

Yolov8-as szegmentációs háló és annak magyarázhatósága EigenCAM típusú modellfüggetlen magyarázó rendszerrel

Nyilas Péter

2024. május 9.

Tartalomjegyzék

1. Bevezetés

A tavalyi témalaboratóriumi dolgozatomban is foglalkoztam közlekedési objektumok detektálásával, a gépi látás (**CV!** (**CV!**)) témakörében, mindez Yolo architektúrájú neurális hálóval akkor a YOLOv5-el, ebben a dolgozatban megfogalmaztam terveket az eredmények javításában, ezek mentén végeztem a következőkben leírt munkámat:

- a. Újabb/nagyobb háló alkalmazása
- b. Áttérni a szemantikus szegmentációra bounding boxról
- c. Új (hardveres) erőforrással és jobb szoftveres kiszolgálás

Ezekhez még hozzáadódott az, hogy vizsgáljam mellé, hogy

- c. a háló döntéseit hogyan lehetne magyarázni, hogy azokat "*debugolhatóvá*" érthetőbbé tegyem.

Szakmai gyakorolatomban is képfeldolgozó neurális hálókkal foglalkoztam és itt megtapasztalhattam hogy az ilyen hálózatok azonban gyakran fekete dobozként működnek, ami komplikálttá teszi azok megértését az emberi felhasználók számára, akik meg szeretnék érteni a működésüket hogy biztonságosabbnak tudhassuk ezeket, és megértésekkel későbbi hibákat javíthatunk. Az interpretálhatóság, vagyis a modell döntéseinek megértés és magyarázata, ezért kulcsfontosságú az ilyen modellek elfogadhatóságában és alkalmazhatóságában.

A YOLOv8 (You Only Look Once version 8) egy népszerű és hatékony ilyen neurális háló, ennek is a szemantikus szegmentációs változatát használom (Yolov8m_seg), amely számos alkalmazásban használatos, például objektumfelismerésben és objektumklasszifikációban. Annak érdekében, hogy jobban megértsük és magyarázni tudjuk ennek hozott döntéseit, olyan magyarázó rendszerekre van szükségünk, amelyek képes betekintést nyerni a modell működésébe.

Egy ilyen eszköz az EigenCAM (Eigen Class Activation Mapping) egy ilyen modellfüggetlen magyarázó rendszer, amely képes vizualizálni, hogy a mély tanuló hálózatok mely részei járultak hozzá a döntéshozatalhoz. Ehhez pedig az aktivációs függvények értékeit kiemeli a hálóból és a képre vetíti amire ezt futtatjuk, majd ezeket kombinálja egy képpé megmutatva azt hogy melyik részei a képnek milyen fontosak a döntések meghozásában.

2. Mély tanulás és interpretálhatóság

A Mesterséges intelligenciát használó megoldásokat azért használjuk gyakran hogy rugalmasabb megoldást adjon akár nehezen algoritmizálható problémáinkra. Azonban ezek a modellek gyakran rendkívül bonyolultak és nehezen értelmezhetőek az emberi számára. Ezért elengedhetetlen az interpretálhatóság, vagyis annak képessége, hogy a modellek döntéseit érthető és ésszerű módon magyarázza meg.

A mély tanulási modellek interpretálhatósága fontos szempont minden alkalmazásban. Például a klinikai diagnosztikában fontos tudni, automatikus döntés miért született, hogy az orvosok megbízhatóan megérthessék és elfogadhassák az eredményt. Forgalmi szituációban miből vett észre vagy nem vett észre objektumokat

Különböző magyarázó módszerek léteznek a mély tanulási modellek interpretálhatóságának növelésére. Ezek közé tartoznak olyan módszerek is amely egy egyszerűbb neurális hálót tanít be hasonló dolgokra mint amire a hálónkat tanítjuk ,ezek a modell-függő magyarázó módszerek. Léteznek azomban modellfüggetlen megoldások például az attribúciós módszerek , a visszafejtési technikák és a modellek folyamatos moniterezésére alkalmas szoftverek. Az egyik ilyen monitorozó módszer az EigenCAM (??).

