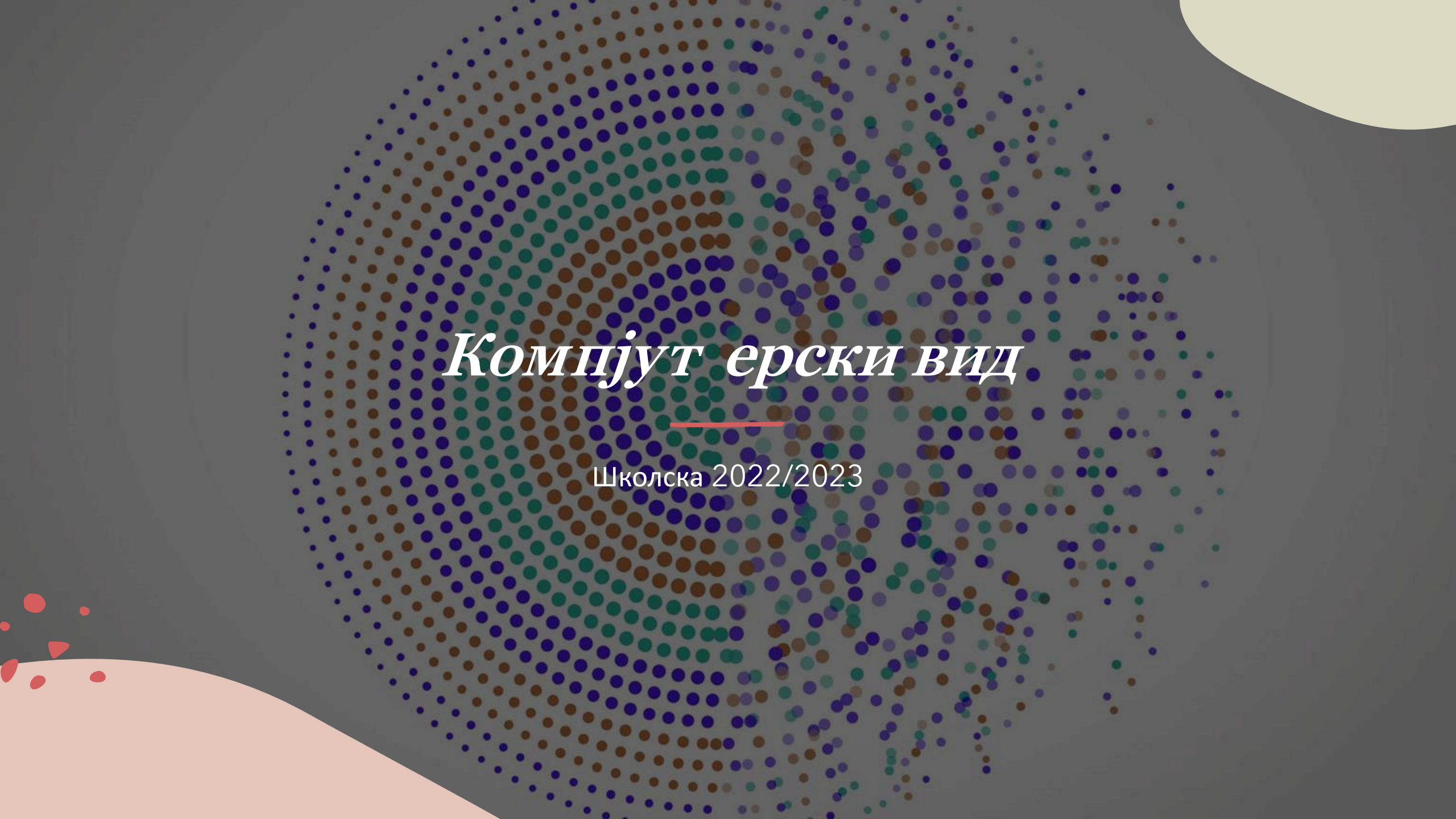




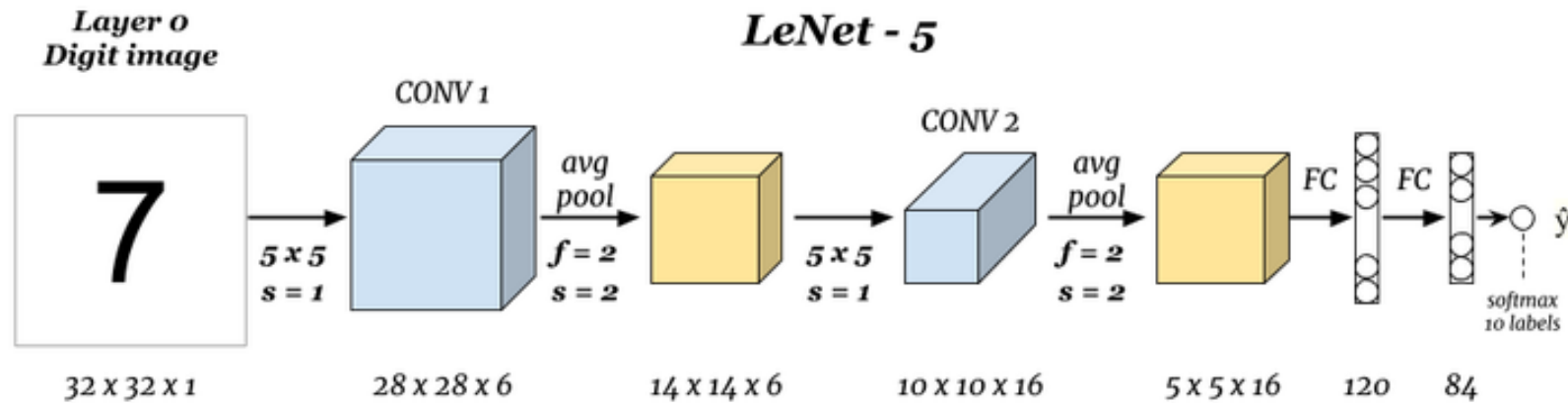
Компјутерски вид

Школска 2022/2023

План рада

1. Неке познатије архитектуре
2. Трансфер знања (енг. transfer learning) и прилагођавање (енг. fine-tuning)
3. Детекција и локализација објеката

LeNet-5 (1998)

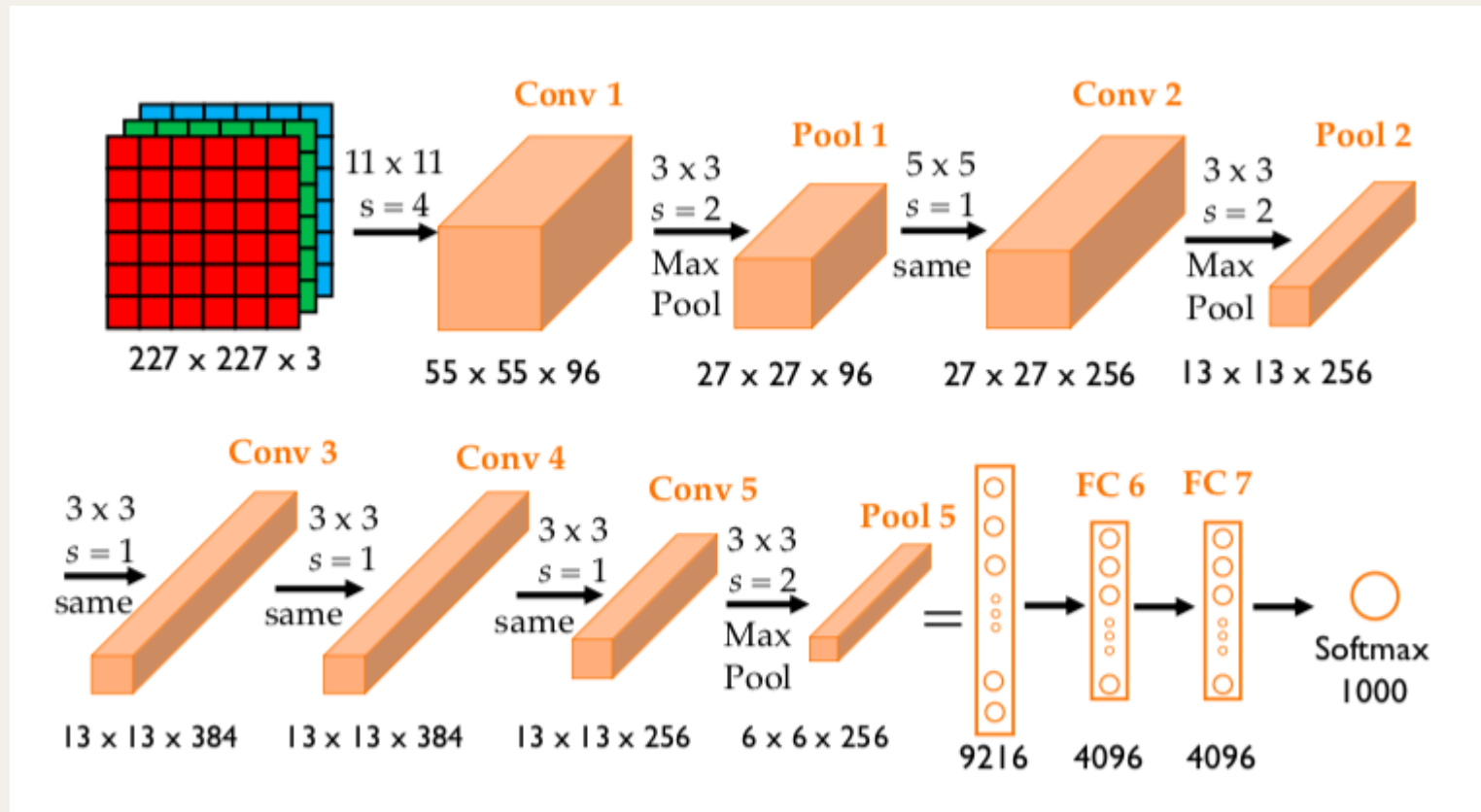


LeNet-5 (1998)

