

# Üzleti Intelligencia

## 9. Előadás: Egyedek felismerése

Kuknyó Dániel  
Budapesti Gazdasági Egyetem

2023/24  
1.félév

**1 Bevezetés**

**2 Szemantikus szegmentálás**

**3 Egyed szegmentáció**

**4 Pózfelismerés**

**1 Bevezetés**

**2 Szemantikus szegmentálás**

**3 Egyed szegmentáció**

**4 Pózfelismerés**

# Objektum lokalizáció

A lokalizáció feladatában a modell nem csak egy címkét rendel hozzá a képhez valamilyen előre meghatározott címkehalmazból, hanem **megadja a keresett objektum kereteződobozának koordinátáit**.

Ezek a koordináták a  $x, y, w, h$  vagyis a bal felső sarok  $x, y$  koordinátái, a doboz szélessége és hosszúsága.

A lokalizáló hálózatnak **két output rétege** (feje) van. Az egyik az objektum osztályba esésének valószínűségét adja meg, a másik pedig a dobozának koordinátáit.

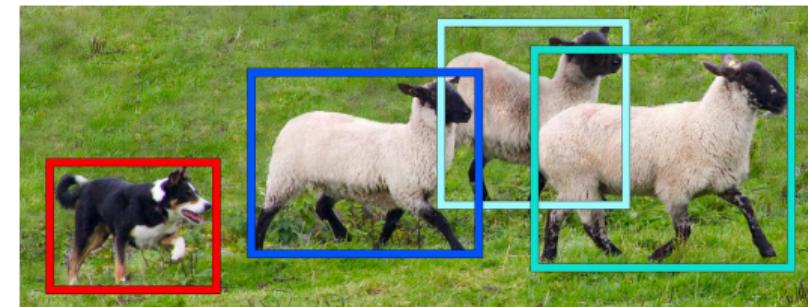


# Objektum detekció

Az objektum detekció a **lokalisáció** általánosítása **tetszőleges számú objektumra** egyetlen képen belül.

Detekció esetén a neurális modell feladata megtalálni az **összes olyan objektumot, amely ismert** a címkehalmazban.

Az objektum detektor outputja kereteződobozok és a hozzájuk tartozó valószínűségeseloszlások halmaza, ezért **multioutput** osztályozási eljárásnak tekinthető.

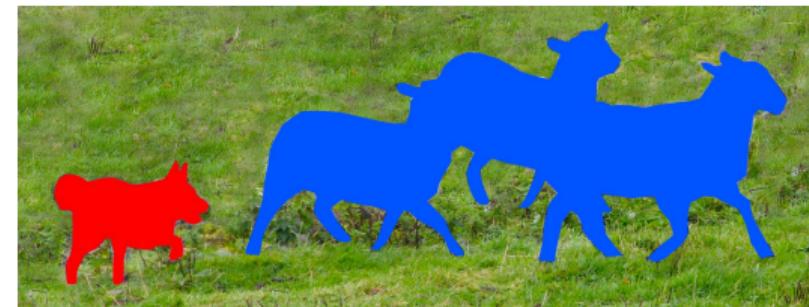


# Szemantikus szegmentáció

A szemantikus szegmentáció problémája nem csak megtalálni, hogy az egyes keresett objektum osztályoknak hol található a kereteződoba a képen belül, hanem **pixel szinten osztályozást végezni a képen** belül a keresett objektumokra.

## Szemantikus szegmentáció

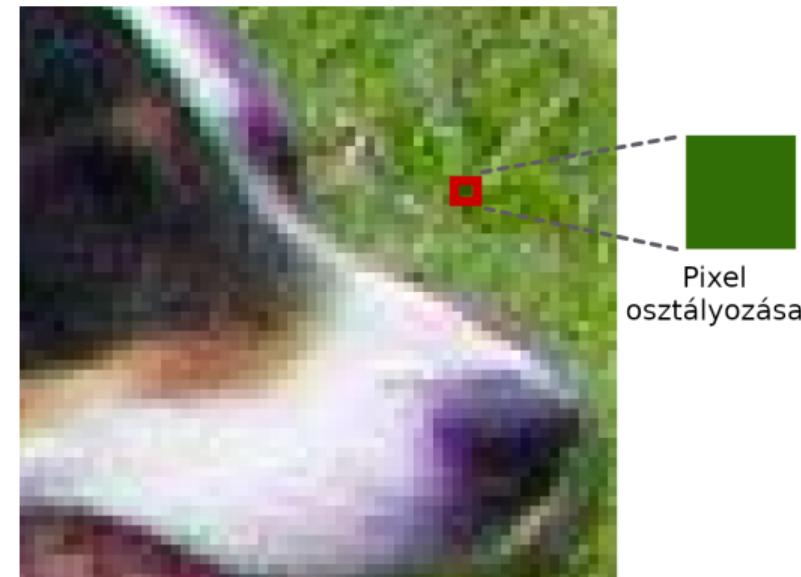
A szemantikus szegmentáció egy sűrű, pixelenkénti osztályozást ad outputként az input képre vonatkozóan **anélkül, hogy megkülönböztetné az objektumok egyedeit**. Az a pixel, amelyik nem becsülhető meg kellően magas valószínűséggel, osztályozatlanul marad.



