

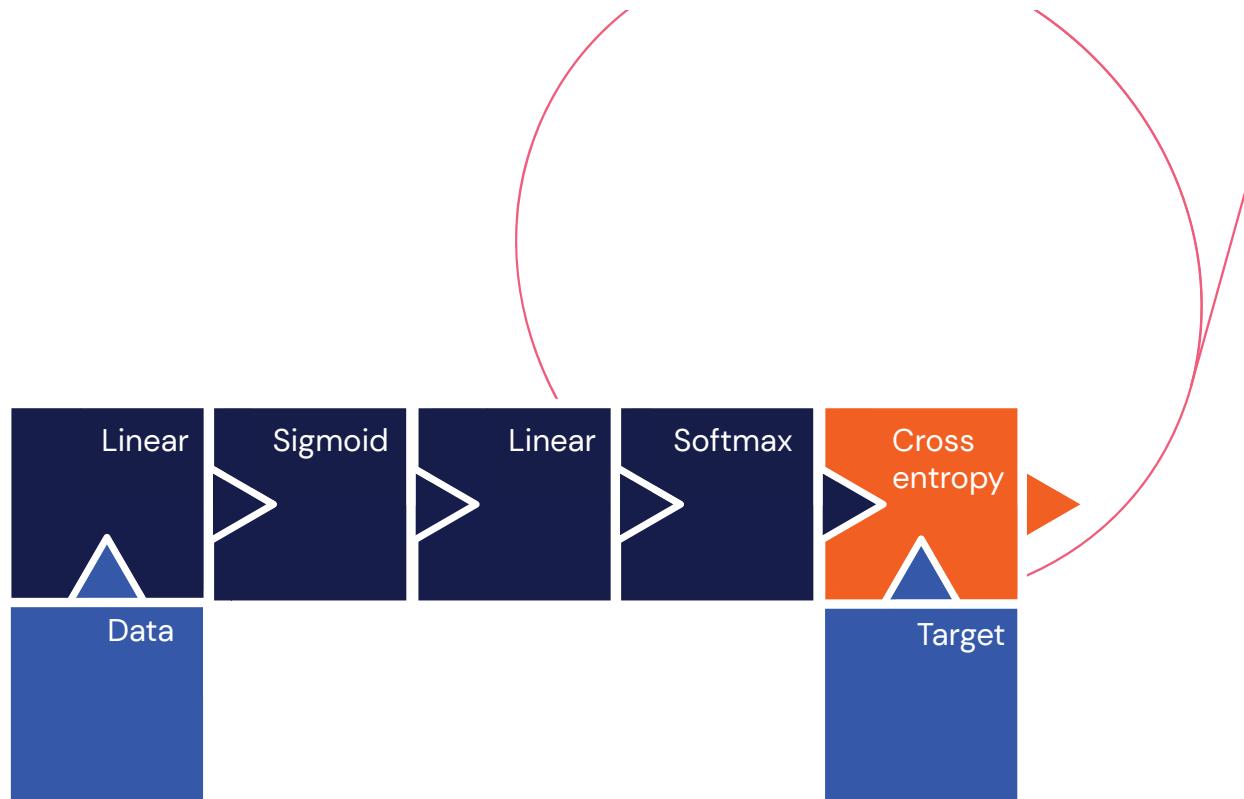
Въведение в дълбокото обучение.
Системи, използващи невронни
мрежи с дълбоко обучение, модели и
структура на конволовационните
невронни мрежи.





През последното десетилетие масовото въвеждане в експлоатация на дълбоки невронни мрежи с конволюционни слоеве доведе до революция в методите и апаратата за компютърно зрение, благодарение на което те може да се използват за приложения в реално време и което позволява например автономното управление на автомобили и др. дейности. С нововъведението да може процесорната логика да е с хиляди изчислителни единици (графични процесори, клъстери с хиляди CPU/GPU предоставящи supercomputing функционалност) направи възможно обучението на невронни мрежи с много слоеве, някои от които трицифрене число.

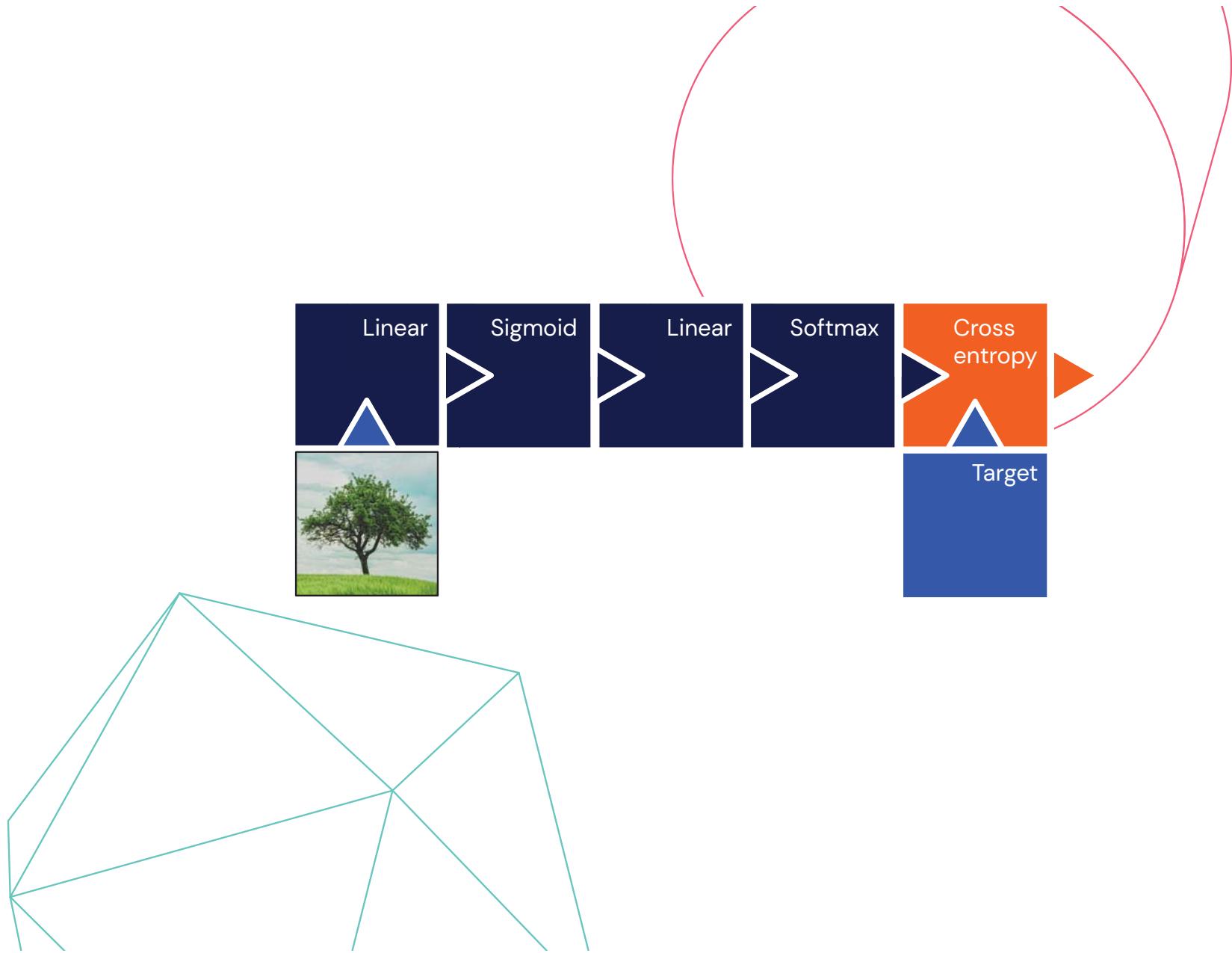




По какъв начин се подават изображения
към една Невронна Мрежа?







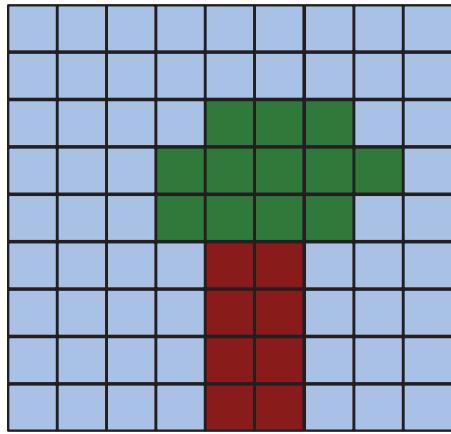
Невронни мрежи с приложение за изображения



Едно дигитално изображение представлява двумерна (2D) решетка от пиксели.



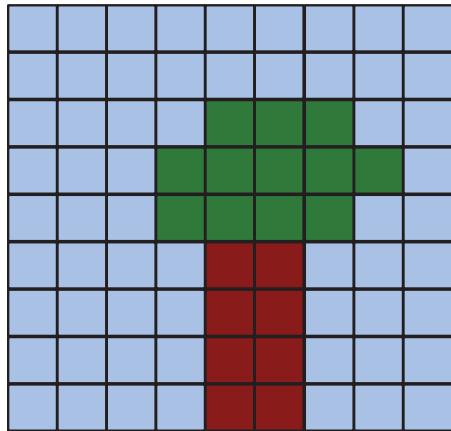
Невронни мрежи с приложение за изображения



Едно дигитално изображение представлява двумерна (2D) решетка от пиксели.



Невронни мрежи с приложение за изображения

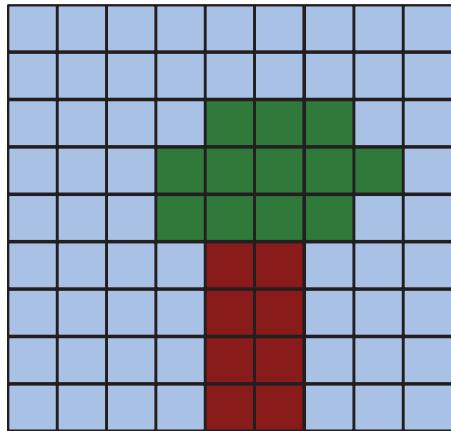


Едно дигитално изображение представлява двумерна (2D) решетка от пиксели.

Невронната мрежа приема/очаква вектор **съставен от числа** за входен образ.



