Budapesti Műszaki és Gazdaságtudományi Egyetem
Villamosmérnöki és Informatikai Kar
Méréstechnika és Információs Rendszerek Tanszék

Amer Jusuf

BETEKINTÉS A MÉLY NEURÁLIS HÁLÓK FEKETE DOBOZ TESZTELÉSÉBE

Konzulensek
Marussy Kristóf, Semeráth Oszkár

BUDAPEST, 2023

Tartalomjegyzék

1. Bevezetés

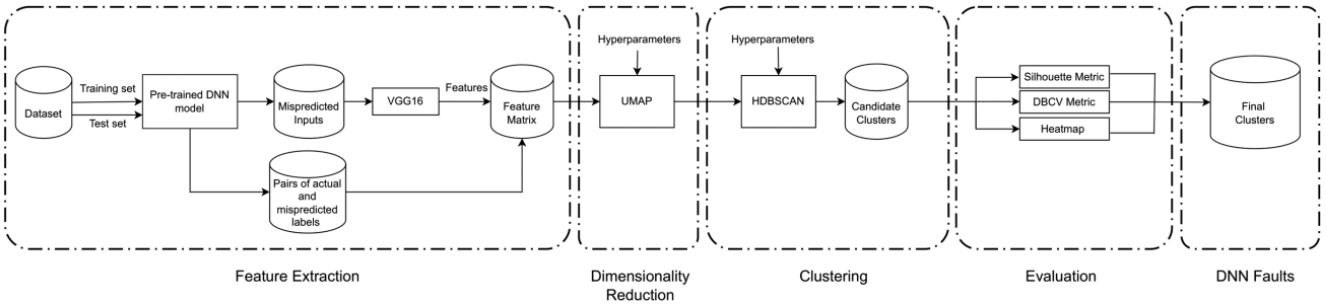
Az autonóm járművek és más kritikus rendszerek fejlesztése során egyre elterjedtebb a mély neurális hálók alkalmazása, például képfelismerés terén. Ezen rendszerek kialakításához nemcsak precíz mérnöki munka és megbízható architektúra szükséges, hanem elengedhetetlen a rendszer alapos tesztelése is.

A mély neurális hálók - azaz olyan neurális hálók, ahol a rejtett rétegek száma megnövelt - tesztelése, kihívást jelent a mérnököknek, a mély neurális hálók komplexitása miatt. A folyamat során, ha sikerül rendszerezni a hibásan tippelt képeket, lehetőség nyílik a neurális háló finomhangolására. Ezt a finomhangolást olyan képekkel végezhetjük, amelyek hasonlóak a hibásan tippelt képekhez, például ha “stop” táblákat nagy százalékban, hibásan ismer fel a mély neurális háló, akkor olyan képekkel lehet érdemes finomhangolni, amiken “stop” tábla látható. Ezáltal a neurális háló tanítása hatékonyabban zajlik, és a rendszer pontossága növekedhet. Ezen felül, az ilyen hibák elemzése segíthet azonosítani, mely rendszerkomponensek érzékenyek a hibás működésre. Ennek eredményeképpen az érintett komponensek köré további biztonsági intézkedéseket vezethetünk be, amelyek hozzájárulnak a kritikus rendszer teljesítményének és megbízhatóságának javításához.

A továbbiakban, a leírt tesztelési módszer elméleti és gyakorlati megközelítését taglalom, aminek alapjául a “*Black-Box Testing of Deep Neural Networks through Test Case Diversity*” [1] című publikációt használtam fel.

2. Irodalomkutatás

2.1. Architektúra



2.1. ábra. Hibák csoportosítása
(Black-Box Testing of Deep Neural Networks through Test Case Diversity [1])

A mély neurális háló (DNN) által, hibásan detektált képek csoportosítására, egy alkalmas architektúrát mutat be a **2.1. ábra**.