# Pixelszintű osztályozás

A szemantikus szegmentáció naív implementációja úgy működne, hogy egy **csúszóablak végigmegy a teljes képen**, és minden egyes pixelt végigáramoltat egy osztályozó neurális hálózaton.

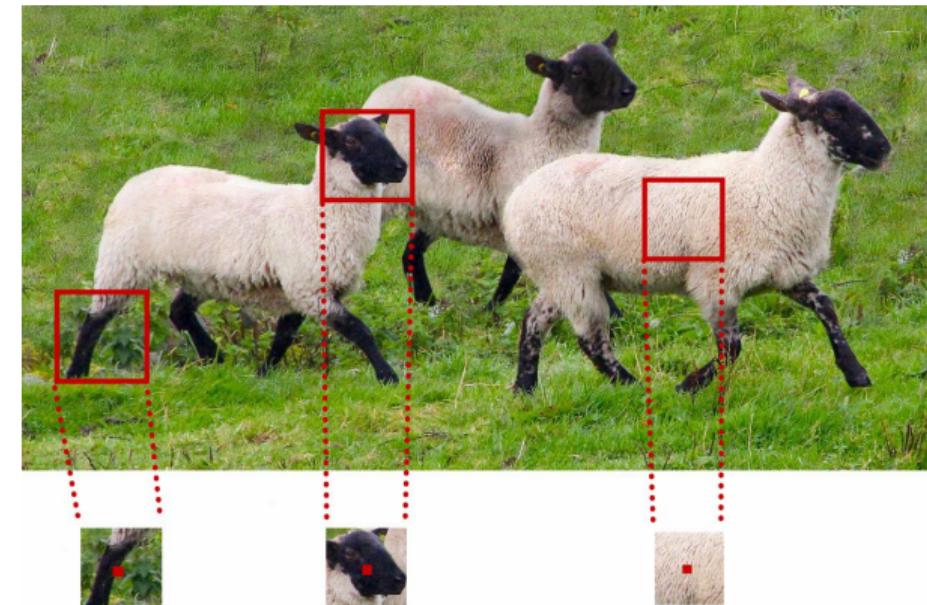
**Ez viszont lehetetlen**, mert egyetlen RGB színadat alapján nem lehet megmondani a képpont szemantikai jelentését.



# Ablakszintű osztályozás

Egy másik naív megoldása a pixelszintű osztályozásnak, ha az algoritmus **végigmozgat egy ablakot a képen**, majd az ablakhoz tartozó régiót beosztályozza egy konvolúciós hálózat segítségével, majd az outputját hozzárendeli a középső pixelhez.

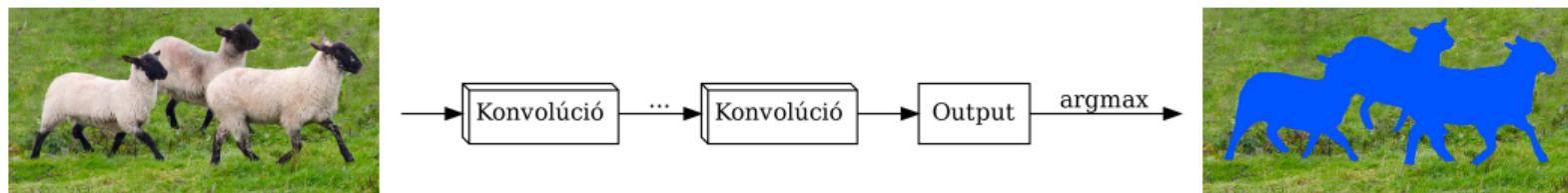
A probléma ezzel a hozzáállással ismételten az, hogy **minden pixelre lefuttatni egy osztályozást rendkívül költséges**, a valóságban nem kivitelezhető.



# Szemantikus szegmentálás konvolúciós hálózattal

Egy intuitív hozzáállás a szemantikus szegmentációhoz, hogy a modell **végigáramoltatja több konvolúciós rétegen az input képet**, majd az outputjából kiválasztja a legnagyobb valószínűségű osztályt minden egyes képontra.

Ebben az esetben az **output méretnek meg kell egyeznie az input mérettel**, ezért az összes konvolúciós szűrőnek azonos méretűnek kellene lennie! A teljes felbontású konvolúció ismételten egy nagyon költséges megoldás.