Невронни мрежи с приложение за изображения

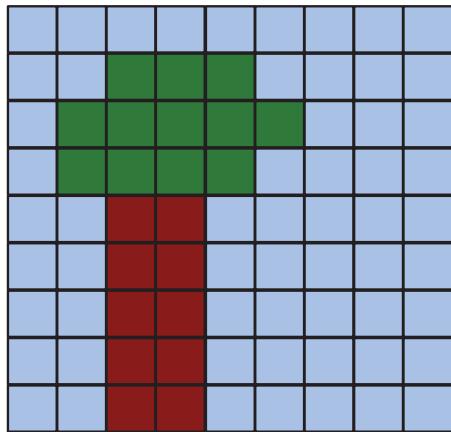


Едно дигитално изображение представлява двумерна (2D) решетка от пиксели.

Невронната мрежа приема/очаква вектор **съставен от числа** за входен образ.



Невронни мрежи с приложение за изображения



Едно дигитално изображение представлява двумерна (2D) решетка от пиксели.

Невронната мрежа приема/очаква вектор **съставен от числа** за входен образ.



Локалност и трансляционна инвариантност



Локалност: околните пиксели са по-строго свързани едни с други (корелирани)

Трансляционна инвариантност: имащи търсен смисъл/значение последователности могат да се появяват на всяко място в изображението



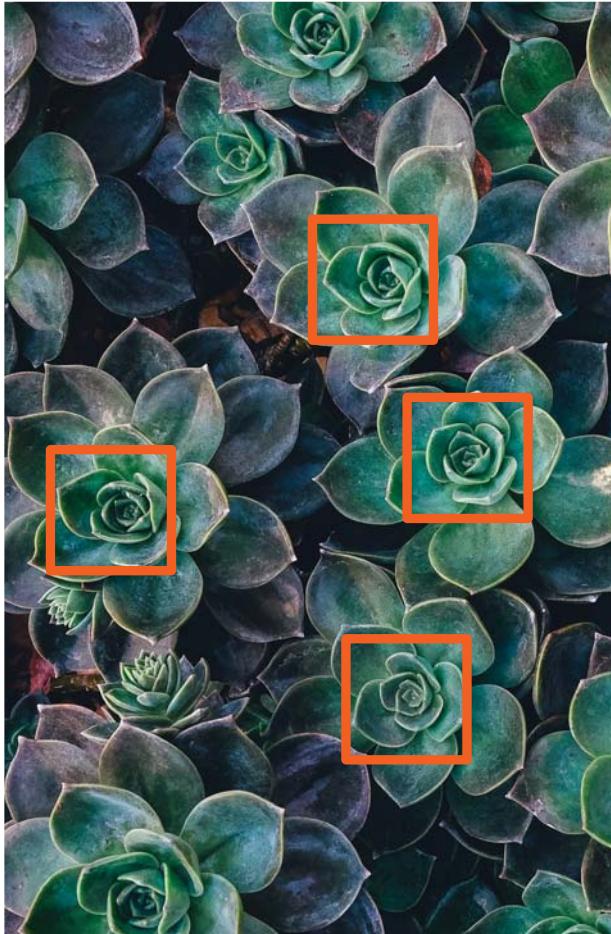




Извличане на полезна информация от топологичната структура



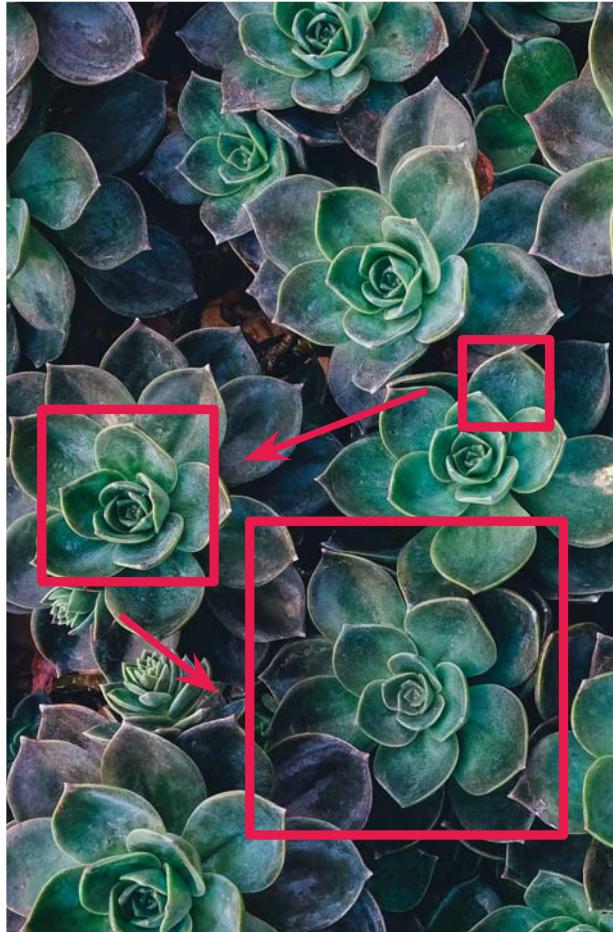
Извличане на полезна информация от топологичната структура



Споделяне на теглата: използват се едни и същи
мрежови параметри за откриване на локални модели но
на много места в изображението



Извличане на полезна информация от топологичната структура



Споделяне на теглата: използват се едни и същи мрежови параметри за откриване на локални модели на много места в изображението

Йерархия: локални елементарни примитиви (признания от ниско ниво) се групират в по-големи и по-абстрактни класове и признания



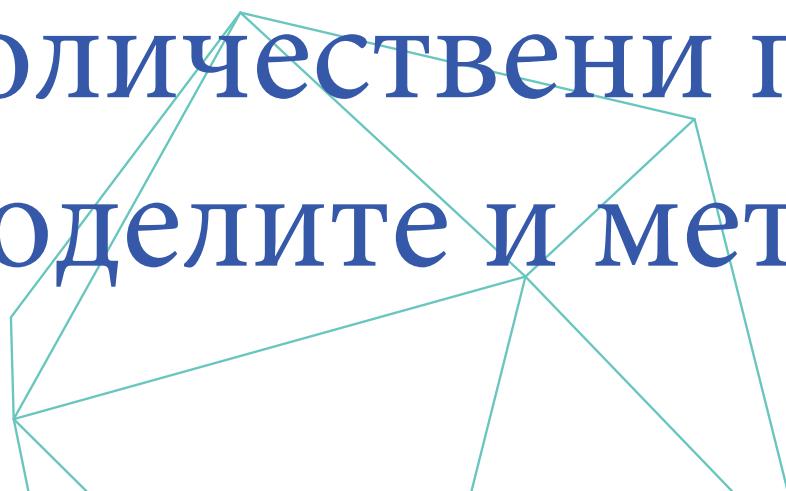
краища на
примитиви
и текстури

части от обекти

обекти



Как повишения обем на
информацията, с която работим
довежда до качествени и
количествени подобрения в
моделите и методиката



