

Mesterséges intelligencia-alapú gépi tanuláson alapuló adatosztályozási technikák összehasonlítása tengeri emlősök víz alatti hangjának elemzésénél

Comparison of data classification techniques based on artificial intelligence-based machine learning in the analysis of the underwater sound of marine mammals

ARADI Attila¹, Dr. VARGA Attila Károly², Dr. TAKÁCS Péter³

¹ ION-technik Kft., Miskolci Egyetem Aradi.Atila@gmail.com

² Miskolci Egyetem, Attila.Varga@uni-miskolc.hu

³ HUN-REN Balatoni Limnológiai Kutatóintézet, ION-technik Kft, takacs.peter@blki.hu

Abstract

In this paper, we investigate and compare different machine learning techniques for classifying underwater sounds, with a focus on shipping noise and marine mammal sounds such as dolphins and cetaceans. The AI-based neural network model can be used to automate the processing of recordings, significantly reducing the possibility of human error. A comparison of different machine learning methods has shown that AI-based approaches are effective tools for identifying cetaceans and dolphins, as well as for recognising the types of vessels that need to be distinguished from background noise. The aim of this research is to demonstrate the effectiveness and applicability of these techniques for acoustic analysis of the marine environment.

Keywords: hydrophone, bioacoustics, machine learning, artificial intelligence, underwater noise, acoustic communication of cetaceans, underwater sound classification, neural networks

Kivonat

Ez a tanulmány különféle gépi tanulási technikákat vizsgál és hasonlít össze a víz alatti hangok osztályozásában, különös tekintettel a hajózási zajokra és tengeri emlősök, például delfinek és cetfélék hangjaira. Az AI-alapú gépi tanulási modellek segítségével automatizálható a hangfelvételek feldolgozása, ami jelentősen csökkenti az emberi hibák lehetőségét. A különböző gépi tanulási módszerek összehasonlítása rámutatott, hogy az AI-alapú megközelítések hatékony eszközök a cetfélék és delfinek azonosításában, valamint a hajótípusok felismerésében, amelyeket a háttérzajok közül kell elkülöníteni. A kutatás célja, hogy bemutassa ezeknek a technikáknak a hatékonyságát és alkalmazhatóságát a vízalatti környezet akusztikai elemzésében.

Kulcsszavak: hidrofon, bioakusztika, gépi tanulás, mesterséges intelligencia, víz alatti zaj, cetfélék akusztikus kommunikációja, víz alatti hangok osztályozása, neurális hálózat (NN)

1. BEVEZETÉS

Az tengerek összetett élőhelyek, ahol a cetfélék, mint a bálnák és a delfinek, alapvetően a hangokra támaszkodnak kommunikáció, tájékozódás és táplálkozás során. [6] Az emberi tevékenységek, például a turizmus és a hajózás, által okozott víz alatti zajok azonban egyre inkább megzavarják ezt a környezetet, ami veszélyezteti a tengeri élővilágot, különösen a cetféléket. [2] A cetfélék hangjai, melyek fontosak a társadalmi interakciók és a túlélés szempontjából, hosszú távolságokat tesznek meg a víz alatt, és ezek tanulmányozása segít a kutatóknak jobban megérteni viselkedésüket és reakcióikat a környezeti változásokra. [25] Az ember által keltett zajok növekedése miatt egyre fontosabbá válik a zaj hatásainak vizsgálata, ami elengedhetetlen a cetfélék populációjának megőrzése és a tengeri ökoszisztémák védelme érdekében. [1] [3]