1 Bevezetés

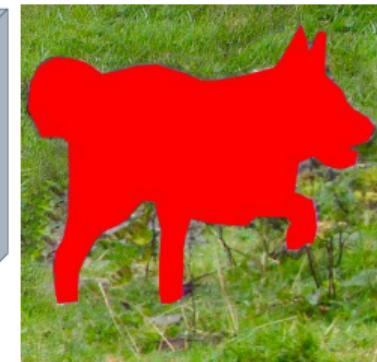
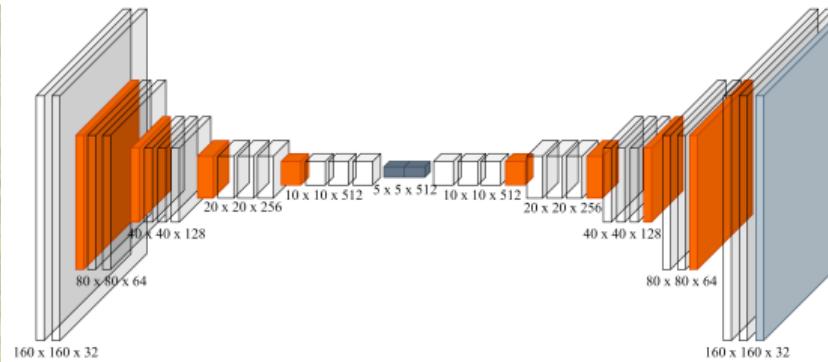
2 Szemantikus szegmentálás

3 Egyed szegmentáció

4 Pózfelismerés

# Teljesen konvolúciós szemantikus szegmentálás

A bevált hozzáállás a szemantikus szegmentáció problémájához, ha olyan konvolúciós hálózaton áramlik át az input adat, ami először lefelé mintavételezi, majd felskálázza a képet, ezzel eliminálva a többszörös teljes felbontású konvolúcióból adódó problémákat.

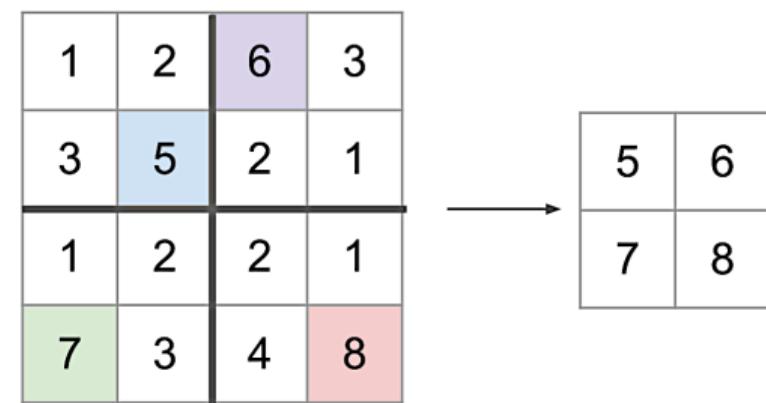


# Mintavételi eljárások

## Lefelé mintavételezés

A lefelé mintázó folyamatok az input adat térbeli dimenzióit hivatottak **csökkenteni** egy konvolúciós hálózatban a számítási teher csökkentése érdekében. Egy ilyen eljárás a **max. pooling**.

Max. pooling során a réteg megőrzi a legnagyobb elemeket minden, a pooling neuronnal kapcsolatban álló bemenet közül.



Input: 4 x 4

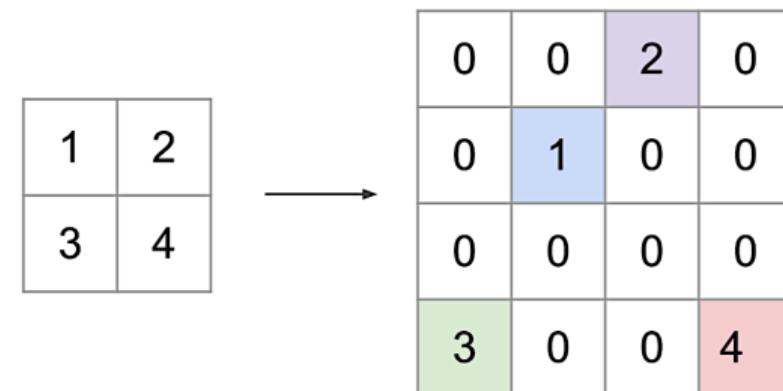
Output: 2 x 2

# Mintavételi eljárások

## Felfelé mintavételezés

A felfelé mintázó folyamatok az input adat térbeli dimenzióit hivatottak **növelni** egy konvolúciós hálózatban azért, hogy vissza lehessen nyerni felbontásban kódolt információt. Egy ilyen eljárás a **max. unpooling**.

A max. unpooling során a lazított területek maximális elemei visszakerülnek azokra a pozíciókra, ahol eredetileg voltak.