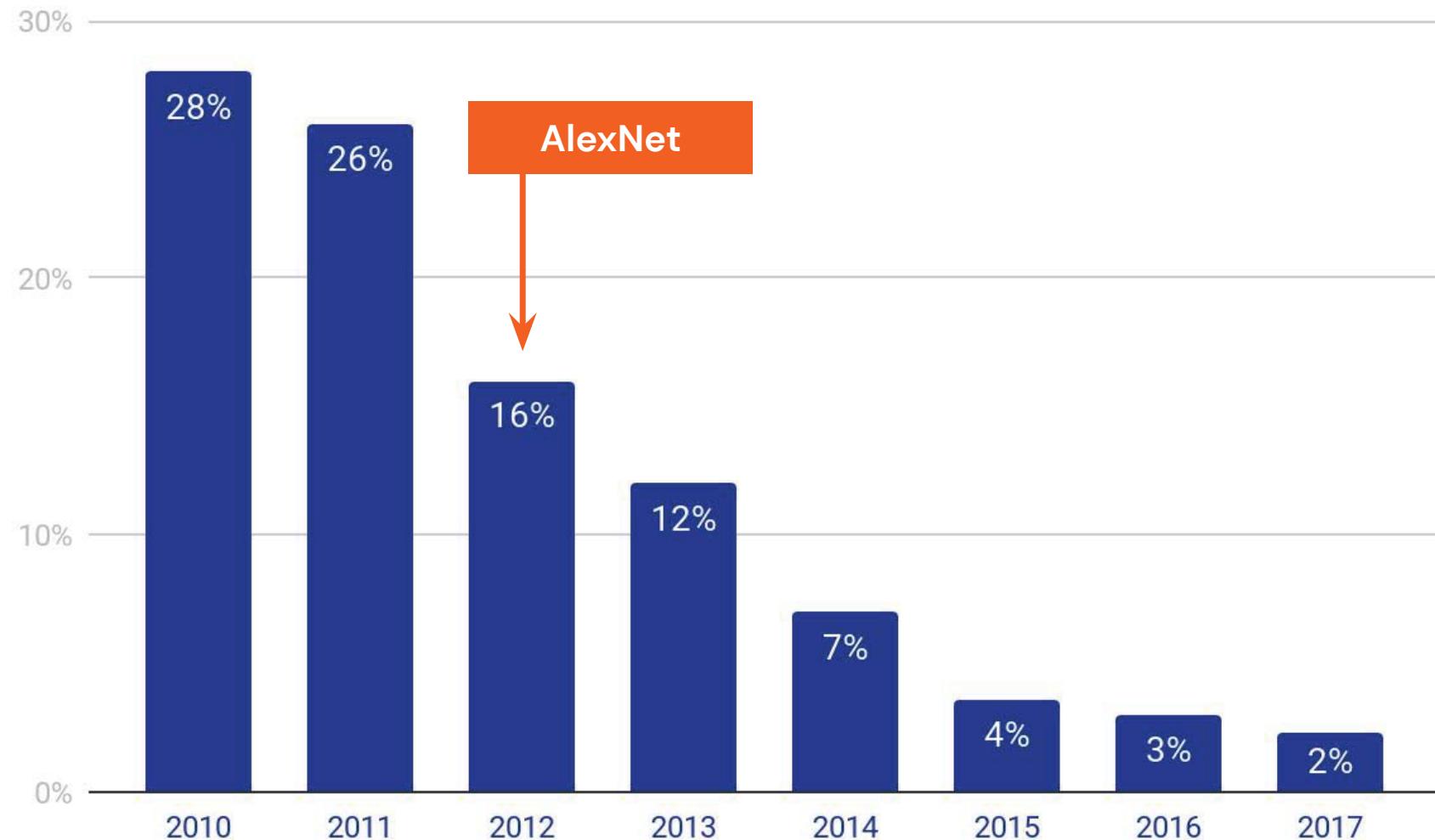
Състезанието ImageNet

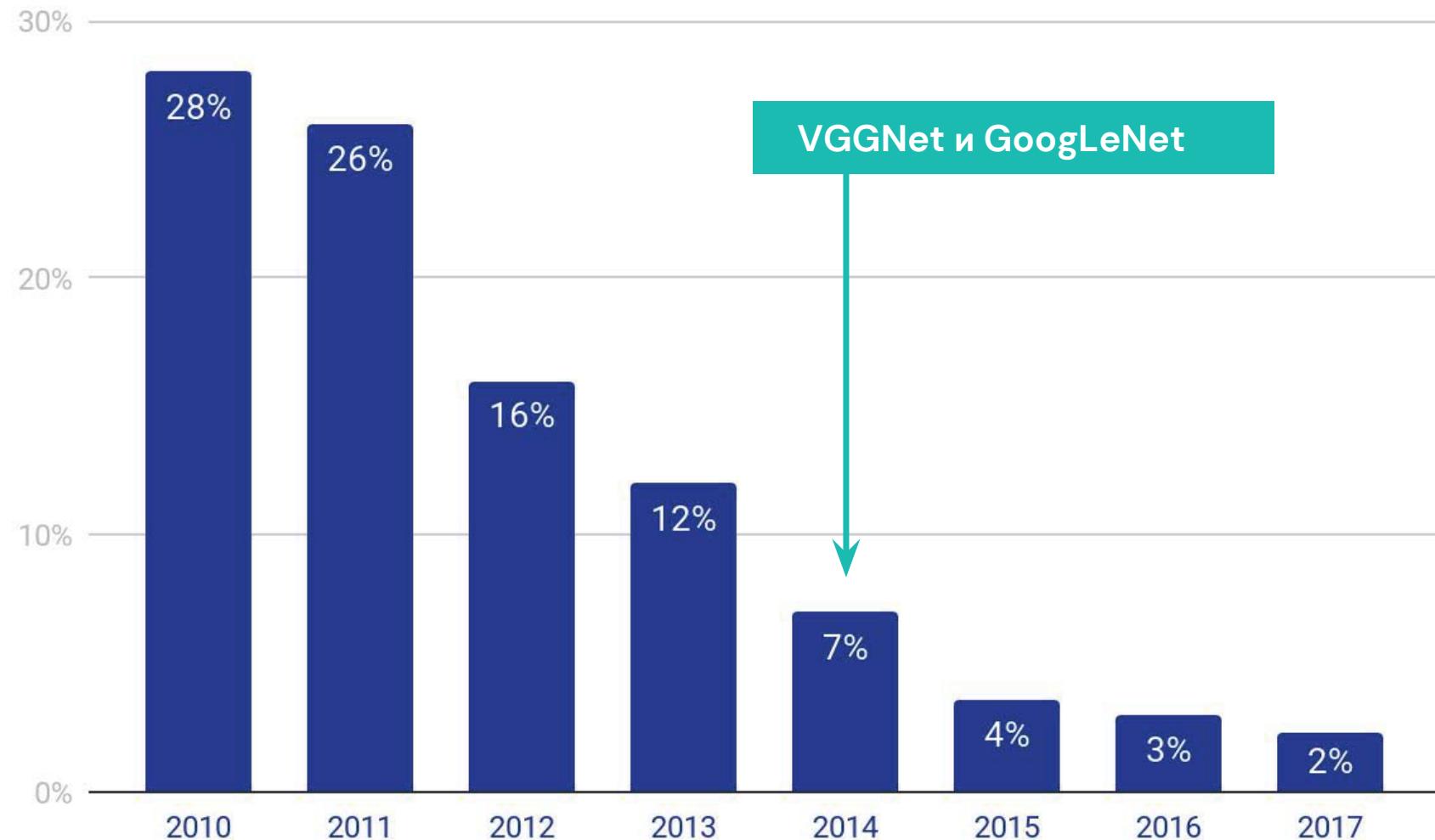
- Еталон за достиженията в компютърното зрение
- Провежда се от 2010 до 2017
- 1.4M изображения, 1000 класа
- Класификация на изображения е задачата на състезанието