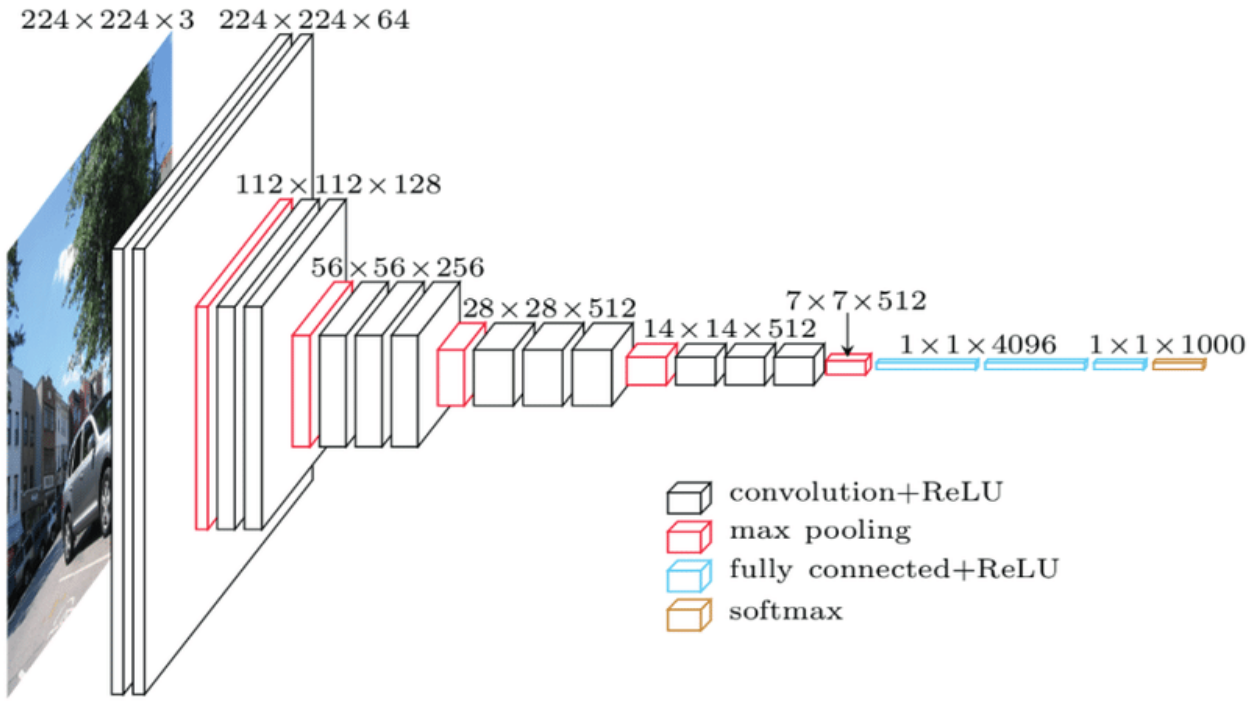
Az architektúra *"Feature Extraction"* fázisában, a DNN által hibásan detektált képekről tárol olyan sokdimenziós vektorokat, amelyek a képek jellemzőinek leírására szolgálnak, mint például élek, színek, textúra stb., ezeknek a vektoroknak, *feature vektorok* a nevük. Az adott architektúrában, a feature vektorok kinyerésének előfeltétele, hogy legyen adott egy DNN, amit tesztelni szeretnénk ("*Pre-trained DNN model*"), illetve egy adathalmaz ("*Dataset*") és az adathalmaz címkéi ("*Pairs of actual and mispredicted labels*"). A hibásan detektált képekből ("*Mispredicted inputs*"), VGG16 segítségével, ki lehet nyerni a feature vektorokat ("*Feature Matrix*").

A sokdimenziós vektorokat UMAP segítségével, amely bonyolult matematikai számítások által, az egymásra *hasonlító* vektorokat a - kétdimenziós síkon - egymáshoz közel, az egymástól *különböző* vektorokat, egymástól távol helyezi el.

A tanulmányom, ezeknek a lépéseknek az elméleti és gyakorlati oldalát mutatja be, nem részletezem a - két dimenzióra redukált - hibásan tippelt képek feature vektorainak a csoportosítását. Habár UMAP dimenzió redukció után is keletkezhetnek hibacsoportok, ezek a hibacsoportok kevésbé informatívak, és gyakran a dataset osztályai szerint csoportosulnak.

2.2. VGG16

A VGG16, egy 14 millió képpel előre betanított mély neurális háló, kifejezetten képfelismerési feladatokra tervezve (CNN).



2.2. ábra. VGG16 architektúra [2]

A VGG16 architektúrát az egyszerűsége és egységes felépítése jellemzi. Összesen 16 réteget tartalmaz, köztük 13 konvolúciós réteget és 3 teljesen összekapcsolt réteget. A konvolúciós rétegek kis, 3x3-as szűrőket használnak, 1-es lépésközzel, és a max-pooling 2x2-es szűrőkkel történik. A kis szűrők használata és több konvolúciós réteg egymásra halmozása segít a hálózatnak abban, hogy hierarchikus jellemzőket tanuljon meg, amelyek fokozatosan növekvő bonyolultságúak. A bemeneti réteg 224*224 pixeles RGB kép.