2. ANYAGOK ÉS MÓDSZEREK

A konvolúciós neurális hálózatok (CNN-ek) a mély neurális hálózatok kategóriájába tartoznak, amelyek a képfelismerési és osztályozási feladatokban nyújtott kivételes teljesítményükről ismertek, és döntő szerepet játszanak a számítógépes látás fejlődésében.[10] A CNN-ek eredetileg az állatok látókérgének felépítése alapján modellezett egyedi architektúrával rendelkeznek, amely lehetővé teszi számukra, hogy a bemeneti képekből önállóan és dinamikusan tanuljanak térbeli mintákat.[4] A CNN felépítése elsősorban konvolúciós rétegekből, pooling rétegekből és teljesen összekapcsolt rétegekből áll, amelyek mindegyike különálló célt szolgál a hálózatban. [14] A konvolúciós rétegek, a CNN alapvető összetevői, konvolúciós műveleteket hajtanak végre a bemeneti adatok szűrésére, kivonva a minták, például élek, textúrák és alakzatok felismeréséhez elengedhetetlen jellemzőket [13]. Ezek a rétegek megtanulható szűrőket alkalmaznak a bemeneti adatokra, jellemzőterképek készleteit létrehozva. [14] A konvolúciós rétegeket követően a pooling rétegek csökkentik a bemeneti térfogat térbeli dimenzióit a következő konvolúciós rétegek számára, az olyan technikák, mint a max pooling és az átlagos pooling segítik a jellemzők absztrakcióját a túllillesztés és a számítási terhelés ellenőrzése érdekében. [9] A hálózat végén a teljesen összekapcsolt rétegek egyesítik a konvolúciós és a pooling rétegekből szerzett magas szintű jellemzőket, hogy megkönnyítsék az osztályozást, hasonlóan a hagyományos többrétegű perceptron (MLP) struktúrákhoz. [11] A CNN tervezése aprólékos tervezést igényel a konvolúciós rétegeken belüli szűrők mennyiségét és méretét, a pooling rétegekben a pooling ablak méretét és a hálózat mélységét illetően, mivel ezek a paraméterek kulcsfontosságú szerepet játszanak a hálózat hatékonyságának és számítási hatékonyságának meghatározásában [7] [21] [8].[12].

A CNN-ek számos előnyt kínálnak:

- Automatikus jellemző-kivonatolás: A hagyományos algoritmusokkal ellentétben, amelyek manuális jellemzőkivonást és kiválasztást igényelnek, a CNN-ek automatikusan tanulnak jellemzőket közvetlenül a képekből, ami a képelemzés robusztusabb és skálázhatóbb megközelítését eredményezi. [14]

- Hierarchikus jellemzőtanulás: A CNN-ek hierarchikus jellemzőreprezentációkat tanulnak, a korai rétegek alapvető jellemzőket, például éleket és textúrákat rögzítenek, a mélyebb rétegek pedig összetettebb mintákat ismernek fel. [10]

- A fordítási invariancia: Ez azt jelenti, hogy a CNN-ek képesek felismerni a képen lévő objektumokat, függetlenül azok helyzetétől. [22]

- Hatékonyság: A konvolúciós rétegekben a közös súlyok csökkentik a paraméterek számát, így a CNN-ek számítási szempontból hatékonyak a hasonló mélységű, teljesen összekapcsolt hálózatokhoz képest. [16]

Erősségeik ellenére a CNN-ek korlátokkal is szembesülnek:

- Magas számítási költségek: A CNN-ek, különösen a mély CNN-ek képzése jelentős számítási erőforrásokat és időt igényel, ami gyakran GPU-k használatát teszi szükségessé a gyakorlati képzési időkhöz. [21]

- Túllillesztés: A CNN-ek könnyen túllilleszkedhetnek a képzési adatokhoz, különösen akkor, ha a képzési példák száma korlátozott.[25]

- Rácsszerű adatokra korlátozva: A szabványos CNN-eket elsősorban rácsszerű topológiájú adatok (pl. képek) feldolgozására tervezték. Ez a korlátozás kevésbé teszi őket alkalmassá más típusú adatok, például gráfok és szabálytalanul strukturált adatok kezelésére.

- Interpretálhatóság: A CNN-ek konkrét döntések meghozatalának megértése továbbra is kihívást jelent, mivel az általuk megtanult jellemzők nagyon absztraktak lehetnek, és az ember számára nem könnyen értelmezhetőek.

Összefoglalva, bár a CNN-ek nagy teljesítményű eszközt jelentenek a képelemzéshez és -osztályozáshoz, tervezésük, megvalósításuk és alkalmazásuk során gondosan figyelembe kell venni előnyeiket és korlátjaikat. A területen folyó kutatások célja, hogy megoldják ezeket a kihívásokat, hatékonyabbá, sokoldalúbbá és értelmezhetőbbé téve a CNN-eket.