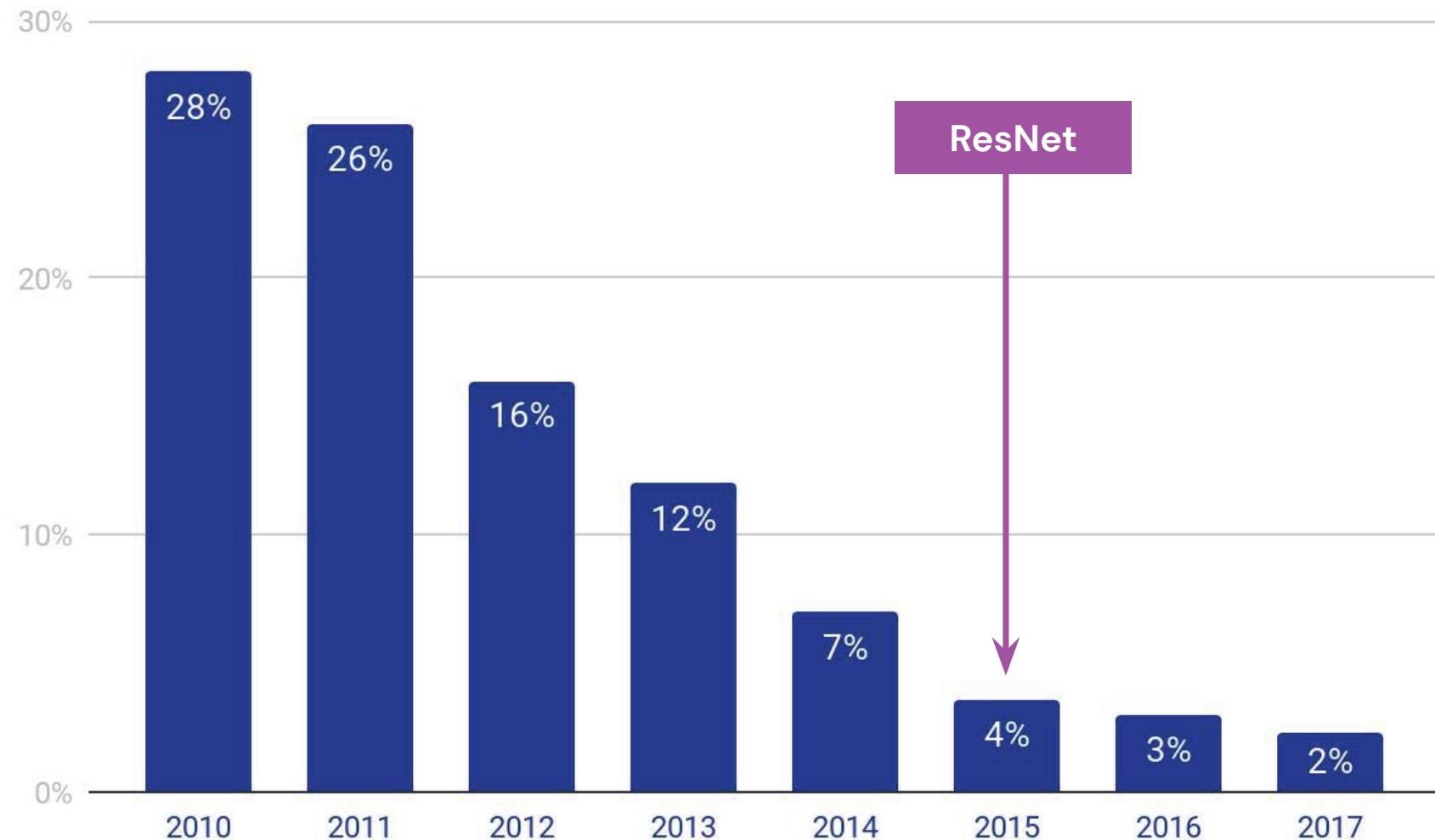








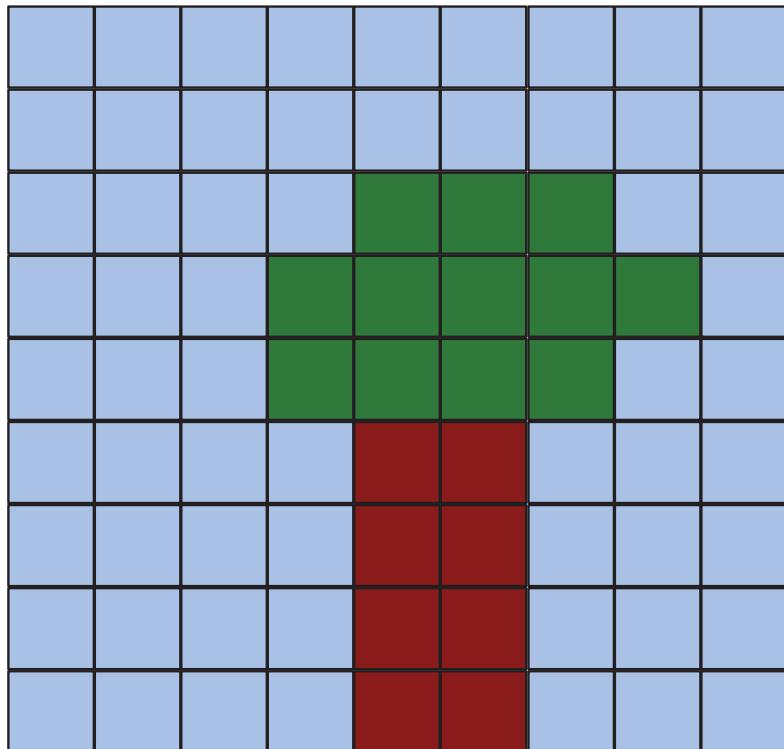




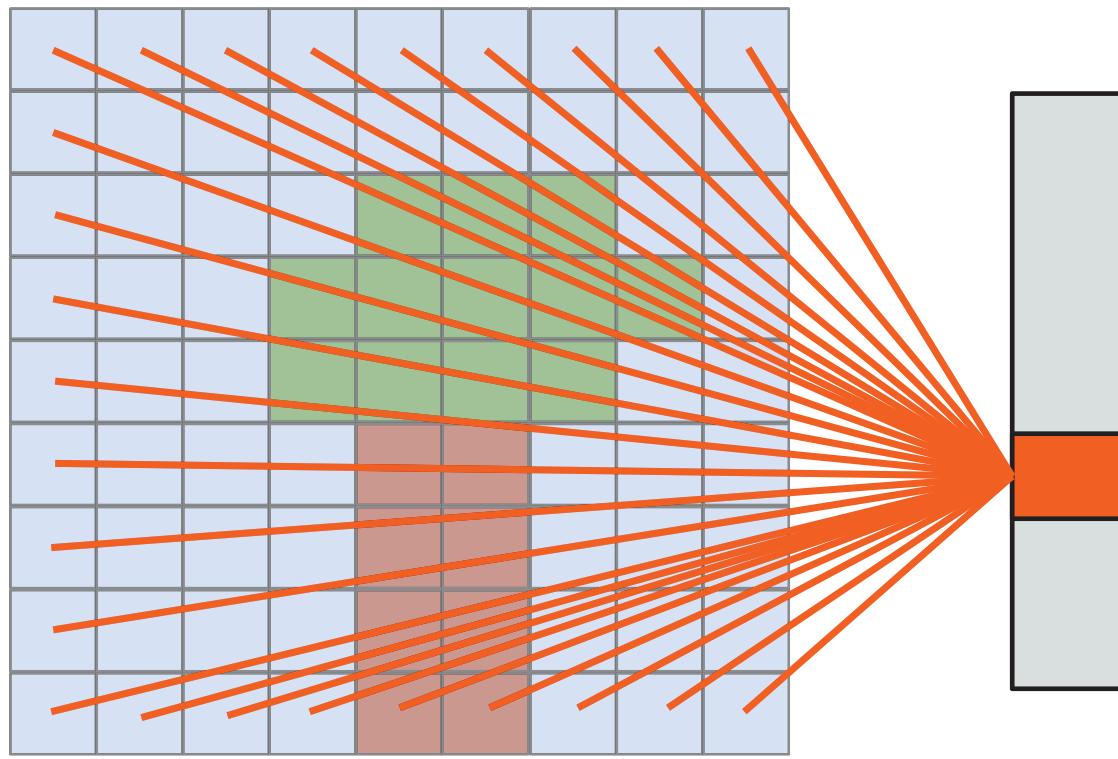
Преход от пълна към локална свързаност



Преход от пълна към локална свързаност



Преход от пълна към локална свързаност

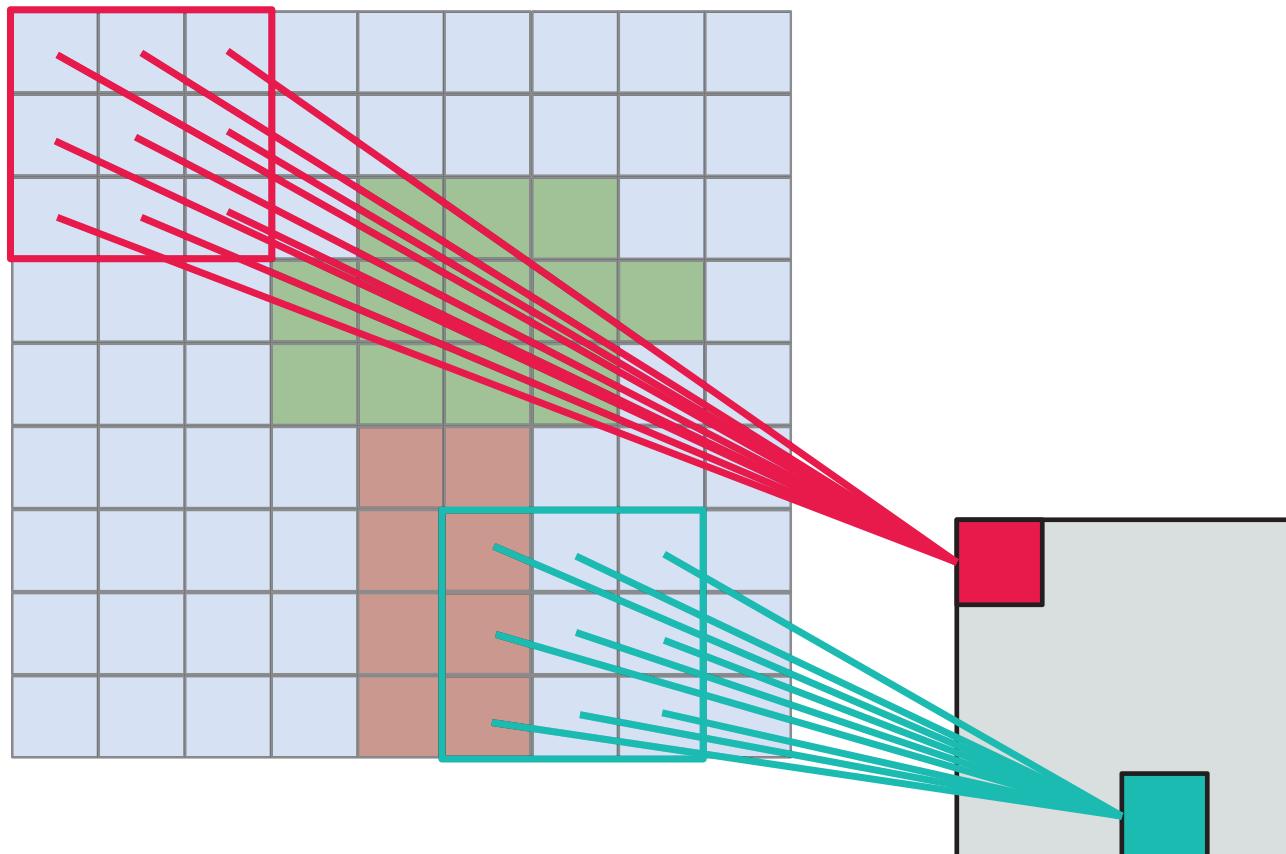


Напълно свързана мрежа

$$y = \sum_{i \in \text{image}} \mathbf{w}_i \mathbf{x}_i + b$$



Преход от пълна към локална свързаност

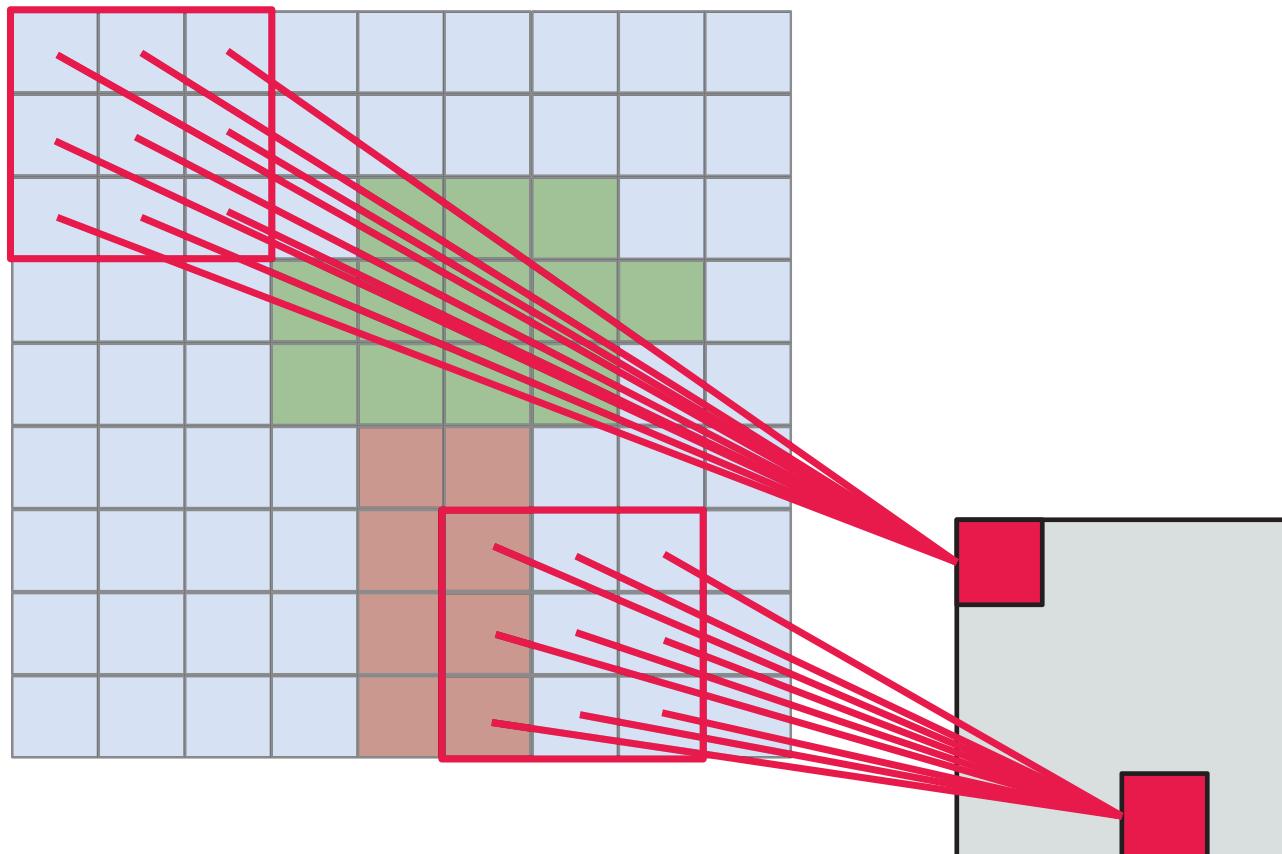


$$y = \sum_{i \in 3 \times 3} \mathbf{w}_i \mathbf{x}_i + b$$

Локално-свързани елементи
в групи от 3×3 рецептивни
полета



