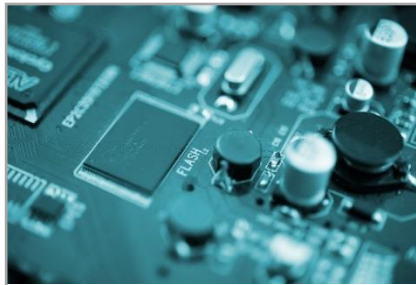




UNIVERZITA
PARDUBICE
FAKULTA
ELEKTROTECHNIKY
A INFORMATIKY

NNUI2

Základy umělé inteligence 2



Základy umělé inteligence 2

Ing. Dominik Štursa

Katedra řízení procesů

FEI, Univerzita Pardubice

Nám. Čs. legií 565

(1. patro, dveře č. 02 043)

e-mail: dominik.stursa@upce.cz

tel.: 466 037 124

Konvoluční neuronová síť (CNN)

Problémy zpracování obrazu

- Klasifikace – zařazení vstupního obrázku do příslušné třídy
- Lokalizace – nalezení objektu ve vstupním obrázku
- Detekce – souběžná lokalizace a klasifikace
- Segmentace – rozdělení vstupního obrázku na segmenty
 - Sémantická – každý pixel je přiřazen do určité třídy
 - Instanční – označuje pixely odpovídající jednotlivým instancím objektu

Problémy zpracování obrazu

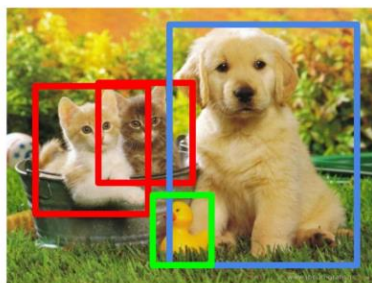
- Klasifikace – zařazení vstupního obrázku do příslušné třídy
- Lokalizace – nalezení objektu ve vstupním obrázku
- Detekce – souběžná lokalizace a klasifikace
- Segmentace – rozdělení vstupního obrázku na segmenty



CAT



CAT



CAT, DOG, DUCK



CAT, DOG, DUCK

Konvoluční neuronová síť – myšlenka

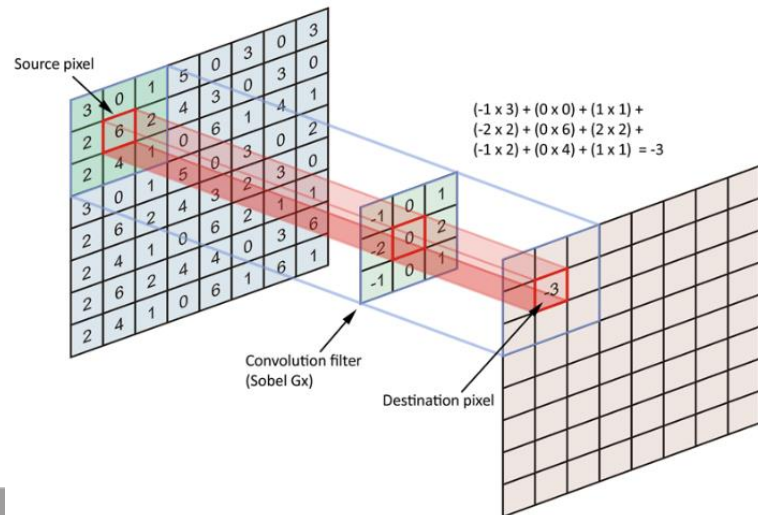
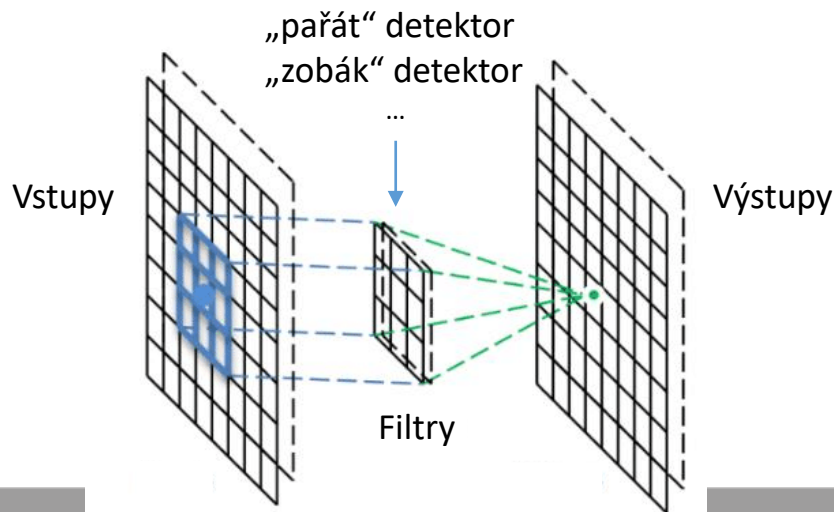
- Typicky člověk vybírá v datech specifické příznaky – hrany, tvary, barvy
- V datech se obecně nacházejí vzory (vlastnosti) definující dané objekty
- Tyto vzory mohou být různých velikostí a různě umístěné v datech
- → návrh skupiny detektorů, které procházejí zpracovávaná data
- Jednotlivé části dat mohou být kódovány stejným detektorem (stejným způsobem)

Konvoluční neuronová síť – myšlenka



Konvoluční neuronová síť - úvod

- Jednotlivé detektory jsou realizované pomocí konvoluční vrstvy
- Konvoluční vrstva je představovaná skupinou filtrů provádějících operaci konvoluce



Konvoluční neuronová síť – konvoluce

- 1 konvoluční vrstva je tvořena definovaným počtem filtrů, které detekují vzory definované velikosti (velikost jádra), váhy filtrů jsou získány trénováním.

Obrázek
6x6

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

Filtr 1 (3x3)

1	-1	-1
-1	1	-1
-1	-1	1

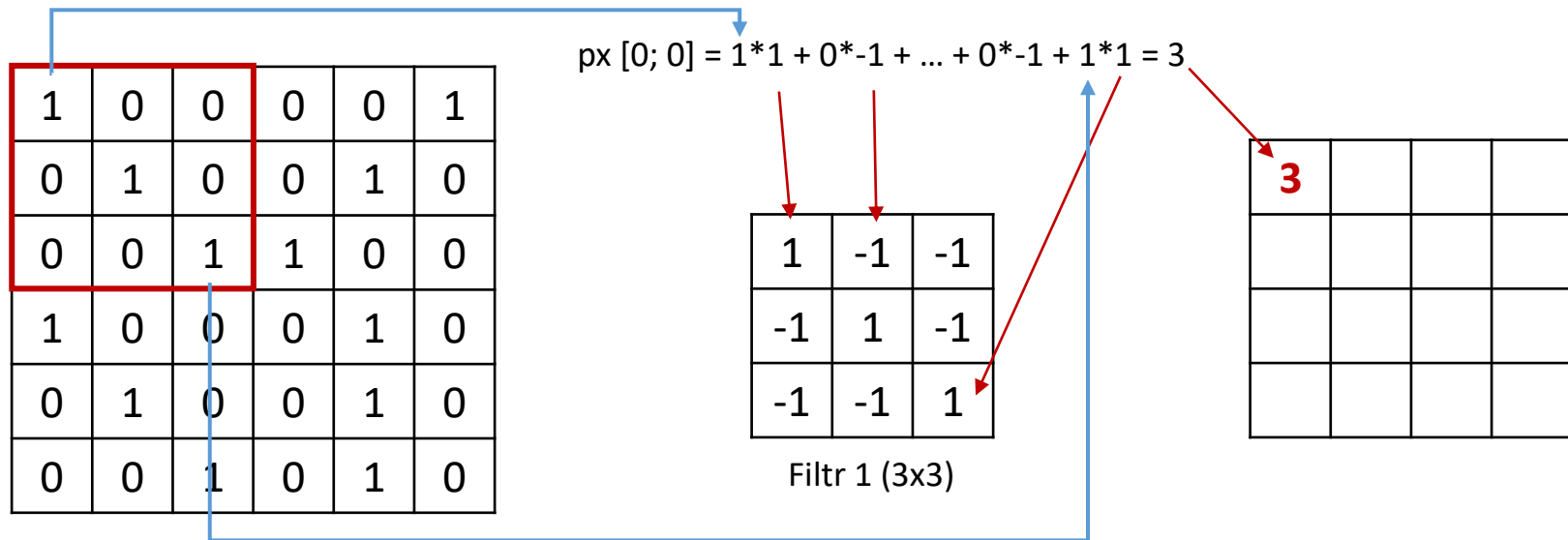
⋮

Filtr n (3x3)

-1	1	-1
-1	1	-1
-1	1	-1

Konvoluční neuronová síť – konvoluce

- Pro každý pixel aplikujeme skalární součin konvolučního filtru s vstupem a získáváme výstupní hodnotu tvořící matici příznaků



Konvoluční neuronová síť – konvoluce

- Pro každý pixel aplikujeme skalární součin konvolučního filtru s vstupem a získáváme výstupní hodnotu tvořící matici příznaků

Stride = 1 (po kolika pixelech brát další blok)

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

1	-1	-1
-1	1	-1
-1	-1	1

Filtr 1 (3x3)

3	-1		

Konvoluční neuronová síť – konvoluce

- Pro každý pixel aplikujeme skalární součin konvolučního filtru s vstupem a získáváme výstupní hodnotu tvořící matici příznaků

Stride = 1 (po kolika pixelech brát další blok)

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

1	-1	-1
-1	1	-1
-1	-1	1

Filtr 1 (3x3)

3	-1	-3	

Konvoluční neuronová síť – konvoluce

- Pro každý pixel aplikujeme skalární součin konvolučního filtru s vstupem a získáváme výstupní hodnotu tvořící matici příznaků

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

1	-1	-1
-1	1	-1
-1	-1	1

Filtr 1 (3x3)

3	-1	-3	-1
-3	1	0	-3
-3	-3	0	1
3	-2	-2	-1

Konvoluční neuronová síť – konvoluce

- Matice příznaků určuje, kde se v původním obrázku vyskytují dominantní příznaky shodné s těmi ve filtru
- Určuje, kde v obrázku jsou přítomné dané tvary

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

1	-1	-1
-1	1	-1
-1	-1	1

Filtr 1 (3x3)

Matice příznaků – filtr 1

3	-1	-3	-1
-3	1	0	-3
-3	-3	0	1
3	-2	-2	-1

Konvoluční neuronová síť – konvoluce

- Matice příznaků určuje, kde se v původním obrázku vyskytují dominantní příznaky shodné s těmi ve filtru
- Určuje, kde v obrázku jsou přítomné dané tvary

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

-1	1	-1
-1	1	-1
-1	1	-1

Filtr n (3x3)

Matice příznaků – filtr n

-1	-1	-1	-1
-1	-1	-2	1
-1	-1	-2	1
-1	0	-4	3

Konvoluční neuronová síť – konvoluce

- Aplikací všech filtrů pak dochází k zisku mapy příznaků

Obrázek
6x6

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

Filtr 1 (3x3)

⋮

Filtr n (3x3)

Mapa příznaků

jednotlivé matice
příznaku za sebou

-1	-1	-1	-1
-1	-1	-2	1
-1	-1	-2	1
-1	0	-4	3

Konvoluční neuronová síť – konvoluce

- Padding (vyplňování) vs. Normálně – dochází ke snížení velikosti

Obrázek 6x6

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



Filtr 1 (3x3)