- Релативно мала мрежа (око 60К параметара)
- Просторна ширина и дужина тензора се смањује са сваком слојем док се дубина повећава
- Понављање Conv-Pool шаблона

Layer		Feature Map	Size	Kernel Size	Stride	Activation
Input	Image	1	32x32	-	-	-
1	Convolution	6	28x28	5x5	1	tanh
2	Average Pooling	6	14x14	2x2	2	tanh
3	Convolution	16	10x10	5x5	1	tanh
4	Average Pooling	16	5x5	2x2	2	tanh
5	Convolution	120	1x1	5x5	1	tanh
6	FC	-	84	-	-	tanh
Output	FC	-	10	-	-	softmax

AlexNet (2012)

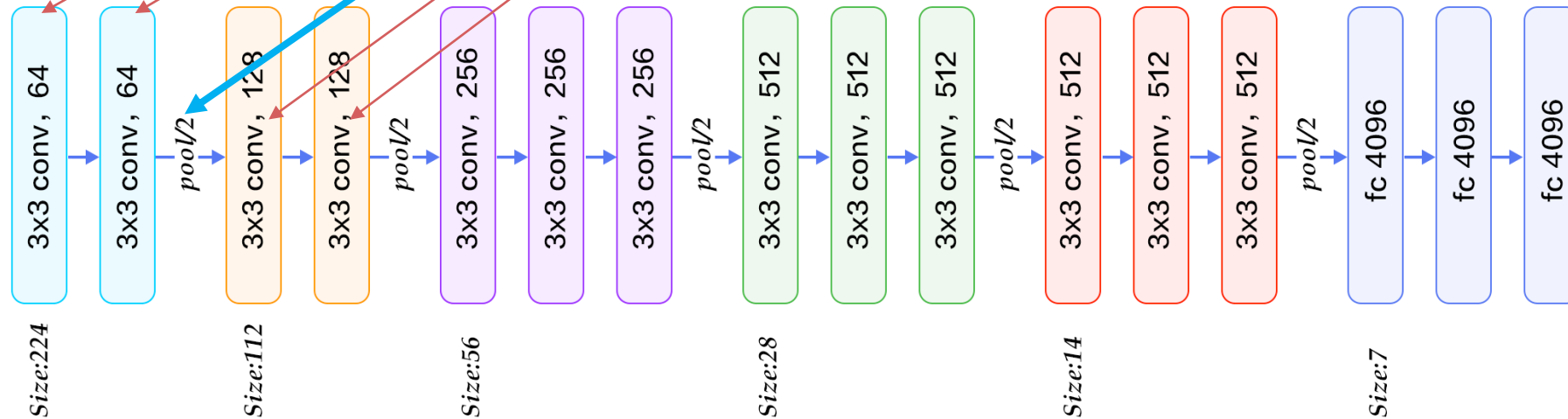
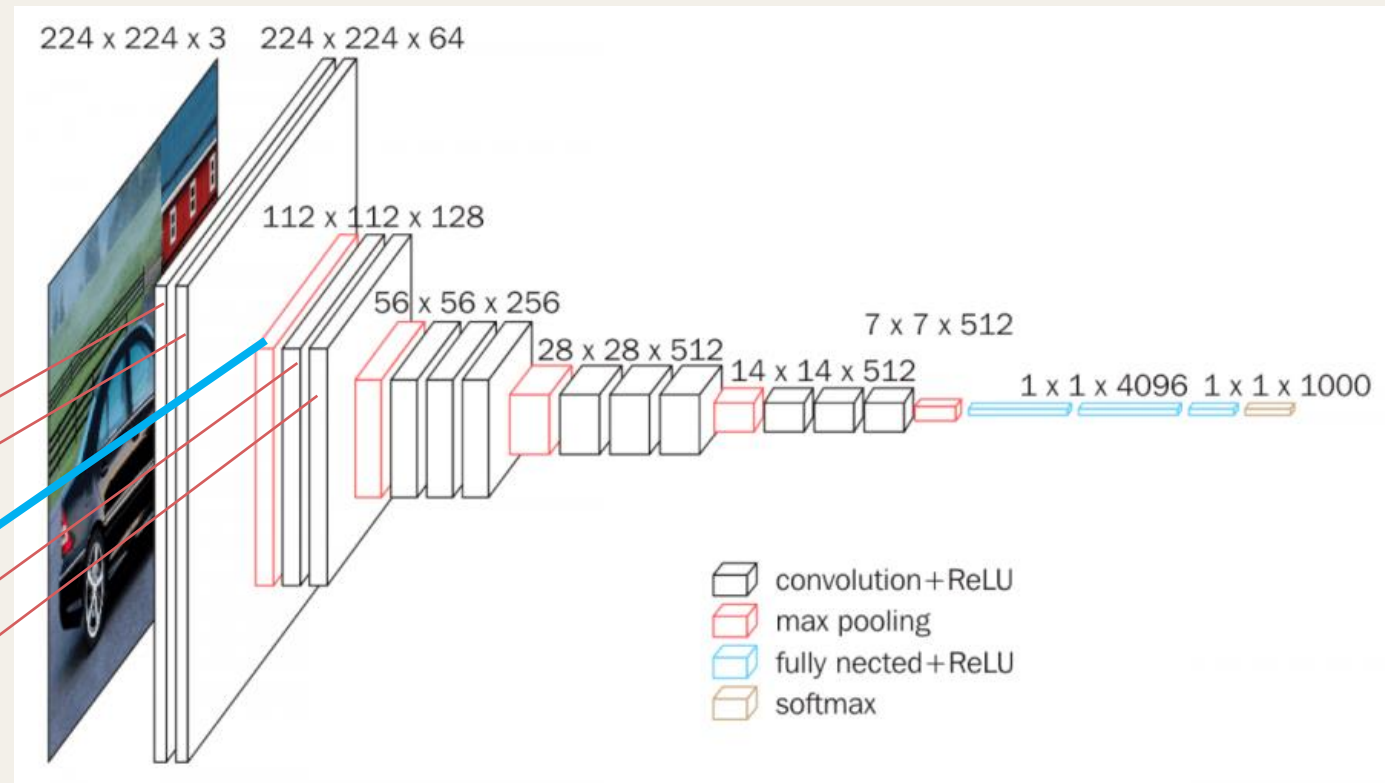


AlexNet (2012)

Layer	# filters / neurons	Filter size	Stride	Padding	Size of feature map	Activation function
Input	-	-	-	-	227 x 227 x 3	-
Conv 1	96	11 x 11	4	-	55 x 55 x 96	ReLU
Max Pool 1	-	3 x 3	2	-	27 x 27 x 96	-
Conv 2	256	5 x 5	1	2	27 x 27 x 256	ReLU
Max Pool 2	-	3 x 3	2	-	13 x 13 x 256	-
Conv 3	384	3 x 3	1	1	13 x 13 x 384	ReLU
Conv 4	384	3 x 3	1	1	13 x 13 x 384	ReLU
Conv 5	256	3 x 3	1	1	13 x 13 x 256	ReLU
Max Pool 3	-	3 x 3	2	-	6 x 6 x 256	-
Dropout 1	rate = 0.5	-	-	-	6 x 6 x 256	-

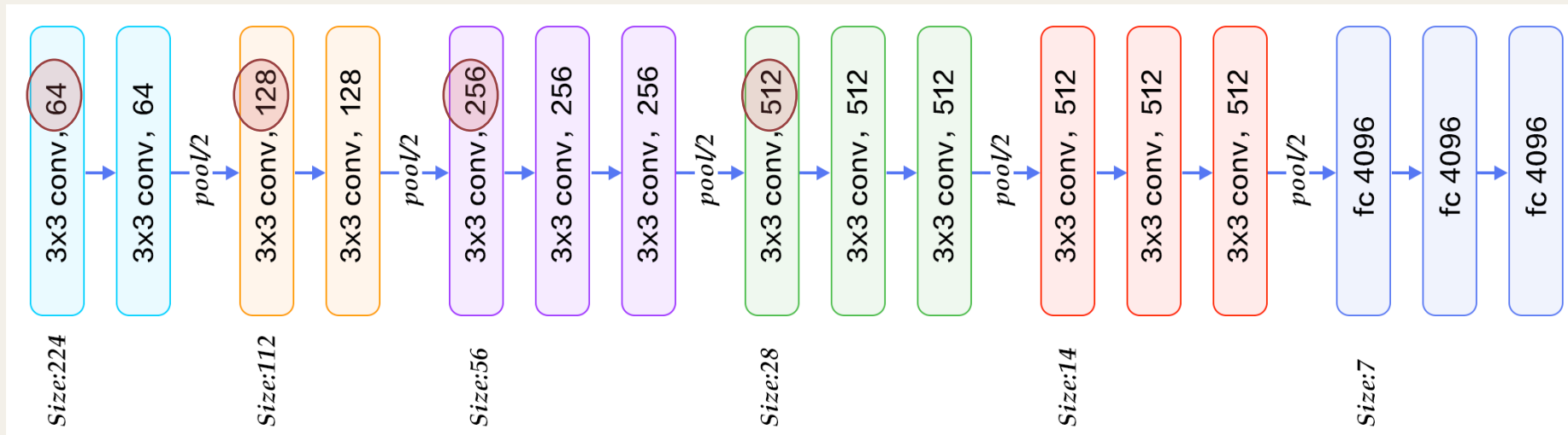
- Релативно велика мрежа (око 60М параметара)
- Коришћење ReLU активационе функције
- „Same” padding – прошири тензор за онолико колико је потребно да се не промени ширина и дужина излазног тензора (у односу на улазни)
- Оригинални рад – алгоритам за паралелно израчунавање на више GPU (у пракси се користи за неке специјалне случајеве).