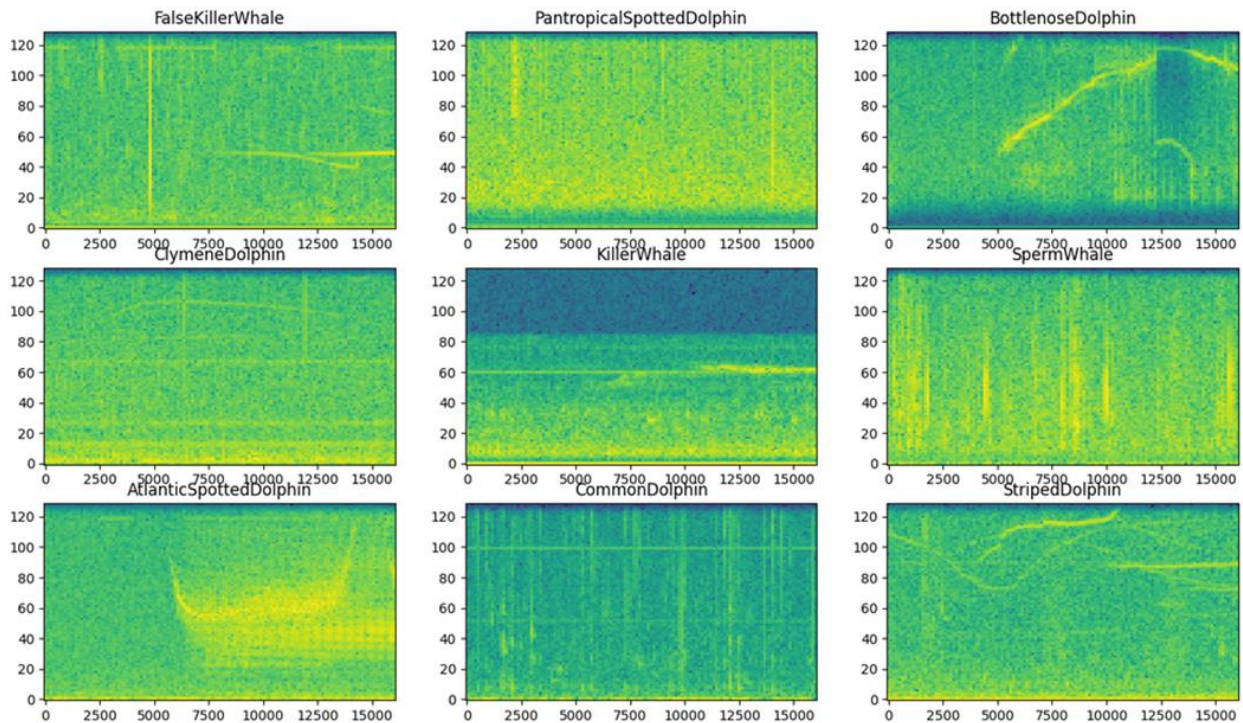
A CNN képzésében részt vevő fajok a 9. ábrát mutatják.

```
[
'Short_Finned(Pacific)PilotWhale' 'Beluga_WhiteWhale' 'AtlanticSpottedDolphin' 'MelonHeadedWhale'
'Walrus' 'White_sidedDolphin' 'Fin_FinbackWhale' 'BowheadWhale' 'FalseKillerWhale' 'Long_FinnedPilotWhale'
'White_beakedDolphin' 'CommonDolphin' 'HumpbackWhale' 'PantropicalSpottedDolphin' 'BottlenoseDolphin'
'StripedDolphin' "Fraser'sDolphin" 'NorthernRightWhale' 'KillerWhale' 'ClymeneDolphin'
'Grampus_Risso'sDolphin' 'SpinnerDolphin' 'SpermWhale'
]
```

9. ábra. Az NN képzéséhez és validálásához használt hangadatbázisban felcímkézett fajok tenzora.