1	-1	-1
-1	1	-1
-1	-1	1



Matice příznaků 4x4

3			

Konvoluční neuronová síť – konvoluce

- Padding = Same – Zachování velikosti obrazu

Obrázek 6x6 + výplň (celkem 8x8)

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

Filtr 1 (3x3)

1	-1	-1
-1	1	-1
-1	-1	1

Matice příznaků 6x6

	3				

Konvoluční neuronová síť – konvoluce

- U barevného obrázku (RGB) se provádí pro každý kanál
- Ziskem jsou mapy příznaků (dominantních vlastností vstupních dat)

Obrázek 6x6x3 (RGB)

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

Filtr 1 (3x3)

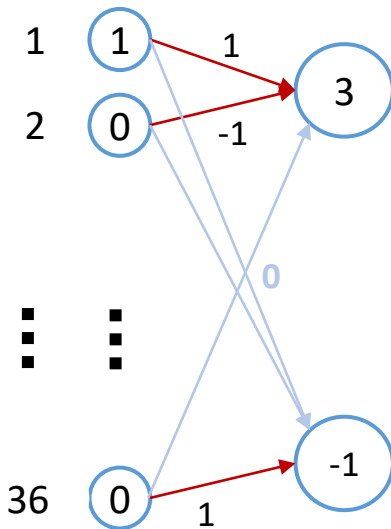
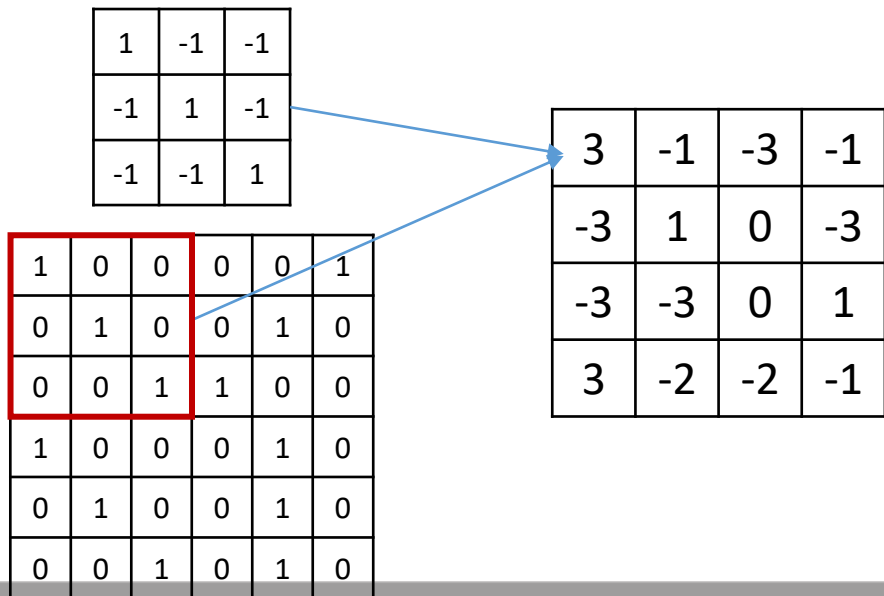
1	-1	-1
-1	1	-1
-1	-1	1

Mapa příznaků 6x6x3

-1	-1	-1	-1
-1	-1	-2	1
-1	-1	-2	1
-1	0	-4	3

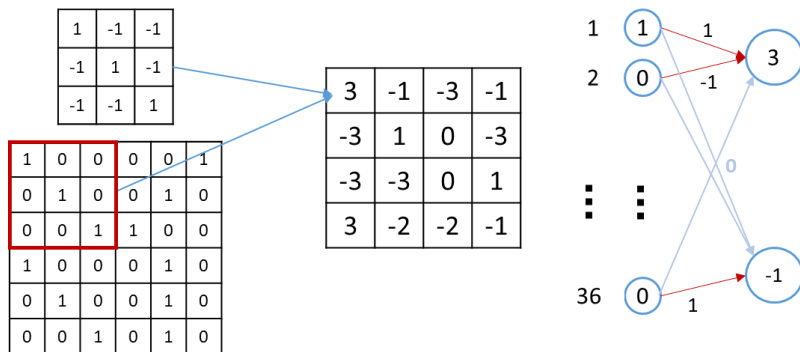
Konvoluční neuronová síť – motivace

- Konvoluční vrstva „sama“ extrahuje klíčové vlastnosti
- Konvoluční vs. plně propojená vrstva



Konvoluční neuronová síť – motivace

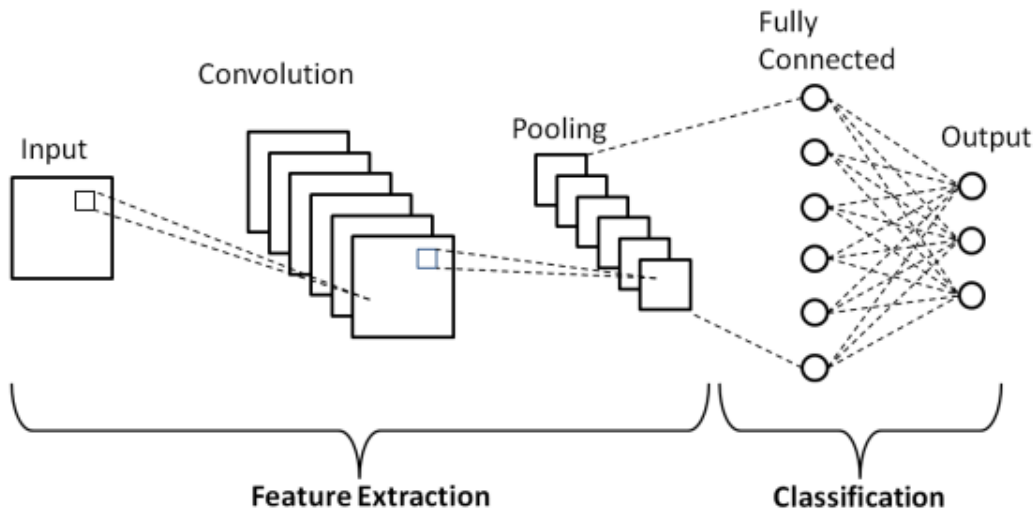
- Konvoluční vrstva není plně propojená (neuron ve skryté vrstvě – příznak je definován počtem vah odpovídající velikosti filtru)
- Dále dochází ke sdílení vah – ještě méně parametrů
- FFNN a CNN se stejným počtem neuronů má méně vah k učení



CNN – Klasifikace

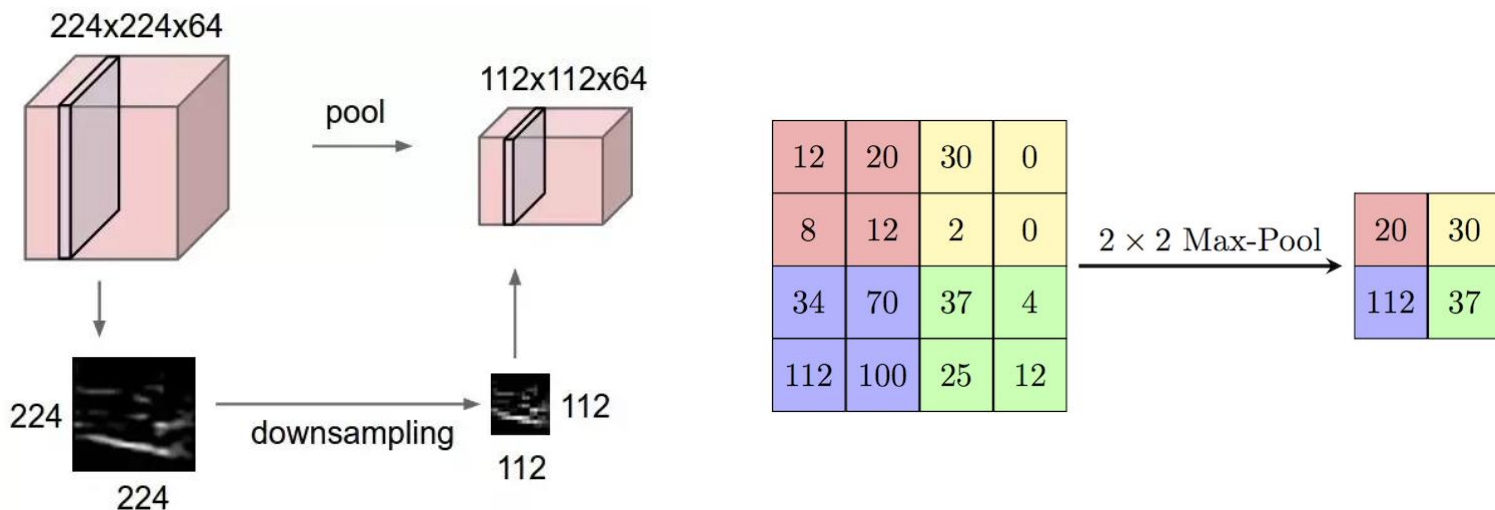
Konvoluční neuronová síť – struktura

- CNN je obecně složená z konvolučních, pooling, flatten, a dense (plně propojených) vrstev



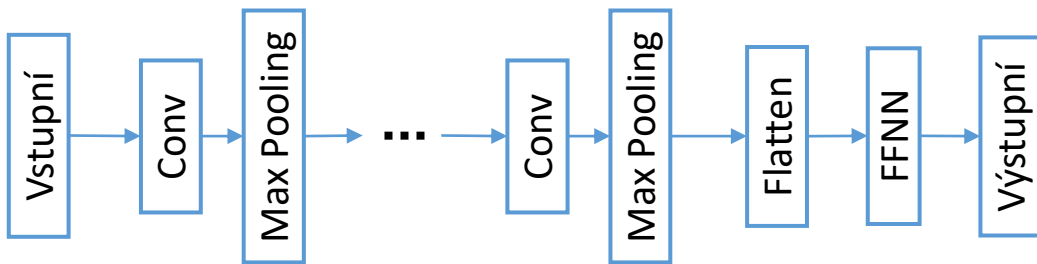
Konvoluční neuronová síť – struktura

- Pooling vrstvy sjednocují skupinu vlastností a vybírají jen tu nejvíce dominantní – max pooling vybírá maximální hodnotu, snižují velikost



Konvoluční neuronová síť – topologie

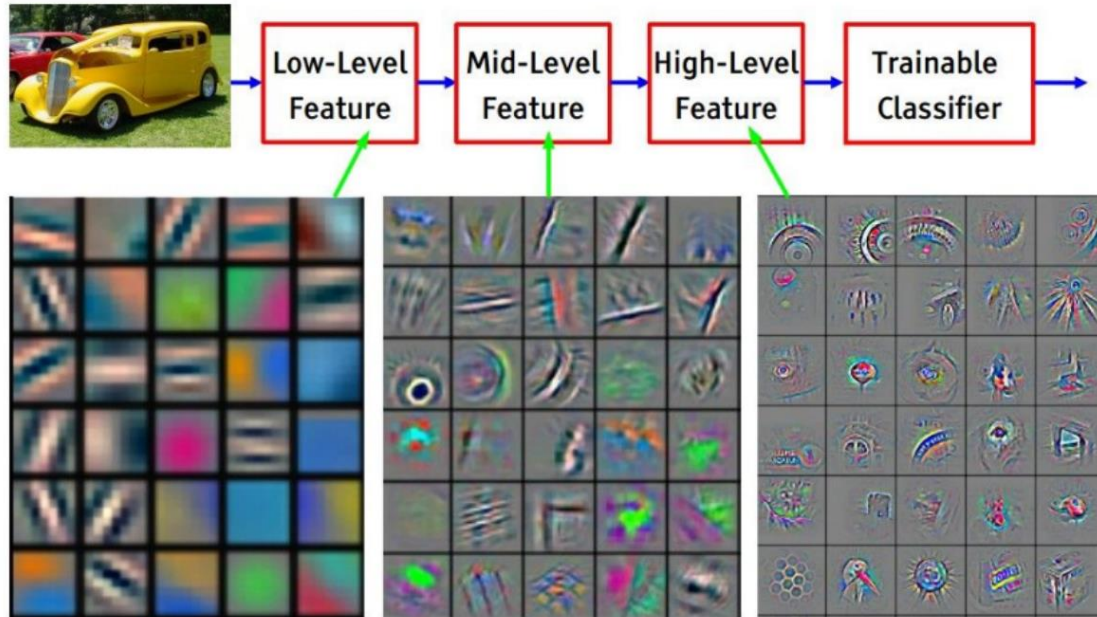
- Jednotlivé CNN se skládají z párů Conv + MaxPooling



