

**USING ARTIFICIAL INTELLIGENCE FOR  
OBJECT DETECTION (FIRST PART)****A MESTERSÉGES INTELLIGENCIA  
FELHASZNÁLÁSI LEHETŐSÉGEI AZ  
OBJEKTUMFELISMERÉSBEN (ELSŐ RÉSZ)**KOLLÁR Csaba<sup>1</sup> – NAGY Barna<sup>2</sup>**Abstract**

The dominance of artificial intelligence is becoming more and more important in everyday life and in the field of scientific discourses, and security technology and security science are no exception. In the first – theoretical – part of our study we deal with machine vision, the history of technical and information science of machine vision, neural networks, and the teaching of networks. In a separate section we discuss the common types of tasks in computer vision (image classification, object localization, object detection), as well as the hardware and development environment (Intel Neural Compute Stick, OpenVINO Toolkit, Raspberry Pi), through which we were able to implement the developments and their results, will be published in the next part of our study. We conclude our theoretical study with an introduction to the concepts of artificial intelligence, machine learning, and deep learning.

**Keywords**

security systems, artificial intelligence, computer vision, Intel Neural Compute Stick, Raspberry Pi

**Absztrakt**

A mesterséges intelligencia dominanciája a mindennapi életben és a tudományos diskurzusok területén is egyre jelentősebb, s ez alól a biztonságtechnika és a biztonság-tudomány sem kivétel. Tanulmányunk első – elméleti – részében a gépi látással, a gépi látás technika- és információtudományi történetével, a neurális hálózatokkal, a hálózatok tanításával foglalkozunk. Külön részben értekezünk a gépi látás gyakori feladattípusairól (képosztályozás, objektum lokalizáció, objektum detektálás), valamint bemutatjuk azt a hardver- és fejlesztőkörnyezetet is (Intel Neural Compute Stick, OpenVINO Toolkit, Raspberry Pi), amelyik révén a tanulmányunk következő részében ismertetésre kerülő fejlesztéseket és annak eredményeit tudtuk megvalósítani. Elméleti tanulmányunkat a mesterséges intelligencia, gépi tanulás, mélytanulás fogalmainak ismertetésével zárjuk.

**Kulcsszavak**

biztonságtechnika, mesterséges intelligencia, gépi látás, Intel Neural Compute Stick, Raspberry Pi

<sup>1</sup> kollar.csaba@phd.uni-obuda.hu | ORCID: 0000-0002-0981-2385 | associate professor/egyetemi docens | Óbudai Egyetem Bánki Donát Gépész és Biztonságtechnikai Mérnöki Kar

<sup>2</sup> nagy.barna@blx.hu | ORCID: 0000-0002-8101-1080 | Software Architect/szoftver architect | Blumenthal Consulting Kft.

## BEVEZETÉS

Egy szervezet információbiztonságának megteremtése és fenntartása egyszerre több területre is feladatot ró. Ennek része az objektum- és területvédelem is, melynek célja, hogy illetéktelenek ne tudjanak fizikailag hozzáférni a különféle információtároló, -továbbító, vagy -feldolgozó eszközökhöz. A megvalósítás során védjük a védendő terület határfelületét a bejutásról, a belépési pontokat az illetéktelen átjutástól, a területen belül pedig keressük az illetéktelen mozgást, vagy egyéb, nem kívánatos tevékenységeket.

Ezeknek a tevékenységeknek az eszköztárából manapság már nem hiányoznak a technikai ellenőrzésű beléptető és a mozgást ellenőrző rendszerek sem. Az ellenőrző rendszerek gyakran tartalmazzak videomegfigyelést, mely egyrészt csökkenti a szükséges élőmunka mennyiségét, másrészt megsokszorozza annak hatékonyságát. Ezek a rendszerek azonban egyre inkább túllépnek a hagyományos megfigyelő szerepükön és egyre komplexebb feladatok végzésére vállalkoznak.

A gépi látás használatával nemcsak az ember (például fáradékonyságból, figyelmetlenségből eredő) hibáit küszöbölhetjük ki, hanem a biztonságtechnika területéről kimutató, üzleti vagy társadalmi célú alkalmazásra is lehetőség nyílik.

Példaként említhető az okos városok és okos falvak koncepciója, ahol az okos jelzőt részben a mesterséges intelligencia, a gépi tanulás, a gépi látás újfajta hasznosítása igazolja. A gépi látás segítségével olyan információk birtokába kerülhetünk, mint például a településre érkező és kilépő gépjárműforgalom nagysága, a gépjárművek típusa, vagy a településen tartózkodók demográfiai adatai (nem, életkor). Egy olyan kor veheti kezdetét, amikor nemcsak régi, hónapokkal, évekkel ezelőtt gyűjtött adatokkal rendelkezhetünk a településről, hanem gyakorlatilag egy azonnali pillanatképünk van az éppen most történő folyamatokról. [1]

A látogatószámolás, nem- és kormeghatározás inkább üzletileg hasznosítható információ, de az arcfelismerés, a rendszámfelismerés vagy az objektumkövetés már a terület fizikai védelmét is szolgálja. [2]

Ezeknek a gépi látást használó feladatoknak van egy komoly problémája. Folyamatos és jelentős számítási kapacitást igényelnek, ami a végponti eszközökben, például egy videokamerában jellemzően nem áll rendelkezésre. A végponti eszközök gyakran csak az adatok digitalizálására és IP hálózaton történő továbbítására képesek. A gépi látást segítő neurális hálózatok (kellő sebességű) futtatása már meghaladja a képességeiket, így az egy központi szerverre marad (amennyiben fel van erre készítve a rendszer). Az már a jelenlegi rendszereknél is megállapítható, hogy a rendszer bővíthetőségét rontja, ha a végpontok „buták” és minden számítás a központi szerverre marad. Például, ha egy videokamera nem tud tömörített videojelet küldeni, akkor néhány példány könnyen elfogyasztja az infrastruktúrában rendelkezésre álló sávszélességet.

Az egyik lehetséges fejlődési irány az, hogy a végponti eszközök válnak egyre „okosabbá”. Ezek az eszközök lesznek képesek olyan gépi látást igénylő feladatokra, amelynek csak a végeredményét, a kimenetét továbbítják a központ felé.

Ehhez viszont szükséges, hogy a végponti eszközök képesek legyenek a gépi látás számításiigényes feladatainak elvégzésére. Az Intel Corporation erre a feladatra fejlesztett

egy feldolgozó egységet (processing unit), mely a felhasználási köre miatt Vision Processing Unit (VPU) nevet kapta.

Írásunkban az vizsgáljuk, hogy hogyan valósítható meg az Intel kis teljesítményű eszközökbe szánt Neural Compute Stick 2 képi feldolgozó segítségével a gépi látás (Computer Vision – CV) két gyakori feladata, az objektumfelismerés és -azonosítás. Több környezetben, gyakorlati példákon vizsgáltuk az eszközt. Feldolgozó teljesítményét összevettük más, általános célú processzorokkal (Intel CPU), így könnyebben meg lehet ítélni, hogy mekkora számítási teljesítménnyel bír a gépi látás feladatainak körében.

Tanulmányunk első része – ahogy arról már az absztraktnál is írtunk – a téma elméleti keretét vázolja fel. Mivel a mesterséges intelligencia és a kapcsolódó fogalmak tartalmukban és tartalmuk értelmezésében is gyorsan és folyamatosan fejlődnek, ezért nem találkoztunk egységesen használt fogalmakkal. Különösen igaz ez a magyar nyelvű szakirodalomra, mely gyakran nem találja a megfelelő és elfogadható kifejezést egy-egy új fogalomra. Ennek ellenére igyekeztünk mindig a magyar kifejezéseket használni, megadva annak az angol megfelelőjét is. Ettől csak akkor tértünk el, ha a magyar változat mindenképpen rontotta volna az érthetőséget.