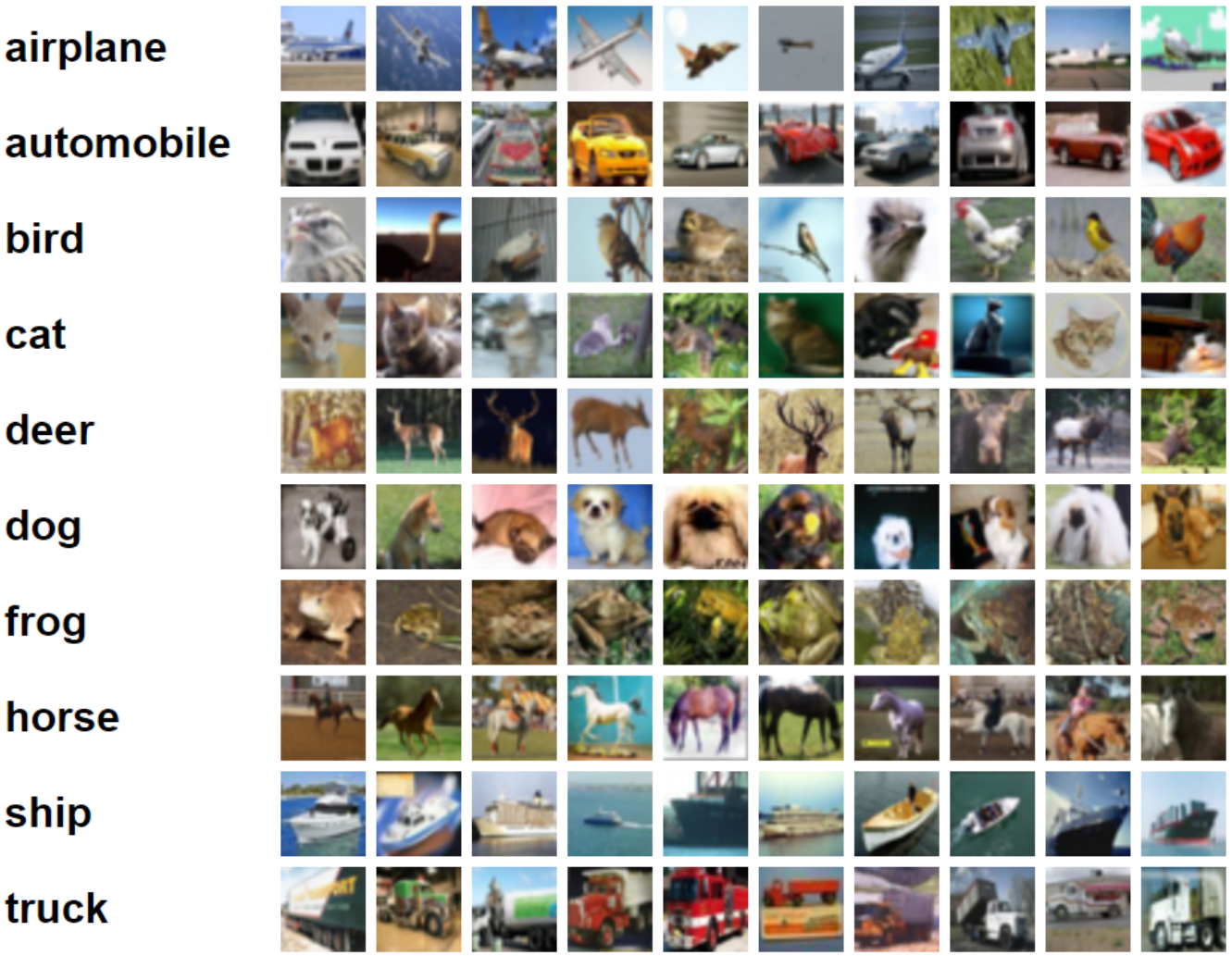
A hibacsoportosító architektúrában, a “fully connected” felső réteg - amely a képek osztályozásáért felelős - le van választva, így a VGG16 felső rétegével leválasztott kimenete egy 1*1*512 dimenziós feature vektor lesz.