A hangklipek időtartama legalább egy másodperc, a mintavételi frekvencia pedig 32 kHz. A kimeneti szekvencia hossza 16000-re van beállítva, így a rövidebb klipek kitöltésével és a hosszabbak levágásával könnyen lehet kötegelni. [22]

Az adatkészlet az időtartományban ábrázolt hullámformákat tartalmazza. Ezen hullámformák időfrekvencia-tartománybeli jelekké történő átalakításához számítsa ki a rövid idejű Fourier-transzformációt (STFT), hogy spektrogramokká alakítsa őket. A spektrogramok a frekvencia időbeli változásait mutatják, és 2D-s képként ábrázolhatók. Ezeket a spektrogram-képeket a neurális hálózatba tápláljuk be a modell képzéséhez. [19][21] A 12. ábra egyes fajok spektrogramjait mutatja.



12. ábra. Normalizált spektrogram példák különböző fajokról. (Y tengely: normalizált frekvencia, X tengely: mintaszám)

Az ebben a tanulmányban használt CNN-modell a hangfájlok spektrogram képekké történő átalakítását használja fel. [21] A modell speciális előfeldolgozó rétegeket tartalmaz, köztük egy átméretező réteget a bemeneti méret csökkentésére a gyorsabb képzés érdekében [14], valamint egy normalizáló réteget a pixelértékek átlag és szórás alapján történő szabványosítására. [18] A normalizáló réteg megvalósításához először az adaptációs módszert kell alkalmazni a képzési adatokra az összesített statisztikák (átlag és szórás) kiszámításához. [17] [23]

A Keras modellt az Adam optimalizáló és a kereszt-entrópia veszteségfüggvény felhasználásával kell jól beállítanunk. [14] A modellt 100 epochán keresztül tanítjuk. A képzési folyamat felgyorsítása érdekében kihasználjuk az Nvidia GeForce RTX 4090 GPU számítási teljesítményét. [16] A képzés során megfigyelt memóiafogyasztás 22174 MiB volt a 24564 MiB-ból, a GPU átlagos kihasználtsága körülbelül 48% volt. [21]

A neurális hálózatok TensorFlow-val vagy bármely más mély tanulási keretrendszerrel történő képzésével összefüggésben az "epoch" a teljes képzési adathalmaz egy teljes átfutását jelenti. Egy epoch alatt a neurális hálózat az adathalmaz minden egyes mintáját pontosan egyszer dolgozza fel és tanulja meg, lehetővé téve számára, hogy frissítse súlyait és torzításait a veszteségfüggvény minimalizálására tett kísérlet során. Az epochok fogalma több okból is kulcsfontosságú a neurális hálózatok képzési folyamatában:

- Tanulási előrehaladás: A modell minden egyes epoch-kal a teljes adathalmazon iterál, és a paraméterek (súlyok és torzítások) beállításával tanul az adatokból. Ez az inkrementális tanulási folyamat idővel gyakran a modell teljesítményének javulásához vezet a képzési adatokon. [22]

- Konvergencia: A modell ismételt alkalmazása a teljes adathalmazra több epochán keresztül lehetővé teszi, hogy a modell tanulási algoritmusai (például a gradiens ereszkedés) konvergáljanak, ami azt jelenti, hogy a modell paramétereinek változásai idővel csökkennek, mivel a modell megtalálja a veszteségfüggvényt minimalizáló optimális vagy közel optimális paraméterkészletet. [23]

- Túlillesztés ellenőrzése: A modell a tényleges jel helyett a képzési adatok zaját tanulja meg, ami gyenge általánosításhoz vezet a nem látott adatokon. Az olyan technikákat, mint a korai leállítás, arra