## A GÉPI LÁTÁS

A gépi látás (Computer Vision) egy olyan interdiszciplináris tudományterület, mely a számítógépek számára teszi lehetővé álló- vagy mozgóképek olyan magasszintű értelmezését, „látását”, mellyel eddig csak emberek rendelkeztek. Fontos hangsúlyozni a magas szintet, mert a használt technológiáktól a háromdimenziós világunk egyfajta megismerését, feltárását várjuk el, úgy, hogy a rendszer bemenete gyakran csak kétdimenziós képek, illetve képek sorozata.

A terület pontos megértéséhez érdemes tudni, hogy az idegennyelvű szakirodalom megkülönbözteti a Machine Vision és Computer Vision fogalmakat, de a magyar nyelvben ugyanazt a kifejezést, a gépi látást használjuk mindkettőre. A fogalmak meghatározását nehezíti, hogy értelmezésük nem letisztult, gyakran szükséges azok definiálása a használatuk előtt.

A Machine Vision (MV) technológiákat – azok használói – általában a Computer Vision részterületének tekintik és jellemzően ipari, szabályozási területen hasznosítják. Fontos tulajdonságuk, hogy általában kontrollált körülmények (előre beállított fényviszonyok, ismert objektumok, jól definiált optikai jellemzők) között végeznek velük ipari vagy ipart támogató tevékenységeket. Elterjedésük a 90-es évekre tehető, amikor a különféle gyártósorok minőségbiztosítási feladatait segítették velük. [3]

A Computer Vision tudomány művelői szerint a két terület között a legfőbb különbség, hogy míg a Machine Vision-t inkább mérésre, értékelésre és az ehhez kapcsolódó vezérlésre, szabályozásra használják, addig a Computer Vision-t értelmezésre, látásra, de mindenképpen valami összetettebb, magasabb szintű tevékenységre. [4]

Tipikus MV felhasználási területek: a mérések, méretezések, pozicionálás, robotok irányítása, kódok olvasása, egyszerűbb jelenségek (pl.: anyaghibák) felismerése, míg a CV-t az alábbi problémakörökben használják: objektumfelismerés, objektumazonosítás, számolás, nyomkövetés, mechanikai elemzések. [5]

A jövőben, várhatóan, a két terület egyre jobban össze fog fonódni, ahogy a CV technológiái egyre inkább megjelennek ipari környezetben is.

### A gépi látás rövid története

A Computer Vision múltja messzire tekint vissza, bár igazán nagy népszerűség csak mostanában övezi. Ennek több oka van. Egyrészt a felhasznált matematikai elmélet is sokat fejlődött. Sikerült olyan területek találni, ahol a gyakorlatban is jól teljesít, valamint az egy-egy áron rendelkezésre álló számítási kapacitás is hatalmasat nőtt az elmúlt évtizedekben.

Az egyik legfontosabb kezdeti mérföldkő Lawrence Roberts PhD tézise volt: „Machine Perception of Three-Dimensional Solids” (Lawrence, 1963), melyben háromdimenziós „információt” próbált kinyerni kétdimenziós képekből. Ő a tézisében még vonalas ábrákkal dolgozott, ám pár évvel később Seymour Papert az MIT AI laboratórium professzora már valós fényképek segítségével szeretett volna áttörést elérni.

A híres „Summer Vision Project” azonban kudarcba fulladt, és egyben rávilágított a CV nehézségeire. (Zbigniew, 2018) A projekt ambiciózus célkitűzését valószínűleg az is fűtötte, hogy a 60-as években nagyon felkapott kutatási téma volt a mesterséges intelligencia. Akkoriban többen azt vízionálták, hogy 25 éven belül a számítógépek olyan intelligensek lesznek, mint az emberek. És ahhoz, hogy ez megtörténjen a gépi látásnak is fejlődnie kellett.

A következő nagy lépés a CV fejlődésében David Marr könyve volt: „Vision: A computational investigation into the human representation and processing of visual information”. David Marr egy új módszert vezetett be, mely „bottom-up” megközelítésként terjedt el. Ennek lényege, hogy a képfeldolgozás egy hierarchikus lépéssorozat, mely alulról építkezik. Első lépésben alacsony szintű feldolgozás hajt végre. Ilyenek például az éldetektálás, sarok detektálás, majd a következő lépésben már komplexebb műveletek következnek, míg végül a háromdimenziós modell megalkotása. (Gomes, 2000) [6]

Ez a fajta „alulról történő építkezés” mind a mai napig meghatározó elv a CV-ben, a deep learning rendszerek is így működnek.

Bár Marr könyve mérföldkő volt, kevés gyakorlati információt tartalmazott. Ezzel szemben Kunihiko Fukushima japán számítógéptudós, egy olyan neurális hálót alkotott – ez volt a Neocognitron (Fukushima, 1980) – mely már képes volt minták felismerésére. A hálózat több (különböző feladatot ellátó) rétegből állt és működését tekintve a mai konvolúciós neurális hálózatok (Convolutional Neural Network – CNN) őseinek tekinthető. (Demush, 2019) [7]

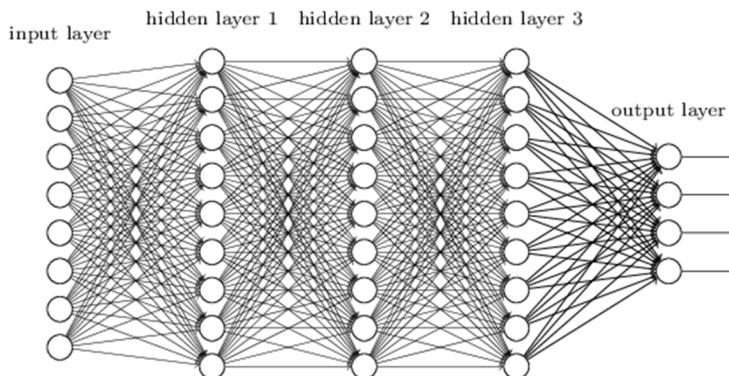
A 90-es évek végén a Computer Vision elfordult az objektumok háromdimenziós modellezése felől (ez volt a CV kutatások kezdeti célkitűzése) és az objektumok felismerése került a középpontba. 2006-ban született meg a Pascal VOC project, mely egyfelől standardizálta a képeket az objektum osztályozási feladatokhoz, másfelől egy workshop-okkal záruló versenysorozatot hirdetett (Visual Object Classes Challenge). A versenyben minden évben összemérték az induló objektumfelismerő metódusokat. (Demush, 2019) [7]

2010-ben indult a Pascal VOC-hez hasonló versenysorozat, az ImageNet Large Scale Visual Recognition Challenge (ILSVRC). A technológiai haladásnak köszönhetően a számok gyorsan nőni kezdtek. Amíg a Pascal VOC 2012-ben még csak 11.530 darab képpel és 20 osztállyal rendelkezett, addig az ImageNet több millió felcímkézett képpel és több

ezer osztállyal várta a jelentkezőket. A versenysorozat indulásakor még 26% körül volt az osztályozás hibaszázaléka, de ez gyorsan lecsökkent pár százalékra. Volt egy másik hozadéka is a versenynek: az objektumfelismerés témakörében egyeduralkodóvá tette a konvolúciós neurális hálózatokat (CNN).