Преход от пълна към локална свързаност

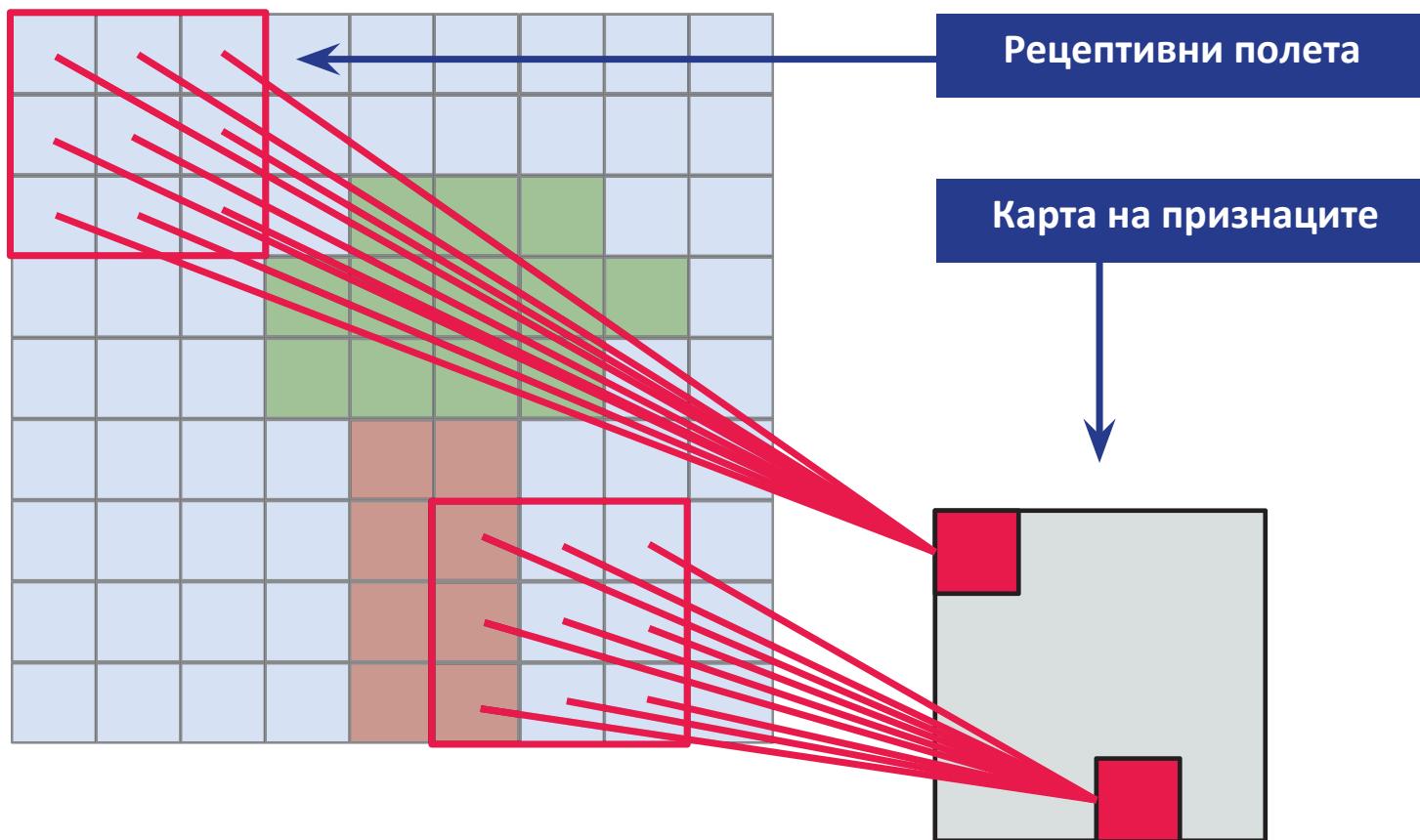


$$y = \mathbf{w} * \mathbf{x} + b$$

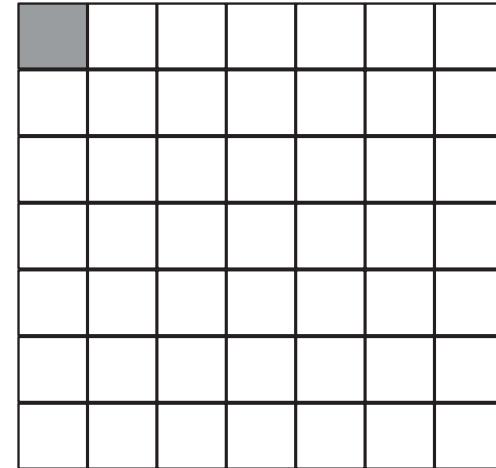
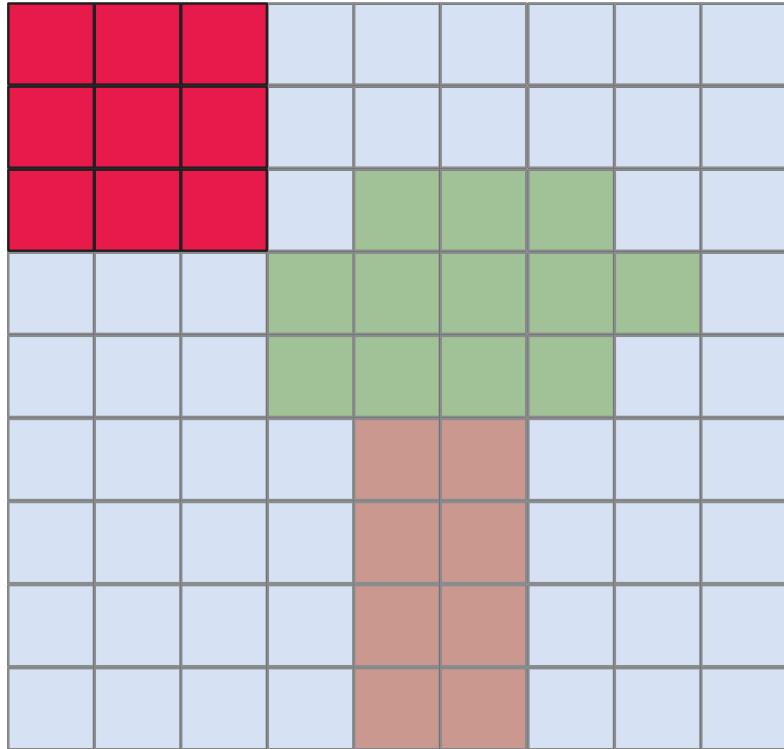
конволовционни елементи с
3×3 рецептивни полета



Преход от пълна към локална свързаност



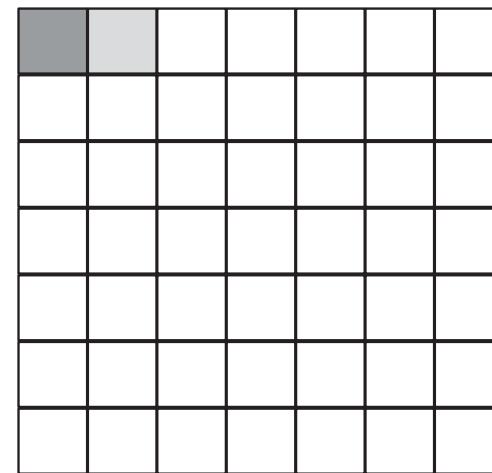
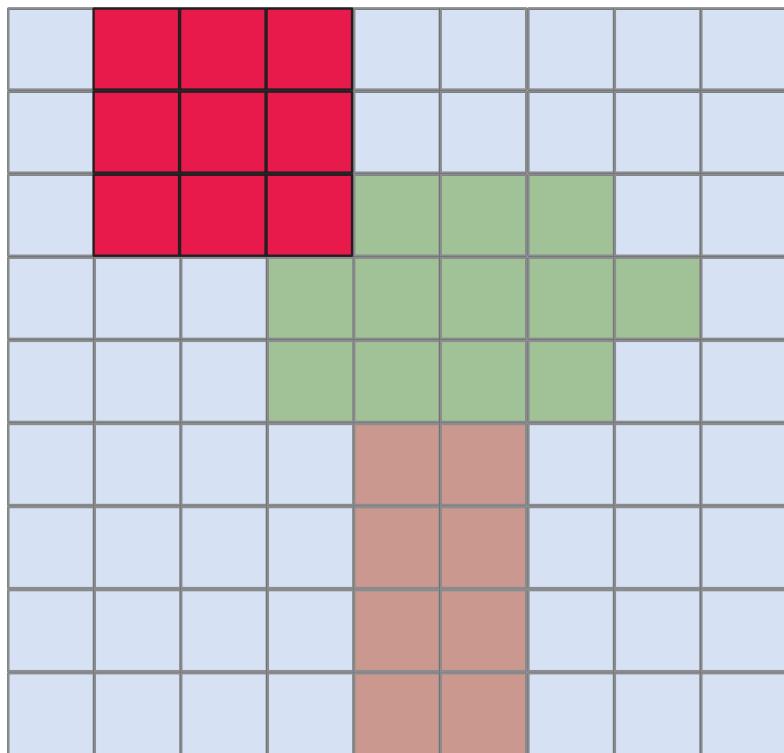
Реализация: прилагане на конволюцията



Ядрото се транслира през изображението и създава изходна стойност за всяка позиция



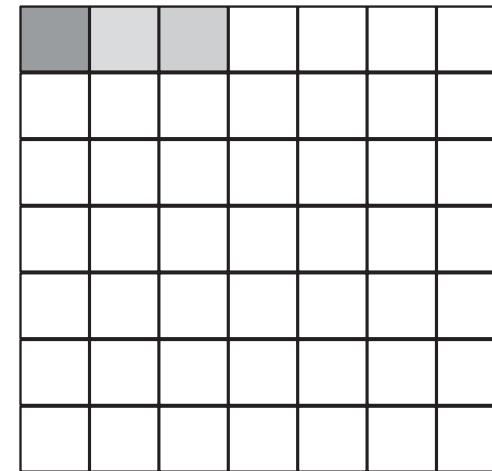
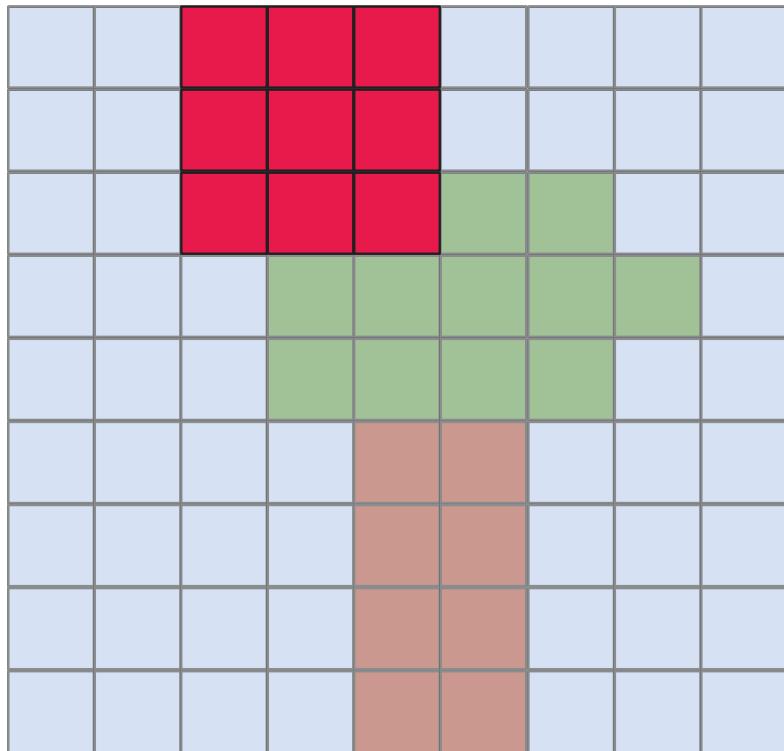
Реализация: прилагане на конволюцията