- Samotných topologií CNN je velké množství a návrh samotné topologie je již značně komplexní
- Běžnou inženýrskou praktikou je výběr z již existujících topologií vhodných pro řešení daného problému

Konvoluční neuronová síť – topologie

- Jednotlivé CNN se skládají z párů Conv + MaxPooling



Konvoluční neuronová síť – vyhodnocení

- K vyhodnocení klasifikačních úloh se používá Confusion Matrix (Matice záměn)

		Actual Values	
		Positive (1)	Negative (0)
Predicted Values	Positive (1)	TP	FP
	Negative (0)	FN	TN

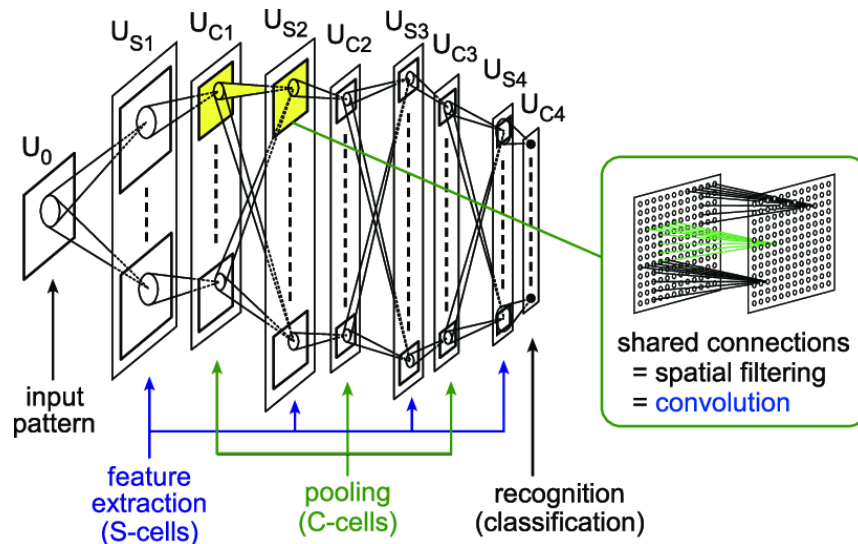
airplane	828	13	12	11	18	0	2	4	85	27
automobile	10	910	0	5	1	1	0	1	11	61
bird	47	1	708	64	88	14	63	4	8	3
cat	3	4	16	768	33	93	50	19	4	10
deer	10	0	39	43	788	12	57	43	6	2
dog	2	0	10	117	29	777	8	33	0	4
frog	7	2	10	54	29	7	888	1	1	1
horse	24	2	14	39	76	17	4	818	2	4
ship	27	13	0	7	3	0	3	0	933	14
truck	19	64	1	7	2	1	1	0	18	887
airplane	airplane	automobile	bird	cat	deer	dog	frog	horse	ship	truck

Konvoluční neuronová síť – historie

- 1959 - Simple and Complex Cells - [David Hubel](#) and [Torsten Wiesel](#)
- Zkoumali lidské zrakovém ústrojí a navrhli, že existují určité druhy buněk, které člověk využívá při rozpoznávání vzorů.
- Simple cell (S-Buňka) reaguje na hrany a pruhy určité orientace v daném perceptivním poli.
- Complex cell (C-Buňka) také reaguje na hrany a pruhy určité orientace, ale od jednoduché buňky se liší tím, že tyto hrany a pruhy mohou být na scéně posunuty a buňka bude stále reagovat.

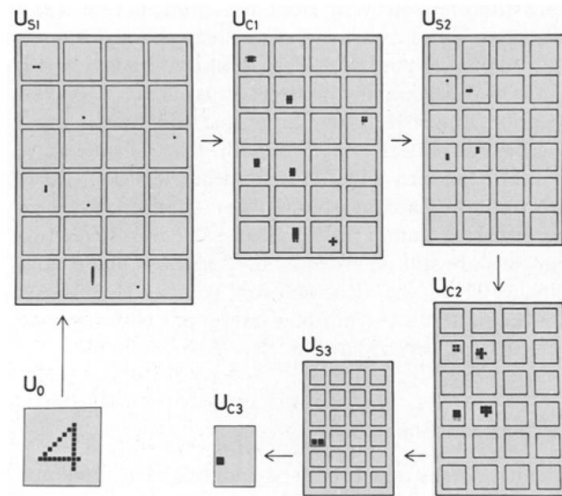
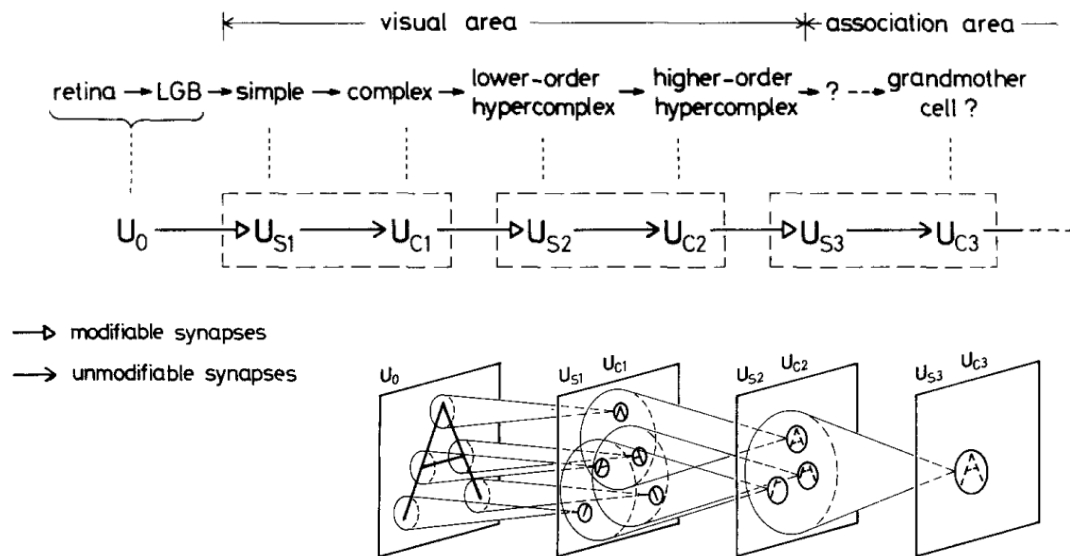
Konvoluční neuronová síť – historie

- 1980 - Neocognitron - Kunihiro Fukushima
 - [Neocognitron: A Self-organizing Neural Network Model for a Mechanism of Pattern Recognition Unaffected by Shift in Position.](#)
 - Model zahrnuje komponenty označované jako S-buňky a C-buňky realizující matematické operace.
 - Celkovou myšlenkou je zachytit koncept "od jednoduchého ke složitějšímu" a přeměnit jej na výpočetní model pro rozpoznávání vizuálních vzorů.



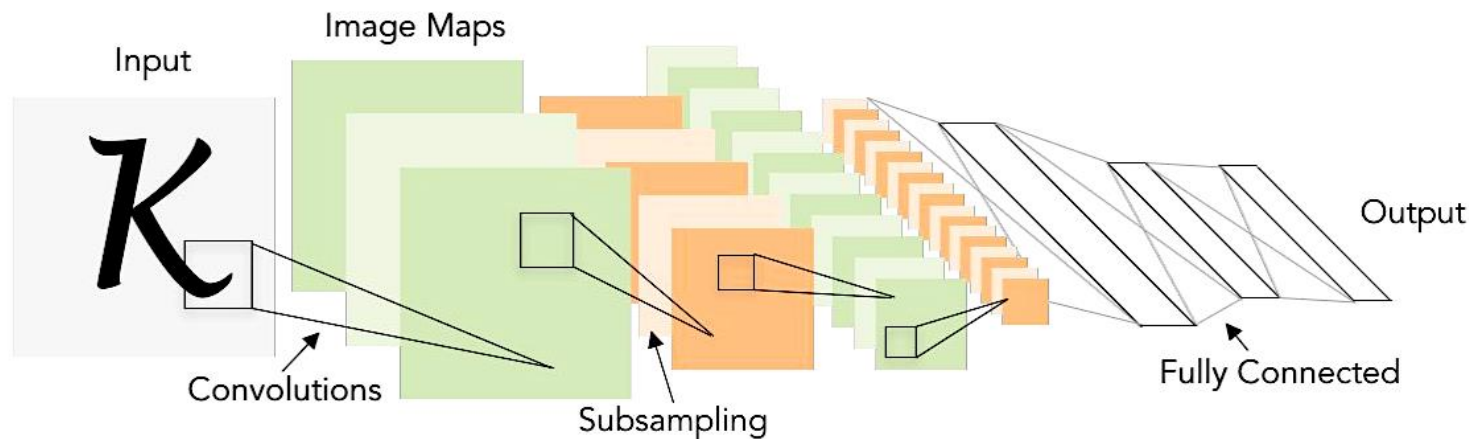
Konvoluční neuronová síť – historie

- 1980 - Neocognitron - Kunihiro Fukushima



Konvoluční neuronová síť – historie

- 1998 - LeNet - Yann LeCun
 - [Gradient-Based Learning Applied to Document Recognition](#)

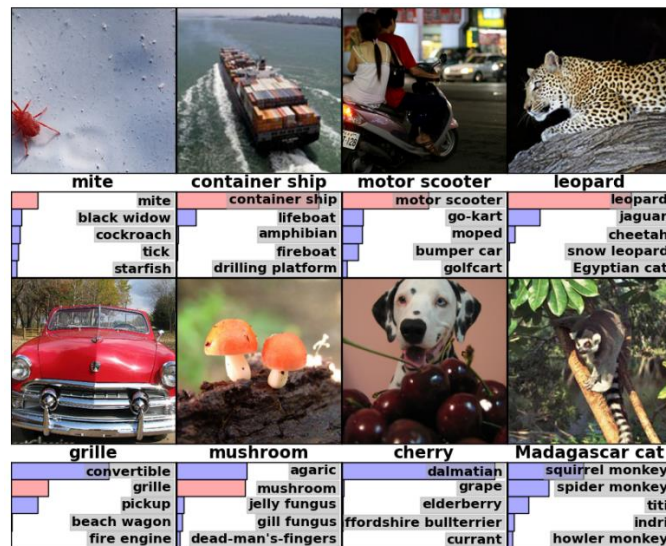
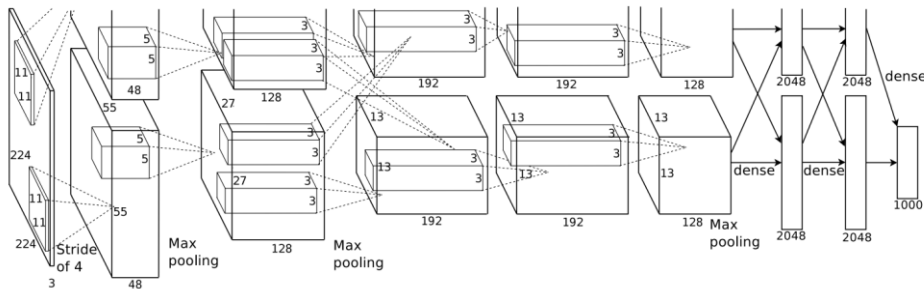