## A neurális hálózatok

A mesterséges neurális hálózat (Artificial Neural Network – ANN) egy olyan nagyszámú, hasonló típusú, önmagában egyszerű felépítésű, valamilyen gráfba szervezett műveleti elemekből (neuronok) álló hálózat, mely elosztott működésű információfeldolgozásra képes.



1. Ábra: A neurális hálózat egy lehetséges topológiája (forrás: [neuralnetworksanddeeplearning.com](http://neuralnetworksanddeeplearning.com))

A hálózatok lehetnek szoftveres és/vagy hardveres megvalósításúak. Közös bennük, hogy rendelkeznek tanulási fázissal (training phase), mely során eltároljuk bennük azokat a bemeneti adatokban rejtve meglévő információkat, melyeket a felhasználás során, később majd előhívunk (inference) belőlük. Ez a két fázis néha nem válik élesen szét, ilyenkor adaptív viselkedésű hálózatokról beszélünk, melyek az információ előhívása során is módosítják belső struktúrájukat, ilyenkor is tanulnak. [8]

Fontos hangsúlyozni, hogy a mesterséges neurális hálózatok tanítása során nem egyszerű adatrögzítés történik. Az előhívás során nem a megtanított adatokat kérjük vissza, és nem is a hálózat bemenetét várjuk vissza a kimeneten.

Ugyan a hálózatok tanítása felfogható egyfajta programozásként is, hiszen korábban nem létező információfeldolgozó képességgel ruházzuk fel a hálózatot, ez azonban nem azonos sem a procedurális programozással, hiszen nem egy algoritmust alkotunk. És nem azonos a deklaratív programozással sem, mert nem a problémát fogalmazzuk meg.

A neurális hálózatok tanítása során az ismert bemeneti (input data) és – felügyelt tanítás során – a kimeneti értékek (output data) alapján a hálózat belső változóit (súlyokat) változtatjuk annak érdekében, hogy majd az előhívás során, egy ismeretlen, de a korábbi bemenetekre a megfelelő módon hasonlító bemeneti mintára az elvárt, egyfajta generalizált választ adjon a hálózat. [8] [9]

A hálózat bemenete és kimenete között általában valamilyen nemlineáris kapcsolat van. Ez a hálózati elemek nemlinearitásával magyarázható. A hálózatok képessége nagy

mértékben függ az elemek (neuronok) topológiájától, a neuronok tulajdonságától, a belső súlyok értékétől.

A hálózat ezen képessége egyébként meg is határozza, hogy milyen problémákra, feladatok használható a neurális háló. Azoknál a feladatoknál teljesíthet jól, aminek a megoldása nem, vagy csak nagyon nehezen, nagyon költségesen algoritmizálható. Továbbá azokban az esetekben, ahol a bemeneti adatok zajosak, zavarosan, nehezen specifikálhatók. Tipikusan ilyen feladatok a felismerési problémák és optimalizálási feladatok.

A hálózat ötlete a biológiai neurális hálózatokból ered. A terület úttörői úgy gondolták, hogy a természetes hálózatok mintájára létrehozhatók és működtethetők mesterséges neurális hálózatok is. Ennek eredményeképp a 40-es években jelentek meg az első tanulmányok és modellek, majd indult el a neurális számítástudomány a fejlődés útján.

1959-ben már gyakorlati alkalmazása volt a neurális hálóknak. A MADALINE egy több rétegű hálózat volt, melyet a telefonvonalak visszhangjainak kiküszöbölésére használtak. Később azonban elcsendesedtek a neurális háló kutatásai. Ennek legfőbb oka az volt, hogy Marvin Minsky és Seymour Papert a Perceptrons című könyvükben (Minsky, 1969.) kimondták, hogy a perceptronok valójában csak nagyon kevés feladatban használhatók, mert csak lineárisan szeparálható osztályozási feladatok megoldására képesek. [8]

További oka a kutatások visszaszorulásának az volt, hogy a Neumann architektúrára épülő számítógépek népszerűsége és gyakorlati használhatósága nagyra nőtt, a neurális háló általános alkalmazhatóságának ígérete azonban beteljesületlen maradt.

Újabb lendületet csak a 80-as években kapott a terület. Az 1982-ben tartott amerikai-japán közös neurális hálózat konferenciának köszönhetően az amerikaiak – attól félve, hogy lemaradnak a japánok mögött – jelentős erőforrást allokáltak erre a kutatási területre.

A Hopfield-háló (Hopfield, 1982.) [10] és a hálózatok tanításában jelentős back-propagation algoritmus publikálása visszafordította az érdeklődést a neurális hálózatok felé. Ennek eredményeképpen elterjedtek a többrétegű (multiple layered) hálózatok és megoldás született azok tanítására is. (Strachnyi, 2019.) [11]

## **A neurális hálózatok tanítása**

A mesterséges neurális hálózatok egyik fontos jellemzője azok tanulási képessége. Ez alatt olyan adaptációs képességet értünk mellyel változtatni tudják viselkedésüket, működésüket. Természetesen – a tanítás során – ez a változtatás egy kívánt cél, egy elvárt működés irányába történik, amely, ha időben változik, akkor (további tanítással) tovább adaptálódhat a hálózat. A tanításoknak két fő típusa van (Horváth, 2006.) [8]:

- felügyelt tanítás (supervised learning) során a meghatározott bemeneti értékekhez az elvárt kimeneti értékek is rendelkezésre állnak. A tanítás során úgy módosítjuk a hálózat belső paramétereit (súlyokat), hogy a hálózat tényleges (aktuális) válasza minél inkább megegyezzen az elvárt válasszal. Ez a gyakorlatban nagy mennyiségű bemenet-kimenet páros (tanító mintapont) használatát jelenti, melyekkel iteratív módon, gyakran a tanítópontok ismételt felhasználásával közelítjük a kívánt eredményt.

- nem felügyelt tanítás (unsupervised learning) esetében csak bemeneti értékeink vannak, az elvárt kimeneti értékek nem állnak rendelkezésre. Nem tudjuk megmondani, hogy adott bemenetre mi a helyes válasz. Ebben az esetben nem előre meghatározott bemenet-kimenet párosok alapján várjuk el, hogy a hálózatunk valamiféle absztrakciós képességgel rendelkezzen, hanem csak a tanításban használt bemenet-terére tudunk olyan megkötéseket tenni, ami segítségével a hálózat működése kedvező irányba módosul. Például megkötések lehetnek, hogy a minták mennyire hasonlítanak egymásra, vannak-e a minták terében olyan tartományok, ahol sűrűsödnek, lehet-e a térben csoportokat találni. A nem felügyelt tanítású hálózatok által megoldott feladatokra gyakori példa a klaszterezés (csoportosítás).