KÉRDÉS: Miért is ilyen fontos a modell interpretálhatósága és a magyarázhatósága?

Válasz:

. Különösen fontos a jogi szabályozásban egy ilyen modell használatakor, hogy a döntések átláthatóak és érthetőek legyenek. Például az Európai Unió által 2016-ban életbe léptetett Általános Adatvédelmi Rendelet (GDPR) előírja, hogy az automatizált döntéshozatalnak átláthatónak kell lennie, és az érintetteknek joguk van tudni, hogy egy algoritmus milyen döntéseket hoz róluk.

3. YOLOv8: Szemantikus szegmentációs háló és működése

A YOLOv8 ([?]) egy hatékony és népszerű szemantikus szegmentációs hálózatcsalád legújjabb példánya, amelyet számos számítógépes látás (**CV!**) feladatban használnak. A "You Only Look Once" (YOLO) megközelítést alkalmazza, amely gyors és pontos objektumdetektálást tesz lehetővé egyetlen neurális háló segítségével.



A YOLOv8 működése során a bemeneti képet egyszerre veszi figyelembe, és az objektumok pozícióját és osztályát egyetlen predikcióval határozza meg. Ez a modell különösen alkalmas valós idejű alkalmazásokhoz, mint például az önvezető járművek vagy a videoelemzés.

A YOLOv8-nak kiváló teljesítménye és pontossága van, ami azt jelenti, hogy gyorsan és pontosan képes azonosítani az objektumokat a képeken és videókon.

1. ábra. Ennek a hálónak én a szegmentációs változatát használtam, amely mostanság egy elég új irány a közlekedési objektumok detektálásában, eddig ugyanis inkább 2–3 dimenziós ún. "bounding boxokat" azaz kereteket használtak az objektumok azonosítására, azonban a szemantikus hálók képesek az objektumok pontosabb azonosítására és lokalizálására, azaz az objektumok pontos körfonalainak meghatározására. Ezáltal a szegmentációs hálók pontosabb és részletesebb információkat nyújtanak az objektumokról, mint a keretes interpretáció.

3.1. Szemantikus szegmentáció

A szemantikus szegmentáció egy olyan számítógépes látás feladat, amelyben a cél az objektumokat tartalmazó kép egyes részeinek (például pixeljeinek) címkézése az objektumokhoz tartozó osztályok szerint. Más szavakkal, minden képpontot hozzá kell rendelni egy osztályhoz vagy kategóriához, például autó, biciklis, gyalogos stb. Ezáltal a szemantikus szegmentáció lehetővé teszi a rendszereknek, hogy pontosan azonosítsák és lokalizálják az objektumokat egy adott képen. Fontos megjegyezni hogy egy nagy különbség a szemantikus és az egyed szegmentáció között, hogy az egyed szegmentációban minden objektumot külön kell azonosítani, míg a szemantikus szegmentációban csak az objektumok osztályait kell meghatározni.