2.3. Adathalmazok

2.3.1 CIFAR10 Dataset

A Cifar10 adathalmaz [3] 60 000 képet és 10 különböző osztályt foglal magába. Minden osztályhoz 6000 kép tartozik, és az adathalmaz fel van osztva tanító- és teszthalmazokra. A tanítóhalmaz 50 000 képből áll, a teszthalmaz pedig a fennmaradó 10 000 képet tartalmazza. A képek mérete 32x32 pixel.

A CIFAR-10 adathalmazt gyakran használják gépi tanulási algoritmusok értékelésére és összehasonlítására.



2.3.1. ábra. Cifar10 adathalmaz osztályai [3]

2.3.2 Clevr v1.0 adathalmaz

Az adathalmaz [4] célja, hogy lehetővé tegye a modellek számára, hogy értelmesen megértse a képeken látható elemeket, és képesek legyenek válaszolni összetett, strukturált kérdésekre. Az adathalmaz tartalmaz absztrakt 3D-s jeleneteket, amelyeken színes, szürkeárnyalatos és bináris képek találhatók.

A kérdések a következő típusokra oszthatók:

- Ellenőrzés: A kérdések egy adott tulajdonság ellenőrzésére irányulnak. Például: "Van-e piros kocka?"
- Kvantifikáció: Ezek a kérdések kvantitatív információra vonatkoznak. Például: "Mennyi kék kocka van?"
- Összetett: Két vagy több összetevőt tartalmazó kérdésekre példa: "Melyik kis dolgotól jobbra található a zöld kocka?"

A CLEVR v1.0 adathalmazt arra tervezték, hogy kihívást jelentsen a vizuális érvelés és a kérdés-válasz feladatok számára, és elősegítse a gépi tanulási modellek fejlesztését ezen a területen. A CLEVR kihívásokat kínál az absztrakt és kompozicionális gondolkodás terén a gépi tanulási modellek számára.

2.3.2 ábra. Clevr v1.0 képek [4]

