network. In *Proceedings of the 5th ACM SIGSPATIAL International Workshop on Mobile Geographic Information Systems (MobiGIS '16)*. Association for Computing Machinery, New York, NY, USA, 37–45. <https://doi.org/10.1145/3004725.3004729>

Maeda, H. – Sekimoto, Y. – Seto, T. – Kashiyama, T. – Omata, H. 2018. Road Damage Detection and Classification Using Deep Neural Networks with Smartphone Images. *Computer-Aided Civil and Infrastructure Engineering* 33(12): 1127–1141. <https://doi.org/10.1111/mice.12387>

Mittal, P. 2024. A comprehensive survey of deep learning-based lightweight object detection models for edge devices. *Artificial Intelligence Review*, 57(242). <https://doi.org/10.1007/s10462-024-10877-1>

Zaidi, S.S.A. Ansari, M.S. Aslam, A. Kanwal, A. Asghar, M. Lee, B. 2022. A survey of modern deep learning based object detection models. *Digital Signal Processing*, 126: 103514. <https://doi.org/10.1016/j.dsp.2022.103514>