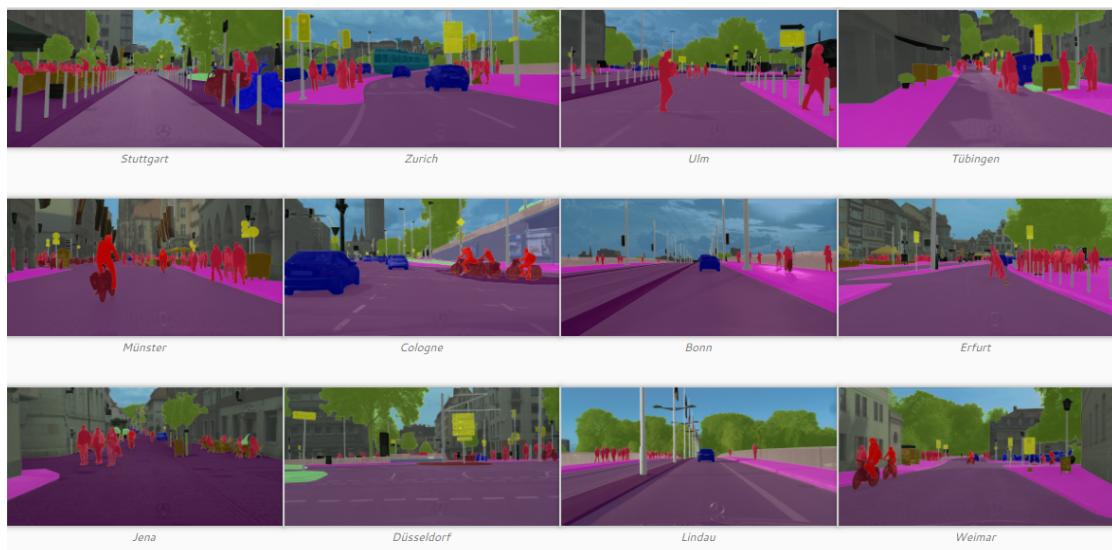
Tehát a szemantikus szegmentációt leírhatjuk úgy, ha egy képet jelölünk I -vel, akkor a szemantikus szegmentáció pedig egy olyan függvény, $F : I \rightarrow L$, ahol L a lehetőséges osztályok halmaza, és F minden képpontot hozzárendel egy osztályhoz.

A szemantikus szegmentáció kiemelt fontosságú az önvezető autók, a videoelemzés, a térképek építése és sok más alkalmazásban, ahol pontos és részletes objektumfel-

ismerésre van szükség.

3.2. Adathalmazok és kísérletek

Adathalmazként (Datasetnek), témalaboratóriumhoz hasonlóan a CityScapes (<https://www.cityscapes-dataset.com>) adathalmazt, annak is a finoman annotált (Fine Annotated), a ??-képen bemutatott, adathalmazát használtam ami osztályszegmentációs maszkokat biztosít a képeihez ami ~5000 kép, közlekedési szituációkban, elég nagy varianciával rendelkezik ahhoz hogy egy robosztus szegmentációs hálót tudjunk tanítani közlekedési objektumok detektálására. A Magyarázat a ??- oldalon található.



2. ábra. CityScapes Examples

Ezeket az adatokat fogjuk a későbbiekkben használni arra hogy megpróbáljuk megérteni a hálónkat az aktivációi alapján. Ezeket a teszteket a Bonn-i halmazon végeztem.

4. EigenCAM: Modellfüggetlen magyarázó

Az EigenCAMindexEigenCAM (Eigen Class Activation Mapping) egy modellfüggetlen magyarázó rendszer, amelyet a mély tanulási modellek interpretálhatóságának növelésére fejlesztettek ki. Az EigenCAM célja, hogy vizualizálja és magyarázza meg a modellek döntéseit a bemeneti adatok alapján, osztálydiszkrimináció nélkül.

Az EigenCAM működése során az algoritmus az egyes osztályokhoz tartozó aktivációs térképeket számolja ki, majd ezeket kombinálja az osztályok jellemzőinek és fontosságának megjelenítéséhez. Ez lehetővé teszi, hogy az emberi felhasználók megértsék, hogy a modell miért döntött úgy, ahogy.

Az EigenCAM előnyei közé tartozik a modellfüggetlenség és a viszonylag egyszerű megvalósítás. Azonban fontos tudni, hogy az EigenCAM csak egy interpretálhatósági eszköz, és nem biztosít teljes képet a modell működéséről.