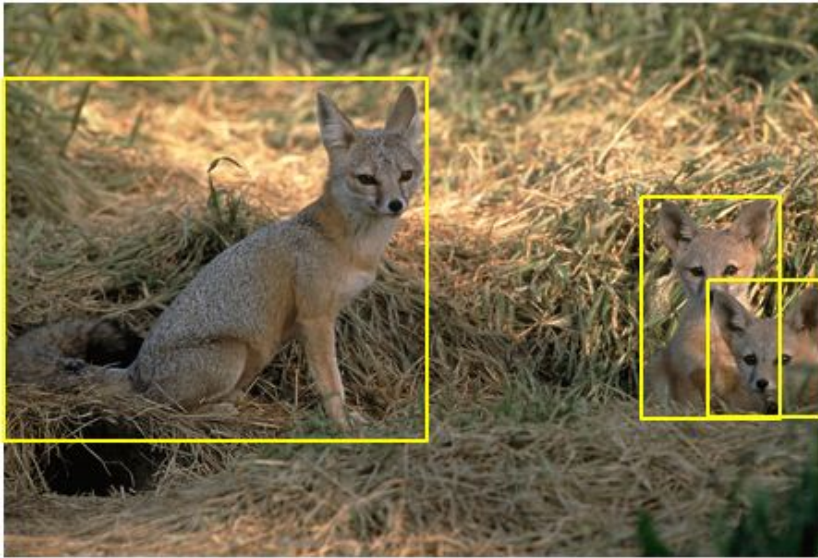
Megjegyezzük, hogy a felügyelt és nem felügyelt tanulás mellett egyes források megkülönböztetik még a megerősítő (visszacsatolásos) tanulást is, amelyiknek a lényege annak vizsgálata, hogy a kumulatív jutalom maximalizálása érdekében milyen cselekvési stratégiát kellene folytatnia az intelligens „ügynöknek”, vagyis magának a gépi tanulónak az adott környezetben.

### A gép látás gyakori feladattípusai

A gépi látás (Computer Vision) gyakorlati alkalmazásait jól definiálható feladatokra lehet osztani. Egy-egy összetett probléma megoldása gyakran bomlik ezekre a részfeladatokra, úgy, hogy a részfeladatokat láncba fűzik. Így egyik részfeladat megoldása (kimenete) egyben inputja (bemenete) a sorban utána következő feladatnak.

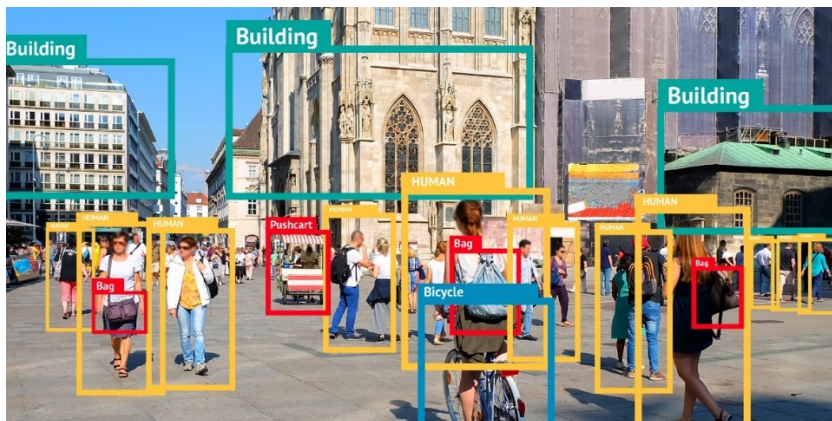
**Képosztályozás.** A képosztályozás (Image Classification, másképp Image Recognition) egy olyan eljárás, mely során egy osztályt vagy címkét rendelünk a bemenetként kapott képhez. Az eljárás a több előre meghatározott osztály közül megmondja, hogy mely, illetve melyek tartozhatnak a képhez. Ez a megvalósításokban általában azt jelenti, hogy kimenetként egy vektort kapunk vissza, melynek minden eleme egy osztályhoz tartozik, értéke pedig megadja, hogy milyen valószínűséggel tartozik a kép abba az osztályba. [12]

**Objektum lokalizáció.** Az objektum lokalizáció (Object Localization) az a művelet, mely során a hálózat megpróbálja az objektum (vagy objektumok) helyét meghatározni egy képen. A helymeghatározás során jellemzően az objektumot befoglaló téglalap paramétereit adja vissza a hálózat. [13]



2. Ábra: Object localization: a képen a megjelölt rókkák (forrás: <https://www.kaggle.com/c/imagenet-object-localization-challenge>)

**Objektum detektálás.** Az objektum detektálás (Object Detection) egy összetett feladat, mely során a hálózat egyrészt lokalizálja (localization) az egyes objektumok helyét, másrészt azonosítja (image classification) azokat. Végeredményben a neurális hálózat a kimenete segítségével megmondja, hogy mely (jellemzően) téglalap által határolt területen mi található. Az azonosítás, jelen esetben image classification egy véges számú lista (label list) alapján történik, a hálózat a klasszikus image classification-höz hasonlóan csak azokat az objektumokat ismeri fel, amire tanítva lett. [12]

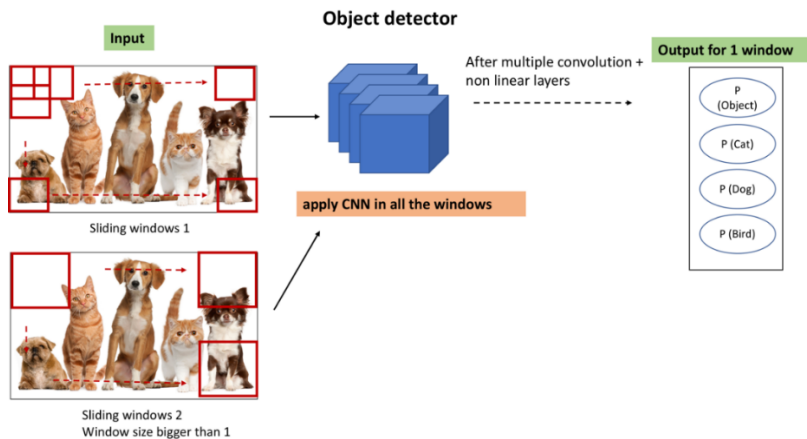


3. Ábra: Példa az objektum detektálás kimenetére. (forrás: <https://bitmovin.com/object-detection>)



Az object detection általában egyszerre több objektum lokalizációját és azonosítását végzi el. A hálózatokkal kapcsolatban az az elvárás, hogy egy többféle tárgyat tartalmazó képről (pl.: videómegfigyelő rendszer) mondja meg, hogy miket tartalmaz. [14]

A legelső megvalósítások az úgynevezett Sliding Window (**csúsztatott ablak**) megközelítést használták, mely során (alkalmazásfüggő módon) különböző alakú és méretű téglalapokkal pásztázták végig a képet, közben pedig vizsgálták az így kimetszett területet. Ez nagyon idő és erőforrásigényes megoldás volt, de megjelentek jobb megközelítések. [15]



4. Ábra: Csúsztatott ablak használata az objektumdetektálásban (forrás: <https://towardsdatascience.com/evolution-of-object-detection-and-localization-algorithms-e241021d8bad>)

A Region-CNN (R-CNN), majd később a Fast R-CNN módszerek nem pásztázták végig a képet, hanem generáltak számos potenciális régiót (region proposal) és azokon dolgoztak tovább. Ez gyorsabb volt, mint a sliding windows megközelítés, de még mindig túl lassú. Ezek a megvalósítások az úgynevezett két lépéses (two-step) megoldások voltak. Első lépésben történt a régió generálás, másodikban pedig az image classification.