VGG-16 (2015)



VGG-16 (2015)

- Велика мрежа (око 140M параметара), недеље за тренирање
- Мање конволуције (3×3 уместо 5×5 или 11×11 у AlexNet-y)
- 92.7% top-5 acc in ImageNet
- Дубља алтернатива VGG-19



Остале архитектуре

- ResNet
- Inception Network
- MobileNet
-
- Обично су доступни већ тренирани модели, видети <https://tfhub.dev/s?subtype=module,placeholder>

Практични савети

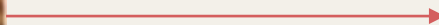
- Имплементација модела на основу рада може бити јако добра практична вежба
- Али.... у пракси је боље користити open-source имплементације модела
 - Није једноставно имплементирати модел само на основу описа из рада
 - Неки детаљи нису експлицитно наглашени и/или се подразумевају
 - Бржи развој пројекта
 - Ако имплементација модела не долази од оригиналних аутора или проверених извора (као што су имплементације доступне у Tensorflow-у) **ОБАВЕЗНО ПРОВЕРИТИ ИСПРАВНОСТ КОДА** (или је не користити)
- Користити аугментацију података

Аугментација података

- Користи се да би се добили бољи модели
- «Обично» повећавање скупа података на коме се учи, доводи до бољих модела
- Користи се у разним ситуацијама (трансфер знања, тренирање од почетка ...)

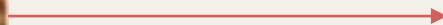
Аугментација података

- Често коришћене методе: *mirroring*



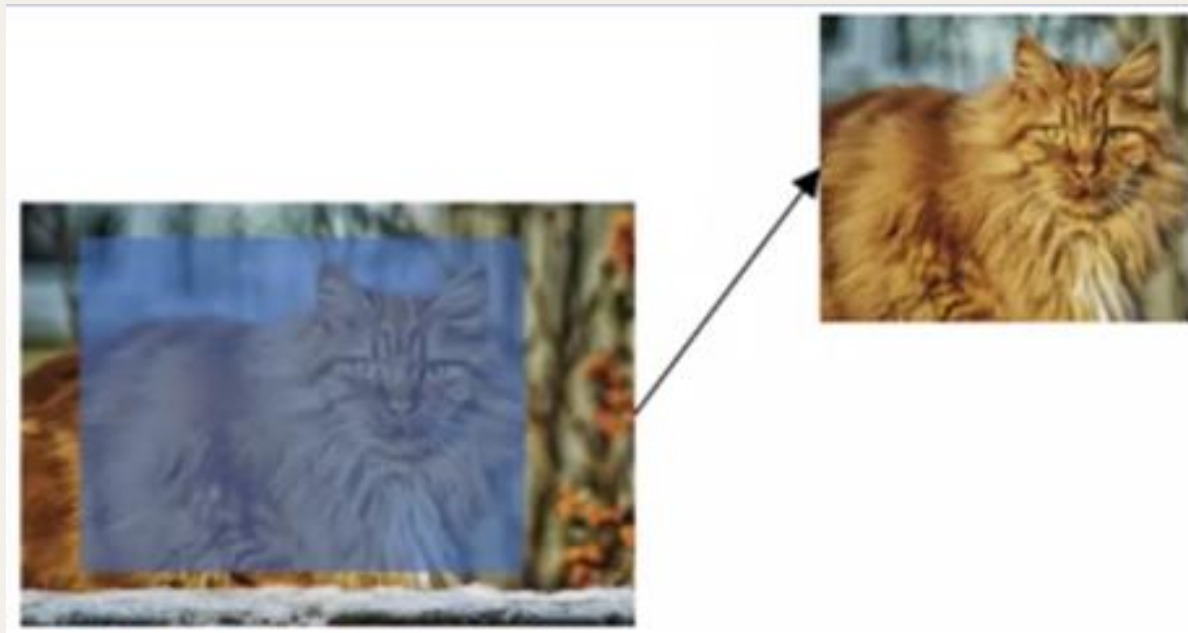
Аугментација података

- Често коришћене методе: *rotation*



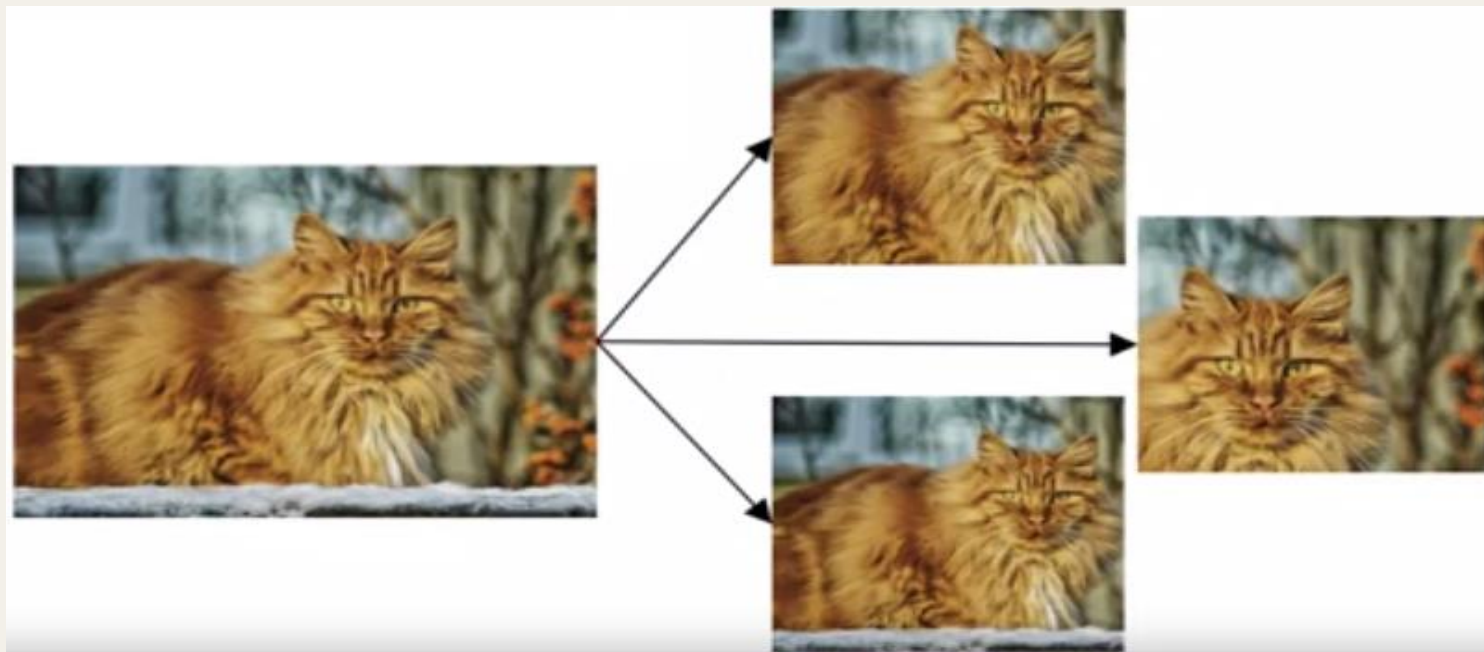
Аугментација података

- Често коришћене методе: *random cropping*



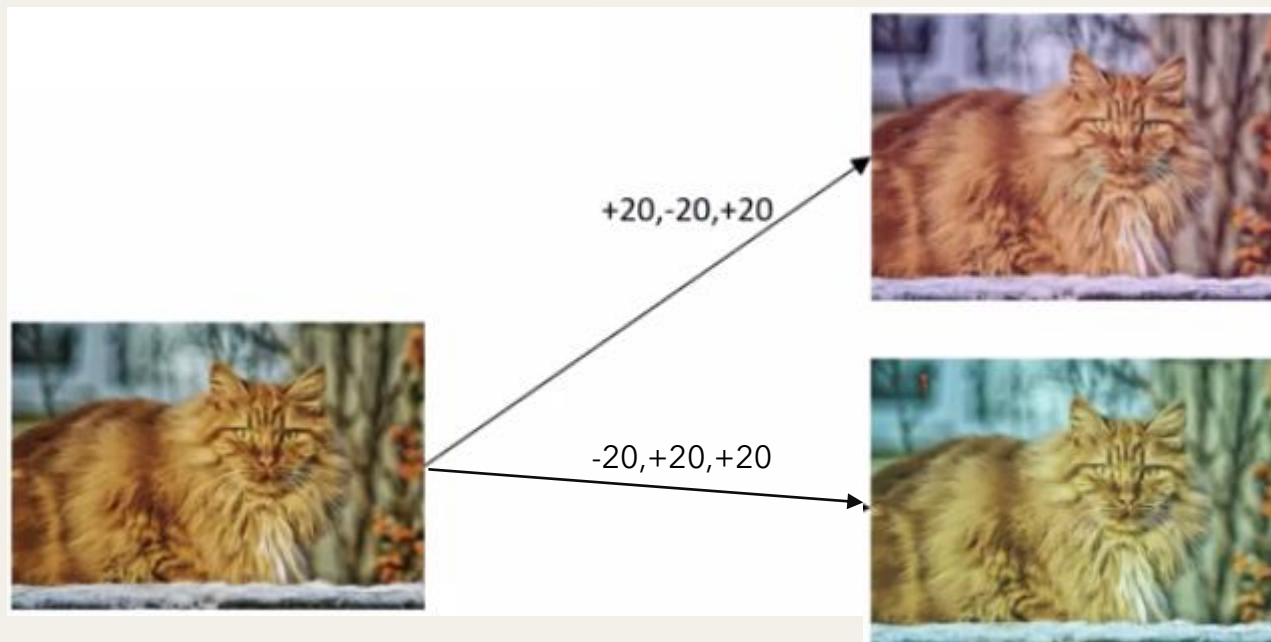
Аугментација података

- Често коришћене методе: *random cropping*
 - Има смисла ако се на крајњој слици и даље налази објекат од интереса