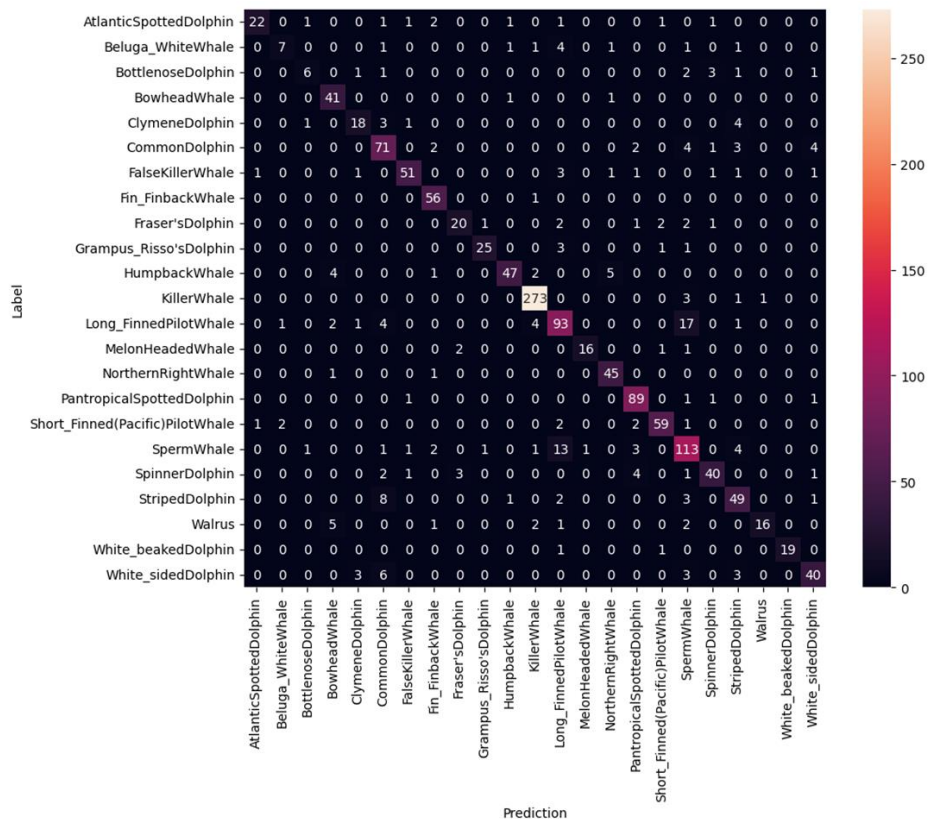
használgják, hogy figyelemmel kísérjék a modell teljesítményét egy validációs halmazon, és leállítsák a képzést, amikor a teljesítmény romlani kezd, ezzel megakadályozva a túlillesztést.

- Hiperparaméter: Az epochok száma a képzési folyamat hiperparamétere, ami azt jelenti, hogy a képzés megkezdése előtt kerül beállításra, és a képzés során állandó marad. Az epochák optimális számának meghatározása kulcsfontosságú a hatékony modellképzéshez, és gyakran kísérletezéssel vagy olyan stratégiák alkalmazásával történik, mint a korai leállítás. A TensorFlow-ban az epochák száma a modell képzési függvényében van megadva, az epochák számának beállítása jelentősen befolyásolhatja a modell tanulási eredményeit, beleértve az új, nem látott adatokra való jó általánosítási képességét. Ebben az esetben a modellt 100 epochán keresztül képezzük.

A korai leállítás egy beszabályozási technika, amelyet a mély tanulási modellek, köztük a TensorFlow-val épített modellek képzésénél használnak a túlillesztés megelőzésére. [15] A technika figyel a modell teljesítményét egy validációs halmazon, és leállítja a képzési folyamatot, ha a modell teljesítménye nem javul tovább vagy romlani kezd, ami azt jelzi, hogy a modell elkezdett túlilleszkedni a képzési adatokhoz. [24] A korai leállítás mindig egy mérőszámot figyel: A korai leállításhoz meg kell adni egy nyomon követendő teljesítménymetriát, például a validálási veszteséget vagy a validálási pontosságot. Ez a metrika annak meghatározására szolgál, hogy javul-e a modell teljesítménye. A Türelem paraméter meghatározza az epochák számát, amelyekkel a képzést folytatni kell, miután a megfigyelt metrika már nem javul. Ha például a türelem értéke 2, akkor a képzés még 2 epochával folytatódik a validálási metrika utolsó javulása után. Ha ezekben az epochákban nem történik javulás a megfigyelt metrikában, a képzés leáll. Opcionálisan a korai leállítás visszaállíthatja a modell súlyait a megfigyelt metrika legjobb értékét mutató epochából, így biztosítva, hogy a modell akkor is megtartja a legjobb tanult paramétereket, ha az optimális pontot követő epochákban túlilleszkedik. 15. ábra. A képzési folyamatot mutatja 100 epochával, a korai leállítás a 13. epochában történt. [25]

A CNN-modell képzése során elért legjobb pontosság 82,6 % volt. A korai leállítás használata segít hatékonyan megtalálni az optimális epochaszámot a képzéshez, számítási erőforrásokat takarít meg, és javítja a modell általánosítását a képzési adatokhoz való túlillesztés megakadályozásával. [23] A betanított CNN-modell veszteség- és pontossági diagramjait a 16. ábra mutatja.