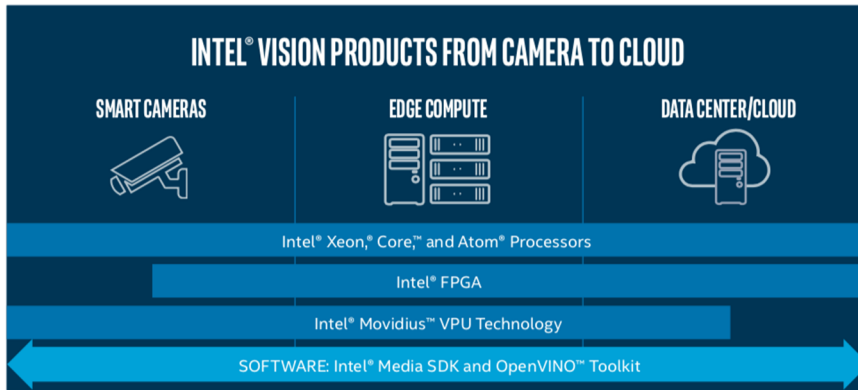
Jelenleg a leggyorsabb megoldást egylépéses (single-shot) objektum detektálás adja. Gyakran valós idejű képfeldolgozásra is alkalmasak, viszont sokkal pontosabban. [16] Az egy lépéses hálózatra példa a YOLO (You Only Look Once) és az SSD (Single Shot Detection) hálózatok. Ezeket később részletezem.

## AZ INTEL COMPUTER VISION MEGOLDÁSAI

Az 1968-ban alapított Intel Corporation nagy múltú hardvergyártó, amely követi (és lehetőségeihez képest alakítja is) az aktuális trendeket. Ezért látva a Computer Vision alkalmazások gyorsuló terjedését, a saját termékpallettáját is az ehhez kapcsolódó igényekhez igazította. Az Intel felismerte, hogy mind a mesterséges intelligencia (MI), mind az erre épülő gépi látás egyre kevésbé az adat- és számítóközpontok (felhők) kiváltsága. A fajlagos számítási kapacitás és a MI technológiák fejlődésével a végponti eszközökben (edge computers) is megjelennek. Ma már nem elképzelhetetlen, hogy egy okoskamera, drón, kisebb robotok vagy más IoT eszközök komolyabb gépi látást igénylő feladatokat is megoldjanak,

bár ez gyakran célhardverek segítségével történik. Az Intel ezekre a (részben hardveres) kihívásokra az alábbi termék csoportokkal válaszol:

1. Intel Xeon, Core és Atom processzorok
2. Intel FPGA
3. Intel Movidius VPU



5. Ábra: az Intel termékcsaládjai Computer Vision feladatokra (forrás: intel.com)

### Intel Movidius Vision Processing Unit (VPU)

Az Intel 2016-ban vásárolta fel a kaliforniai Movidius céget, mely speciális célprocesszorairól volt ismert. A vállalat Myriad nevű processzorai olyan kifizetésű célhardverek, melyek mesterséges intelligencia gyorsítóval (AI accelerator) rendelkeznek, és ezzel támogatják a gépi látást igénylő feladatokat. Ezt a processzor osztályt – hogy megkülönböztessék a többi célprocesszortól – VPU-nak, Visual Processing Unit-nak nevezték el.

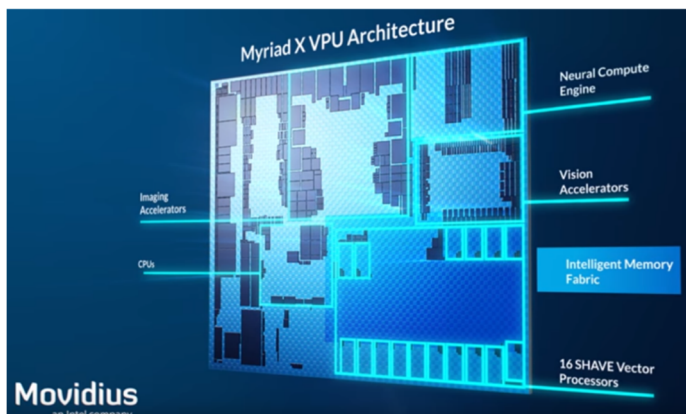
A hardverfejlesztő vállalatok érdekeltek abban, hogy termékeik kipróbálását, tesztelését és a velük való fejlesztést minél inkább támogassák. Ez a támogatás processzorok, mikrokontrollerek esetében általában fejlesztőlapot (development board) és fejlesztő környezetet (development environment) is jelent. Az Intel a Movidius VPU esetében több eszközzel is megjelent.

**Az Intel Neural Compute Stick 2 eszköz.** Az Intel vállalat Movidius VPU-t használó saját készítésű eszköze az Intel Neural Compute Stick 2 (NCS2), mely egy USB csatlókba illeszthető - ezen keresztül történik a kommunikáló és tápellátás – mesterséges intelligencia feladatokat támogató modul.



6. Ábra: Az Intel Neural Compute Stick 2 (forrás: intel.com)

Az NCS2 a Movidius™ Myriad™ X harmadik generációs VPU-t tartalmazza, mely akár 4 TOPS (Trillion Operations Per Second) teljesítményre is képes, csupán 1,5 W fogyasztás mellett. A VPU valójában nem egy darab processzor, hanem egy komplett SoC (System-On-Chip). [17]



7. Ábra: Myriad X VPU architektúra (forrás: tomshardware.com)

### Az OpenVINO Toolkit

Az Intel a Computer Vision-t támogató eszközeinek (CPU-k, GPU-k, VPU-k, FPGA-k) szoftveres támogatását két szoftvercsomaggal segíti:

- Intel Media SDK
- OpenVINO Toolkit (Open Visual Inference and Neural network Optimization)

Az OpenVINO egy ingyenes csomag, mely közösségi támogatással és az Intel támogatásával rendelkező verzióban is elérhető. Az Apache Licence 2.0 licenccel letölthető Windows, Linux és macOS platformokra is. A csomag az alábbi főbb komponenseket tartalmazza [18]:

- Model Optimizer: egy parancssori modell konvertáló eszköz, mellyel ismert modellező eszközök modelljeit lehet olyan köztes formátumra (Intermediate Representation - IR) hozni, hogy azt az Intel eszköze is megértse
- Inference Engine: egy API, mely segítségével egy alkalmazásból is meghívható a neural stick.
- OpenCV könyvtár: az OpenCV közösségi verziója, újrafordítva az Intel eszközeire
- Model Downloader: parancssori eszköz, mellyel előre tanított modellek tölthetők le az eszközre.

Az Intel az OpenVINO szoftverfejlesztői csomaggal több hardvertípust is támogat, azonban ez a támogatás korlátos, azaz csak bizonyos generációk, bizonyos technológiákkal rendelkezők vannak a támogatottak listáján. Ennek az az oka, hogy az egyes hardvertípusokhoz külön-külön készült plugin, melyek csak bizonyos, elsősorban újabb utasításkészleteket vagy technológiákat használnak a neurális hálózatok futtatására.

- (1) A processzorai közül az alábbiakat támogatja:
  - a. Intel Xeon család: AVX és AVX512 utasításkészlettel
  - b. Intel Core család: AVX2 utasításkészlettel
  - c. Intel Atom család: SSE utasításkészlettel
- (2) A grafikus gyorsítói közül:
  - a. Intel HD Graphics
  - b. Intel Iris Graphics
- (3) FPGA kártyái közül:
  - a. Intel Arria 10 FPGA 10
- (4) VPU moduljai közül:
  - a. Intel Movidis Myriad 2 lapkával szereltek
  - b. Intel Movidis Myriad X lapkával szereltek

Ezzel a technológiai megkötések a processzorok fiatalabb generációkra szűkíti a támogatottak listáját. A támogatás hiánya azonban nem azt jelenti, hogy biztosan nem lehet futtatni hálózatokat. Csupán annyit jelent, hogy nem garantált, hogy a Model Optimizer eszköz olyan hálózatmodellre fog fordítani, aminek minden rétege megfelelően implementált a használt processzor utasításkészleteivel. [19]

Az általunk elérhető és a vizsgálatokban használt processzorok nem mindegyike van a támogatottak között, de a kiválasztott neurális hálózatok mégis futtathatók voltak rajtuk.