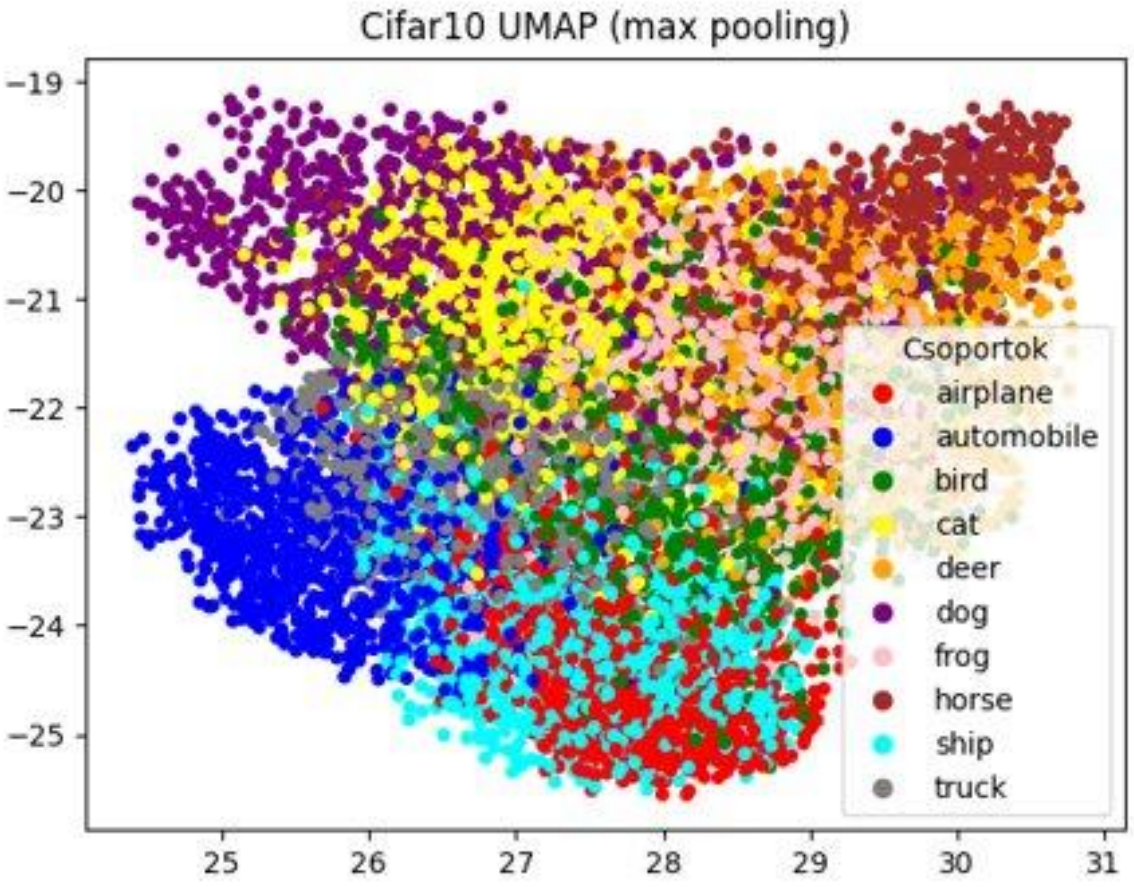
3. Gyakorlati mérés

3.1. Cifar10 adathalmaz feature vektorai

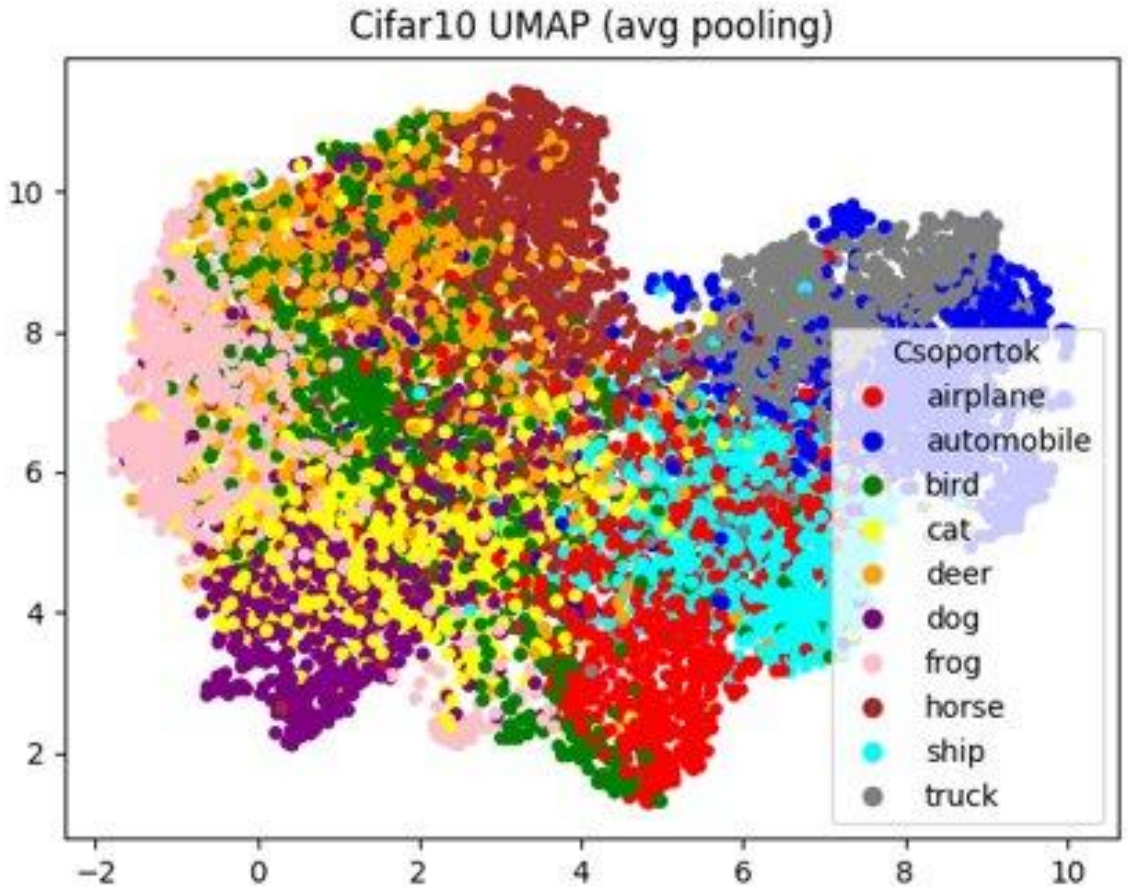
Kezdetben a Cifar10 [3] adathalmaz képeiből nyertem ki feature vektorokat, VGG16 segítségével. Ehhez a Cifar10 32*32 pixeles RGB képeit kellett átméretezni 224*224 pixeles RGB képekre, hogy a VGG16 bemeneti formátumnak megfeleljenek a képek.