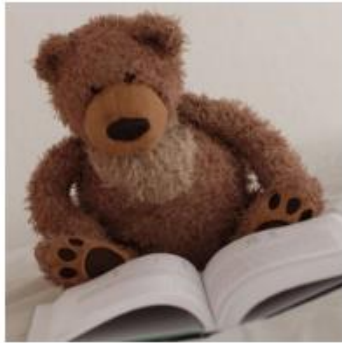


Аугментација података

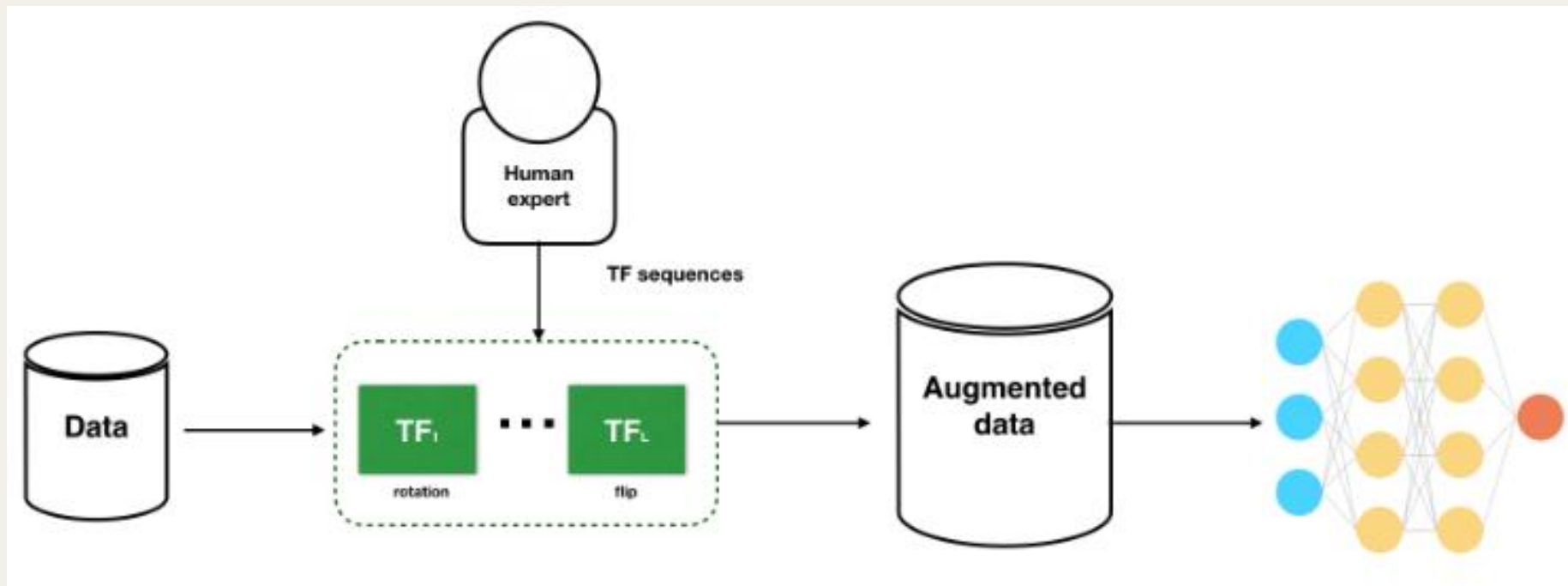
- Често коришћене методе: *color shifting*



Аугментација подат ака

Original	Flip	Rotation	Random crop
			
• Image without any modification	• Flipped with respect to an axis for which the meaning of the image is preserved	• Rotation with a slight angle • Simulates incorrect horizon calibration	• Random focus on one part of the image • Several random crops can be done in a row
Color shift	Noise addition	Information loss	Contrast change
			
• Nuances of RGB is slightly changed • Captures noise that can occur with light exposure	• Addition of noise • More tolerance to quality variation of inputs	• Parts of image ignored • Mimics potential loss of parts of image	• Luminosity changes • Controls difference in exposition due to time of day

Аугментација података



План рада

1. Неке познатије архитектуре
2. Трансфер знања (енг. transfer learning)
3. Детекција и локализација објеката

Трансфер знања (енг. transfer learning)

- Практична ограничења:
 - Доступна је (углавном) мала количина података
 - Тренирање модела може трајати јако дуго (нпр. месецима)
 - Почетна иницијализација модела није «добра» за решавање конкретног проблема
 - Модел дивергира у поновљеним тренинзима

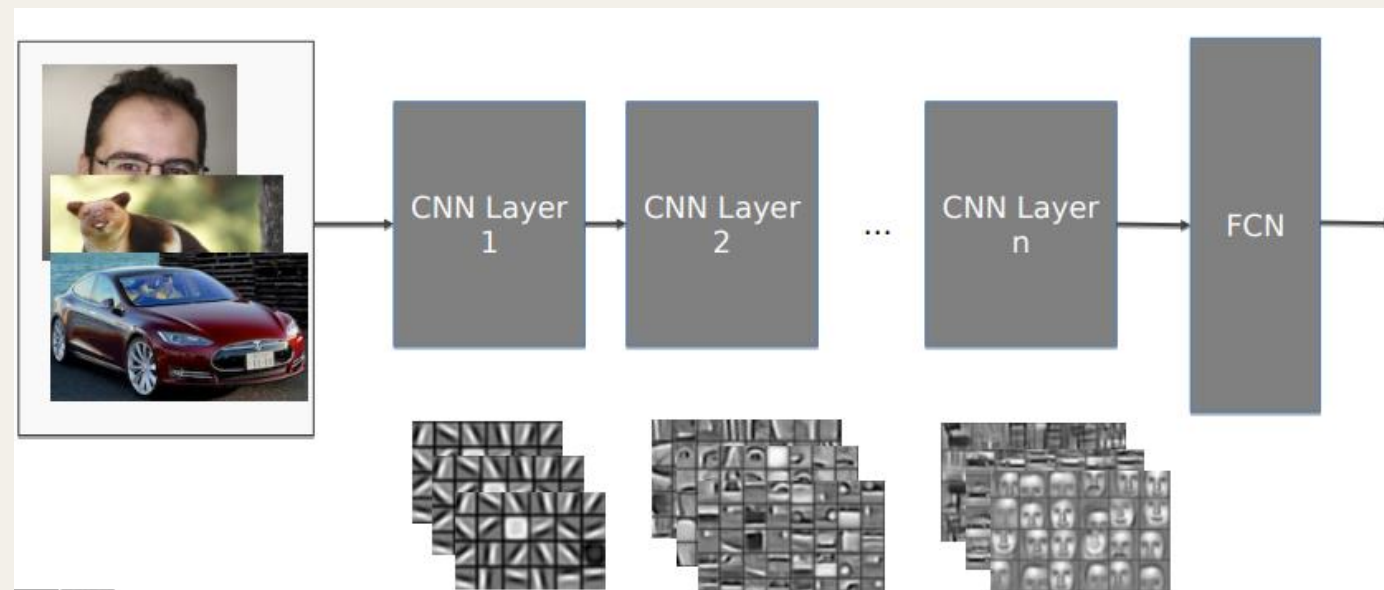
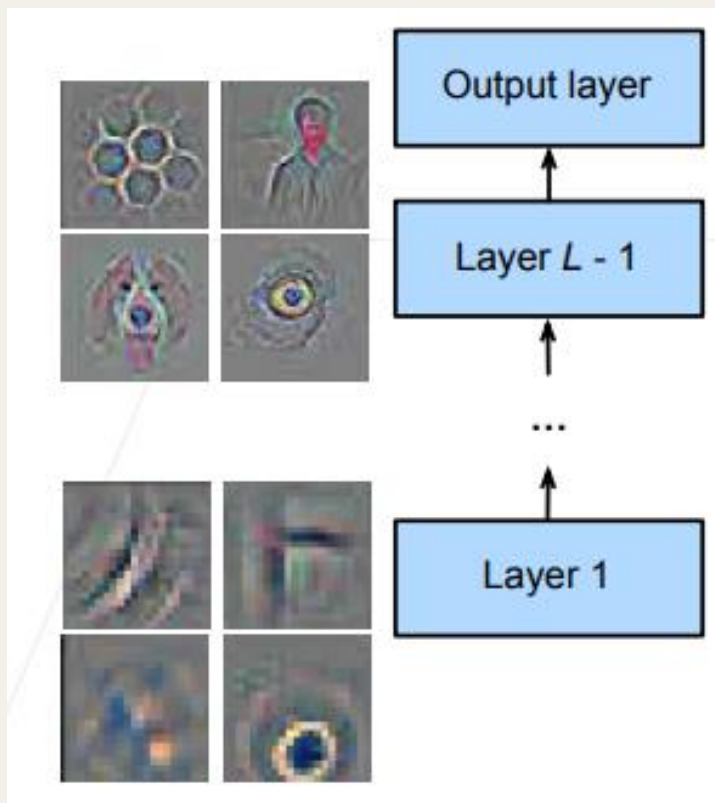
Трансфер знања (енг. *transfer learning*)

- Идеја трансфера знања
 - Знање стечено приликом решавања неког (познатог) проблема, искористити при решавању новог проблема