Számítógép	Processzor	Utasításkészlet kiegészítés	Hiányzó utasításkészlet
Mac mini "Core i5" 2.3 (Mid-2011)	Core i5 (I5-2415M) (2nd generation)	Intel® AVX	AVX2
MacBook Pro 13-Inch "Core i5" 2.6 Mid-2014	Core i5 (I5-4278U) (4th generation)	Intel® SSE4.1, Intel® SSE4.2, Intel® AVX2	

1. Táblázat: a használt Intel processzorok és utasításkészleteik (forrás: intel.com)

## AZ EDGE COMPUTING

Az elosztott információs rendszerek architektúrájának egyik paradigmája, hogy a számítási és a tárolási kapacitás egy részét, közelebb hozzuk a végpontokhoz, melyek forrásai és/vagy fogyasztói lehetnek az adatoknak. Ezek a végpontok a rendszer alkalmazásától függően lehetnek személyi számítógépek, mobil eszközök, beágyazott eszközök, lokális szerverek, de az IoT világ erősödésével jelenthetnek szenzorokat (érzékelők), beavatkozókat (aktuátorok) vagy csak egyszerű felhasználói interfészeket.

A cél minden esetben az, hogy csökkenjen a rendszerek válaszsideje, csökkenjen a szükséges sávszélesség, a szükséges adatkommunikáció mennyisége, vagy épp a központi szerverek erőforrásigénye. [20]

Megoszlanak a vélemények, hogy az edge computing körébe tartozik-e az az alkalmazási területe, mikor csupán a tárolási kapacitást hozzuk közelebb a felhasználás helyéhez.

Erre példa a 90-es évek vége felé megjelenő CDN-ek (Content Delivery Network), melyek az egyre növekvő mennyiségű és üzletileg egyre fontosabb adatok globális kiszolgálását végezték, illetve végzik most is. Ezek az adatok jellemzően statikus média tartalmak voltak: képek, videók, egyéb statikus webes tartalmak. Az alapelgondolás az volt, hogy a felhasználók által fogyasztandó ritkán változó, de nagy sávszélességet igénylő tartalmat, nem egy távoli központi szerverről szolgálják ki, hanem a felhasználó infrastruktúrájához közel létesített CDN szerverről. Ezek a szerverek egyfajta cache-ként működtek, főleg azokat a tartalmakat tárolták, melyek a hozzá tartozó felhasználók számára szükségesek voltak. Ezzel nemcsak az adatkommunikációt gyorsították, hanem az olykor jelentős távközlési költségeken is faragtak. [21]

A 2000-es évek elejére a webes alkalmazásokkal kapcsolatos elvárások megváltoztak. A weboldalak sokkal inkább dinamikus felépítésűek lettek. A felhasználói interakciók száma, melyeket már nem lehetett statikus tartalommal kiszolgálni megnőtt. Ez azt eredményezte, hogy az edge computing fejlődésével nem csak a statikus tartalmat tárolták közel a felhasználóhoz, hanem a dinamikus tartalomért felelős üzleti logikát is. Az így létrejövő distributed application service-ek, már a számítási kapacitást is közelebb vitték a felhasználás helyéhez. (Davis, 2004)

Az edge computing-nak a következő nagy lökést az IoT megjelenése adta. Az egyes végponti eszközökben lévő szenzorok, aktuátorok, vagy maguk a processzorok folyamatosan kommunikálnak a külvilággal. Ez a kommunikáció jelenthet nagyon kevés adatot pl. egy hőmérsékletmérő szenzor esetében, és igényelhet jelentős sávszélességet pl. egy videokamera esetén. [22]

Ez az adatfolyam általában valamilyen feldolgozást igényel, mielőtt tárolásra kerül vagy egyéb üzleti logika fut le a hatására. Ez a feldolgozás lehet egyszerű adattömörítés, szűrés vagy más komplexebb feladat.

A hálózatba kötött IoT eszközök száma egyre nő, az adatcenterek, illetve egyéb szerverek felé irányuló adatfolyam mind nagyobb sávszélességet igényel, ráadásul a feldolgozásuk is (a növekvő üzleti elvárások miatt) egyre erőforrásigényesebb. Ez már kisebb hálózatok esetén is könnyen kapacitásproblémákhoz vezethet, de az edge computing megoldást kínálja erre a problémára.

A fajlagos számítási kapacitás egyre olcsóbb, így nemcsak a nagy szerverközpontok teljesítménye nőhet, hanem a végpontok is egyre intelligensebbek lehetnek. Az intelligens végpontok sok olyan előfeldolgozást el tudnak végezni, mellyel mind a hálózati infrastruktúrát, mind a központi gépeket tehermentesíteni tudják. Itt a hangsúly gyakran a hálózati kapcsolaton van. Egy videokamera például a nyers videó stream helyett tömörített streamet is küldhet, sőt megoldható, hogy csak akkor küldjön képet, ha mozgást is érzékelt. Ezzel már jelentős adatmennyiséget tud megtakarítani, de még tovább fejleszthető, hogy csak akkor küldjön képi információt, ha nem ismerte fel a látószögébe került személyt vagy tárgyat. Ez utóbbi már komolyabb computer vision (gépi látás) feladat, melyet a mai eszközök mesterséges neurális hálózattal végeznek. [20]

Az Intel Corporation hardver tervező és gyártó vállalat természetesen látja az edge computing kihívásait és saját termékpalettáját is ehhez igazította. Ennek egyik eleme a

Computer Vision területére fejlesztett Movidius VPU egysége, mely a gépi látást igénylő feladatokat segíti edge eszközökben.

## AZ INTEL NEURAL COMPUTE STICK 2 HASZNÁLATA EGY EDGE COMPUTING DEVICE ESZKÖZBEN

A Movidius VPU működésének vizsgálatához egy népszerű egykártyás számítógépet a Raspberry Pi-t választottunk. A miniszámítógép teljesítményét, hardveres interfészeit és fogyasztását tekintve megfelel egy IoT eszköznek.

Környezet, számítógép	USB feszültség	Energiafogyasztás	Áram	USB feszültség	Energiafogyasztás	Áram
	idle <sup>3</sup>			peak <sup>4</sup>		
Raspberry Pi3	5,040V	0,594W	118mA	4,943V	3,041W	615mA
MacMini 2011	5,073V	0,613W	121mA	4,917V	3,136W	637mA

2. Táblázat: A fejlesztés során használt eszközök fontosabb paraméterei

Az egykártyás számítógép (single-board computer – SBC) modelljei közül a Raspberry Pi 3B vett részt a vizsgálatunkban, amely az alábbi főbb paraméterekkel rendelkezik:

<b>Modell</b>	Raspberry Pi 3B
<b>Main chip</b>	Broadcom BCM2837 - ARM core (64 bit) - VideoCore IV (32 bit)
<b>Memória</b>	1024MB SDRAM (shared with GPU)
<b>USB interfészek</b>	4x USB 2.0
<b>Wifi (internal)</b>	b/g/n single band 2.4 GHz
<b>Háttértár</b>	Micro SD csatlakozó
<b>Tápellátás</b>	5V DC power input
<b>Operációs rendszer</b>	Raspbian GNU/Linux 10 (buster) (release date: 2020-02-13)

