

KONVOLÚCIÓS NEURONHÁLÓK

A tananyag az EFOP-3.5.1-16-2017-00004 pályázat támogatásával készült.

SZÉCHENYI  2020



MAGYARORSZÁG
KORMÁNYA

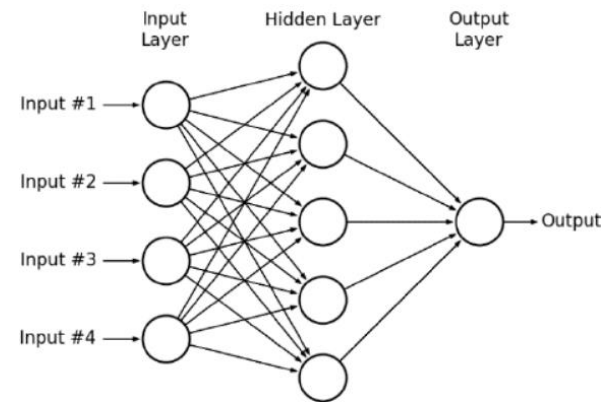
Európai Unió
Európai Szociális
Alap



BEFEKTETÉS A JÖVŐBE

1. motiváció

- A klasszikus neuronháló struktúra a „fully connected” háló
 - Két réteg között minden neuron kapcsolódik minden neuronnal
 - Az egyes inputoknak teljesen egyforma szerepük van
 - Ha az inputokat véletlenszerűen permutáljuk (de minden egyes tanító vektort ugyanúgy!), akkor a háló ugyanolyan eredményesen fog tanulni
- Nagyon sok feladatnál a jellemzők sorrendjének tényleg nincs szerepe
 - Pl. korábbi feladat: (láz, ízületi_fájdalom, köhögés) → influenza
 - Nyilván pont annyira tanulható, mint (ízületi_fájdalom, köhögés, láz) → influenza
- Vannak azonban feladatok, ahol a jellemzők elrendezése fontos

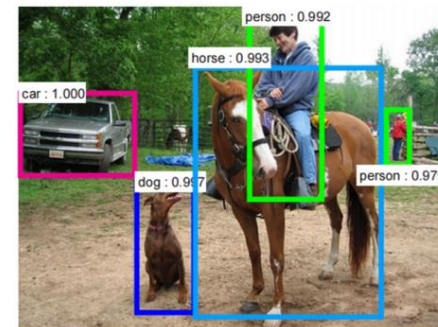


2. motiváció

- Vannak azonban feladatok, ahol a jellemzők elrendezése, egymáshoz való viszonya (topológiája) fontos
 - Pl. képi alakfelismerés: Nem ugyanazt látjuk, ha a pixeleket összekeverjük
 - Egy „fully connected” háló viszont ugyanúgy fog tanulni a kétféle képtípuson

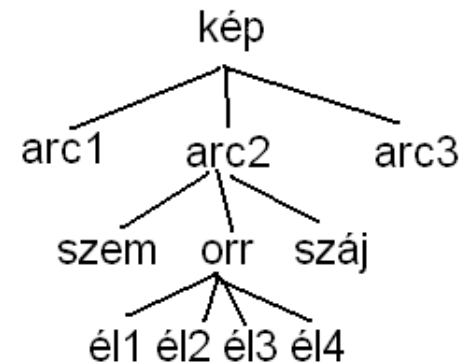
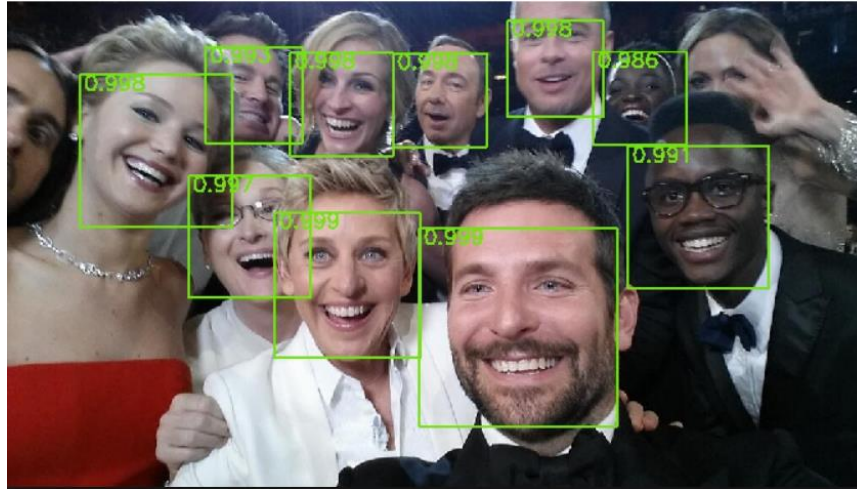