Input: 2 x 2

Output: 4 x 4

## Felfelé mintavételezés

Az output tartalmazza a szűrő másolatait a bemenet súlyozásával, összegezve ott, ahol az átfedés van az outputban. A felfelé mintavételezés művelete értelmezhető tetszőleges dimenziószámra. Ebben az esetben az  $unpool(\cdot, \cdot)$  esetén az első paraméter az input, a második pedig a szűrő. A szűrő súlyai taníthatóak.

$$unpool \left( \begin{bmatrix} a \\ b \end{bmatrix}, \begin{bmatrix} x \\ y \\ z \end{bmatrix} \right) = \begin{bmatrix} a \cdot x \\ a \cdot y \\ a \cdot z + b \cdot x \\ b \cdot y \\ b \cdot z \end{bmatrix}$$

1 Bevezetés

2 Szemantikus szegmentálás

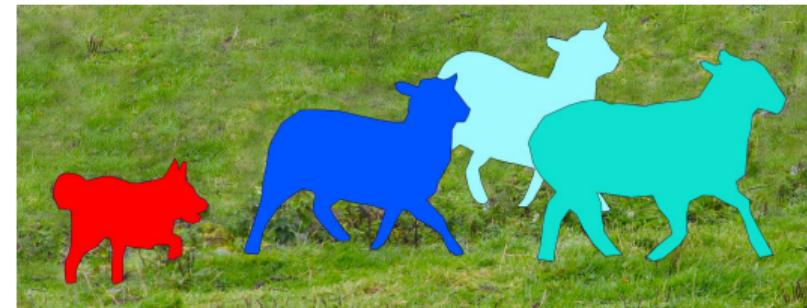
3 Egyed szegmentáció

4 Pózfelismerés

# Az egyed szegmentáció alapjai

## Egyed szegmentáció

Az egyed szegmentáció az egy képen található objektumokat **azonosítja** és **szegmentálja** olyan módon, hogy minden objektumhoz külön **maszkot rendel**.

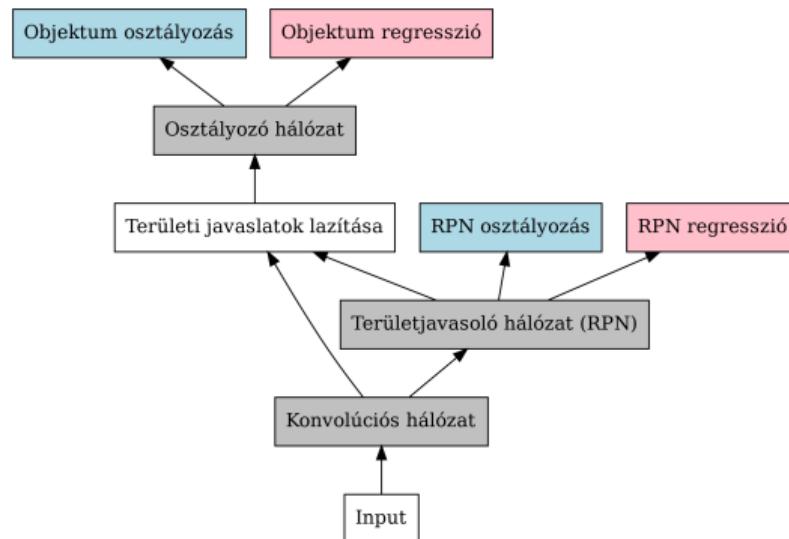


Ez azt jelenti, hogy nem csak az objektumokat azonosítja (detekció), hanem a minden egyes objektumot **körülvevő pixel területeket is elkülöníti**. Az eredmény egy olyan maszk vagy maszkok halmaza, amely pontosan meghatározza a képen található egyedek **körvonalait** és **elhelyezkedését** a képen.

# Objektum detektor és egyed szegmentáló architektúra

- **RPN osztályozás:** A kereteződoboz objektum / nem objektum?
- **RPN regresszió:** A javasolt doboz és a kereteződoboz közötti transzformáció megbecslése.
- **Objektum osztályozás:** A javaslatok osztályának megbecslése.
- **Objektum regresszió:** A javasolt doboz és objektum doboz közötti transzformáció megbecslése.

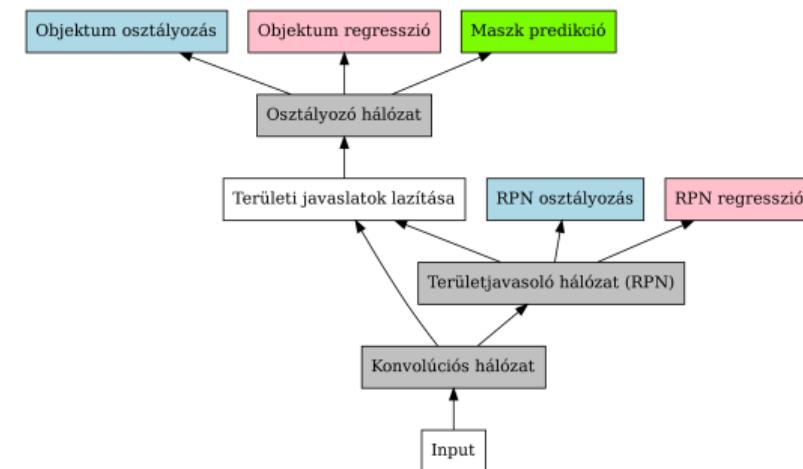
## Objektum detektor hálózat (Faster R-CNN)