3. Táblázat: A Raspberry Pi 3B főbb paraméterei (forrás: <https://www.raspberrypi.org/products/raspberry-pi-3-model-b>)

A Movidius VPU-t tartalmazó kártyák közül az Intel saját termékét választottuk, az Intel Neural Compute Stick 2-t (NCS2), mely az alábbi főbb paraméterekkel rendelkezik:

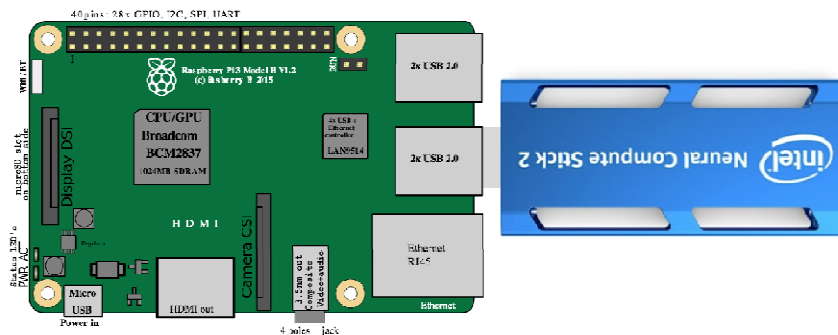
<b>Modell</b>	Intel Neural Compute Stick 2
<b>VPU</b>	Intel® Movidius™ Myriad™ X

<sup>3</sup> Nyugalmi (alap)állapot

<sup>4</sup> Performancia tesztekénél mérték értékek

<b>VPU nums</b>	1x
<b>Processor number</b>	MA2485
<b>Memory</b>	4Gbit DDR3 (1600 MHz)
<b>USB interfész</b>	USB 3.1
<b>Model data type</b>	half-precision floating point (FP16)

4. Táblázat: Az Intel Neural Compute Stick 2 főbb paraméterei (forrás: intel.com)



8. Ábra: A Raspberry Pi<sup>5</sup> és az Intel Neural Compute Stick 2<sup>6</sup> konfigurációja

Az NCS2 kipróbálásához összerakott infrastruktúrámban kevés elemből állt. A Raspberry Pi-hez csak az NCS2 csatlakozott USB interfész segítségével, valamint egy 2500 mA-es tápegység az erre a célra kialakított mikro USB csatlakozón keresztül. Beviteli eszközöket, illetve monitort külön nem csatlakoztattunk, minden adatforgalom a beépített WiFi csatolón keresztül történt. A Linux konzolt SSH kapcsolaton, a grafikus felületet pedig VNC<sup>7</sup>-n keresztül értük el. A mintaalkalmazások fejlesztéséhez és kipróbálásához az OpenVINO Toolkit fejlesztőeszköz 2020.1<sup>8</sup>-es verzióját használtuk. Ez a kísérleti/fejlesztési időszakban elérhető legfrissebb verzió volt.

## MESTERSÉGES INTELLIGENCIA, GÉPI TANULÁS, MÉLYTANULÁS

Az utóbbi években a mesterséges intelligencia (MI), különösen annak bizonyos részterületei sok új alkalmazásban, valamint korábban szinte elképzelhetetlen eszközökben (lásd: Edge Computing) jelentek meg. Célszerű pontosítani az ezzel kapcsolatos fogalmakat, hogy segítsük a téma megértését akkor is, mikor a kifejezéseket (gyakran hibásan) egymás helyettesítésére használják. A fogalmak definiáláskor feltüntettük azok angol elnevezését is, mert jellemzően még a magyar szakirodalomban is az angol terminológiával találkozhatunk.

<sup>5</sup> A kép forrása: [https://upload.wikimedia.org/wikipedia/commons/e/e4/RaspberryPi\\_3B.svg](https://upload.wikimedia.org/wikipedia/commons/e/e4/RaspberryPi_3B.svg)

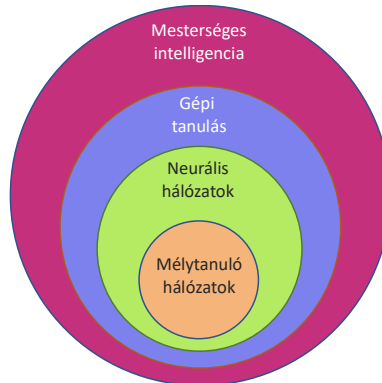
<sup>6</sup> A kép forrása: <https://www.engadget.com/2018-11-14-intel-neural-compute-stick-2.html>

<sup>7</sup> Virtual Network Computing (VNC) – egy grafikus képernyőmegosztó rendszer, mely hálózati kapcsolaton keresztül továbbítja a kliens felől érkező egér és billentyűzet eseményeket, valamint a szerver felől küldött grafikus képernyőmódosításokat.

<sup>8</sup> Telepítési utasítás elérhetősége: [https://docs.openvino toolkit.org/2020.1/\\_docs\\_install\\_guides\\_installing\\_openvino\\_raspbian.html](https://docs.openvino toolkit.org/2020.1/_docs_install_guides_installing_openvino_raspbian.html)

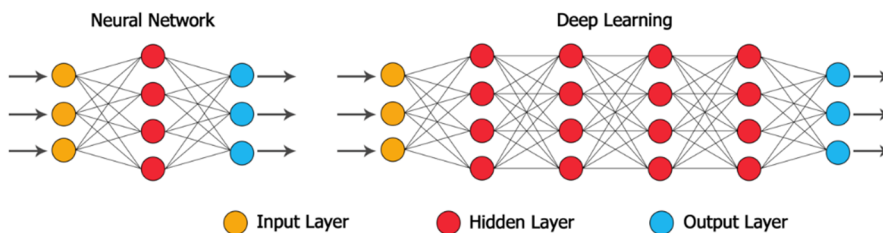
A mesterséges intelligencia (artificial intelligence/AI) fogalmának egy lehetséges definíciója szerint, a mesterséges intelligencia egy számítógép vagy számítógép vezérlésű rendszer, mely olyan feladatokat hajt végre, ami korábban csak humán intelligenciával volt lehetséges. (Copeland, 2020) Egy másik értelmezés szerint ez annak a tudománya, ami azt kutatja, hogyan lehet olyan intelligens gépeket vagy programokat készíteni, melyek eddig az embernek tulajdonított módon oldanak meg problémákat. [23]

A gépi tanulás (machine learning/ML) a mesterséges intelligencia egy részterülete, ami olyan rendszerek fejlesztésével foglalkozik, melyek automatikusan, tapasztalatokból tudnak tudást felhalmozni.



9. Ábra: Az egyes fogalmak kapcsolódása (saját szerkesztés)

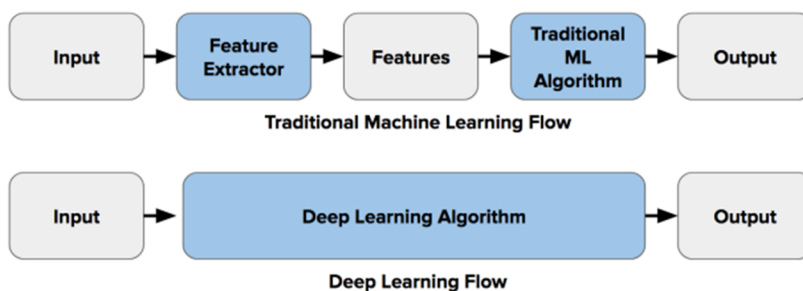
A neurális hálózatok (neural networks/NN) olyan nagy számú, hasonló működésű mesterséges neuronokból álló hálózatok, amelyek egyfajta elosztott működésű információfeldolgozást végeznek.