- A pixelek elrendezésében az agyunk számára fontos információ van, amit viszont a fully connected háló nem használ ki!
 - A pixelek együtt alkotnak képet
 - A közeli pixelek kapcsolata jellemzően szorosabb (együtt alkotnak objektumokat)

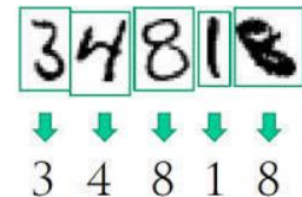


3. motiváció

- A képek jellegzetesen hierarchikusan épülnek föl
 - Egyszerűbb, lokális építőelemektől a nagyobb, összetettebb objektumok felé haladva



- A korai neuronhálós alakfelismerési kísérletek az ilyen összetett képek felismerését eleve reménytelennek tartották
 - Csak egyszerűbb feladatokkal próbálkoztak, pl. karakterfelismerés
- Most már összetett, „valódi” képekkel is kísérleteznek
 - Ezek felismerésében a konvolúciós háló jelentős javulásokat hozott



4. motiváció

- Vannak arra utaló kutatások, hogy az emberi agy is hierarchikusan dolgozza fel a képeket
 - A vizuális agykéregben bizonyos neuronok a képen látható egyszerű objektumok, pl. különböző irányban álló élek látványára „tüzelnek”
 - Vagy erre utal pl. az ún. „Thatcher illúzió” – a fejjel lefele álló kép esetén az agyunk megállapítja, hogy a száj, orr, szem rendben van és a helyén van (együtt kiadnak egy arcot), de nem veszi észre a finom részletekben rejlő hibát