Ядрото се транслира през изображението и създава изходна стойност за всяка позиция



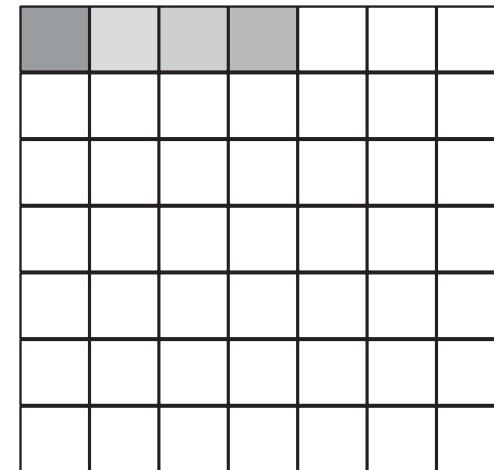
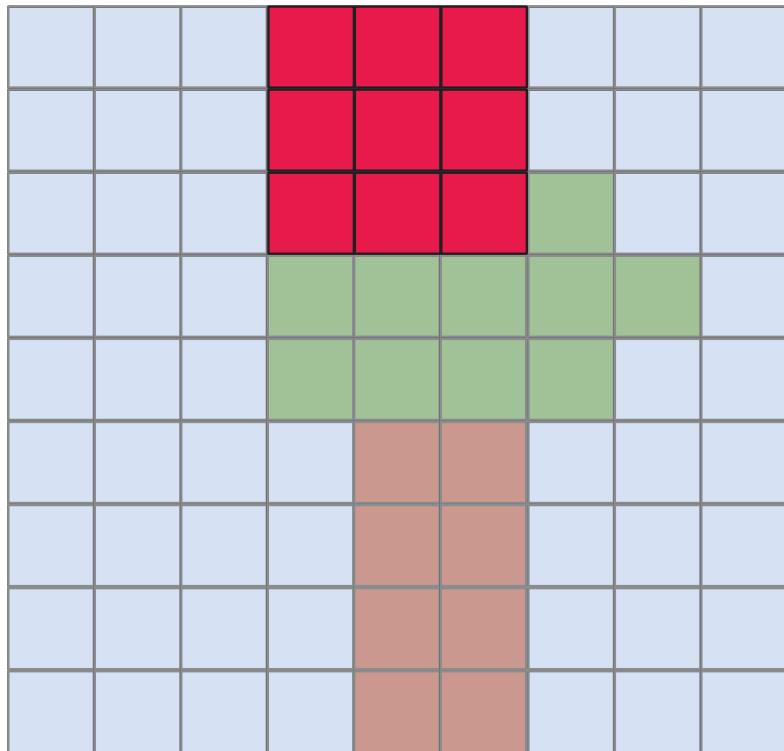
Реализация: прилагане на конволюцията



Ядрото се транслира през изображението и създава изходна стойност за всяка позиция



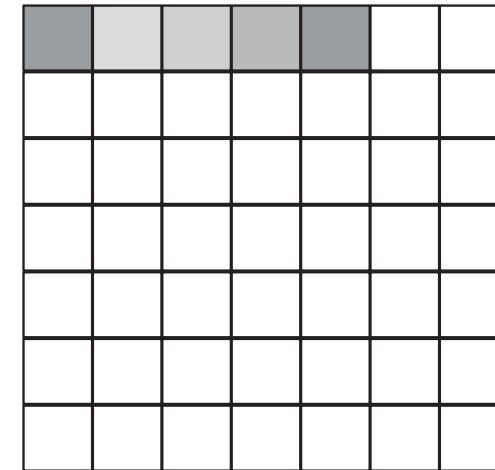
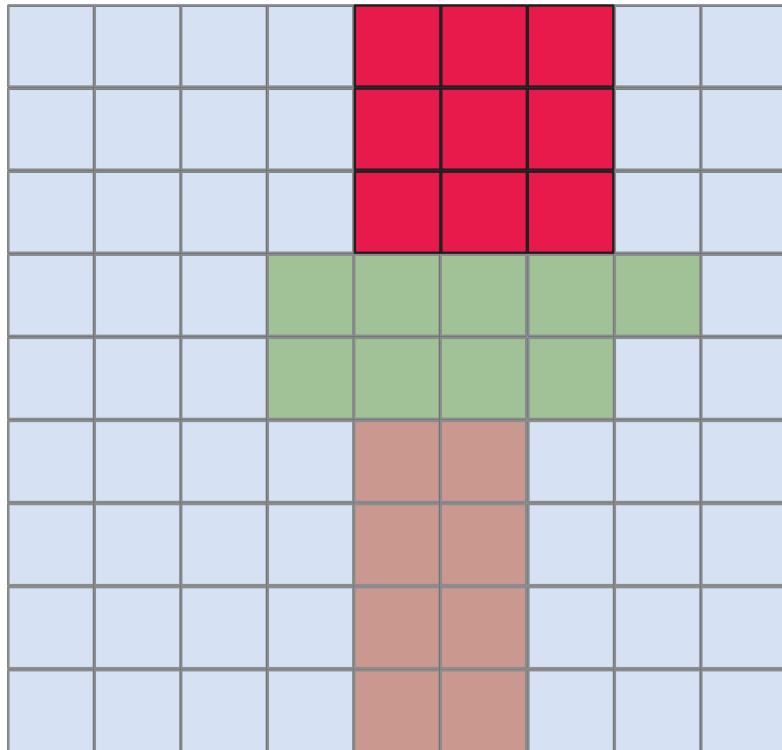
Реализация: прилагане на конволюцията



Ядрото се транслира през изображението и създава изходна стойност за всяка позиция



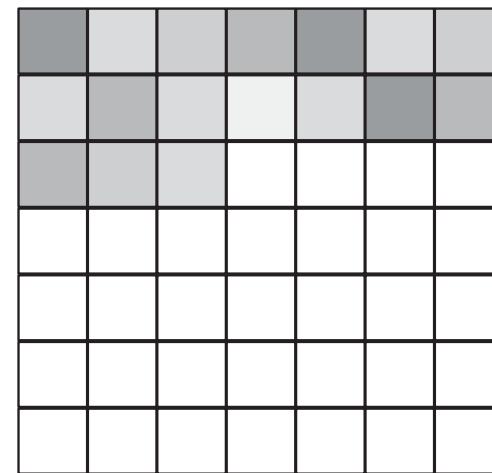
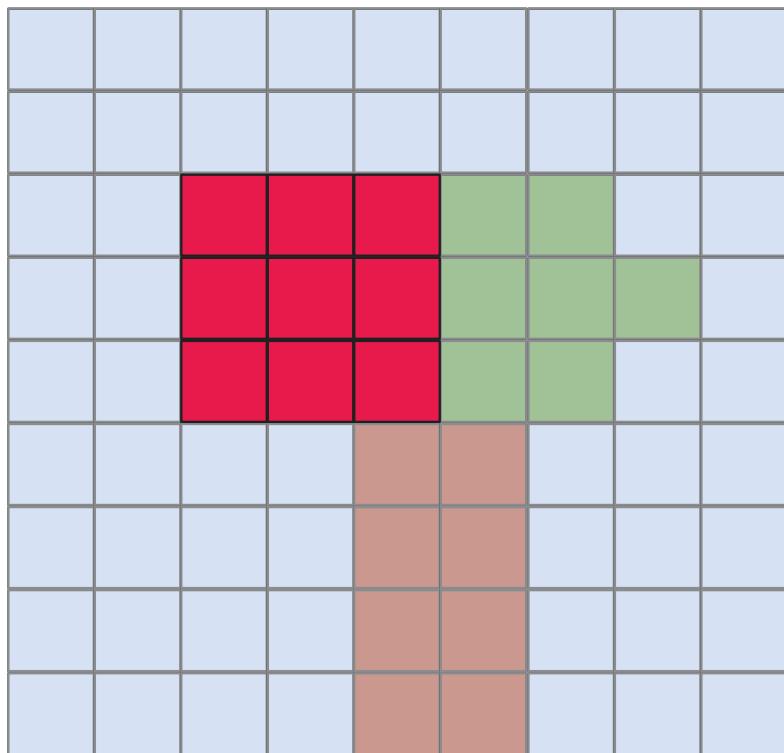
Реализация: прилагане на конволюцията



Ядрото се транслира през изображението и създава изходна стойност за всяка позиция