Az NxN táblázatként ábrázolt zavarmátrix összefoglalja az osztályozó modell által tett pontos és pontatlan előrejelzéseket, lehetővé téve annak értékelését, hogy a modell milyen teljesítményt nyújt a tesztadathalmazon belüli egyes kategóriák osztályozásában. A 17. ábra a CNN-modell zavarmátrixát szemlélteti.



17. ábra. A CNN-modell konfúziós mátrixa.

Kész modellként ezt a CNN-modellt használhatjuk egy ismeretlen hangfájl előrejelzésére. Ez a modell a végtermék, és a gyakorlati munka során használható.

A modell előrejelzési példája a következő 18. ábrán látható.

```

6 export = ExportModel(model)
7 export(tf.constant(str(data_dir/'White_sidedDolphin/7800700U_32000_mono_16bit.wav')))
8
Out[33]: {'predictions': <tf.Tensor: shape=(1, 23), dtype=float32, numpy=
array([[ -6.287754 , -5.243652 , -4.0659657 , -1.377963 , -0.9377544 ,
-0.19001298, -2.1791933 , -7.156084 , -6.1525946 , -5.5500803 ,
-1.9471575 , -1.6961814 ,  0.6640844 , -5.2282 , -4.492587 ,
-4.549725 , -3.4521031 ,  2.4977608 , -6.4903836 ,  2.3285284 ,
-3.245285 , -0.7267522 ,  5.0455155 ]], dtype=float32)>,
'class_ids': <tf.Tensor: shape=(1,), dtype=int64, numpy=array([22])>,
'class_names': <tf.Tensor: shape=(1,), dtype=string, numpy=array([b'White_sidedDolphin'], dtype=object)>}
```

18. ábra. A CNN-modell gyakorlati alkalmazása a fajok előrejelzésére egy ismeretlen hangfájlból.

Ez a tömb tartalmazza az egyes osztályok normalizált valószínűségeit, a lehető legtöbb lehetséges osztályra mutatva.

A TensorFlow modell előrejelzésének kimenete, amelyet általában osztályozási feladatokban látunk. A kimenet három fő összetevőre tagolódik: `predictions`, `class_ids` és `class_names`, amelyeket a következőkben ismertetünk:

-jóslatok: Ez egy (1, 23) alakú tenzor, ami azt jelzi, hogy a modell 23 különböző osztályra vonatkozóan adott előrejelzést egy mintára. E tenzor minden egyes értéke a modell bizalmi pontszámát vagy logitját jelenti az egyes osztályokra vonatkozóan, a Softmax függvény alkalmazása előtt. Például a legmagasabb érték ebben a tenzorban az "5,0455155", ami arra utal, hogy a modell az ennek a pontszámnak megfelelő osztályra vonatkozóan a legmegbízhatóbb.

-class_ids: Ez az (1,) alakú tenzor tartalmazza a minta megjósolt osztályazonosítóját, a "predikciók" tenzor legmagasabb megbízhatósági pontszáma alapján. Ebben az esetben a megjósolt osztályazonosító a `22`, ami azt jelenti, hogy a modell szerint a minta az ehhez az azonosítóhoz tartozó osztályba tartozik.

-class_names: Ez az (1,) alakú, string típusú tenzor tartalmazza a megjósolt osztály ember által olvasható nevét. A kimenet szerint a modell azt jósolja, hogy a minta a `White_sidedDolphin` osztályba tartozik.