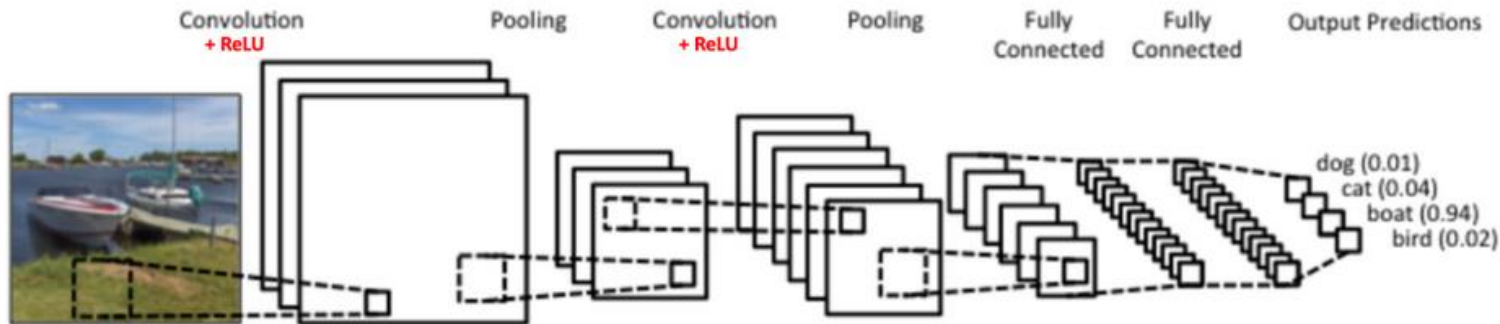
Konvolúciós neuronháló

- A fenti motivációkhoz igazodva a konvolúciós neuronháló
 - Az inputot hierarchikusan dolgozza fel
 - ebből adódóan szükségszerűen mély lesz, azaz sok rétegből fog állni
 - Az egy egyes rétegek inputja lokális, azaz a kép kis részleteire koncentrálnak
 - De fölfelé haladva egyre nagyobb részeket fed le
 - A magasabb rétegek egyre nagyobb képrészeket fednek le, de ezzel egyidejűleg egyre rosszabb felbontásúak (a finom részletek egyre kevésbé számítanak)
 - Kevésbé lesz érzékeny az objektumok pontos pozíciójára (ebben segít majd a konvolúció)
- A konvolúciós mély háló fő alkalmazási területe a alakfelismerés, de más olyan feladatban is szóba jöhet, ahol az input kép, vagy ahhoz hasonlóan hierarchikus felépítésű (pl. beszéd felismerésben is használják)

A konvolúciós neuronháló építőelemei

- Egy tipikus konvolúciós háló felépítése

- <https://ujjwalkarn.me/2016/08/11/intuitive-explanation-convnets/>



- Konvolúciós neuronok (szűrők)
- Pooling (egyesítés) művelet
- A fentiek többször ismételve
- A háló tetején általában néhány „fully connected” réteggel, és persze a szokásos kimenő réteggel
- A tanítás szokványosan a backpropagation algoritmussal történik (ebben a pooling lépés okoz némi bonyodalmat)



A konvolúciós lépés

- A konvolúciós neuronok pontosan ugyanolyan neuronok, mint amilyenekről eddig beszéltük. A fő különbségek:
 - Lokalitas: a neuronok a bemenő képnek egyszerre csak egy kis részletét dolgozzák fel
 - Konvolúció: ugyanazt a neuront a kép több különböző pozícióján is kiértékeljük (mivel ezek a kiértékelések ugyanazokat a súlyokat használják, szokás ezt „weight sharing” tulajdonságként is emlegetni)
- A neuronok által végzett művelet nagyon hasonlít a képfeldolgozásból ismert szűrők hatására, ezért gyakran szűrőnek (filter) is hívjuk ezeket a neuronokat
 - De most a kimeneten valamilyen nemlineáris aktivációs függvényt is alkalmazni fogunk
 - Valamint az együtthatókat (súlyokat) nem kézzel állítjuk be, hanem tanuljuk

A konvolúciós lépés (2)

- Konvolúciós neuron működésének szemléltetése:
 - Input kép, szűrőmátrix (neuron súlymátrixa), és az eredmény:

1	1	1	0	0
0	1	1	1	0
0	0	1	1	1
0	0	1	1	0
0	1	1	0	0

1	0	1
0	1	0
1	0	1

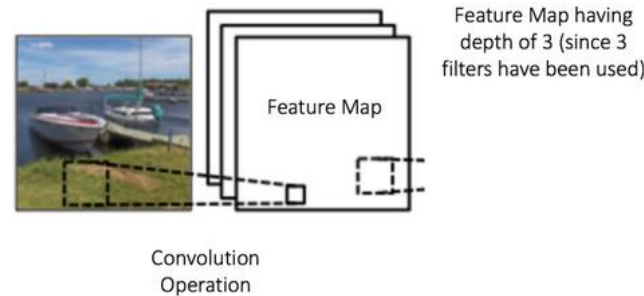
1	1 _{x1}	1 _{x0}	0 _{x1}	0
0	1 _{x0}	1 _{x1}	1 _{x0}	0
0	0 _{x1}	1 _{x0}	1 _{x1}	1
0	0	1	1	0
0	1	1	0	0

4	3	

- A kiértékelést minden pozíción elvégezhetjük, ekkor a fenti példához hasonlóan egy mátrixot kapunk
 - De a bejárt tartományt esetleg korlátozhatjuk is
- Stride: a szűrő lépésköze. Ha pl. 2, az azt jelenti, hogy minden második pozíciót átugrunk
- Zero padding: A szűrő szélső pixelekre való illesztéséhez szükség lehet a kép kitoldására (jellemzően 0 értékekkel) – ez tervezési kérdés