# Objektum detektor és egyed szegmentáló architektúra

- **RPN osztályozás:** A kereteződoboz objektum / nem objektum?
- **RPN regresszió:** A javasolt doboz és a kereteződoboz közötti transzformáció megbecsélése.
- **Objektum osztályozás:** A javaslatok osztályának megbecsélése.
- **Objektum regresszió:** A javasolt doboz és objektum doboz közötti transzformáció megbecsélése.
- **Maszk predikció:** minden területi javasatra egy  $28 \cdot 28$  bináris maszkot becsül meg.

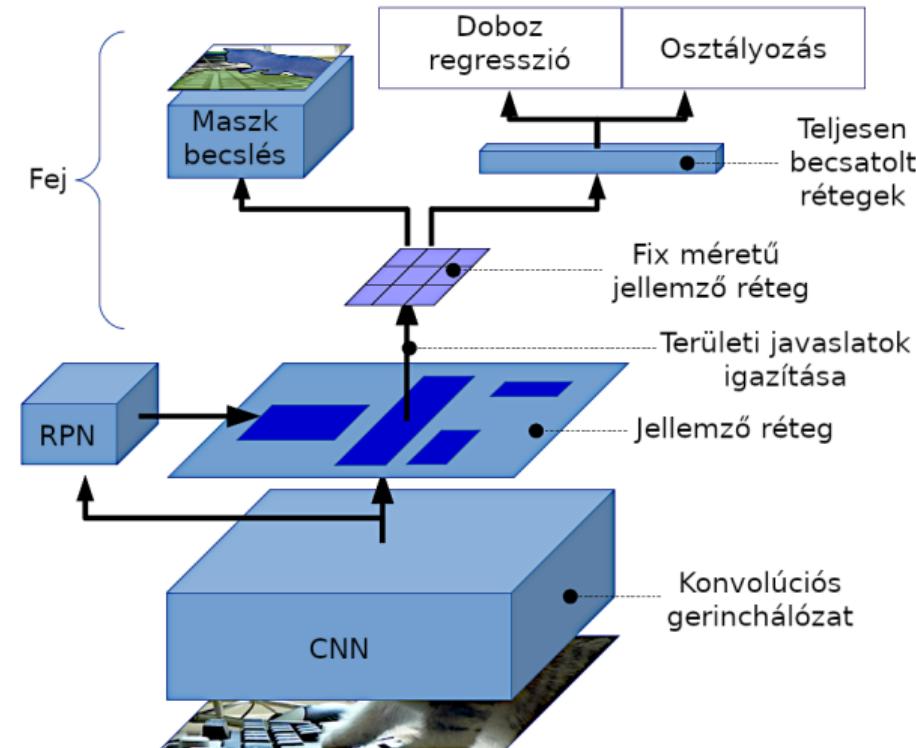
Egyed szegmentáló hálózat (Mask R-CNN)



# Mask R-CNN architektúra

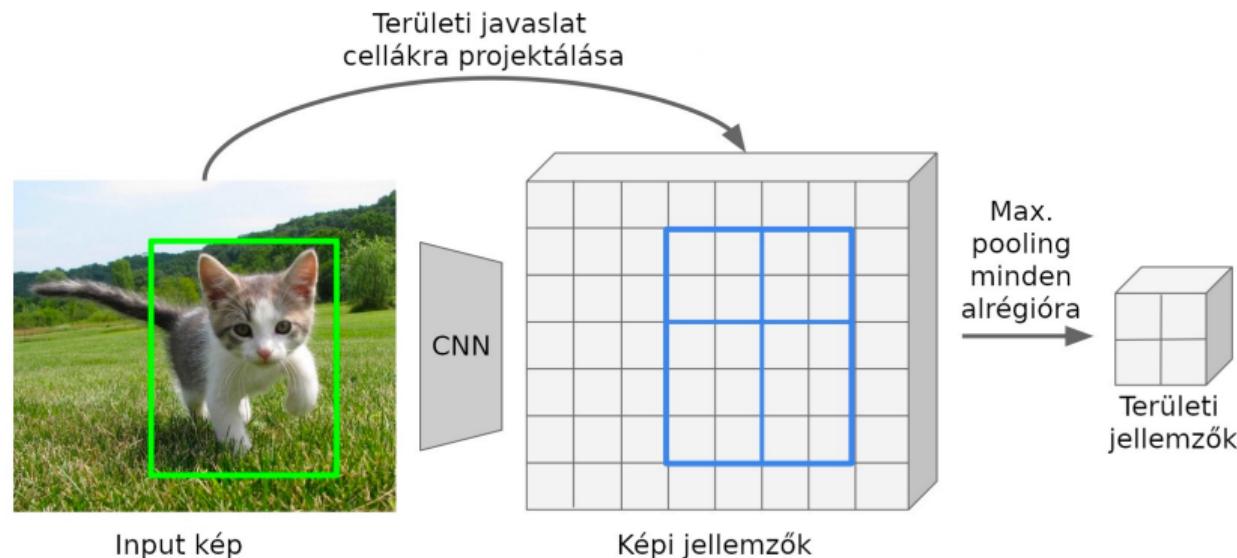
A Mask R-CNN architektúra egy továbbfejlesztése a Faster R-CNN keretrendszernek, ami az egyed szegmentálásra specializálódott.