10. Ábra: Különbség a neurális hálózat és a mélytanuló neurális hálózat között (forrás: <https://alphabold.com/neural-networks-and-deep-learning-an-overview/>)

A mély tanuló hálózatok (deep neural networks/DNN) olyan mesterséges neurális hálózatok, melyek egyrészt több rejtett réteggel (hidden layers) rendelkeznek, mint a hagyományos neurális hálózatok. Másrészt nincs szükségük „manuális” feature extraction-re, mert ezt a munkát elvégzik a rétegek (hidden layers). Egyik legismertebb mélytanuló hálózat a konvolúciós neurális hálózat (Convolutional Neural Networks/CNN). [24]





11. Ábra: Különbség a mélytanulás és a hagyományos gépi tanulás között (forrás: Haritha Thilakarathne<sup>9</sup>)

## ZÁRÓ GONDOLATOK

Tanulmányunk első – elméleti – részében a téma fogalmi keretét tekintettük át, valamint bemutattuk azokat az eszközöket, amelyek segítségével a konkrét – tanulmányunk következő részében ismertetésre kerülő – gyakorlati feladatot végeztük el. Ebben a második részében írunk továbbá a gépi látás neurális hálózatokkal történő támogatásáról is, illetve bemutatjuk a fontosabb keretrendszereket is.

## FELHASZNÁLT FORRÁSOK

- [1] Cs. Kollár, „Az okos város és az okos vidék szimbiózisa: Utópia, fikció, vagy realitás?,” Kecskemét, 2019.
- [2] L. Gábor, „Intelligencia az IP-alapú videorendszerekben,” Detektor Plusz, pp. 26-27, vol. 4., 2013.
- [3] „Computer Vision vs. Machine Vision – What’s the Difference?,” Appen.com, [Online]. <https://appen.com/blog/computer-vision-vs-machine-vision/>. [Hozzáférés dátuma: 2020.04.28.].
- [4] „MACHINE VISION VS. COMPUTER VISION,” Intec Automation Inc., 2019. [Online]. <https://www.intecautomation.com/blog/machine-vs-computer-vision/>. [Hozzáférés dátuma: 2020.04.25.].
- [5] „Computer Vision vs. Machine Vision,” AIA Weboldal, [Online]. [https://www.visiononline.org/vision-resources-details.cfm/vision-resources/Computer-Vision-vs-Machine-Vision/content\\_id/4585](https://www.visiononline.org/vision-resources-details.cfm/vision-resources/Computer-Vision-vs-Machine-Vision/content_id/4585). [Hozzáférés dátuma: 2002.04.28.].
- [6] H. Gomes, „Marr’s Theory: From primal sketch to 3-D models,” 2000. [Online]. [http://homepages.inf.ed.ac.uk/rbf/CVonline/LOCAL\\_COPIES/GOMES1/marr.html](http://homepages.inf.ed.ac.uk/rbf/CVonline/LOCAL_COPIES/GOMES1/marr.html). [Hozzáférés dátuma: 2020.04.29.].
- [7] R. Demush, „A Brief History of Computer Vision (and Convolutional Neural Networks,” [Online]. <https://hackernoon.com/a-brief-history-of-computer-vision-and-convolutional-neural-networks-8fe8aac79f3>. [Hozzáférés dátuma: 2020.05.01.].
- [8] G. Horváth, „Neurális hálózatok”, Budapest: Panem Könyvkiadó Kft., 2006.
- [9] G. Horváth, „Neurális hálózatok és műszaki alkalmazásaik”, Budapest, 1998.

<sup>9</sup> Webes elérhetőség: <https://naadispeaks.wordpress.com/2018/08/12/deep-learning-vs-traditional-computer-vision>

- [10] J. J. Hopfield, „*Neural Networks and Physical Systems with Emergent Collective Computational Abilities*,” 1982. [Online]. [https://www.researchgate.net/publication/16246447 Neural Networks and Physical Systems with Emergent Collective Computational Abilities](https://www.researchgate.net/publication/16246447_Neural_Networks_and_Physical_Systems_with_Emergent_Collective_Computational_Abilities). [Hozzáférés dátuma: 2020.04.14.]
- [11] K. Strachnyi, „*Brief History of Neural Networks*,” 2019. [Online]. <https://medium.com/analytics-vidhya/brief-history-of-neural-networks-44c2bf72eec>. [Hozzáférés dátuma: 2020.04.12.]
- [12] Z. Luo, 10 szeptember 2017. [Online]. <https://luozm.github.io/cv-tasks>.
- [13] H. Gao, 27 augusztus 2017. [Online]. <https://towardsdatascience.com/object-localization-in-overfeat-5bb2f7328b62>.
- [14] P. Ganesh, 12 augusztus 2019. [Online]. <https://towardsdatascience.com/object-detection-simplified-e07aa3830954>.
- [15] A. Rohan, „*Convolutional implementation of the sliding window algorithm*,” 2020. [Online]. <https://medium.com/ai-quest/convolutional-implementation-of-the-sliding-window-algorithm-db93a49f99a0>. [Hozzáférés dátuma: 2020.05.30.]
- [16] J. Browniee, „*Machine Learning Mastery*,” 5 július 2019. [Online]. <https://machinelearningmastery.com/object-recognition-with-deep-learning/>.
- [17] P. Alcorn, „*Intel Unveils Movidius Myriad X Vision Processing Unit*,” 2017. [Online]. <https://www.tomshardware.com/news/intel-movidius-vpu-ai-inference,35327.html>. [Hozzáférés dátuma: 2020.04.25.]
- [18] „*Release Notes for Intel® Distribution of OpenVINO™ toolkit*,” Intel Corporation, [Online]. <https://software.intel.com/content/www/us/en/develop/articles/openvino-relnotes.html>. [Hozzáférés dátuma: 2020.04.15.]
- [19] „*Supported Devices webpage*,” Intel Corporation, [Online]. [https://docs.openvino-toolkit.org/2020.1/\\_docs\\_IE\\_DG\\_supported\\_plugins\\_Supported\\_Devices.html](https://docs.openvino-toolkit.org/2020.1/_docs_IE_DG_supported_plugins_Supported_Devices.html). [Hozzáférés dátuma: 2020.05.27.]
- [20] K. Shaw, „*What is edge computing and why it matters*,” 2019. [Online]. <https://www.networkworld.com/article/3224893/what-is-edge-computing-and-how-it-s-changing-the-network.html>. [Hozzáférés dátuma: 2020.05.05.]
- [21] „*The Debate in a Nutshell: CDN vs Edge Computing*,” BelugaCDN, [Online]. <https://www.belugacdn.com/cdn-vs-edge-computing>. [Hozzáférés dátuma: 2020.05.05.]
- [22] L. Moné, „*IoT Devices, Sensors, and Actuators Explained*,” 2020. [Online]. <https://www.leanix.net/en/blog/iot-devices-sensors-and-actuators-explained>. [Hozzáférés dátuma: 2020.05.08.]
- [23] „*Artificial Intelligence vs. Machine Learning vs. Deep Learning: What’s the Difference*, *Medium online magazin*,” Serokell, 2020. [Online]. <https://medium.com/ai-in-plain-english/artificial-intelligence-vs-machine-learning-vs-deep-learning-whats-the-difference-dcce18efe7f>. [Hozzáférés dátuma: 2020.05.25.]
- [24] H. Thilakarathne, „*Deep Learning Vs. Traditional Computer Vision*,” 2018. [Online]. <https://naadispeaks.wordpress.com/2018/08/12/deep-learning-vs-traditional-computer-vision/>. [Hozzáférés dátuma: 2020.05.14.]