A konvolúciós lépés (3)

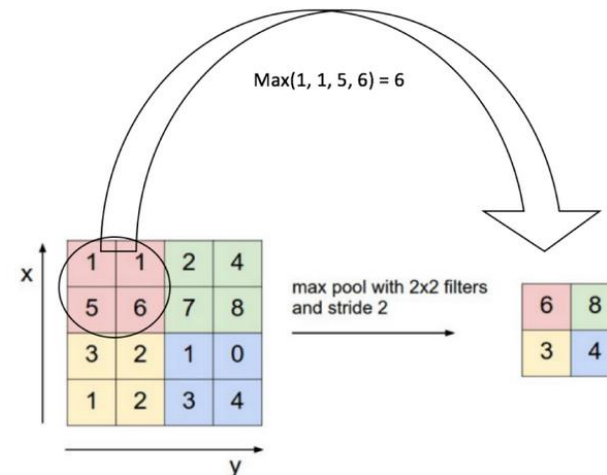
- A konvolúciós szűrőktől avagy neuronoktól azt várjuk, hogy valamilyen többé-kevésbé absztrakt fogalmat tanuljanak meg (pl. egy él detektálása, vagy egy orr detektálása)
 - Ehhez általában egyetlen neuron nem elég, tehát párhuzamosan több neuront tanítunk ugyanazon az input pozíción. Pl. három neuron esetén 3 kimeneti mátrixot kapunk, ez lesz kimenet „mélysége (depth)”



- A kimenetre újabb rétegeket fogunk tanítani, tehát úgy tekintjük, hogy az adott réteg jellemzőket nyert ki a háló többi rétege számára
 - Ezért hívják a kimenetet az ábrán „feature map”-nek

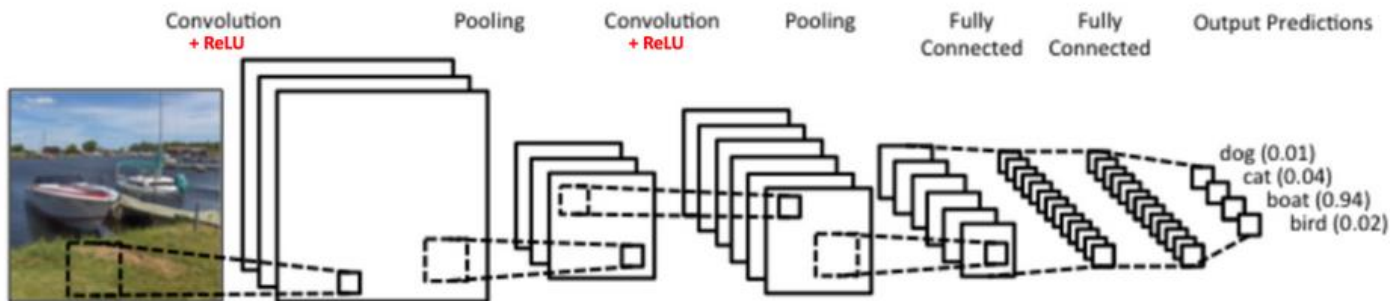
Az egyesítési (pooling) lépés

- Magasabbra menve egyre kevésbé akarjuk a finom részleteket megőrizni
 - Pl. ha az „orr-detektáló” jelzett, hogy talált egy orrot, akkor annak a pontos pozíciójára már nem lesz szükségünk magasabb rétegekben
- Ennek megfelelő műveletet végez a pooling:
 - Egy kis környezetben a kimenő értékeket egyesíti egyetlen értékké
 - Pl. a maximumot véve
- Előnyei:
 - Eltolás-invariancia a pooling régión belül (bárhol jelzett a háló, a max. ugyanakkora lesz)
 - Fokozatosan csökken a kimenő mátrix mérete (downsampling) – a paramétercsökkentés mindig hasznos!
 - Följebb lépve ugyanakkora szűrővel egyre nagyobb területet fedünk le egyre rosszabb felbontással – hierarchikus feldolgozás