Válasz:

. Az aktívációs függvények a modell layereinek kimeneti függvényei ahol "x" az input, "W" a súlymátrix és "b" a bias vektor. Az utolsó layerben egy lineáris aktivációs (??) függvényt használunk, míg az összes többi layerben egy szivárgó rektifált lineáris aktivációt (??) használunk:

$$(\text{LReLU}): \varphi(x) = \begin{cases} x, & \text{ha } x > 0 \\ 0.1x, & \text{különben} \end{cases} \quad (1)$$

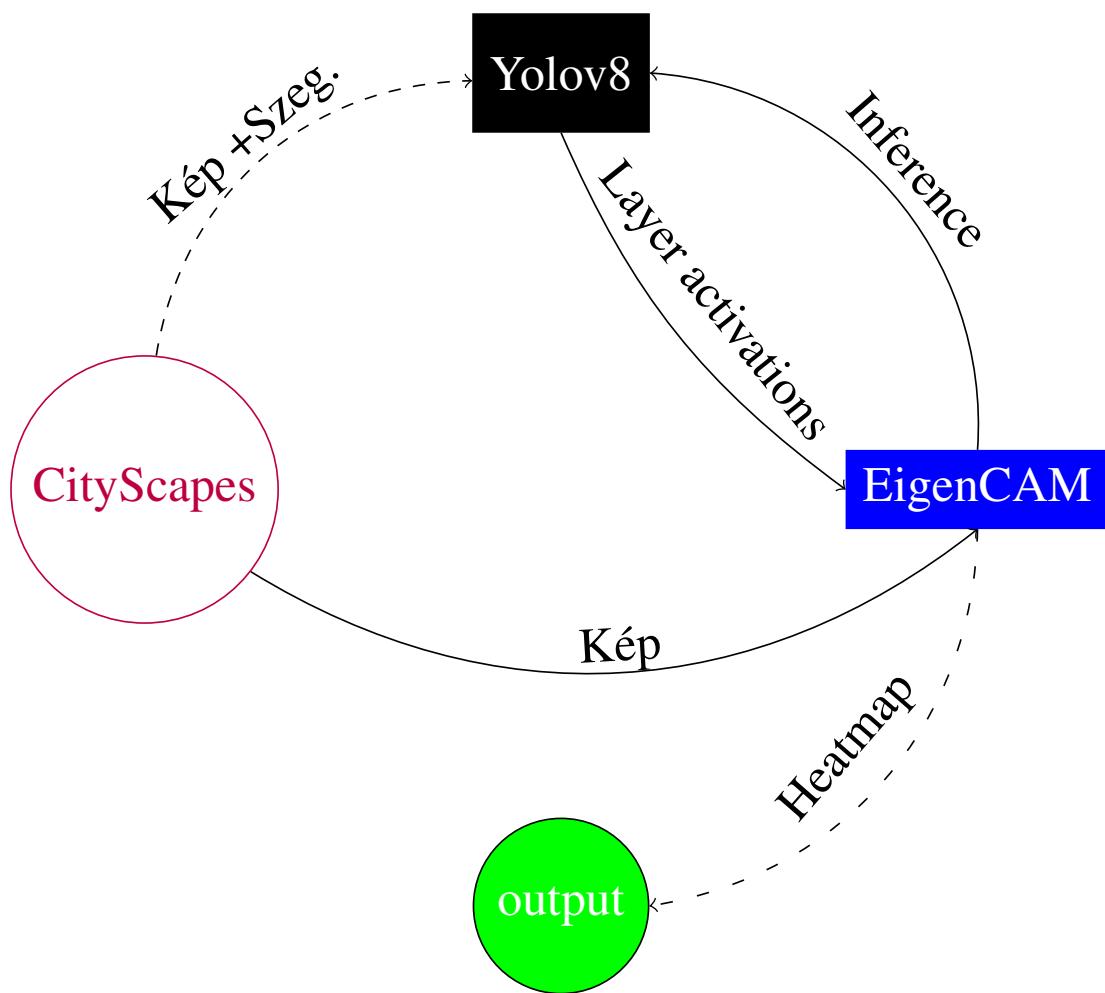
$$\text{RELU}: \varphi(x) = \begin{cases} x, & \text{ha } x > 0 \\ 0, & \text{különben} \end{cases} \quad (2)$$

$$\text{Sigmoid}: \varPhi(x) = \frac{1}{1 + e^{-x}} \quad (3)$$

5. Az EigenCAM alkalmazása a YOLOv8-ra

Az EigenCAM alkalmazása a YOLOv8 szemantikus szegmentációs hálóra lehetővé teszi számunkra, hogy megértsük, hogy a modell miként dönt az objektumok azonosításáról és lokalizálásáról a képeken és videókon.