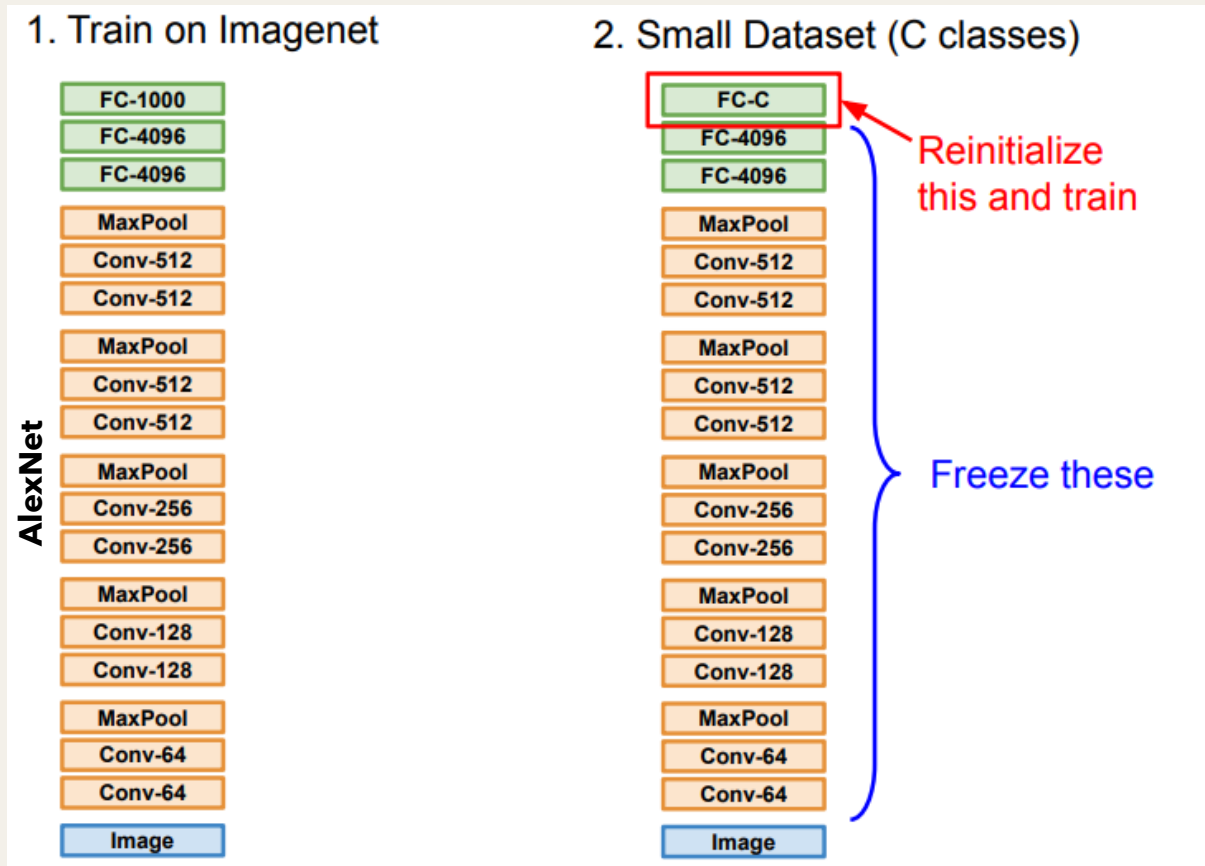


Лакше је научити руски ако већ говорите српски, шпански ако већ говорите француски, немачки ако већ говорите енглески итд.

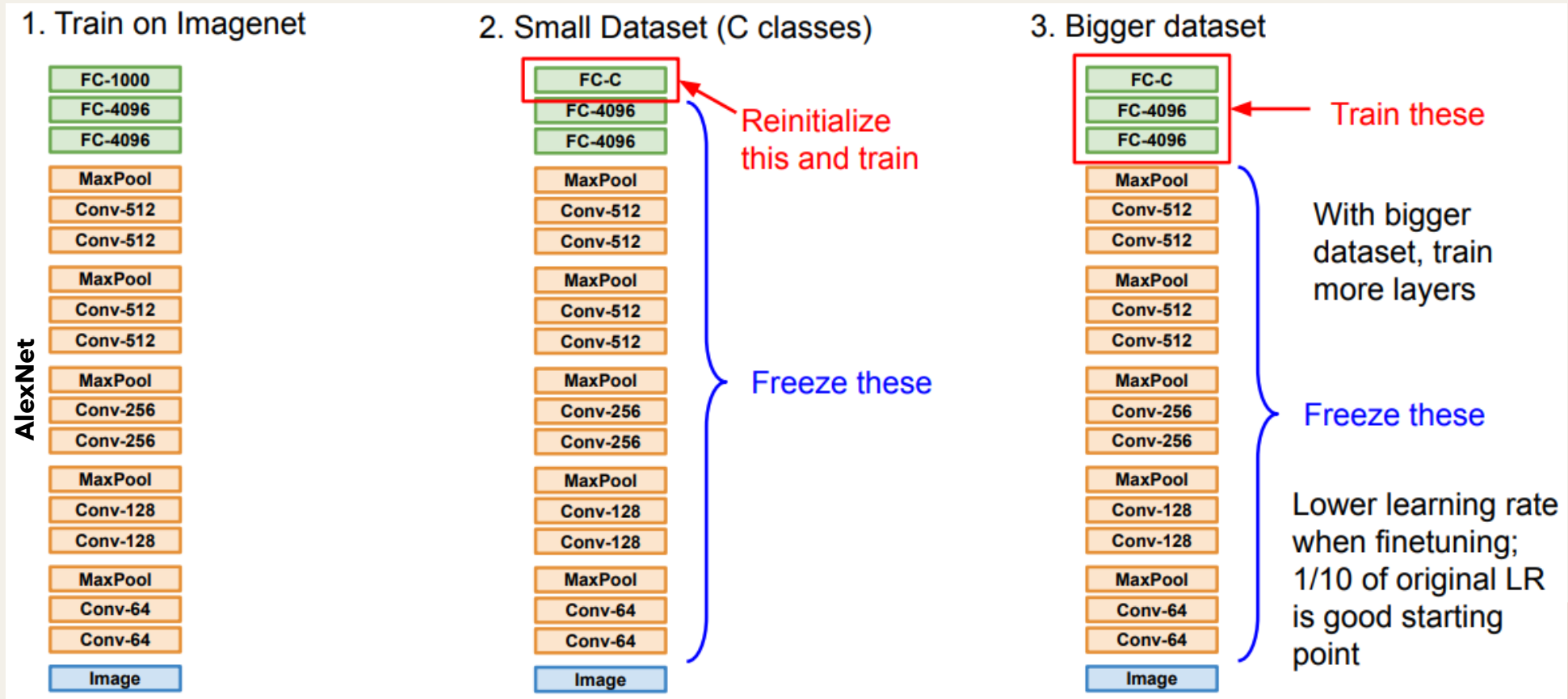
Трансфер знања (енг. *transfer learning*)



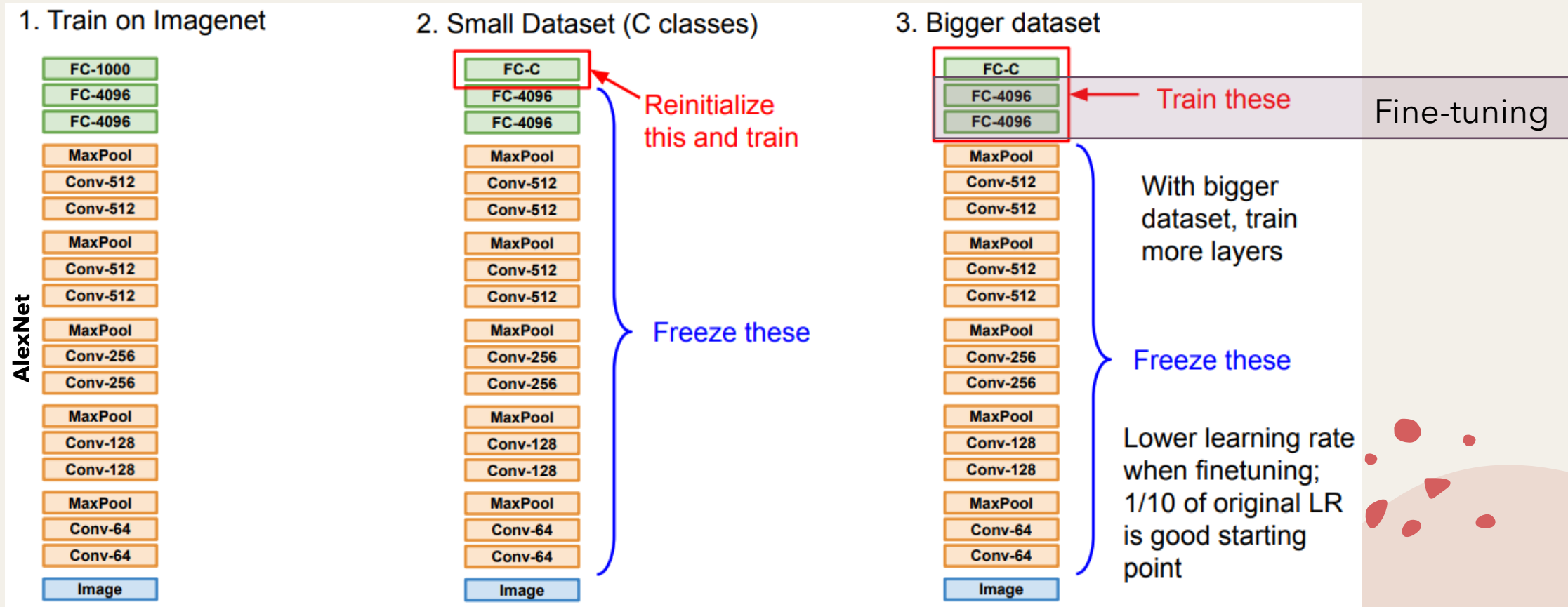
Трансфер знања (енг. *transfer learning*)