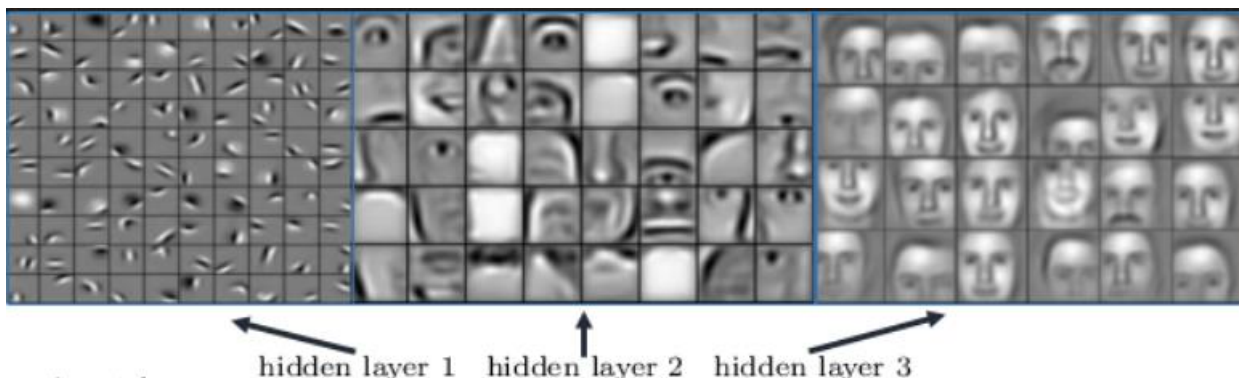


Hierarchikus feldolgozás

- A fenti konvolúció+pooling lépésekből sokat pakolunk egymásra



- Azt várjuk, hogy az egyre feljebb eső rétegek egyre absztraktabb fogalmakat tanulnak meg, nyernek ki
- Ez többé-kevésbé így is van (példa egy arcfelismerő hálóból):



- Végül a végső osztályozást teljesen kapcsolt rétegekkel végezzük el

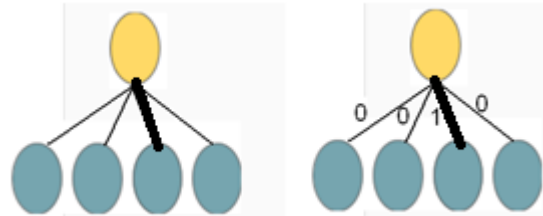


Összegzés

- A konvolúciós háló fő tulajdonságai és előnyei, hátrányai
 - Konvolúció előnyei: teljes input helyett lokális feldolgozás, több pozíción ugyanazokkal a súlyokkal - kevesebb paraméter, eltolás-invariancia
 - Hátrány: nagyobb műveletigény, mint a teljesen kapcsolt hálónál
 - Pooling előnyei: kevesebb paraméter, eltolás-invariancia, hierarchikusan egyre kisebb felbontású, de egyre nagyobb térrészt lefedő input
 - hátrány: a rész-objektumok pontos pozíciója elvész, néha pedig fontos lenne
 - Hierarchikus feldolgozás előnye: az összetett képek hierarchikusan épülnek fel, ésszerű az elemzést is így végezni
 - Hátrány: szükségszerűen mély hálót kapunk, a nagyon mély hálók tanítása problematikus lehet

Technikai megjegyzés

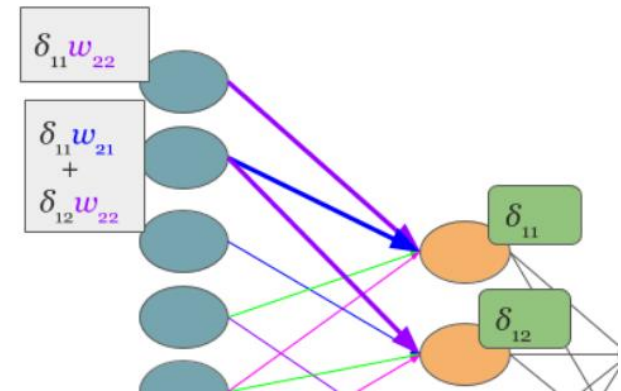
- Konvolúciós hálók esetén a derivált kiszámolása némileg bonyolultabb, mint egy sima előrecsatolt hálóban. A két bonyodalmat okozó lépés a konvolúció, illetve a pooling művelete.
- A derivált átvitele a pooling műveleten viszonylag egyszerű:
 - Pl. Max pooling esetén jelölje a maximumot adó inputot a vastag vonal:



- Ez esetben a kimentet a 3. input adja, a másik 3 neuron hozzájárulása 0
- Ezért a derivált értéke a 3. neuron esetén 1 lesz, a többi esetén 0
- Vagyis a hibát a „győztes” útvonalon terjesztjük csak vissza, a többi útvonalon nem

Technikai megjegyzés (2)

- A derivált visszavezetése a konvolúció műveletén jóval bonyolultabb: Egy input többször is felhasználásra kerül
 - Nem csupán a különböző neuronok által
 - Hanem ugyanazt azt inputot ugyanaz a neuron is többször felhasználja, különböző súlyokkal (ld. ábra, 1D-konvolúcióra)
- Az adott neuron hibája ezen hibaértékek összege lesz
- Hosszadalmas levezetés után azt kapjuk, hogy a hiba-visszaterjesztésnél is egy konvolúciós műveletet kell elvégezni
- További részletek itt:
 - <https://grzegorzwardys.wordpress.com/2016/04/22/8/>





2. Megjegyzés

- A háló struktúrájának speciális megválasztásával külső tudást viszünk a rendszerbe. Például a konvolúció+pooling műveletekkel *explicit* módon visszük be azt a tudást, hogy az eltolás nem változtatja meg egy objektum azonosságát.
 - Hátrány: pontos tudással kell rendelkezünk arra nézve, hogy hogyan kell a modellt megfelelően módosítani
- Ugyanezt a tudást *implicit* módon is be lehetne vinni, például a data augmentation technikával (rengeteg mesterséges példát generálni, ahol ugyanaz az objektum különböző pozíciókon szerepel, de kép címkéje mégis ugyanaz)
 - Hátrány: Jelentősen megnő a tanítópéldák száma és a tanítási idő
- A tanítóadatok mennyiségének növekedésével elvileg egyre csökkenni fog a különbség a speciális hálók (CNN) és az egyszerűbb hálók között

Felismerési példa

- Komplex képfelismerési feladat: többféle objektum is lehet a képen
 - ImageNet adatbázis: 1,2 millió kép, 1000 felismerendő alakzat (címke)
 - Az algoritmus 5-öt tippelhet, helyesnek tekintjük a választ, ha a helyes címke ezek között van



cheetah

cheetah

leopard

snow leopard

Egyptian cat



Let Train is like a plane, with in-train magazine and a seat that you can plug your headphones into and listen to