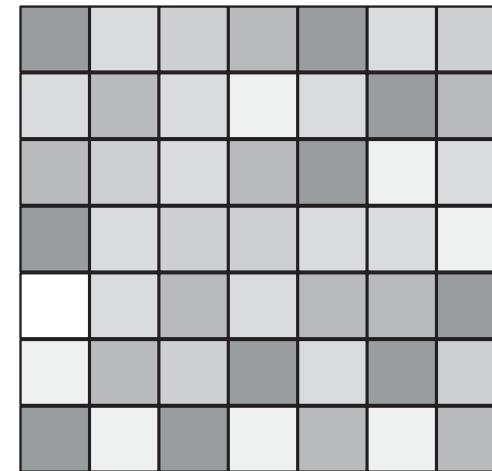
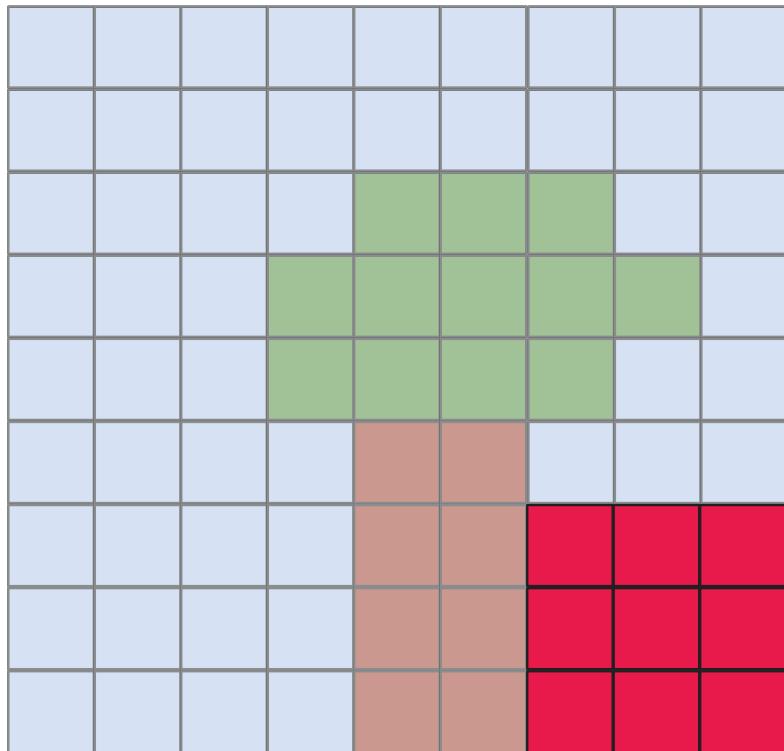
Реализация: прилагане на конволюцията



Ядрото се транслира през изображението и създава изходна стойност за всяка позиция



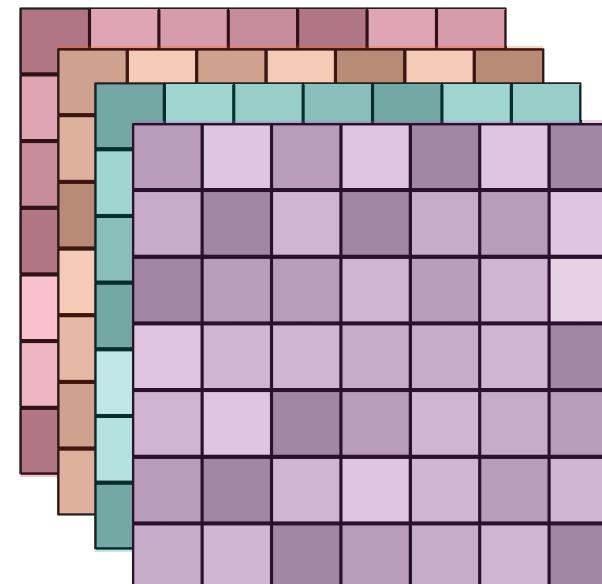
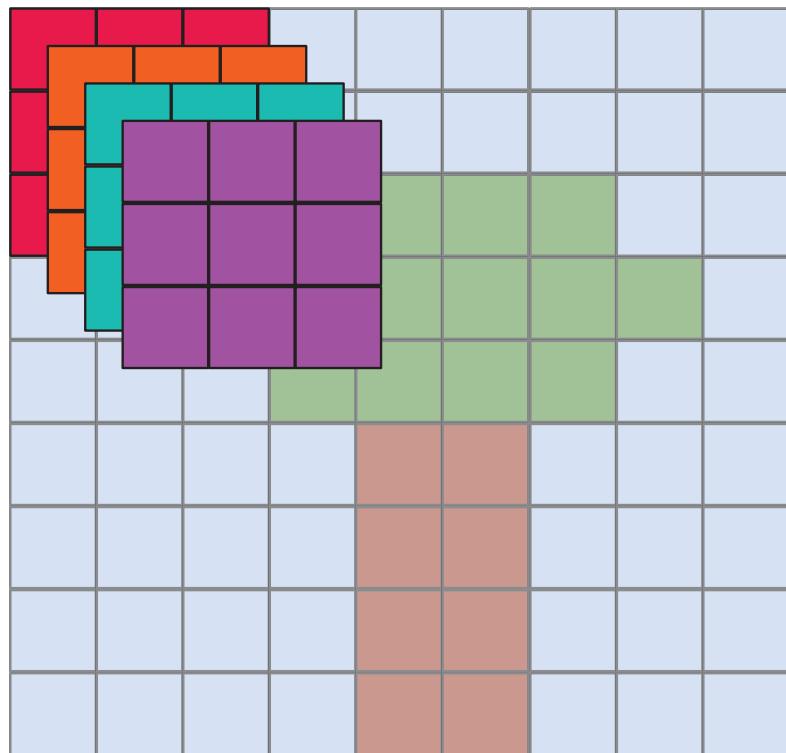
Реализация: прилагане на конволюцията



Ядрото се транслира през изображението и създава изходна стойност за всяка позиция



Реализация: прилагане на конволюцията



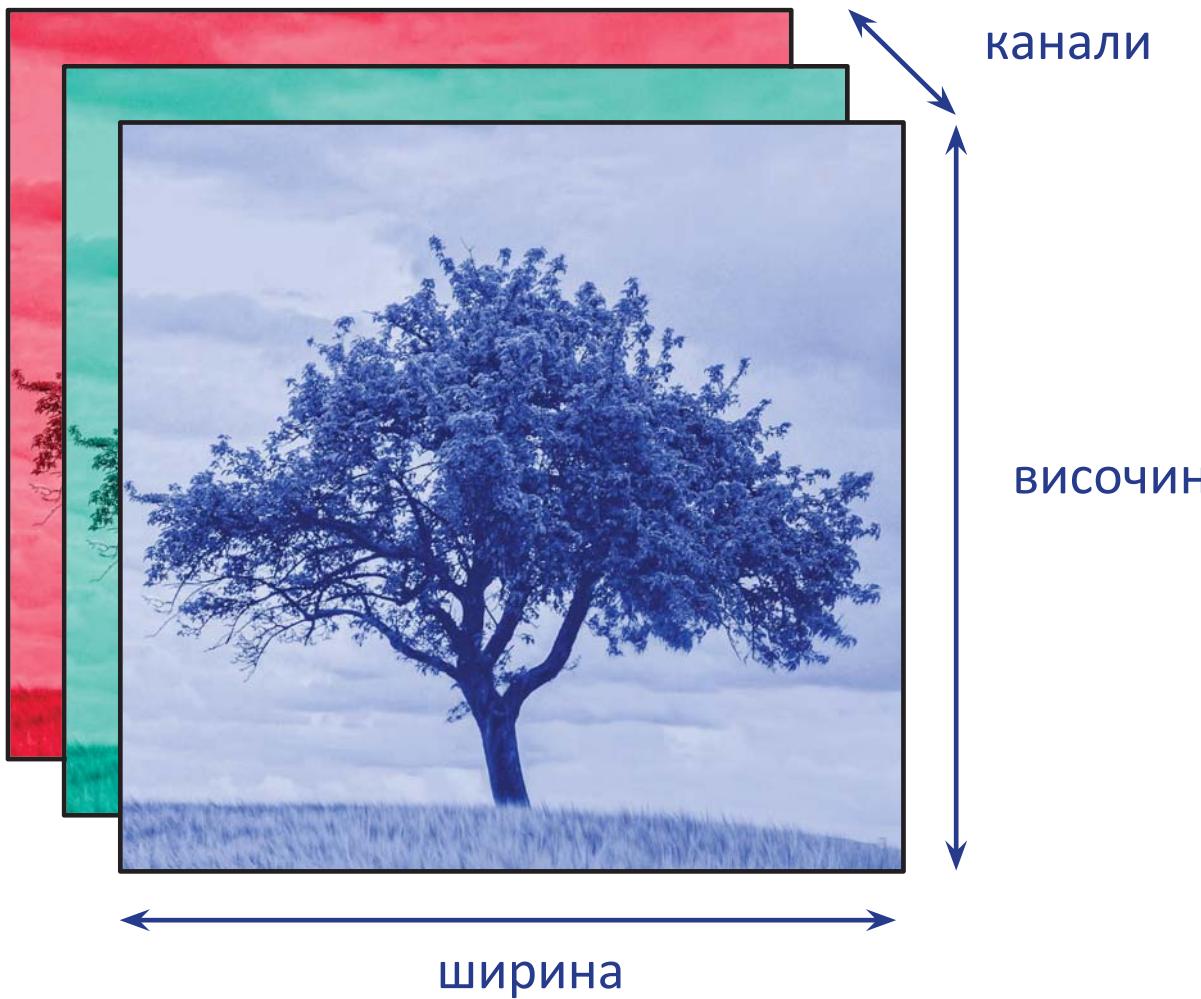
Прилагат се и се варират множество ядра, с които се извличат множество карти на признаките, които се групират в **канали**



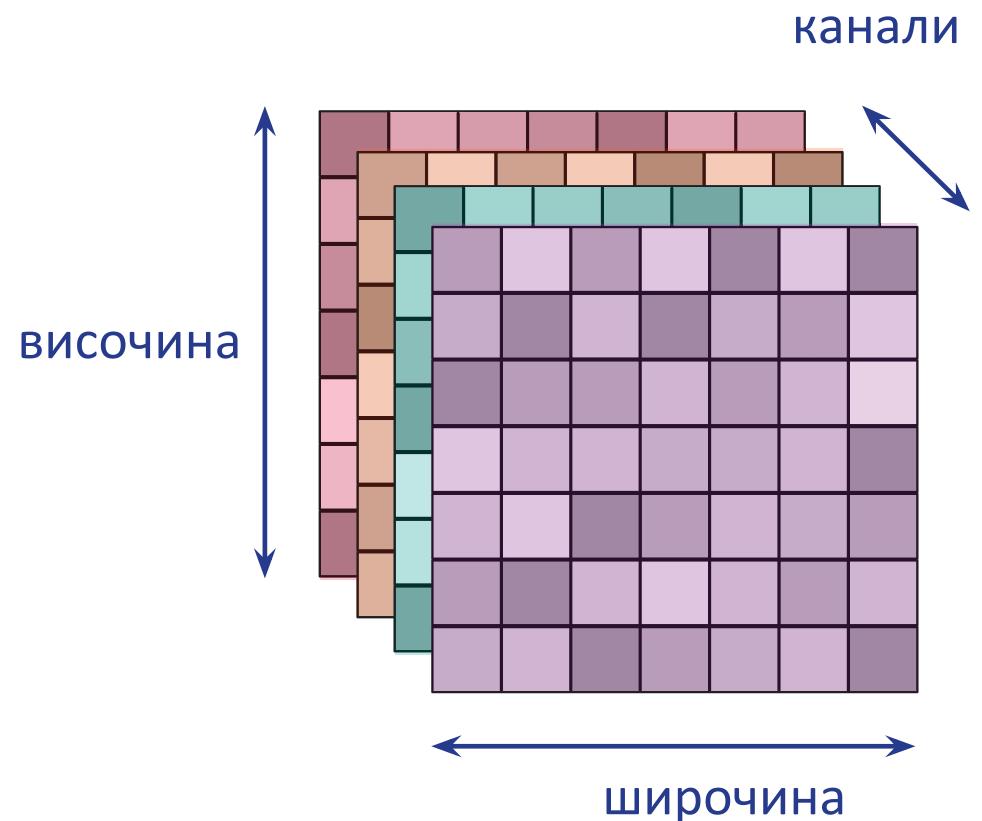
Входовете и изходите представляват тензори