Konvoluční neuronová síť – historie

- 2009 – [IMAGENET](#) – Fei-Fei Li & Team
 - První rozsáhlý dataset (1000 tříd; 1,3 milionu obrázků) vyzívající k soutěžení v klasifikaci obrazových dat mezi výzkumníky
 - Díky soutěžení vznikají návrhy nových topologií klíčových pro rozvoj aplikace neuronových sítí pro zpracování obrazu
- 2012 – [AlexNet](#) – Alex Krizhevsky
 - ReLu aktivační funkce, velký počet filtrů v konvolučních vrstvách
 - Paralelní trénování architektury
 - Augmentace dat
 - Zavedení Dropout vrstvy – nastavení výstupu neuronů s určitou pravděpodobností na hodnotu 0. (Omezení vzájemných vztahů neuronů → neuron se nemůže spoléhat na přítomnost jiných neuronů)

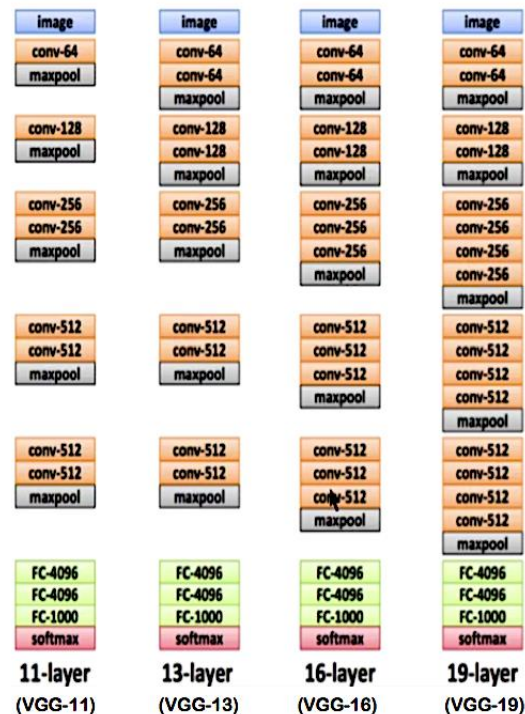
Konvoluční neuronová síť – historie

- 2012 – [AlexNet](#) – Alex Krizhevsky
 - Překrývající se části z Pooling vrstev



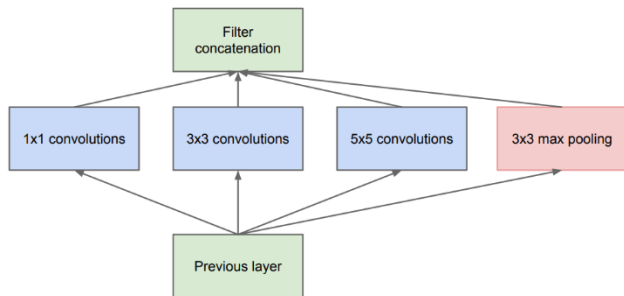
Konvoluční neuronová síť – historie

- 2013 – ZFNet – Zeiler and Fergus
 - Založený na AlexNet, ale měnící nastavení jednotlivých vrstev
- 2014 – [VGGNet](#) – Visual Geometry Group
 - Přineslo hlubší architekturu CNN, která dosahovala nižší chybovosti v soutěži na datasetu ImageNet.
 - Nová filozofie – zvětšením hloubky lze modelovat více nelinearit ve funkci → zohledňování hloubky jako kritické složky při návrhu topologie.

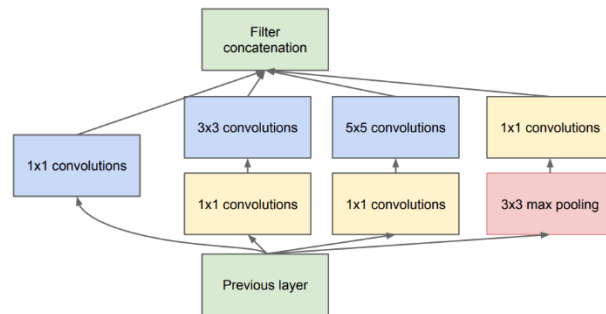


Konvoluční neuronová síť – historie

- 2014 – [GoogLeNet](#)
 - Trend zvětšování hloubky sítě, ale bez použití plně propojených vrstev (12x méně parametrů než u AlexNet, 28x méně parametrů než u VGG)
 - Zavedení „inception“ modulu - cílem je aproximovat optimální lokální struktur CNN. Umožňuje použít v jednom bloku více velikostí filtrů, místo abychom byli omezeni na jednu velikost filtru, které pak spojíme a předáme další vrstvě.

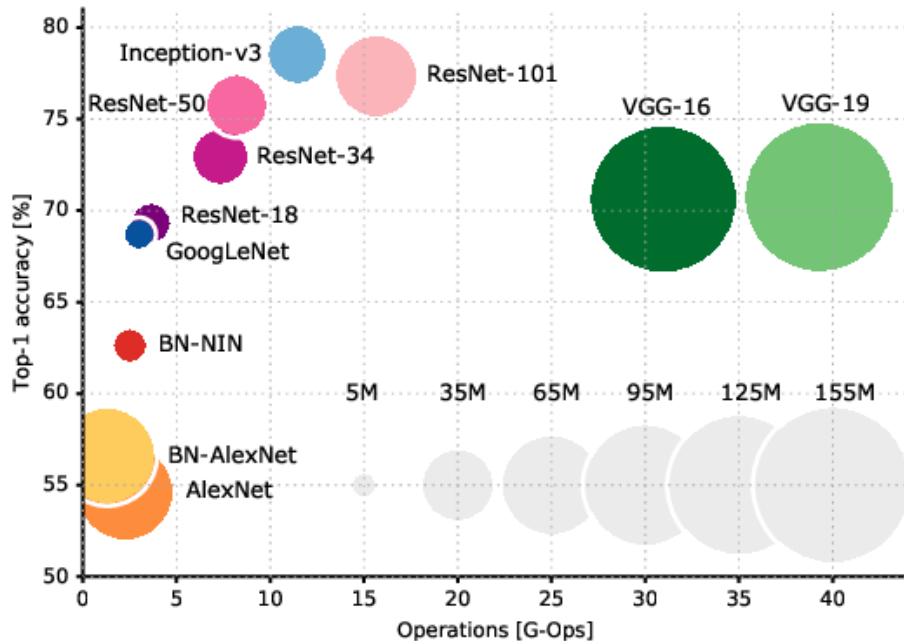
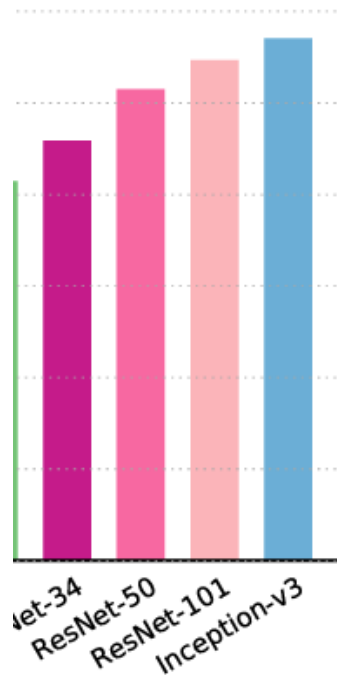


(a) Inception module, naïve version



(b) Inception module with dimension reductions

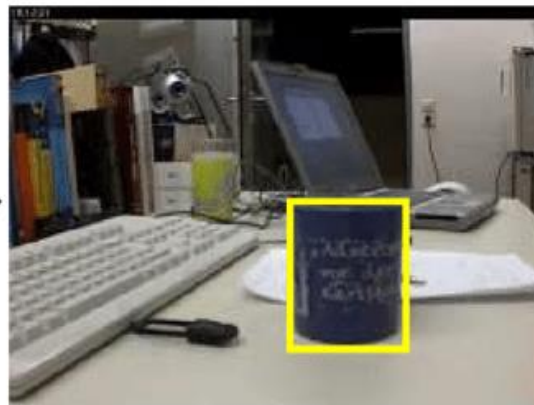
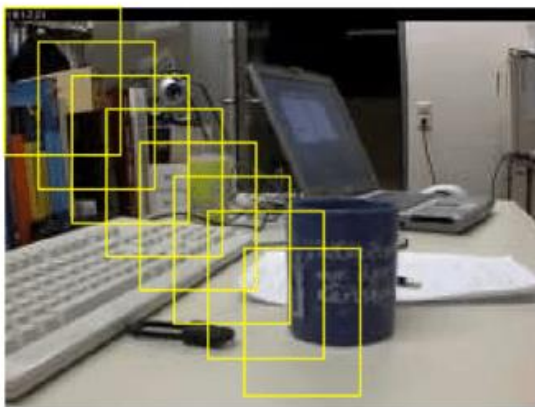
Konvoluční neuronová síť – porovnání



CNN – Detekce

Konvoluční neuronová síť – detekce

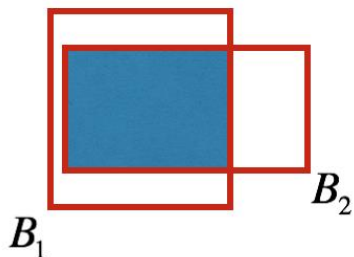
- Základní přístup pro detekci objektů v obraze je použití klasifikátoru na tzv. pohyblivé okno (sliding window)
- Uvažují se tzv. ohraničující obdélníky pro detekci pozice objektu



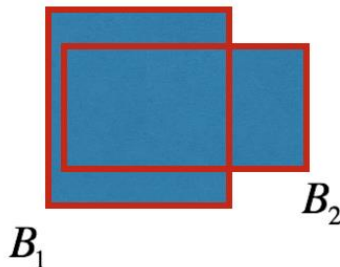
Konvoluční neuronová síť – vyhodnocení

- K vyhodnocení detekčních úloh se používá metrika IoU - Intersection over Union

Intersection



Union



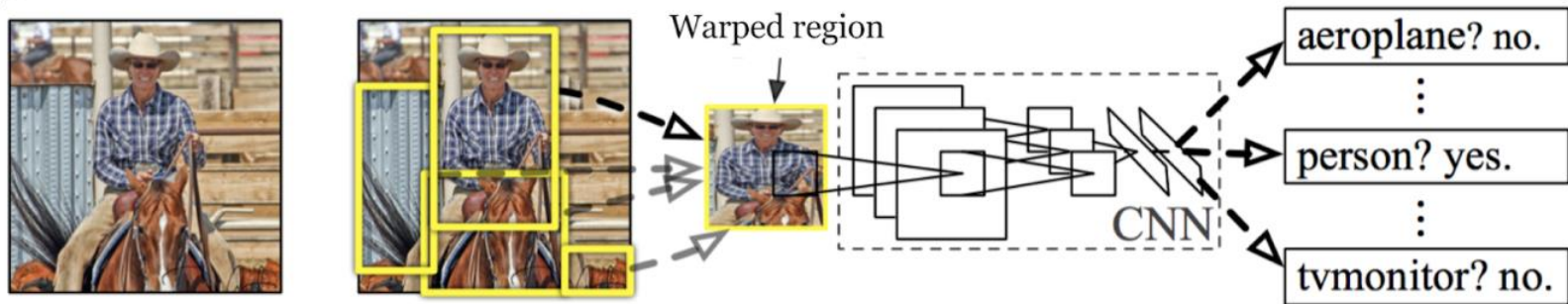
Intersection over Union

$$IoU = \frac{B_1 \cap B_2}{B_1 \cup B_2} = \frac{\text{Intersection}}{\text{Union}}$$

A diagram illustrating the IoU calculation. It shows two overlapping rectangles, B_1 and B_2 . The intersection is shaded blue, and the union is shaded light blue. The ratio of the intersection area to the union area is represented by the fraction above the diagram.