Ezt követően, a VGG16 felső rétegeit leszedve - két különböző kinyerési módszerrel (max pooling és average pooling) - megkaptam az 512 dimenziós feature vektorokat, amiket UMAP segítségével 2 dimenziós vektorokra redukáltam.

A két dimenziósra redukált feature vektorok a síkban:



3.1. ábra. Cifar10 kétdimenzióra redukált feature vektorai (max pooling)



3.2. ábra. Cifar10 kétdimenzióra redukált feature vektorai (avarage pooling)

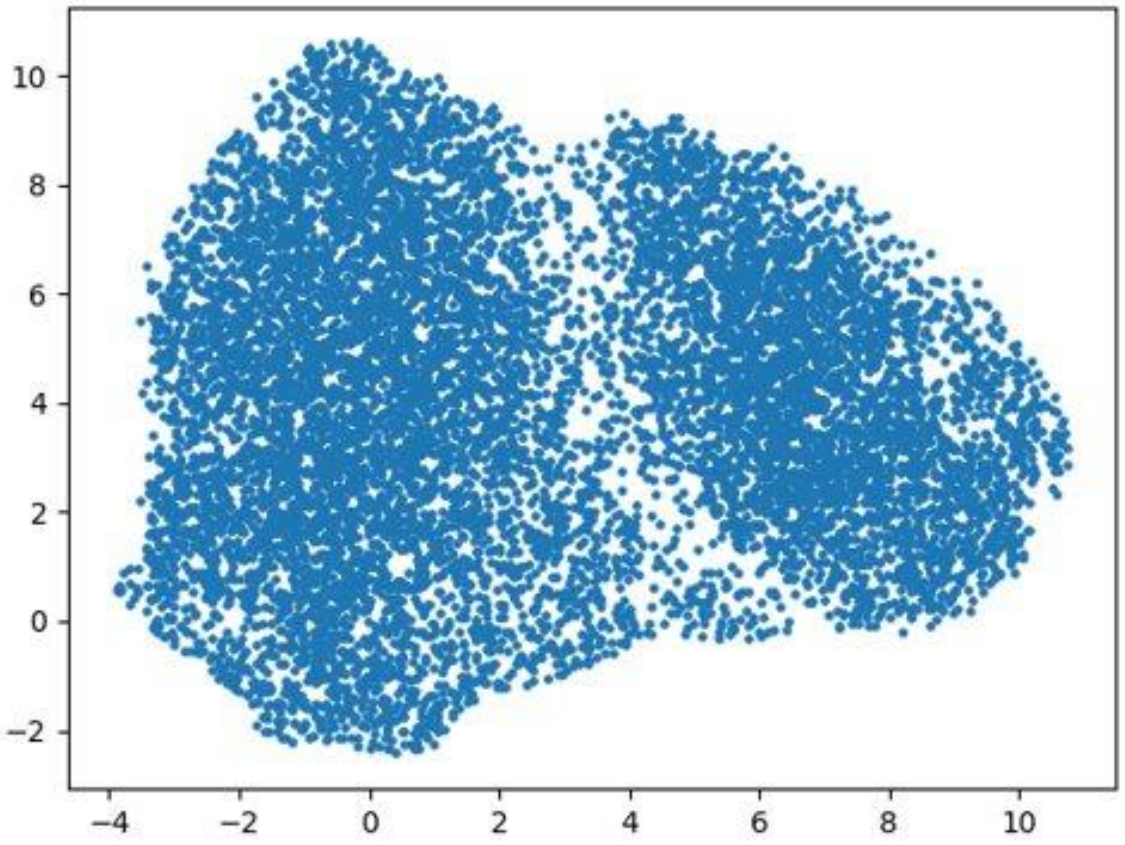
A képeken (**3.2. ábra**, **3.3. ábra.**) az x és y tengely menti számok, nem mértékegységet jelölnek, nem informatívak (az UMAP bonyolult algoritmusai után keletkezett 2 dimenziós vektorok (x,y) kordinátáival vannak kapcsolatban)

3.2. Clevr v1.0 adathalmaz feature vektorai

Miután betekintést nyertem a feature vektorok kinyerésébe, VGG16 segítségével, a Clevr v1.0 adathalmazon megismételtem a folyamatot. Autonóm járművek mély neurális hálói által feldolgozott képekben, nagy szerepet játszanak az objektumok egymáshoz vett relációi. A relációk vizsgálatára, jól felhasználhatóak a Clevr v1.0 [4] adathalmaz képei.