bullet train

bullet train

passenger car

subway train

electric locomotive



hand glass

scissors

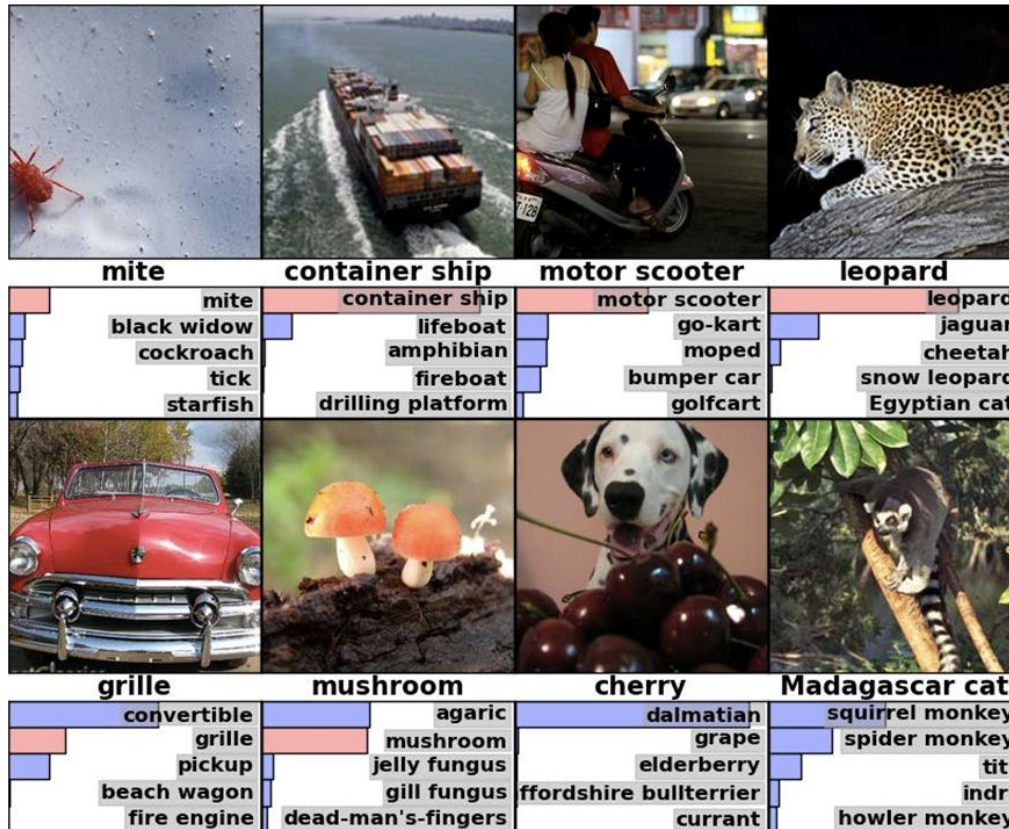
hand glass

frying pan

stethoscope

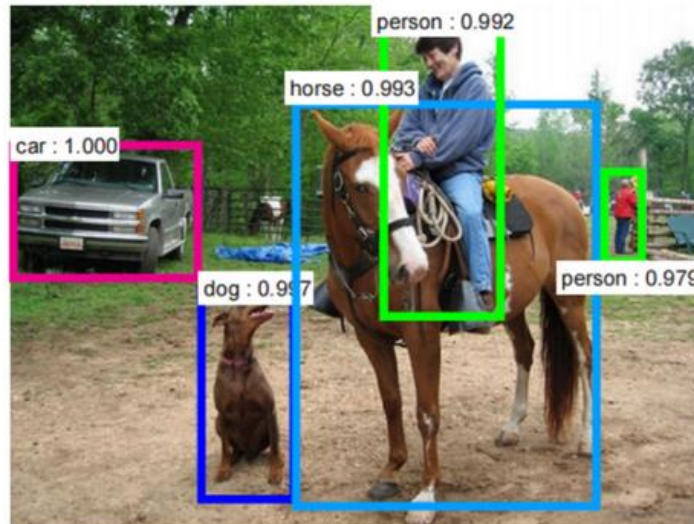
Felismerési példa (2)

- Konvolúciós hálók előtti legjobb eredmény: 26% hiba
- Első konv. háló (2012): 16% -os hiba
- jelenleg a legjobb eredmény: a hiba <5%



Objektum-detektálás

- Nagyon sok gyakorlati alkalmazásban nem csak egyetlen objektum lehet a képen, hanem több is (pl. önvezető autók)
- Ilyenkor a kimenet nem egyetlen címke, hanem:
 - Meg kell találni a képen levő objektum(ok) pozícióját
 - És azonosítani kell őket
- Párhuzamosan kell elvégezni az objektumok osztályozását és detektálását
- Erre is konvolúciós hálókat használnak, de speciális módosításokkal



KÖSZÖNÖM A FIGYELMET!

A tananyag az EFOP-3.5.1-16-2017-00004 pályázat támogatásával készült.

SZÉCHENYI  2020



MAGYARORSZÁG
KORMÁNYA

Európai Unió
Európai Szociális
Alap



BEFEKTETÉS A JÖVŐBE