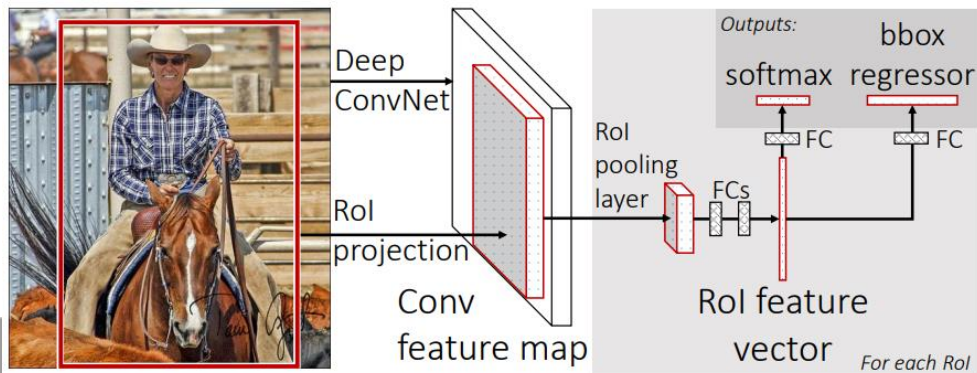
Konvoluční neuronová síť – R-CNN

- 2014 – [R-CNN](#)
 - 2 Fázový detektor (přístupuje k problému rozdělením na fáze)
 - Projde obrázek CNN a určí návrh oblasti zájmu, pak každý návrh projde sítí pro klasifikaci



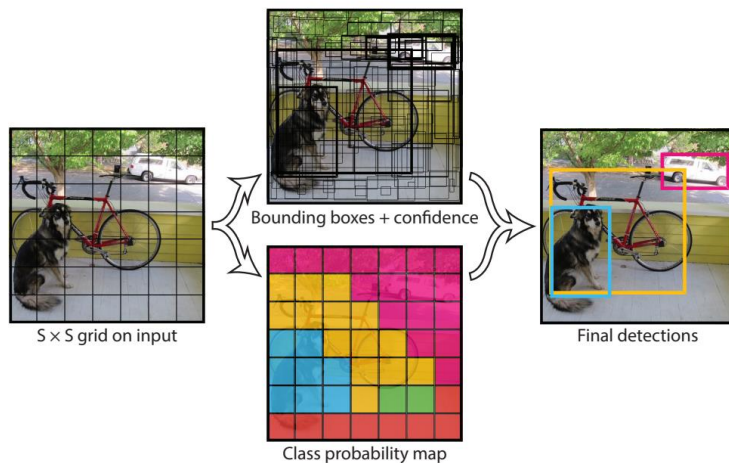
Konvoluční neuronová síť – Fast R-CNN

- 2015 – [Fast R-CNN](#)
 - Kombinuje 2 fáze → Jednou projet obrázek CNN + až potom řešit jednotlivé regiony
 - Postup detekce:
 - Celý vstupní obrázek projde jednou hlubokou CNN (např. VGG-16). Výstup = Feature mapa (popis obrázku).
 - Na feature mapě se aplikují tzv. Region Proposal Regions (RoIs) — oblasti, kde by mohly být objekty
 - Pro každý RoI: Použije se speciální vrstva → RoI Pooling: Převzorkuje fixní velikosti (např. 7×7). (Pro FFNN)
 - Výstup z RoI Poolingu jde do plně propojené vrstvy (klasifikační head).



Konvoluční neuronová síť – YOLO

- 2016 – [YOLOv1](#)
 - You Only Look Once – Přináší nový přístup kombinující fáze z 2 fázových detektorů
 - Obrázek se rozdělí na 7×7 (v originále) buněk a každá buňka předpovídá několik bounding boxů včetně pravděpodobnosti, že tam objekt je, a klasifikaci do tříd.



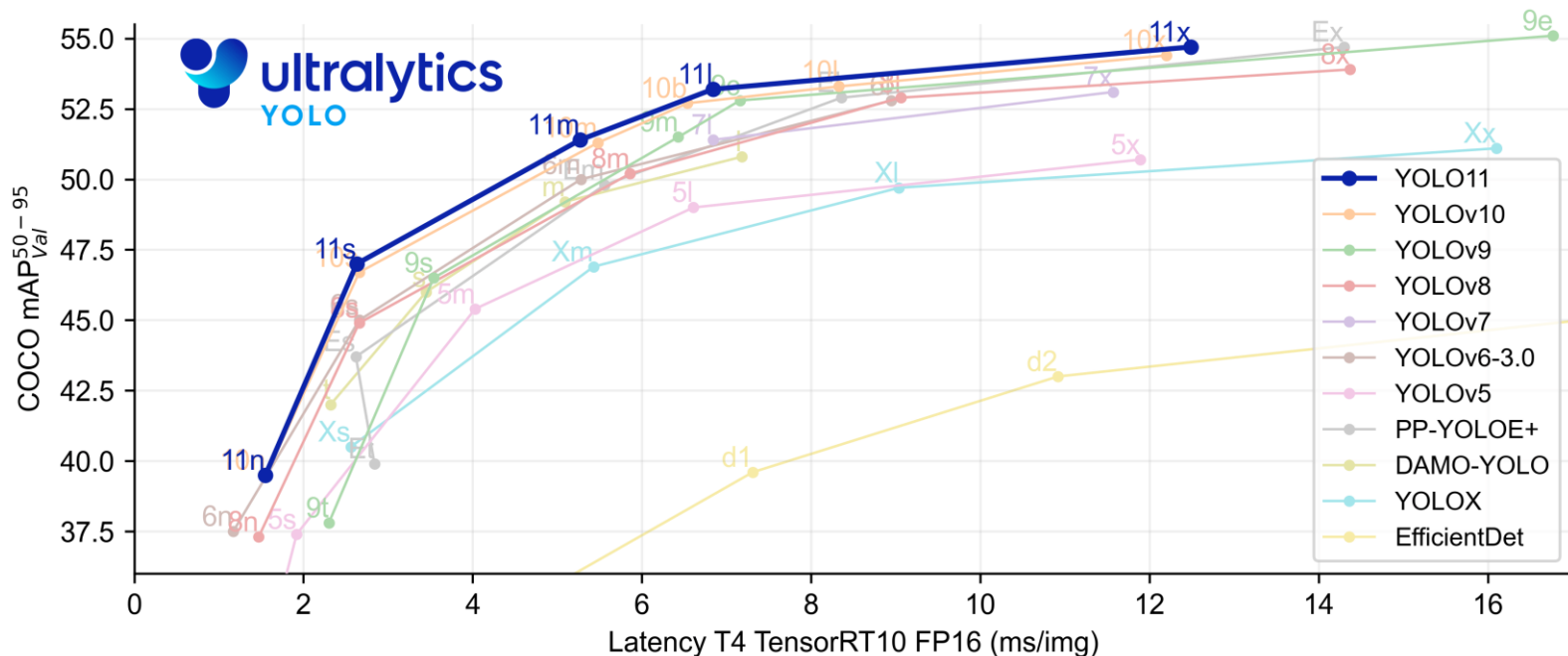
Konvoluční neuronová síť – YOLO

- 2018 – [YOLOv3](#)
 - Navazuje na v2, s vylepšením díky využití tzv. „anchor boxů“ místo bounding boxů
 - Bounding box**
 - Výsledný obdélník kolem objektu. Vzniká po výpočtu/predikci modelu.
 - Anchor Box**
 - Předdefinované tvary boxů, které model používá jako „startovní šablony“ pro predikci bounding boxů.
 - Využití více úrovní map příznaků
 - Použití „multi-scale“ přístupu (tři úrovně detekce) → lepší detekce menších obj.
 - Počátek zavádění tzv. „páteřních“ sítí (Backbone)
 - Část modelu, která extrahuje feature mapy.
 - Typicky předtrénovaná na ImageNet (klasifikace).
 - Detektor pak přidává jen „hlavičku“ (head) pro detekci objektů.

Konvoluční neuronová síť – YOLO

- [YOLO](#), [YOLOv8](#)
 - Rozšíření zaměřující se zejména na zlepšení přesnosti a udržení real-time inference
 - Přejít pod Open-source platformu Ultralytics
 - Různé implementace + snadné vlastní implementace (PyTorch, připravené skripty)
 - Různé velikosti sítě (U, S, M, L, X)
 - Testování a optimalizace backbone
 - Pokročilá augmentace pro zlepšení schopnosti generalizovat

Konvoluční neuronová síť – porovnání



CNN – Segmentace

Segmentace obrazových dat

- Pokročilejší metoda, přiřazující třídu (label) jednotlivým pixelům
- Instanční vs Sémantická



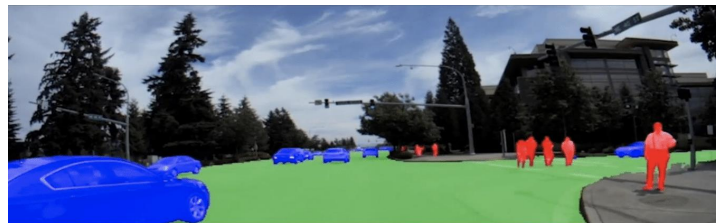
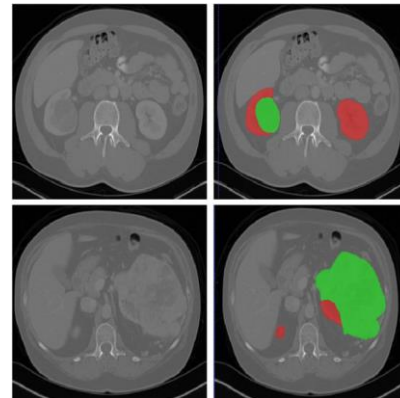
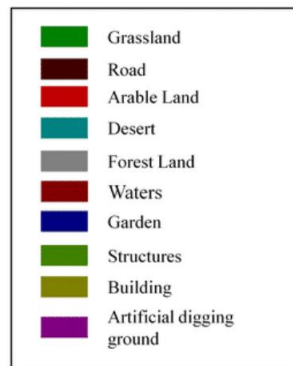
Semantic Segmentation