A konkrét Működési folyamata a ?? ábrán látható. A CityScapes adathalmazon tanítottuk a YOLOv8m-seg hálót, majd az EigenCAM segítségével vizualizáltuk az aktivációkat, amelyek a háló a képek feldolgozása közben produkál (ezt az EigenCAM adja a hálónak egy olyan folyamatot keresztül amit Inference-nek nevezünk), ezeket az EigenCAM kivezeti és rávetíti a bemenő képre (ezt a folyamatot lásd a ?? képen ésa ?? pontban), a különböző héjak aktivációit külön-külön. Amikor végez a kép előállításával azt HEATMAP formájában az output mappájába helyezi.



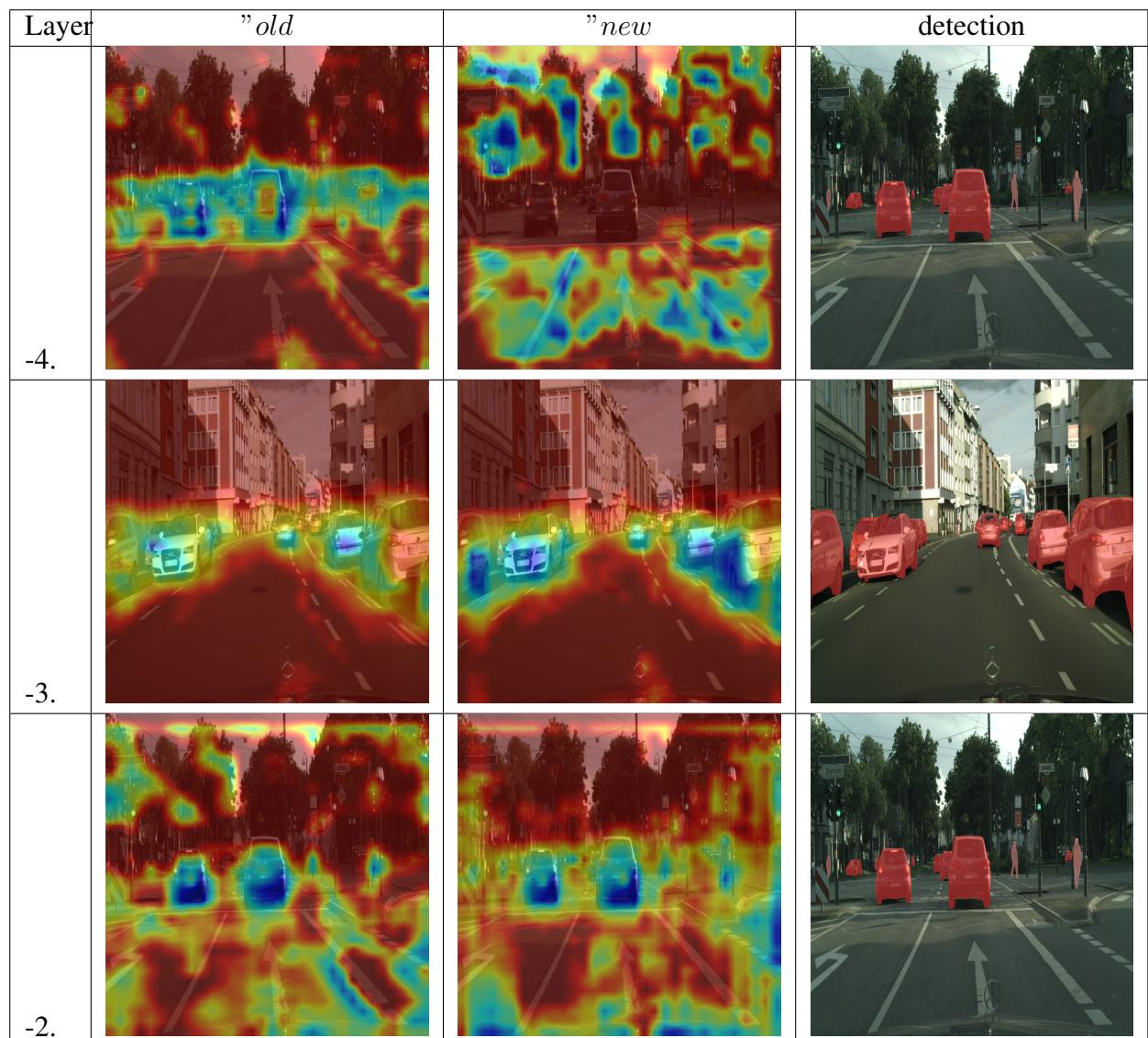
- Normál Szaggatott: Tanítási időben

- Teli nyíl: Inference (EigenCAM használata) közben
- Laza szaggatott nyíl: Inference idő után

6. Magyarázhatosági eredmények és értékelés

Tanítottam két hálót, legyen az egyik "old" a másik "new," oldegy yolov8m-seg háló 20 epochon keresztől tanult a teljes adathalmazon, a másik pedig 40 epochon keresztől tanult ugyanazon az adathalmazon.

Az EigenCAM által nyújtott kimenet mint amit ábrázoltam a ?? táblán a két hálónak a detekciója kevessé különbözik kizárolag abban milyen bizonyosságban képesek megmondani átlagosan az objektumok osztályát.



1. táblázat. Héj aktivációk összehasonlítása a két háló között.

6.1. Magyarázat

Unortodox módon a heatmappeken az alacsony aktivációs értékeket a vöröses árnyalatok míg a magas aktivációs értékeket a kékes árnyalatok jelölnek.