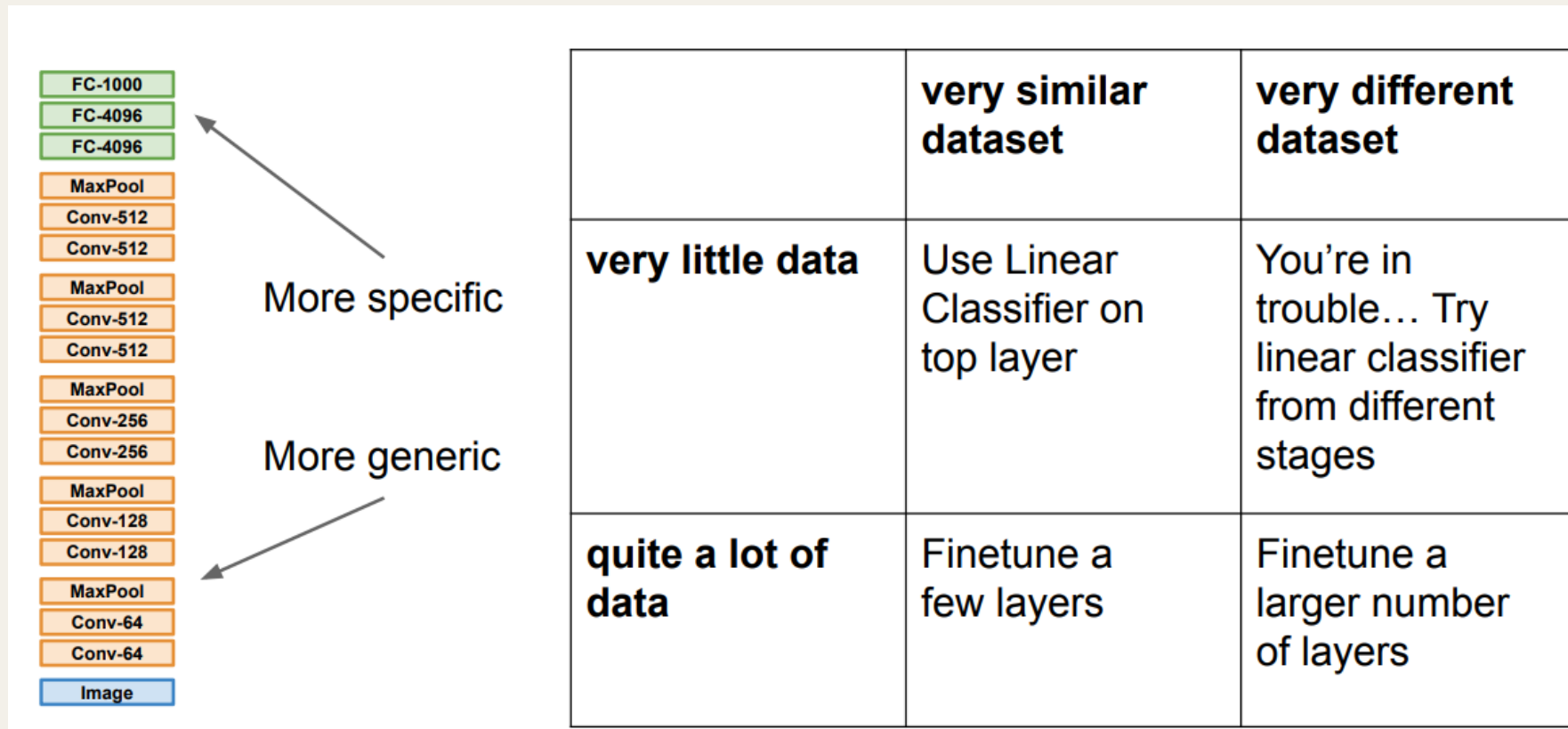
Трансфер знања (енг. *transfer learning*)



Трансфер знања (енг. *transfer learning*)



Трансфер знања (енг. *transfer learning*)



Трансфер знања (енг. transfer learning)

- Уместо иницијализације мреже случајним вредностима, користити већ трениране мреже
- Знање (тежине) научене у првим слојевима мреже задржати
 - Не мењати њихове тежине током тренирања
 - Користи се научено знање (филтери) из претходног задатака
- Заменити један (или више) слојева у мрежи
 - Обучити их на основу доступних података
- Ако постоји довољно велика количина нових података, цела мрежа може да се до-тренира (енг. fine-tune) користећи јако мали learning rate
 - Мали learning rate је важан да се не би „уништило“ оно што је претходно научено

Трансфер знања (енг. transfer learning)

- Врло често се користи у пракси али....
 - Погледати рад [Rethinking ImageNet Pre-training](#)
 - Тренирање од почетка може дати исто тако добре резултате (иако ће трајати много дуже)
 - Проширивањем скупа података могу се (понекад) постићи бољи резултати
- Скупови података (нови и стари задатак) треба да буду слични да би трансфер имао смисла

Радови за читање

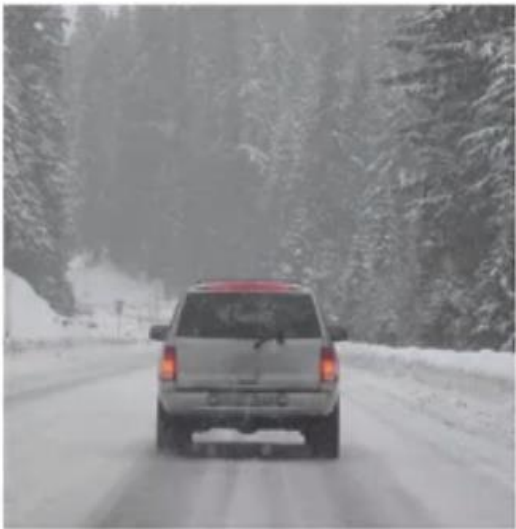
- Krizhevsky et al., 2012. ImageNet Classification with Deep Convolutional Neural Networks
- <https://arxiv.org/pdf/1608.08614.pdf>
- <http://adas.cvc.uab.es/task-cv2016/papers/0002.pdf>
- <https://proceedings.neurips.cc/paper/2020/file/0607f4c705595b911a4f3e7a127b44e0-Paper.pdf>

План рада

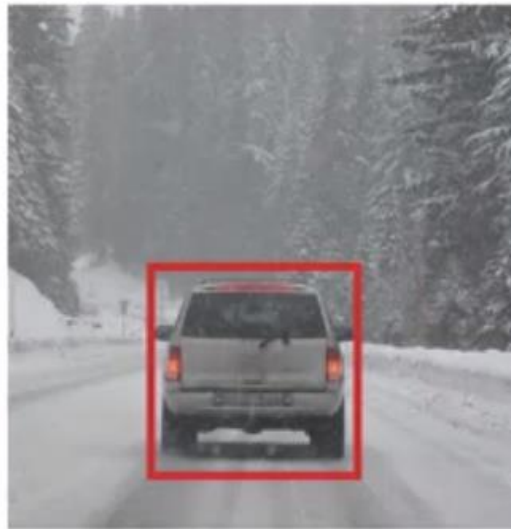
1. Неке познатије архитектуре
2. Трансфер знања (енг. transfer learning)
3. Детекција и локализација објеката