Instance Segmentation

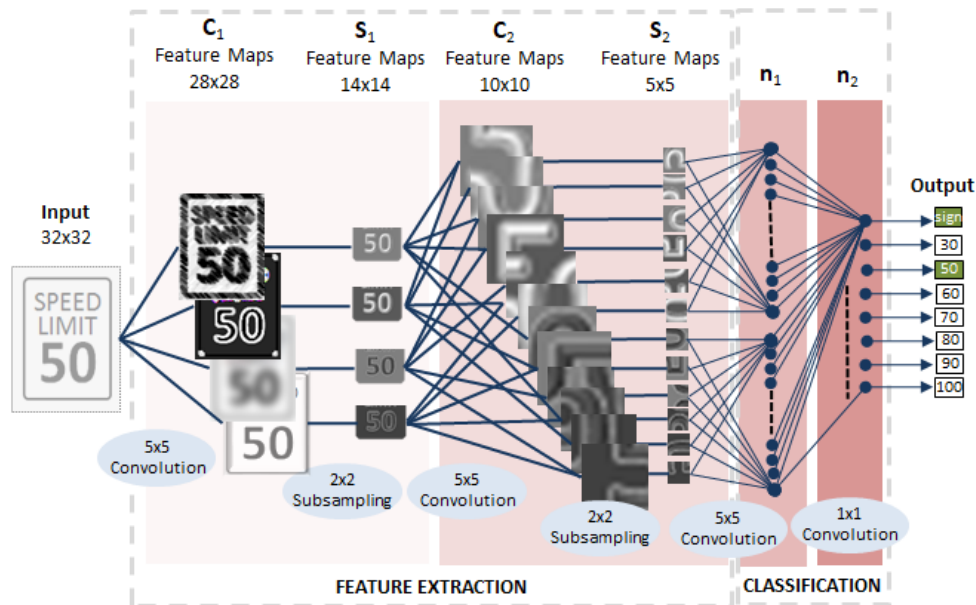
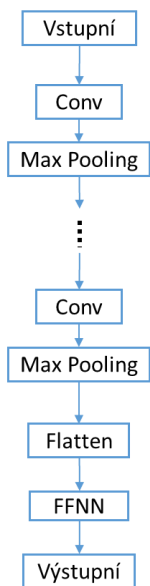
Segmentace obrazových dat – užití

- Medical imaging, Self-driving cars, Remote Sensing



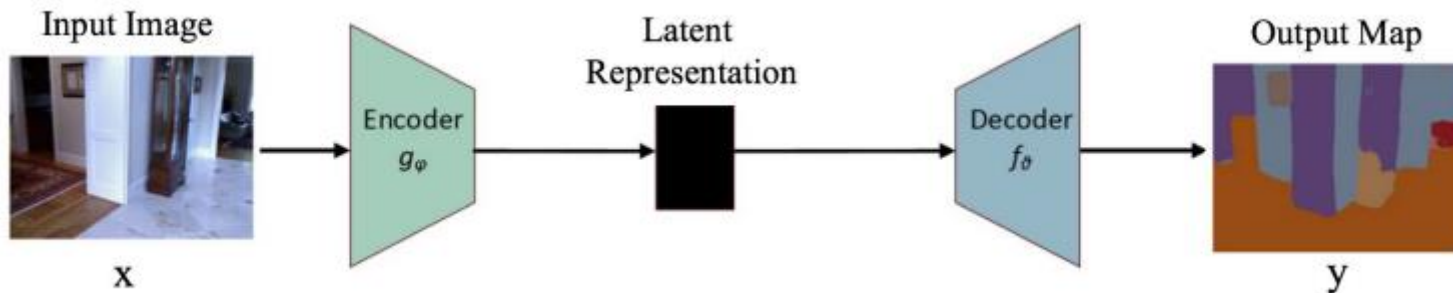
Segmentace pomocí CNN

- Konvoluční vrstvy produkují výstupní data – zpracované obrázky



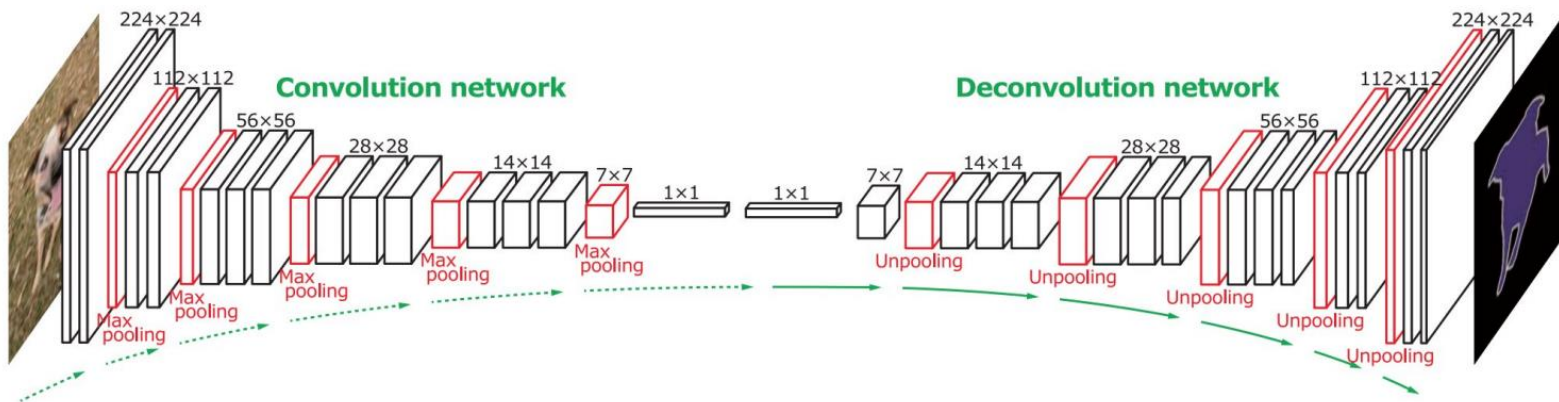
Segmentace pomocí CNN

- U segmentace vyžadujeme na výstupu označený obrázek.
- Modely založené na principu Enkodér-Dekodér nebo Auto-Enkodér.
- Jsou založené na vlastnosti konvolučních sítí kódovat vstupní informaci (obrázek).



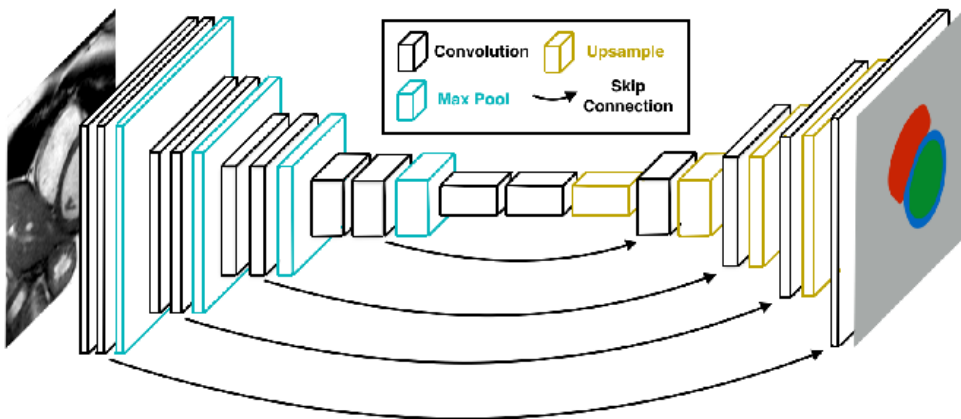
Segmentace pomocí CNN

- Enkodér-Dekodér schéma – postupné kódování vlastností obrázku a následné jeho postupné dekódování pro vygenerování segmentovaného obrázku



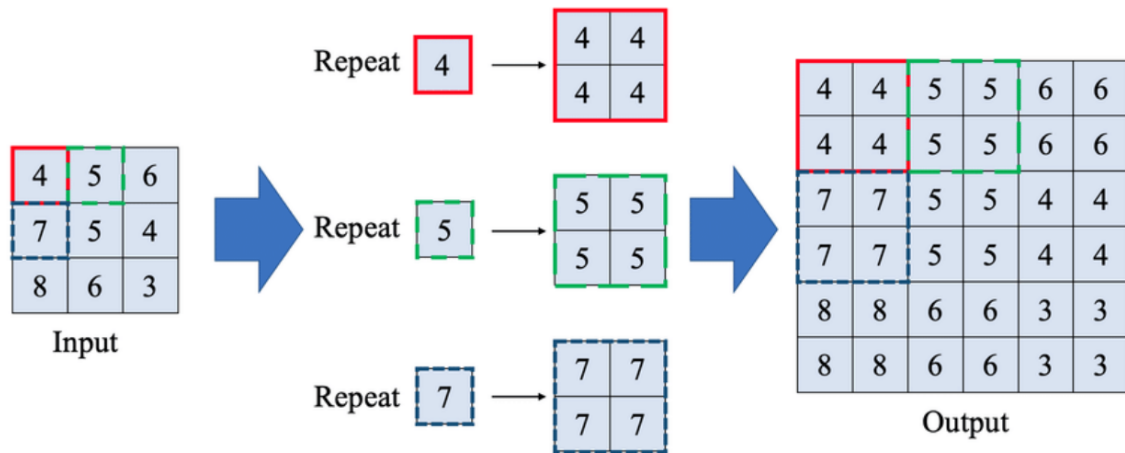
Segmentace pomocí CNN

- Skip spojení – přenášejí určité klíčové vlastnosti mezi enkodérem a dekodérem v dané hloubce



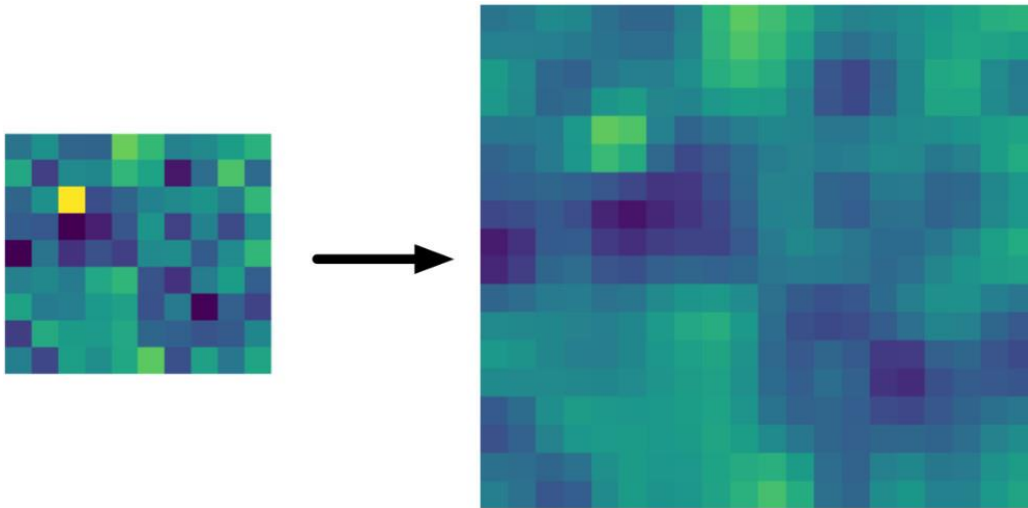
Segmentace pomocí CNN

- UpSampling vrstva – zvětšuje původní rozměr vstupu
- Metody: **replikace**, průměrování, Unpooling