Layer	Magyarázat
-4	magyarázat
-3	magyarázat
-2	magyarázat

2. táblázat. Magyarázat

7. Az EigenCAM alkalmazásának gyakorlati haszna

Gyakorlati alkalmazások és lehetséges felhasználási területek az EigenCAM és hasonló magyarázó rendszerek alkalmazására. Lehetőségek az interpretálhatóság javítására és az emberi felhasználók bizalmának növelésére.

8. Jövőbeli irányok és kutatási lehetőségek

Lehetséges fejlesztési irányok az EigenCAM és a YOLOv8 interpretálhatóságának javítására. Új kutatási lehetőségek a mély tanuló modellek magyarázhatóságának területén.

9. Összegzés és következtetés

Az EigenCAM és a YOLOv8 magyarázhatóságáról szóló dokumentum végén összefoglaljuk a főbb tanulságokat és eredményeket. Megvizsgáljuk az elért eredményeket és azok lehetséges hatásait a jövőbeli kutatásokra és alkalmazásokra. ossz:indexjegyzek)

Formai elem	Megvalósítás
irodalomjegyzék	itt
tartalomjegyzék	itt
rövidítésjegyzék	itt
indexjegyzék	itt
táblázat	itt és itt
hivatkozás táblázatra	itt
vektor-grafikus kép	itt
raszter-grafikus kép	itt
hivatkozás képre	itt
tikz ábra	itt
hivatkozás ábrára	itt
képlet	itt
hivatkozás képletre	itt
képletecsoporthoz	itt
hivatkozás képletecsoporthoz egy képletére	itt
fejezet	itt
hivatkozás fejezetre	itt
lista	itt
hivatkozás lista elemre	itt
hivatkozás oldalszámra	itt
hivatkozás irodalomra	itt
saját makró használata	itt