A hálózat egy **maszk fejjel** (output hálózattal) képes osztályozni az egyes pixeleket. A hálózatnak ezen feje a területi javaslatok igazítása után képes pixelszintű maszkokat megbecsülni minden objektumra.



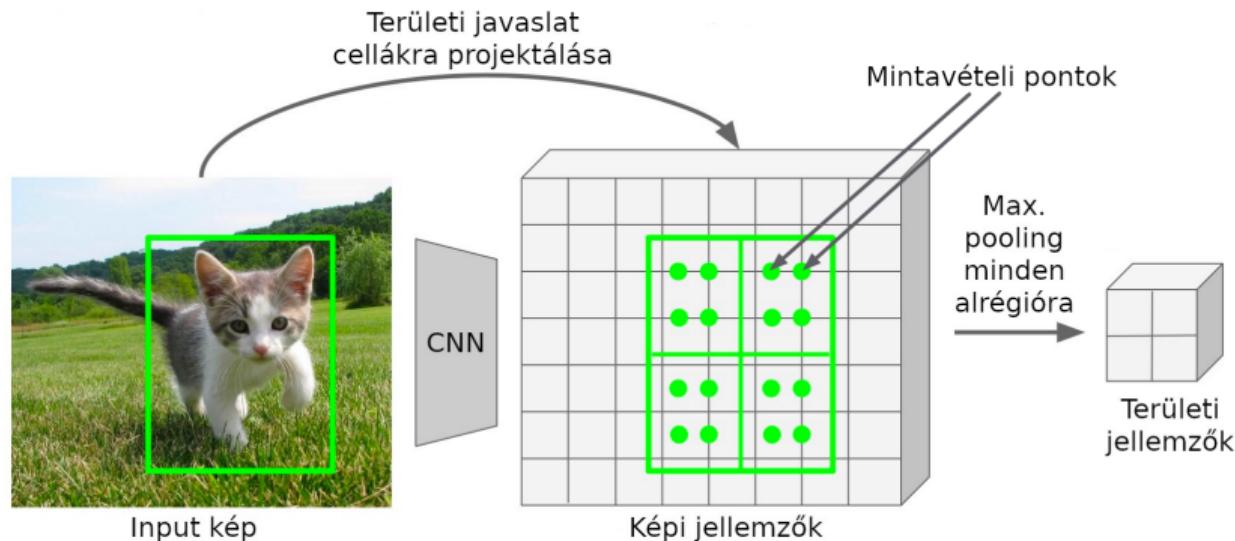
# Területi javaslatok lazítása

A területi javaslat lazítás (RoI pooling) felosztja az inputként érkező javasolt területet egy  $n \cdot n$  méretű cellára, amelyeket egy előre definiált háló határoz meg. A felosztott régiókon **max. pooling** segítségével dimenziót csökkent majd átadja az eredményt a következő rétegnek.



# Területi javaslatok igazítása

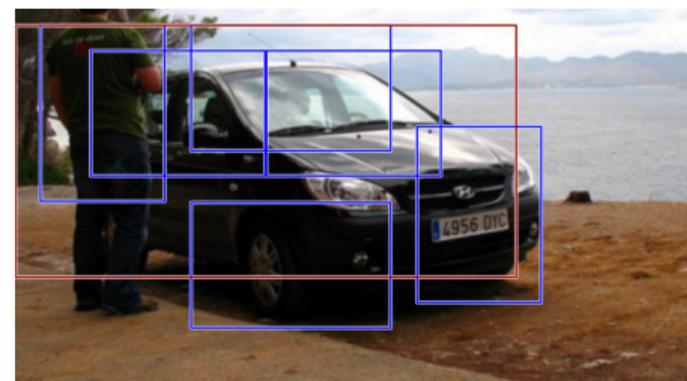
A területi javaslat igazítás (RoI align) a lazítás továbbfejlesztésének tekinthető. Az igazító eljárás **nincs előre meghatározott cellahálóhoz kötve**, ezért a becsült értékek pontosabban illeszkednek az objektumokra. A mintavételezés itt **egyenletes intervallumokban történik, bilineáris interpoláció segítségével**.