Класификација – локализација – детекција

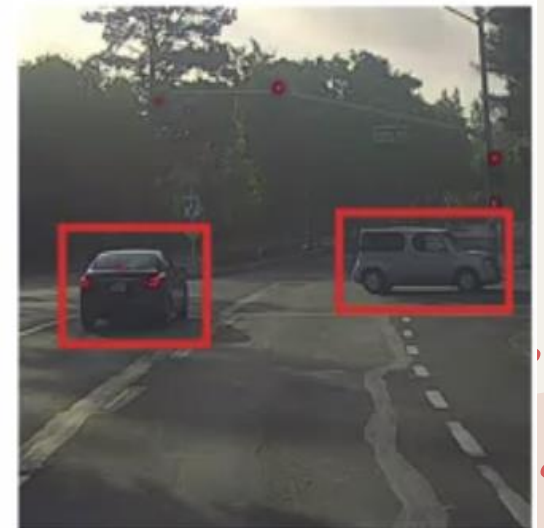
Image classification



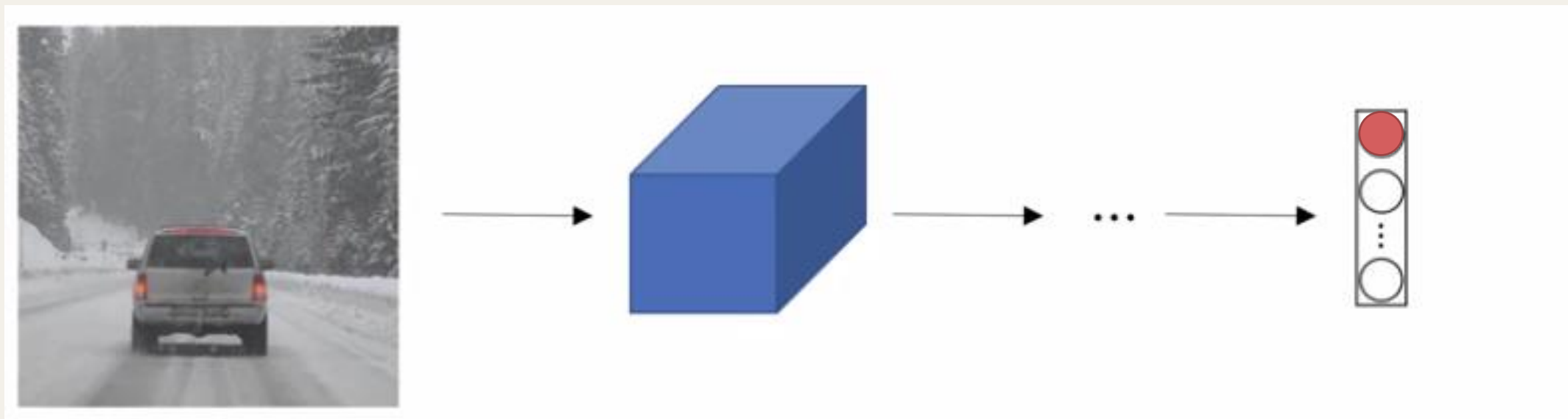
Classification with localization



Detection

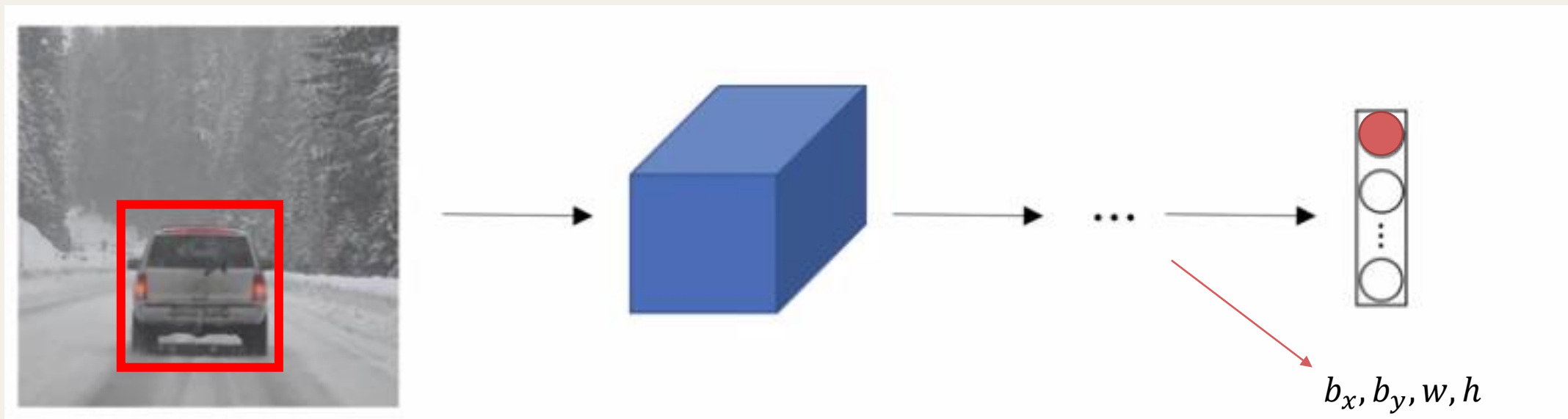


Класификација



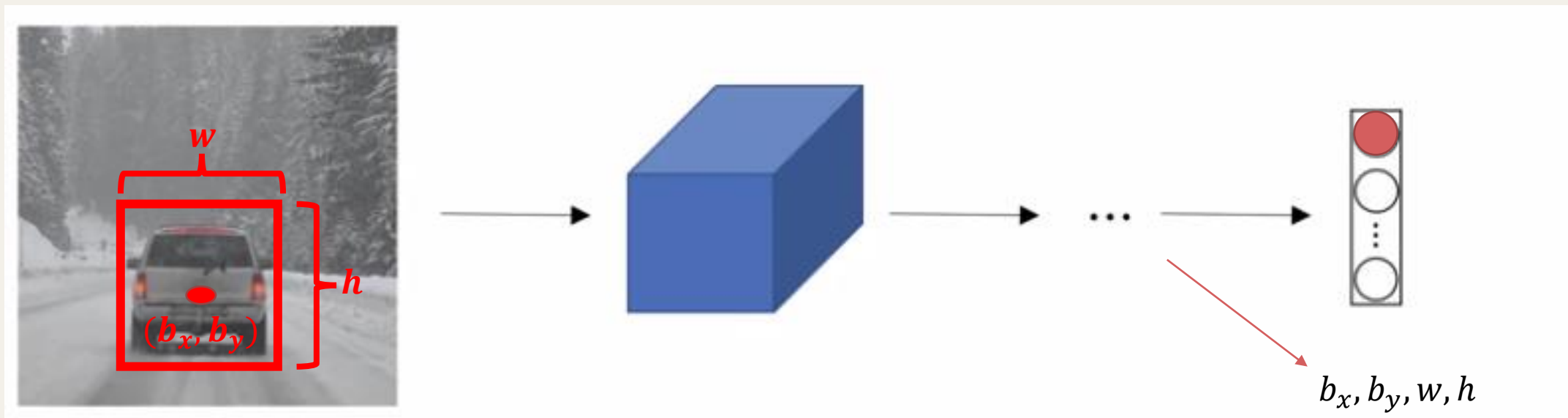
- Модел предвиђа класу целе слике
- „На слици се налази ауто“ али не знамо где нити колико их има

Локализација



- „На слици се налази ауто“ на локацији (b_x, b_y) , ширине w и висине h

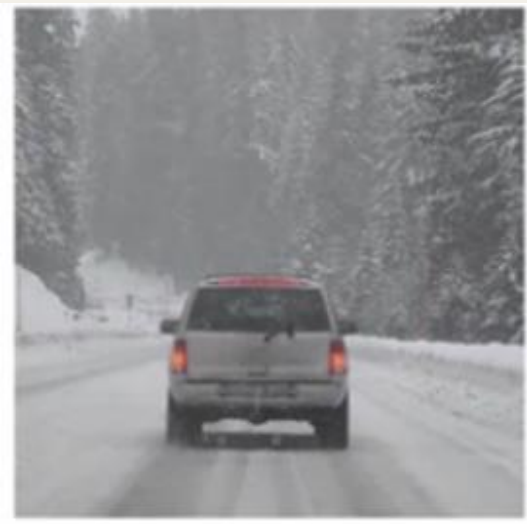
Локализација



- „На слици се налази ауто“ на локацији (b_x, b_y) , ширине w и висине h

Класификација

C_1 - Car
 C_2 - Person
 C_3 - Dog

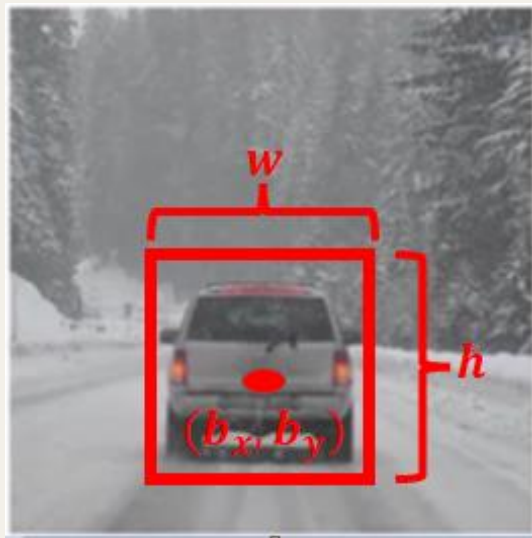


Облик анотације

$\begin{bmatrix} 1 \\ 0 \\ 0 \end{bmatrix}$

Локализација

C_1 - Car
 C_2 - Person
 C_3 - Dog

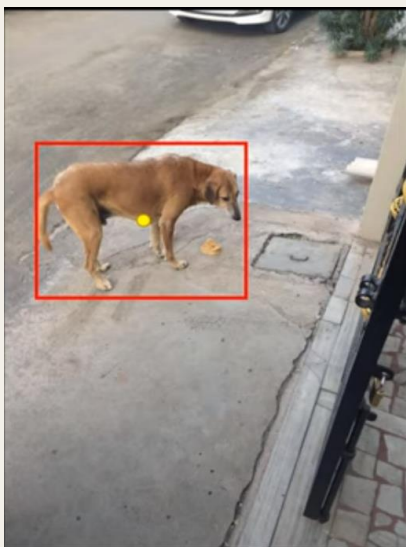


Облик анотације

$$\begin{bmatrix} Pc \\ b_x \\ b_y \\ w \\ h \\ C_1 \\ C_2 \\ C_3 \end{bmatrix} \longrightarrow \begin{bmatrix} 1 \\ 50 \\ 70 \\ 35 \\ 40 \\ 1 \\ 0 \\ 0 \end{bmatrix}$$

Локализација

C_1 - Car
 C_2 - Person
 C_3 - Dog



Облик анотације

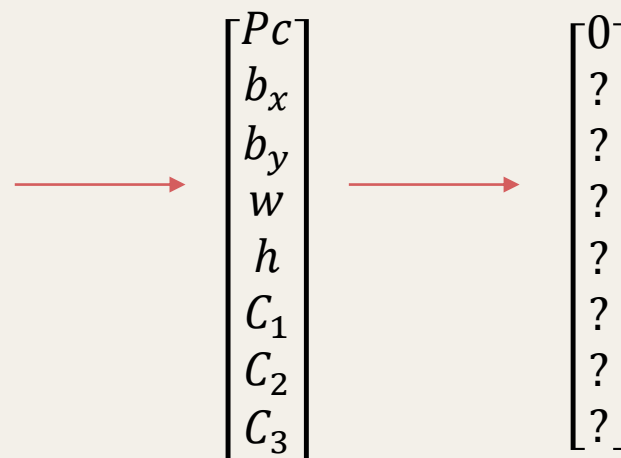
$$\begin{matrix} \longrightarrow & \begin{bmatrix} Pc \\ b_x \\ b_y \\ w \\ h \\ C_1 \\ C_2 \\ C_3 \end{bmatrix} & \longrightarrow & \begin{bmatrix} 1 \\ 120 \\ 70 \\ 50 \\ 25 \\ 0 \\ 0 \\ 1 \end{bmatrix} \end{matrix}$$