A Clevr v1.0 képekből - a képek előfeldolgozása után - kinyertem a feature vektorokat leválasztott felső rétegű VGG16 segítségével.

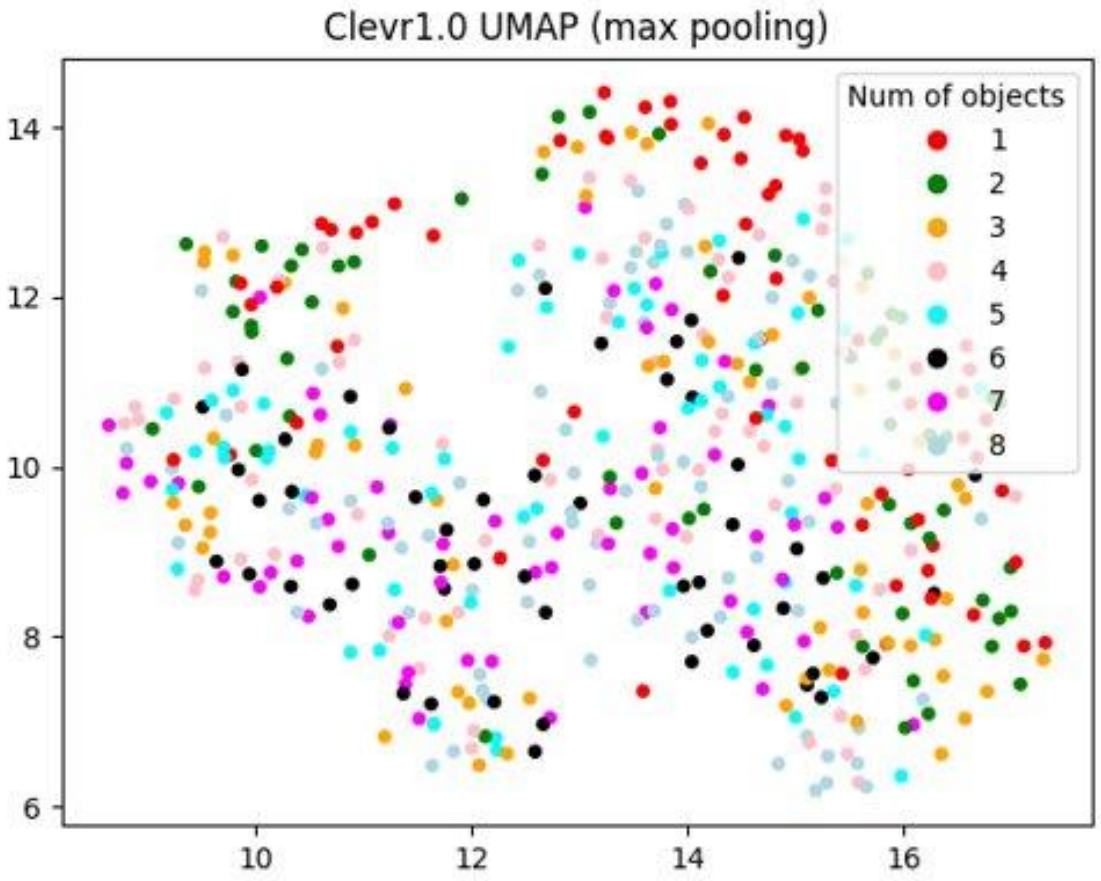
A két dimenziósra redukált feature vektorok a síkban:



3.4. ábra. Clevr v1.0 kétdimenzióra redukált feature vektorai

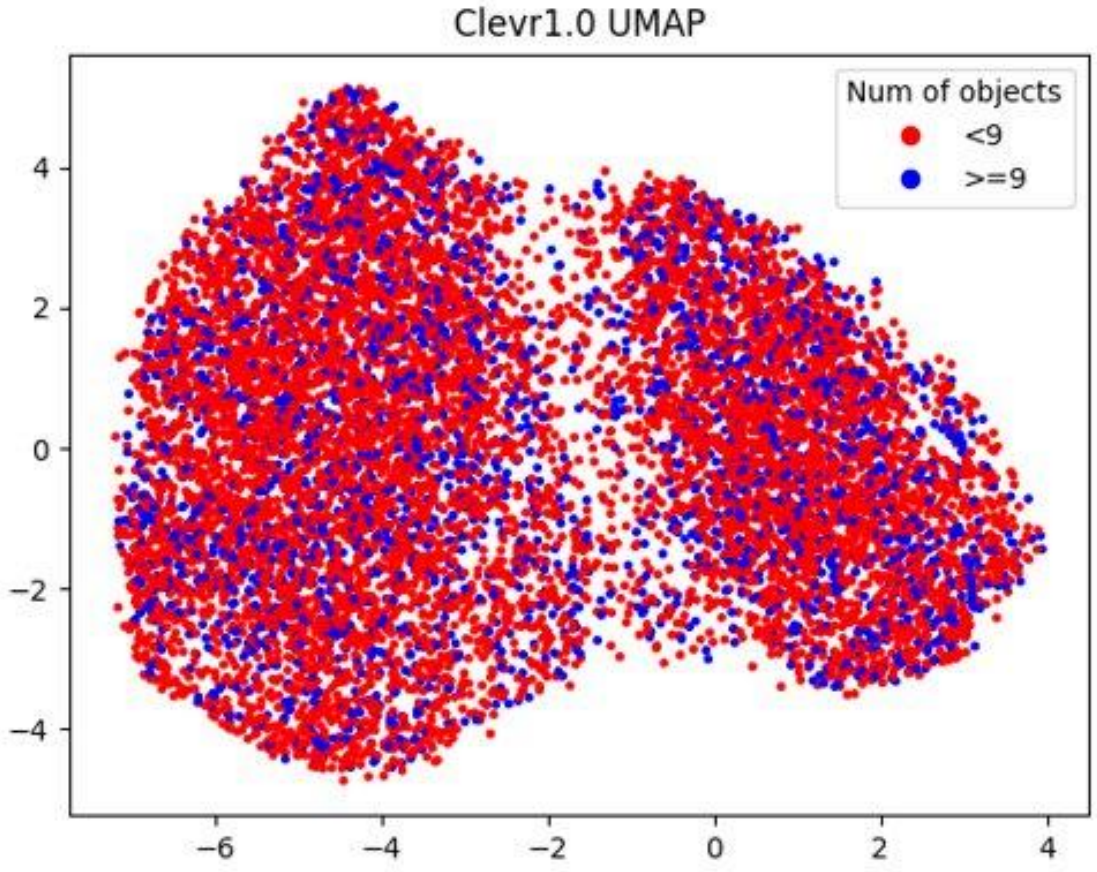
Az ábrán megfigyelhető, hogy a Clevr v1.0 adathalmaz képeinek feature vektorai, két nagyobb osztályba csoportosulnak. A feature vektorok, két dimenziós pontjait a képen látható objektumok száma szerint jelöltem kétféleképpen:

1. Darabszám - Minden számhoz egy szín, 500 képre:



3.4. ábra. Clevr v1.0 kétdimenzióra redukált feature vektorai (Darabszám)

2. “Zsúfoltság” - 9-nél több vagy kevesebb objektum van a képen, 2 szín, 10 000 képre:



3.5. ábra. Clevr v1.0 kétdimenzióra redukált feature vektorai (“Zsúfoltság”)

Az x és y menti számok, az előző méréshez hasonlóan, itt sem informatívak (az UMAP bonyolult algoritmusai után keletkezett 2 dimenziós vektorok (x,y) kordinátáival vannak kapcsolatban)

4. Konklúzió

A bevezetésben bemutatott architektúra feature extraction és dimensionality reducation komponens elméleti és gyakorlati megközelítését mutattam be.

A Cifar10 képek, VGG16 segítségével kinyert feature vektorok, dimenzió csökkentett vektorai csoportosíthatóak.

A Clevr v1.0 képek, VGG16 segítségével kinyert feature vektorok, dimenzió csökkentett vektorai két clusterbe csoportosulnak. Mérésem alapján, ezek a clusterek, függetlenek a képeken látható objektumok számától.

Irodalomjegyzék & források

[1] Zohreh Aghababaeyan , Manel Abdellatif , Lionel Briand , Fellow, IEEE, Ramesh S , and Mojtaba Bagherzadeh: Black-Box Testing of Deep Neural Networks through Test Case Diversity - https://ieeexplore.ieee.org/stamp/stamp.jsp?arnumber=10041782

[2] VGG 16 architektúra - https://www.researchgate.net/profile/Timea-Bezdan/publication/333242381/figure/fig2/AS:760979981860866@1558443174380/VGGNet-architecture-19.ppm

[3] Cifar10 adathalmaz - https://www.cs.toronto.edu/~kriz/cifar.html

[4] Clevr v1.0 adathalmaz - https://cs.stanford.edu/people/jcjohns/clevr/