# Nem-maximális elnyomás (NMS)

Az NMS feladata utólag feldolgozni a kereteződobozokra adott predikciót úgy, hogy eltávolítsa a redundáns vagy átfedésben álló dobozokat. Az NMS folyamata:

- ① **Rendezés:** Dobozok rendezése a becslések valószínűsége szerint.
- ② **Iteráció:** A legbiztosabb doboz és az összes többi doboz közötti IoU kiszámolása.
- ③ **Küszöbölés:** Azok a dobozok, amelyek adott küszöb feletti valószínűségekkel rendelkeznek, redundánsnak számítanak.
- ④ **Eldobás:** A redundáns kereteződobozok eldobása.
- ⑤ **Ismétlés:** A folyamat megismétlődik az összes többi kereteződobozra.



# Nem-maximális elnyomás (NMS)

Az NMS feladata utólag feldolgozni a kereteződobozokra adott predikciót úgy, hogy eltávolítsa a redundáns vagy átfedésben álló dobozokat. Az NMS folyamata:

- 1 Rendezés:** Dobozok rendezése a becslések valószínűsége szerint.
- 2 Iteráció:** A legbiztosabb doboz és az összes többi doboz közötti IoU kiszámolása.
- 3 Küszöbölés:** Azok a dobozok, amelyek adott küszöb feletti valószínűségekkel rendelkeznek, redundánsnak számítanak.
- 4 Eldobás:** A redundáns kereteződobozok eldobása.
- 5 Ismétlés:** A folyamat megismétlődik az összes többi kereteződobozra.



1 Bevezetés

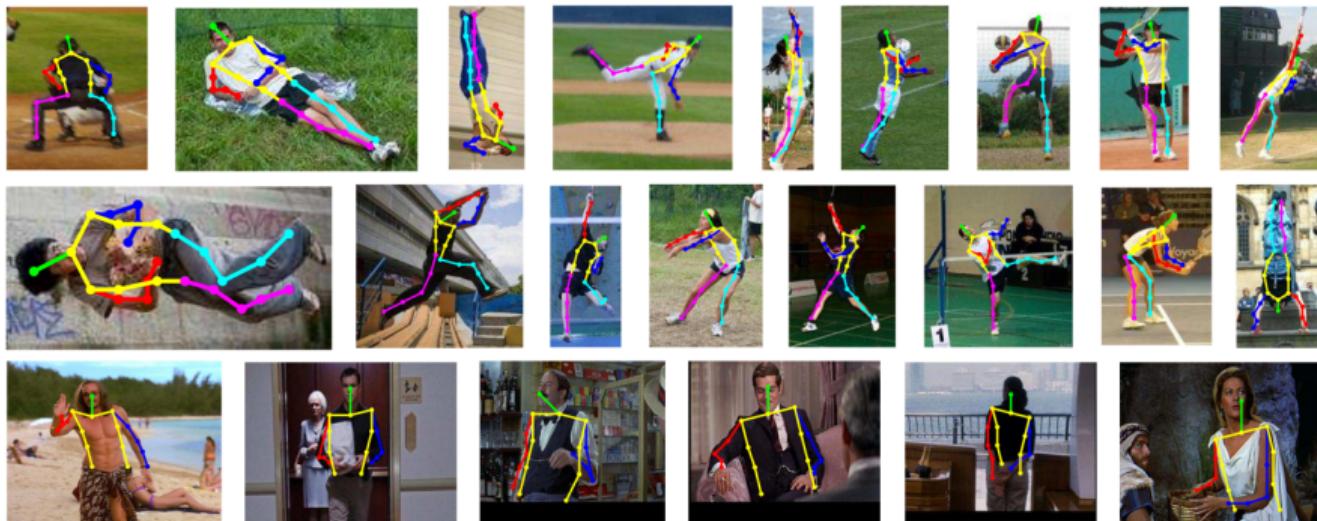
2 Szemantikus szegmentálás

3 Egyed szegmentáció

4 Pózfelismerés

# Emberi pózfelismerés alapjai

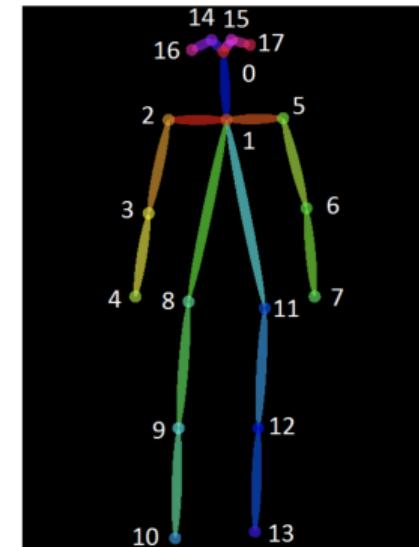
A pózfelismerés célja az emberi végtagok lokalizálása képeken és videókon. A modell felismeri az emberi test fontosabb pontjait (**kulcsPontjait**) és ezeket köti össze a póz rekonstruálása érdekében. Fontosabb **kulcsPontok** a fej, fülek, nyak, vállak, csípők stb...