Mindezeket összevetve a modell kimenete azt mutatja, hogy az adott bemeneti minta esetében a modell a legnagyobb biztonsággal azt jósolja, hogy a minta a "White_sidedDolphin" osztályba tartozik, ami megfelel a "22" osztályazonosítónak. A logitok mind a 23 osztályra vonatkozóan jelentősen eltérnek, a megjósolt osztály logitja jelentősen magasabb, mint a többi, ami azt mutatja, hogy a modell mennyire bíz a jósolatában. [31]

3. EREDMÉNYEK

A modellek összehasonlításához értékelnünk kell a teljesítményüket. A modellt az adatok 80%-án képezzük ki, és a fennmaradó 20%-on értékeljük, amely a képzés során nem használt tartalékhalmazként szolgál. Minden egyes előre elkészített gyakori és elterjedt gépi tanulási modellt teszteltünk a Sklearn segítségével, és kiválasztunk néhányat a további vizsgálathoz, mivel ezek a modellek gyorsan betaníthatók a mi adathalmazunkon.

A hagyományos gépi tanulási modellek, mint a támogató vektoros gépek (SVM), a k legközelebbi szomszédok (kNN) és a „véletlen erdők” egyedi előnyöket kínálnak a mély neurális hálózatokhoz képest különböző feladatokban. [15] Azonban a hangok osztályozása során még a gyengébben képzett mély neurális hálózatok által elért teljesítményszintet sem érik el. A többrétegű perceptron (MLP) modell jó teljesítményt nyújt, és finoman beállítható a pontosság érdekében, miközben viszonylag gyorsan betanítható. [16]

A többrétegű perceptron (MLP) figyelemre méltó pontosságot mutat a mi adathalmazunkkal; azonban az adathalmaz viszonylag kis számú gyakorló mintája korlátozza a tökéletes betanulást. [20] A konvolúciós neurális hálózatok (CNN) ideálisak a mély tanulással a hangadatok osztályozására. Ennek oka, hogy a hangadatokból kinyert jellemzők, mint például a spektrogramok, erősen hasonlítanak a képekre, egy olyan területre (képfeldolgozás), ahol a CNN-ek kiválóan felismerik és megkülönböztetik a különböző mintákat. [21]

Az MLP (többrétegű perceptron) osztályozó kiemelkedik azzal a képességével, hogy kimagaslóan jó teljesítményű a feltáró elemzésre, a hiperparaméterek optimalizálására és a modell felépítésére fordított kisebb erőforrás igényével. Következésképpen a gépi tanulási modelljeink hatékonyan működnek, lehetővé téve az új hangfájlok osztályozását, beleértve a nagyméretűeket is, amelyekkel a modell korábban nem találkozott. Ez lehetővé teszi a cettfélék hangjainak automatikus felismerését az új hangfájlokban, majd fajuk meghatározását,

megkönnyítve ezzel az osztályozásukat emberi beavatkozás nélkül, nagy mennyiségű és nagy méretű hangfájlok esetében is. [31]

KÖSZÖNETNYILVÁNÍTÁS

Szeretnénk köszönetet nyilvánítani az ION-technik Kft.-nek., az ION Alkalmazott Kutatási NP Kft.-nek és a KDP Kooperatív Doktori Program, Kulturális és Innovációs Minisztérium, Nemzeti Kutatási, Fejlesztési és Innovációs Alapnak.