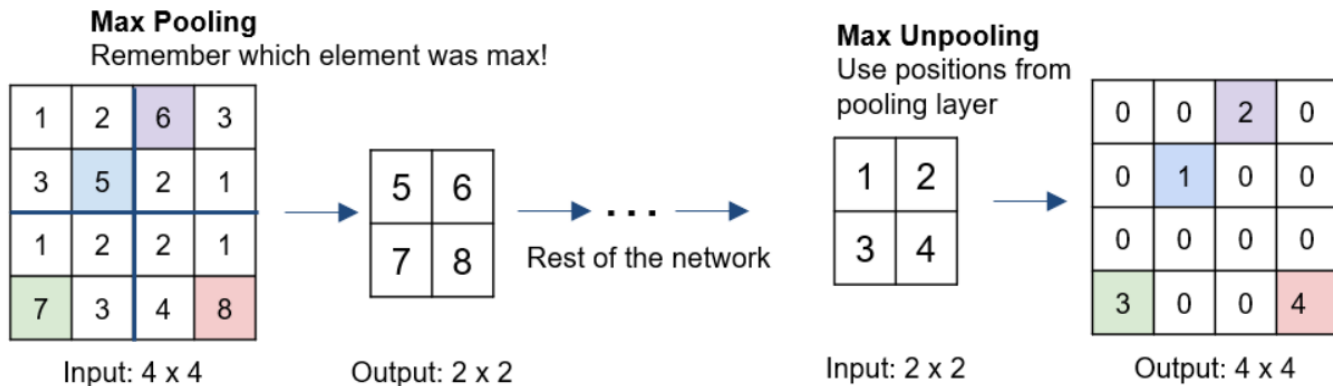
Segmentace pomocí CNN

- UpSampling vrstva – zvětšuje původní rozměr vstupu
- Metody: replikace, **průměrování**, Unpooling



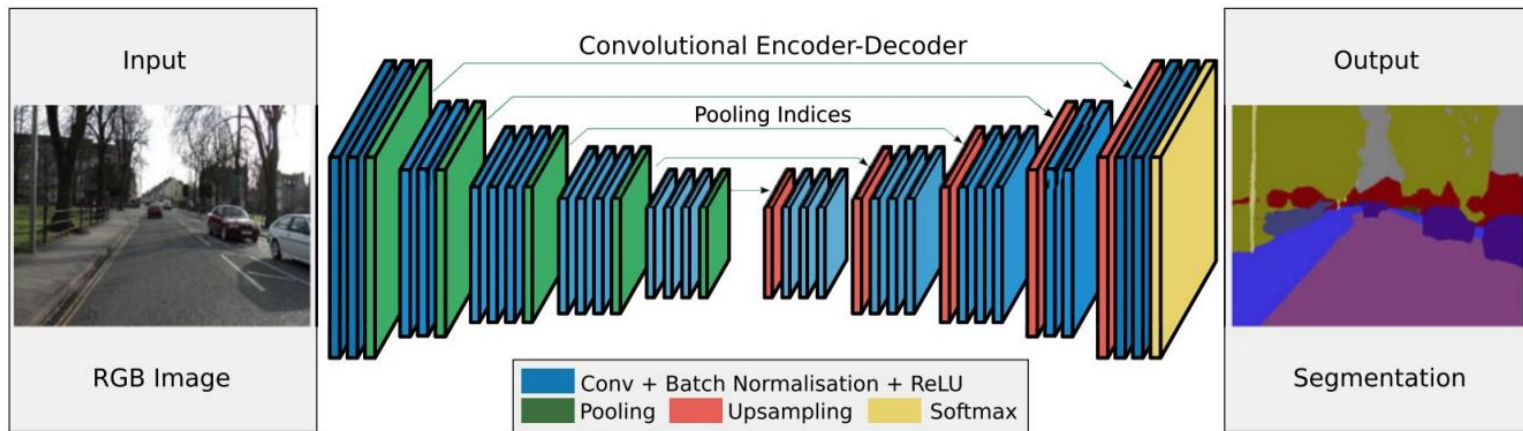
Segmentace pomocí CNN

- UpSampling vrstva – zvětšuje původní rozměr vstupu
- Metody: replikace, průměrování, **Unpooling**