# Kulcspontok

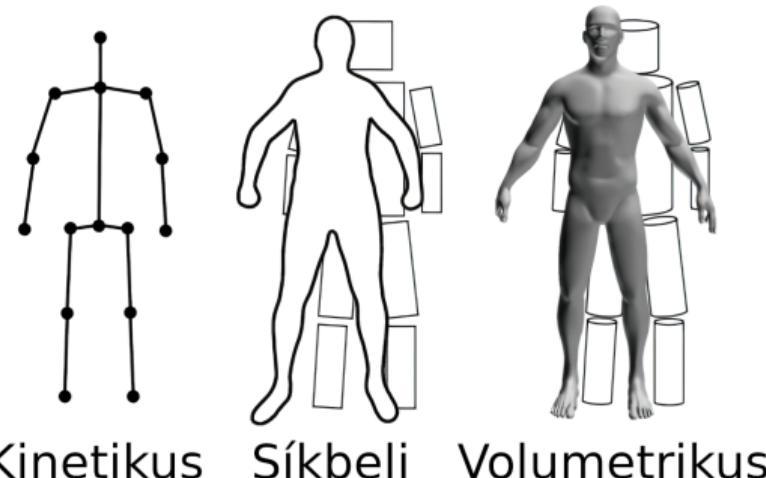
A leginkább elterjedt egységes jelölés 18 különböző kulcspontot különböztet meg (ez nem minden modellnél azonos):

- |               |             |
|---------------|-------------|
| ① Orr         | ⑨ Jobb térd |
| ② Nyak        | ⑩ Jobb boka |
| ③ Jobb váll   | ⑪ Bal csípő |
| ④ Jobb könyök | ⑫ Bal térd  |
| ⑤ Bal váll    | ⑬ Bal boka  |
| ⑥ Bal könyök  | ⑭ Jobb szem |
| ⑦ Bal csukló  | ⑮ Bal szem  |
| ⑧ Jobb csípő  | ⑯ Jobb fül  |
|               | ⑰ Bal fül   |



# Testmodellek fajtái

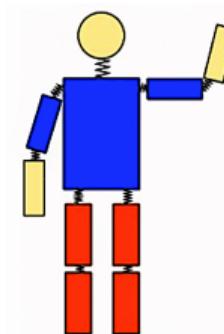
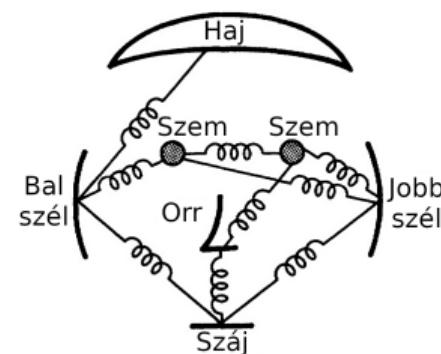
- **Kinetikus:** Legjobb alkalmazása dinamikus, mozgással és testtartással kapcsolatos információ bányászására szolgál.
- **Síkbeli:** Egy egyszerűsített reprezentáció amely abban az esetben hasznos, amikor valamilyen mozgás a síkban történik.
- **Volumetrikus:** Egy összetett, 3D reprezentációja az emberi testnek, ami a test geometrikus elrendezését és térbeli eloszlását hivatott a lehető leg pontosabban rekonstruálni.



# Klasszikus hozzállások a pózbecsléshez

Az eredeti, **Pictorial Structures** keretrendszeren alapuló hozzállás az emberi testet különálló, merev részekként fogta fel, amely egy **deformálható, laza konfigurációban helyezkednek el a képen**. minden rész egy sablonként jelenik meg, amelyet az algoritmus megpróbál valamely képrészhez hozzátársítani.

Az ebből adódó struktúra nagyon **jól tudja modellezni a különböző részek egymáshoz illeszkedését** ami fontos az emberi pózbecslésben, viszont **nagyon függ a képi jellemzőktől**. Ha valamely testrész nem tökéletesen felismerhető, az illesztés összeomlik.



# Modern megközelítés: OpenPose architektúra

Az OpenPose egyike a legkorszerűbb valós idejű pózfelismerő modelleknek az egyike. Első lépésben minden testrészhez egy **magabiztosági térképet** becsül meg ami megadja, a kép mely részén a legvalószínűbb a jelenléte. Ezután a becsült testrészeket **irányított kapcsolási mezőkkel** párosítja őket kettésével.



1. Input kép



3. Testrész kapcsolási mező



4. Kétoldalú összekapcsolás



5. Eredmények felépítése

# OpenPose részletei

Az OpenPose minden testrész magabiztosági mezőt egyenként becsül meg a testrészekre, majd a kapcsolási mezők segítségével meghatározza ezeknek egymáshoz való viszonyát. A kapcsolási mezőket és a magabiztosági jellemzőket egymástól függetlenül is lehetséges megbecsülni.