IRODALMI HIVATKOZÁSOK

- [1] "Soundscapes in the North Adriatic Sea and their impact on marine biological resources," Retrieved from <https://www.italy-croatia.eu/web/soundscape>
- [2] M.V. Valueva és mások, "A maradékszámrendszer alkalmazása a konvolúciós neurális hálózat megvalósításának hardverköltégeinek csökkentésére, A konvolúciós neurális hálózatok ígéretes eszköz a mintafelismerés problémájának megoldására.", *Mathematics and Computers in Simulation*. Elsevier BV. 177: 232-243., 2020.
- [3] Z. Wei, "Shift-invariáns mintafelismerő neurális hálózat és annak optikai architektúrája". *Proceedings of Annual Conference of the Japan Society of Applied Physics.*, 1988.
- [4] Z. Wei, "Párhuzamos elosztott feldolgozási modell helyi térinvariáns összeköttetésekkel és annak optikai architektúrája". *Applied Optics*. 29 (32): 4790-7. 1990.
- [5] A. Mouton és mások, "Mesterséges intelligencia kutatás", *Communications in Computer and Information Science*. Cham: Springer International Publishing. 1342: 267-281., 2020.
- [6] Van Den Oord és mások, "Deep content-based music recommendation.", *Curran Associates, Inc.* pp. 2643-2651., 2013.
- [7] Collobert és mások, "A unified architecture for natural language processing: deep neural networks with multitask learning.", *Proceedings of the 25th international conference on machine learning. ICML '08.* pp. 160-167., 2008.
- [8] Avilov és mások, "Deep Learning Techniques to Improve Intraoperative Awareness Detection from Electroencephalographic Signals,". 2020 *Az IEEE Engineering in Medicine & Biology Society (EMBC) 42. éves nemzetközi konferenciája.* Montreal, QC, Kanada: IEEE. 2020: 142-145.
- [9] Tsantekidis és mások, "Forecasting Stock Prices from the Limit Order Book Using Convolutional Neural Networks,". 2017 *IEEE 19th Conference on Business Informatics (CBI)*. Thessaloniki, Görögország: IEEE: 7-12.
- [10] K. Fukushima, "Neocognitron.", *Scholarpedia*. 2 (1): 1717., 2007.
- [11] K. Fukushima, "Neocognitron: "A Self-organizing Neural Network Model for a Mechanism of Pattern Recognition Unaffected by Shift in Position in Position,". *Biológiai kibernetika*. 36 (4): 193-202., 1980.
- [12] Matusugu és mások, "Subject independent facial expression recognition with robust face detection using a convolutional neural network," *Neural Networks*. 16 (5): 555-559., 2003.
- [13] Mcfee és mások, "librosa: Proceedings of the 14th python in science conference, pp. 18-25. 2015. <https://zenodo.org/badge/latestdoi/6309729>.
- [14] F. Chollet és mások, "keras. gitHub.", Retrieved from <https://github.com/fchollet/keras>., 2015.
- [15] F. Pedregosa és mások, "scikit-learn: gépi tanulás Pythonban. *Journal of Machine Learning Research*, 12(Oct), 2825-2830., 2011.
- [16] M. Abadi, A. Agarwal, P. Barham, E. Brevdo, Z. Chen, C. Citro, G. S. Corrado, A. Davis, J. Dean, M. Devin, S. Ghemawat, I. Goodfellow, A. Harp, G. Irving, M. Isard, R. Jozefowicz, Y. Jia, L. Kaiser, M. Kudlur, J. Levenberg, D. D. Mané, M. Schuster, R. Monga, S. Moore, D. Murray, C. Olah, J. Shlens, B. Steiner, I. Sutskever, K. Talwar, P. Tucker, V. Vanhoucke, V. Vasudevan, F. Viégas, O. Vinyals, P. Warden, M. Wattenberg, M. Wicke, Y. Yu, X. Zheng., "TensorFlow: Large-scale machine learning on heterogeneous systems," 2015. szoftver elérhető a [tensorflow.org](https://www.tensorflow.org) oldalon. <https://doi.org/10.5281/zenodo.4724125>.
- [17] J. D. Hunter, "Matplotlib: Computing in Science & Engineering, vol. 9, no. 3, pp. 90-95, 2007. <https://doi.org/10.5281/zenodo.592536>.
- [18] C.R. Harris és mások, "Array programming with NumPy," *Nature* 585, 357-362 (2020). DOI: 10.1038/s41586-020-2649-2.
- [19] Mckinney, "Data structures for statistical computing in python, " *Proceedings of the 9th Python in Science Conference, Volume 445, 2010.* Pandas. <https://doi.org/10.5281/zenodo.3509134>
- [20] J. Shrey, "Best of Watkins Marine Mammal Sound Database, " Retrieved from <https://www.kaggle.com/datasets/shreyj1729/best-of-watkins-marine-mammal-sound-database>.
- [21] L. Chien-Liang és mások, "Time Series Classification with Multivariate Convolutional Neural Network," *IEEE Transactions on Industrial Electronics*. PP. 1-1. 10.1109/TIE.2018.2864702 2018.
- [22] M. Neralla és mások, "Design and Performance Analysis of Short Time Fourier Transform Processor," *Ijrasnet Journal For Research in Applied Science and Engineering Technology Vol. 10./IV.* 2022