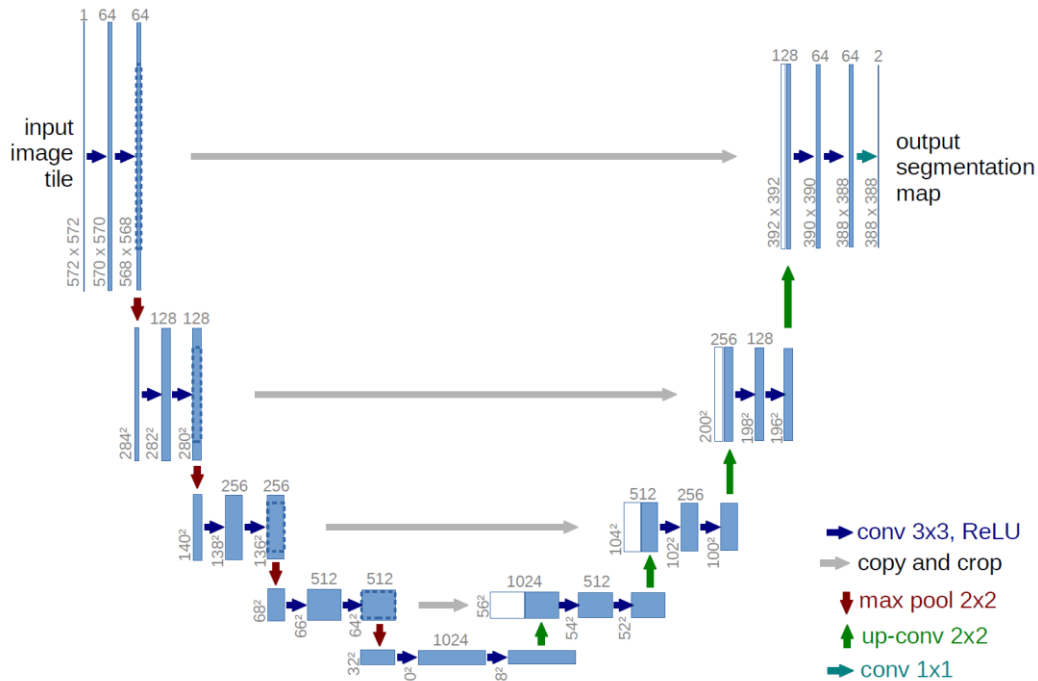
Segmentace pomocí CNN – architektury

- SegNet



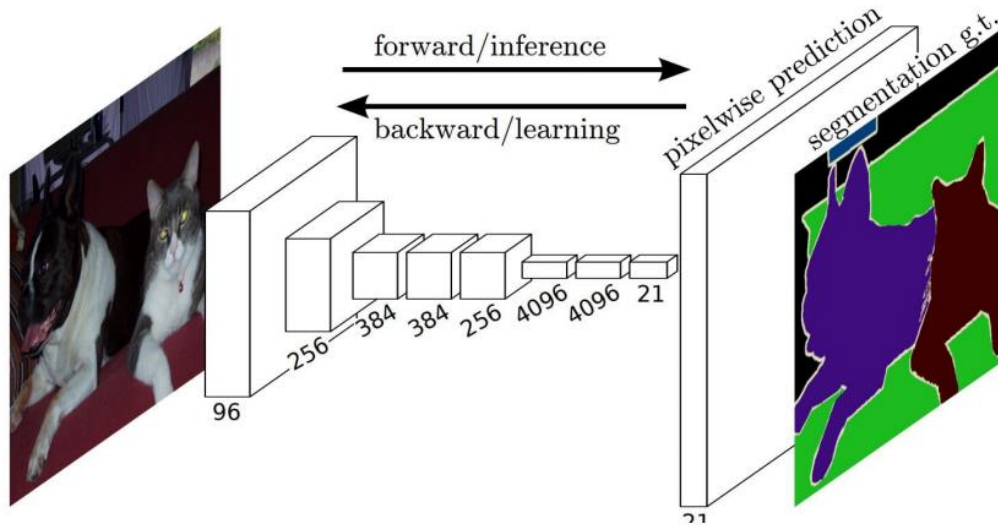
Segmentace pomocí CNN – architektury

- U-Net



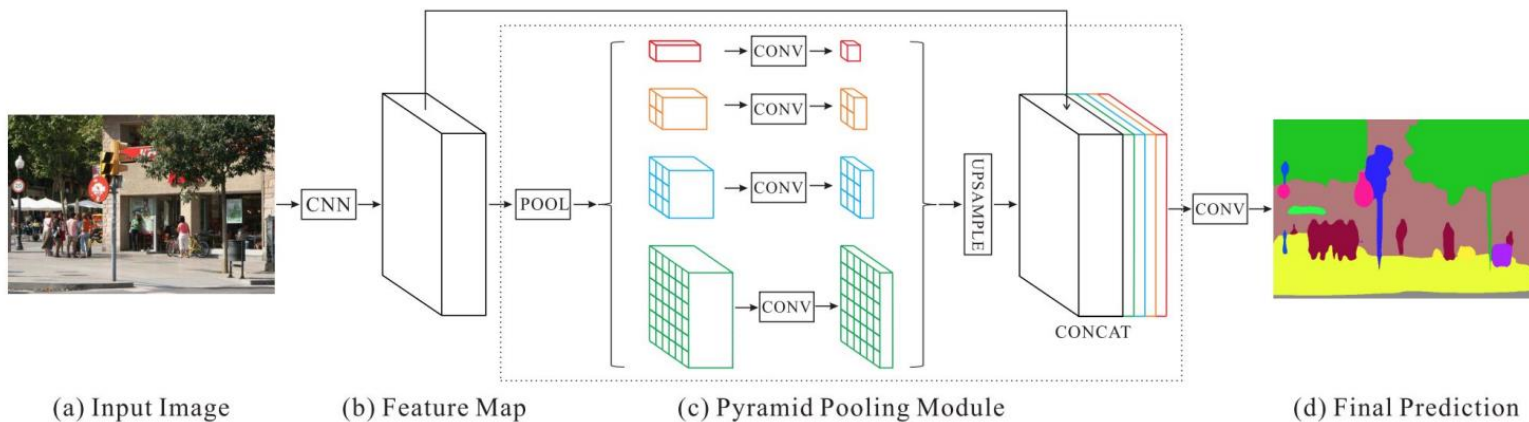
Segmentace pomocí CNN – architektury

- FCN – Fully Convolutional Network



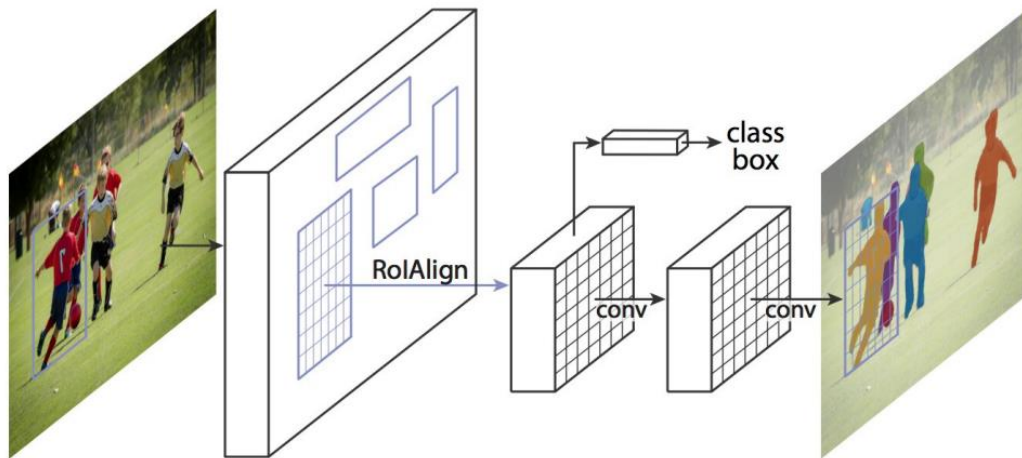
Segmentace pomocí CNN – architektury

- Pyramid Network modely



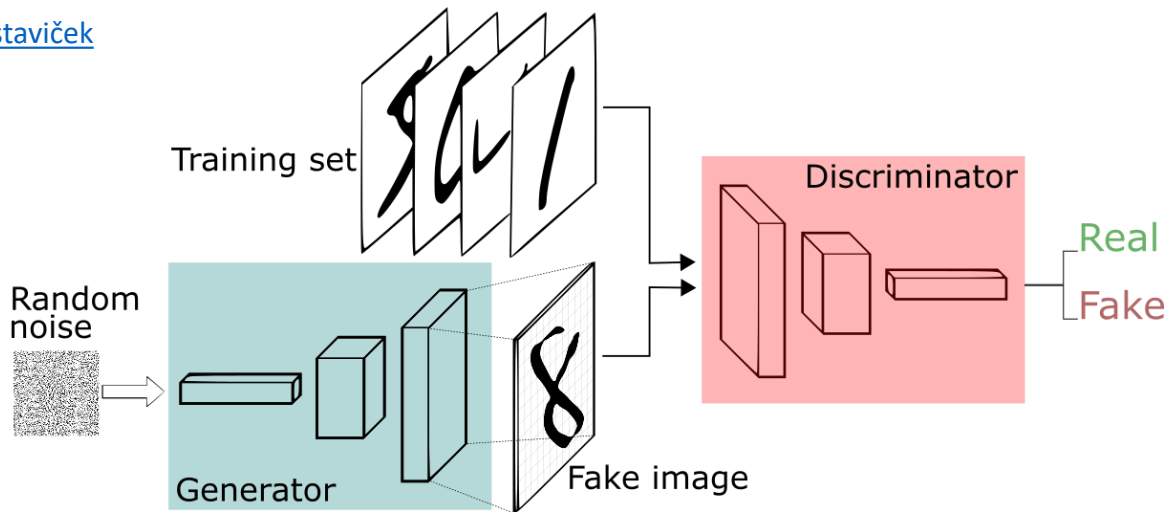
Segmentace pomocí CNN – architektury

- Mask R-CNN
 - Objekty jsou klasifikovány a lokalizovány pomocí ohraničujícího boxu a sémantické segmentace, která klasifikuje každý pixel do daných kategorií. Každá oblast zájmu dostane segmentační masku.



Segmentace pomocí CNN – architektury

- Segmentační úloha vede k zisku nového obrázku, který nemusí zachycovat pouhé přiřazení do třídy, ale může vytvářet nový obrázek
- GANs - Generative Adversarial Networks
 - [Generování kreslených postav](#)
 - [Stárnutí obličejů](#)
 - [Vybarvování obrázků](#)



Datasety

- Datasety jsou nedílnou součástí při práci s neuronovými sítěmi.
- V obrazovém zpracování mohou vstupní obrázky nabývat 2D, 2.5D (RGB+D), 3D rozměrů.
- Podle dané úlohy existují různé datasety, na kterých se provádí testování vlastních topologií vzhledem k ostatním. Přehledné shrnutí dostupných datasetů bylo např. popsáno v [článku](#) z 2020.
- Kromě vytvořených „obecných“ datasetů je ale často pro specifickou úlohu vytvořit datasety vlastní.
 - Vytvoření obrázků trénovací a testovací množiny.
 - Ruční označení dat (získ tzv. ground truth) - [bounding boxy](#), labelling pro segmentaci ([přehled 2022](#)), vlastní označování (vlastní aplikace)

Augmentace

- Vlastní datasety jsou často vytvořeny z několika stovek, tisíců obrázků, které nepokrývají dostačující část pro správné natrénování neuronové sítě.
- Pomocí augmentace můžeme dataset uměle rozšířit (původní obrázky orotovat, ztmavit, posunout, ...)
- Použití více dat pro trénování vede k lepší generalizaci sítě.
- [Tensorflow Data Generator](#)

Děkuji za pozornost.

Přednáška pro mikrocertifikát
!!! 5. Test - 22. 4. 2024 !!!

Cvičení – společné

- Organizace další výuky
- Výpočet konvoluce a operace max-pooling
- Záznam do deníku

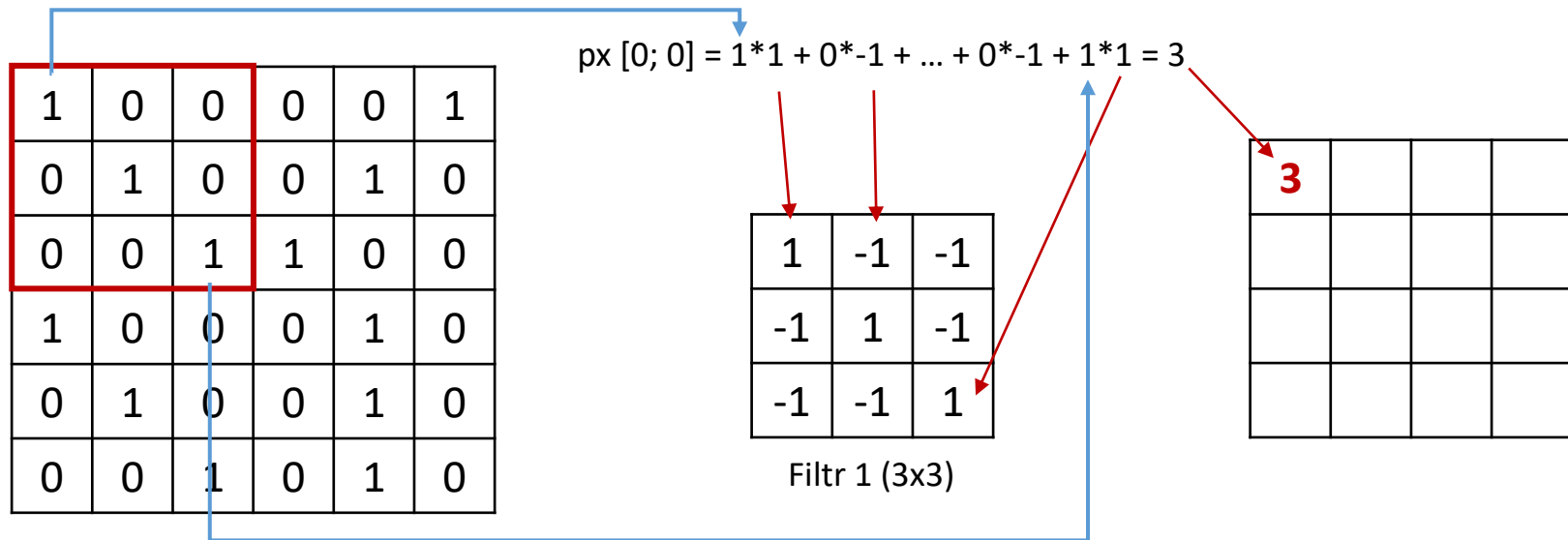
Organizace další výuky

- 22. 4. 2024
 - Přednáška pro mikrocertifikát
 - 5. test na CNN, kompletace deníků
- 29. 4. 2024
 - [Vybrané téma přednášky](#)
 - Prezentace deníku (Zp)
- 6. 5. 2024
 - Předtermín zkoušky
 - Prezentace deníku (Zp)



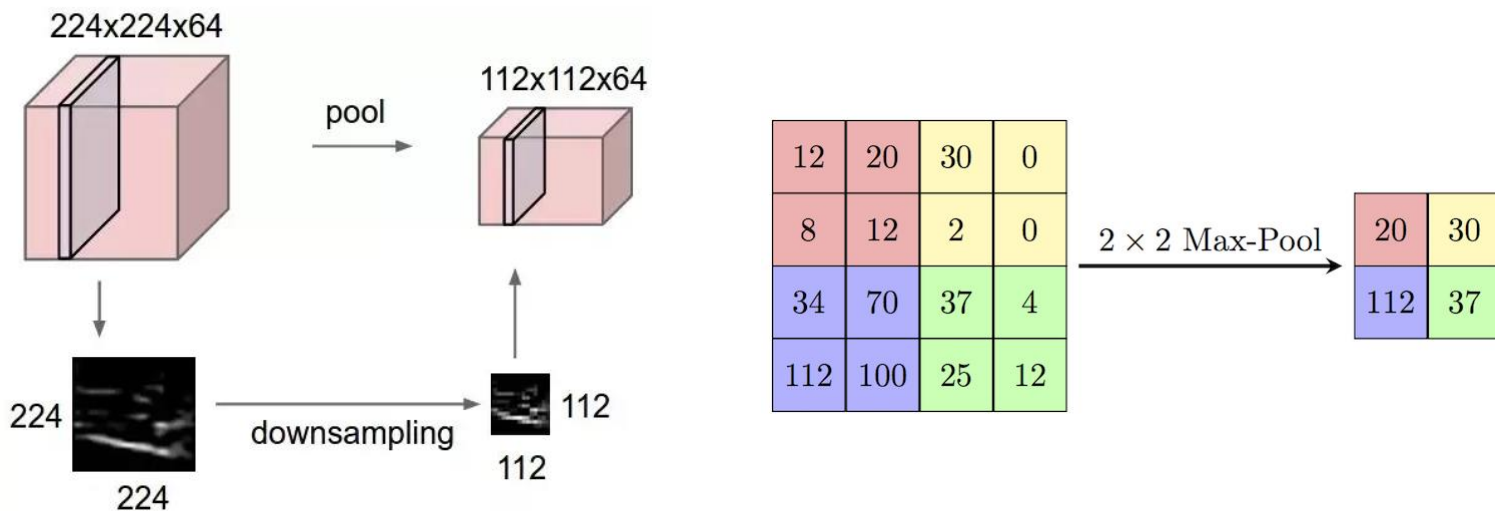
Konvoluční neuronová síť – konvoluce

- Pro každý pixel aplikujeme skalární součin konvolučního filtru s vstupem a získáváme výstupní hodnotu tvořící matici příznaků



Konvoluční neuronová síť – Max-Pooling

- Pooling vrstvy sjednocují skupinu vlastností a vybírají jen tu nejvíce dominantní – max pooling vybírá maximální hodnotu, snižují velikost



Konvoluční neuronová síť – konvoluce

- Pro každý pixel aplikujeme skalární součin konvolučního filtru s vstupem a získáváme výstupní hodnotu tvořící matici příznaků

Vstupní obraz (5x5)

1	0	0	3	0
1	1	1	1	1
0	2	1	1	0
1	2	2	1	1
0	1	0	1	1

1	1	2
0	0	0
0	3	1

Filtr 1 (3x3)

1	-1
0	1

Filtr 2 (2x2)

Konvoluční neuronová síť – konvoluce

- Pro každý pixel aplikujeme skalární součin konvolučního filtru s vstupem a získáváme výstupní hodnotu tvořící matici příznaků

Vstupní obraz (4x4)

1	0	1	1
0	1	1	0
2	1	1	1
0	0	0	3

0	1	1
0	1	0
0	1	1

Filtr 1 (3x3)

Cvičení– samostatně

- Záznam do deníku:
 - Samostatně navrhnete a několik vlastních architektur konvolučních neuronových sítí (CNN) pro klasifikaci dopravních značek na datech z ([GTSRB dataset](#)) pomocí knihovny KERAS nebo pyTorch.
 - Využijte úprav pro přípravu dat z minulého úkolu.
 - Navrhnete alespoň 5 různých topologií CNN (Každý model natrénujte 10×)
 - Vytvořte boxplot pro přesnost/valid loss všech 5 topologií
 - Zobrazte nejlepší modelu nad test daty (např. klasifikační report, confusion matrix)