Локализација

C_1 - Car
 C_2 - Person
 C_3 - Dog



Облик анотације

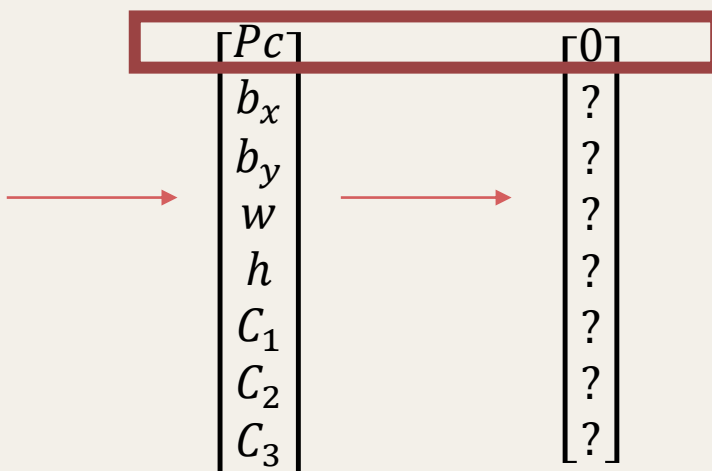


Локализација

C_1 - Car
 C_2 - Person
 C_3 - Dog



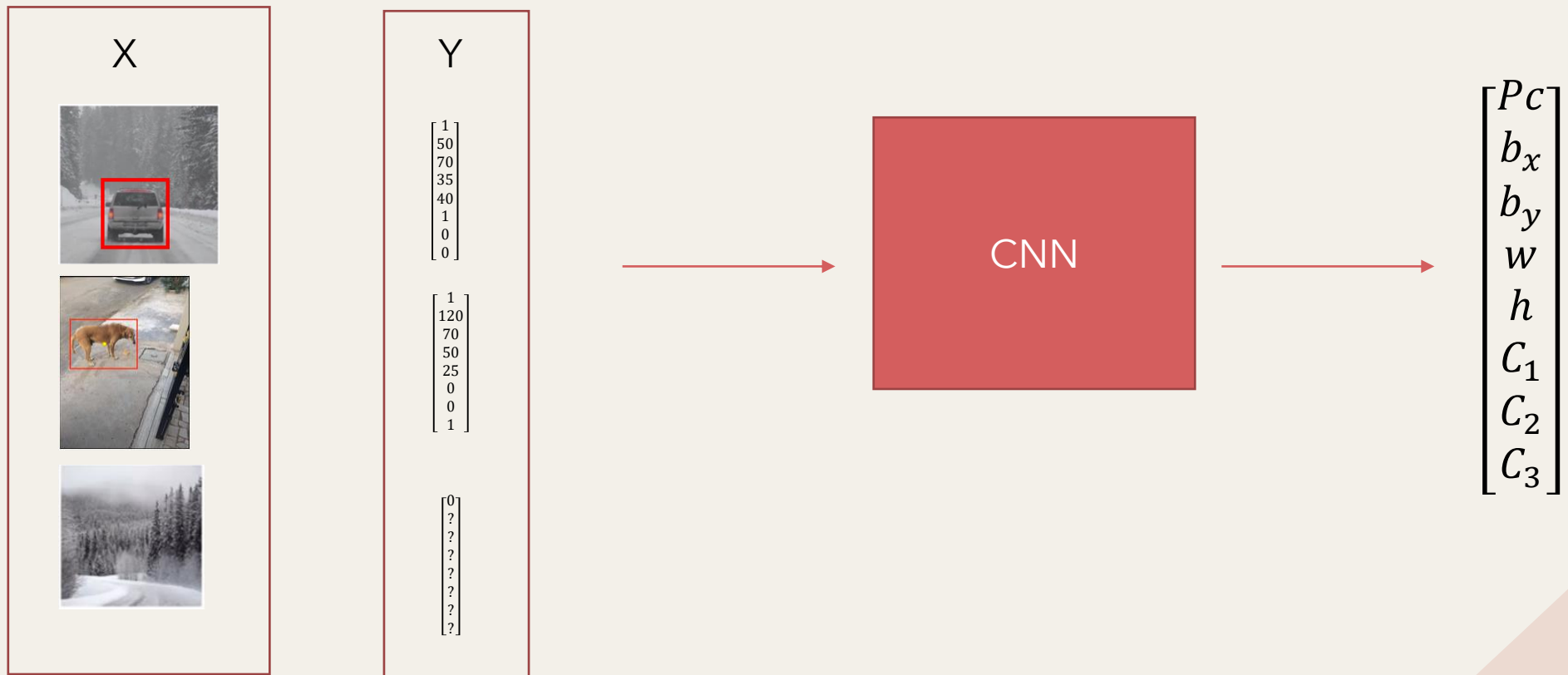
Облик анотације



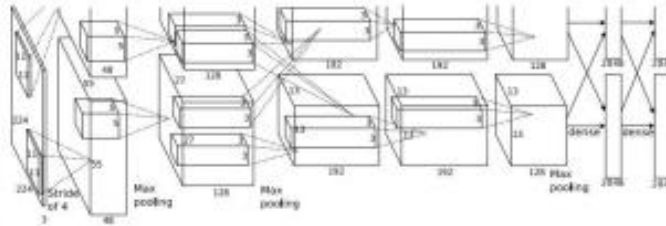
0 означава да нема објекта (background) па остали елементи овог вектора нису битни

Локализација

Тренирање

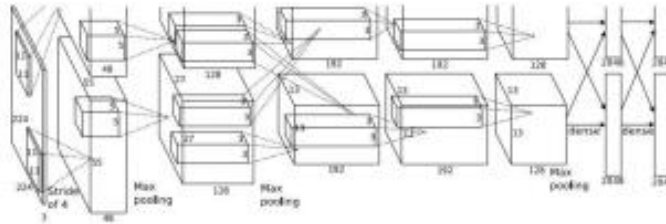


Детекција – локализација више објеката



CAT: (x, y, w, h)

4 numbers

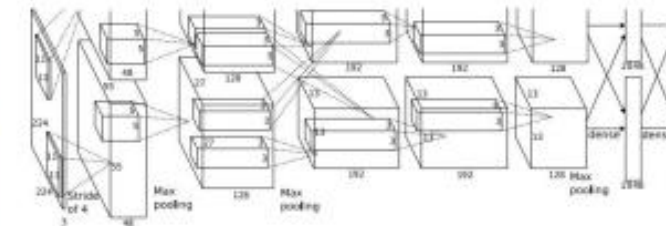


DOG: (x, y, w, h)

DOG: (x, y, w, h)

CAT: (x, y, w, h)

12 numbers



DUCK: (x, y, w, h)

DUCK: (x, y, w, h)

Many
numbers!

....

Детекција – локализација више објеката

- Може да се користи принцип клизајућих прозора (sliding windows, видети template matching)



DOG - 0
Background -1



DOG - 1
Background -0

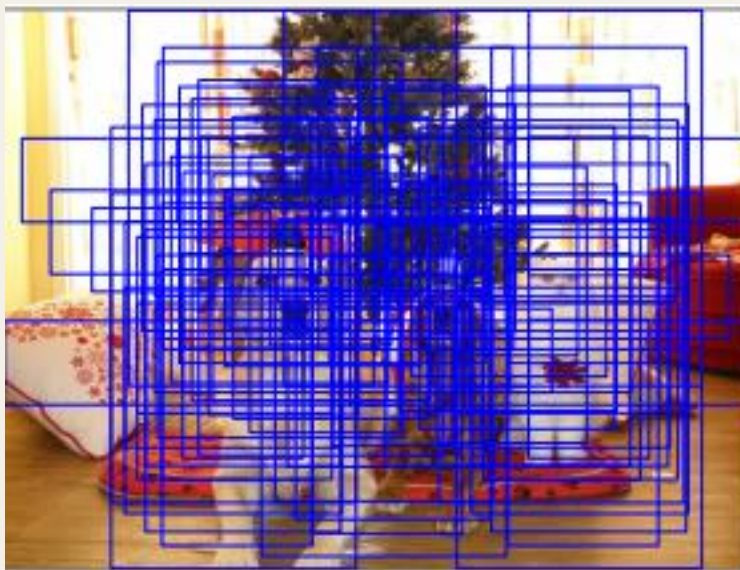


DOG - 1
Background -0

Детекција – локализација више објеката

- Може да се користи принцип клизајућих прозора (sliding windows, видети template matching)

Недостатак: потребно је применити мрежу на много локација на слици, узимајући у обзир да се објекат од интереса може налазити у различитим размерама на слици



DOG - 0
Background - 1

Детекција – локализација више објеката

- Две групе алгоритама
 - Region-based detection (R-CNN, Fast R-CNN, Faster R-CNN)
 - Detections without proposals (YOLO)

Детекција – поређења алгоритама

Accurate object detection is slow!

	Pascal 2007 mAP	Speed	
DPM v5	33.7	.07 FPS	14 s/img
R-CNN	66.0	.05 FPS	20 s/img



1/3 Mile, 1760 feet

~500m

Детекција – поређења алгоритама

Accurate object detection is slow!

	Pascal 2007 mAP	Speed	
DPM v5	33.7	.07 FPS	14 s/img
R-CNN	66.0	.05 FPS	20 s/img
Fast R-CNN	70.0	.5 FPS	2 s/img



176 feet

~50m

Детекција – поређења алгоритама

Accurate object detection is slow!

	Pascal 2007 mAP	Speed	
DPM v5	33.7	.07 FPS	14 s/img
R-CNN	66.0	.05 FPS	20 s/img
Fast R-CNN	70.0	.5 FPS	2 s/img
Faster R-CNN	73.2	7 FPS	140 ms/img



Детекција – поређења алгоритама

Accurate object detection is slow!

	Pascal 2007 mAP	Speed	
DPM v5	33.7	.07 FPS	14 s/img
R-CNN	66.0	.05 FPS	20 s/img
Fast R-CNN	70.0	.5 FPS	2 s/img
Faster R-CNN	73.2	7 FPS	140 ms/img
YOLO	63.4	45 FPS	22 ms/img



2 feet



~0.6m