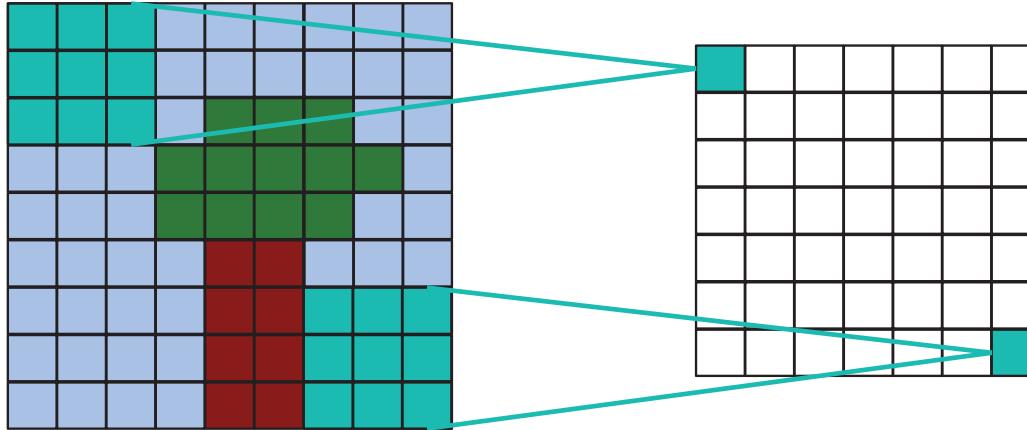
Входовете и изходите представляват тензори



Входовете и изходите представляват тензори



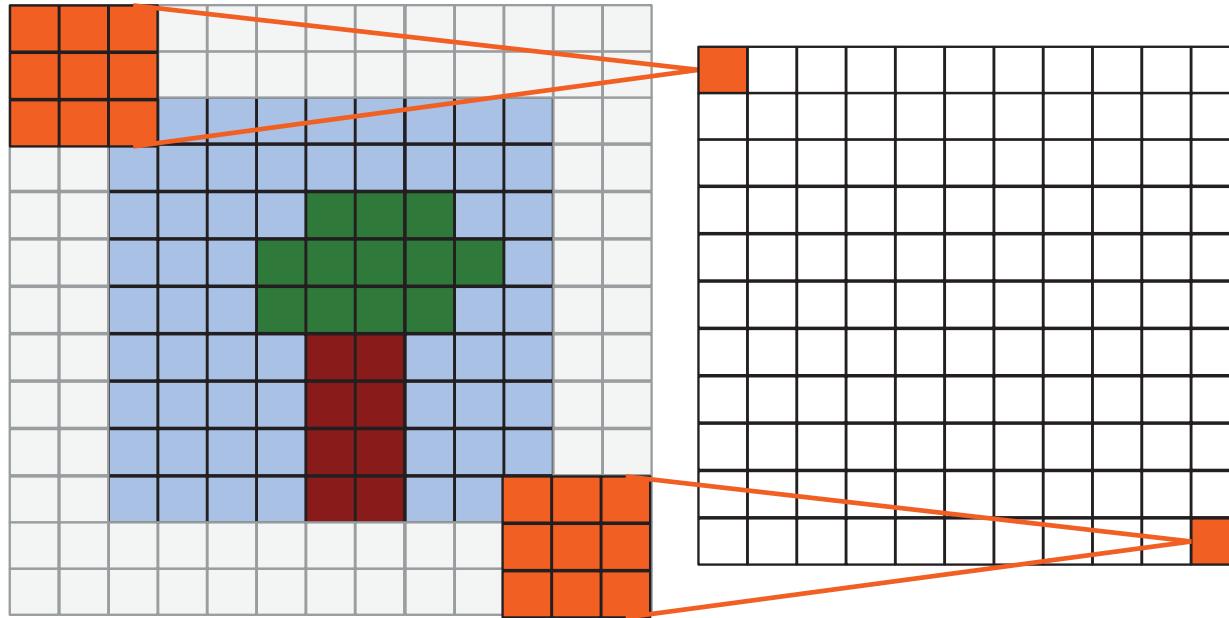
Варианти на прилагане на конволюцията



Валидна конволюция: размер на изходния слой = размер на входа – големина на ядрото + 1
output size = input size - kernel size + 1



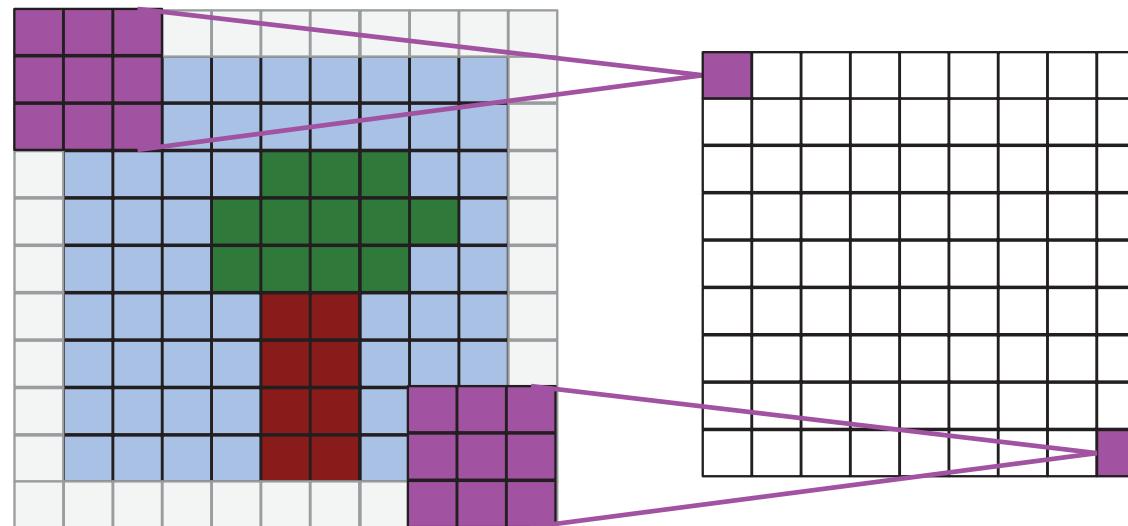
Варианти на прилагане на конволюцията



Пълна конволюция: output size = input size + kernel size - 1



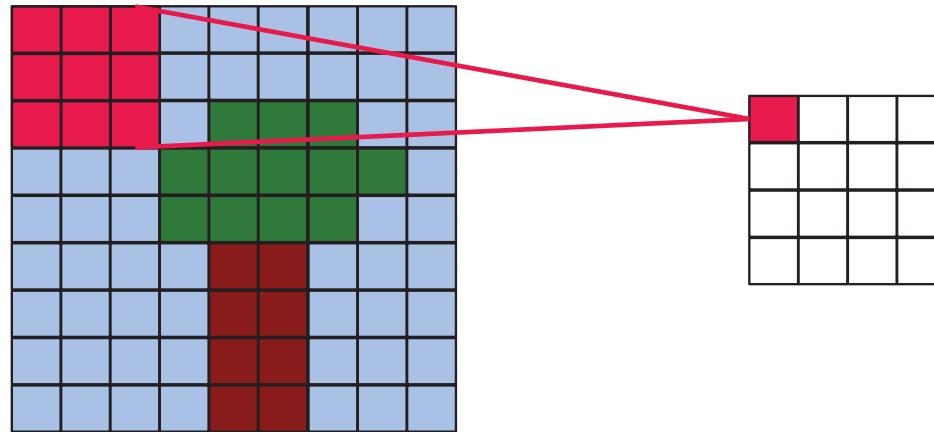
Варианти на прилагане на конволюцията



Същата/Еднаква конволюция: output size = input size



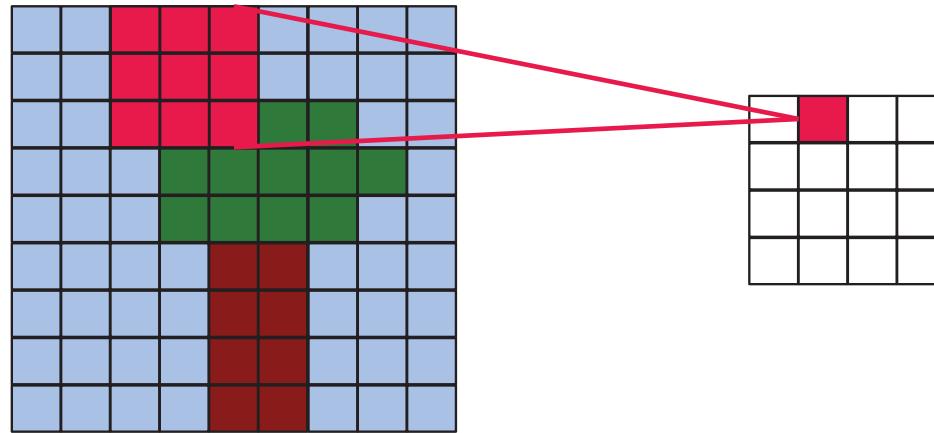
Варианти на прилагане на конволюцията



Разкрачена (Strided) конволюция: ядрото се транслира по изображението със стъпка > 1



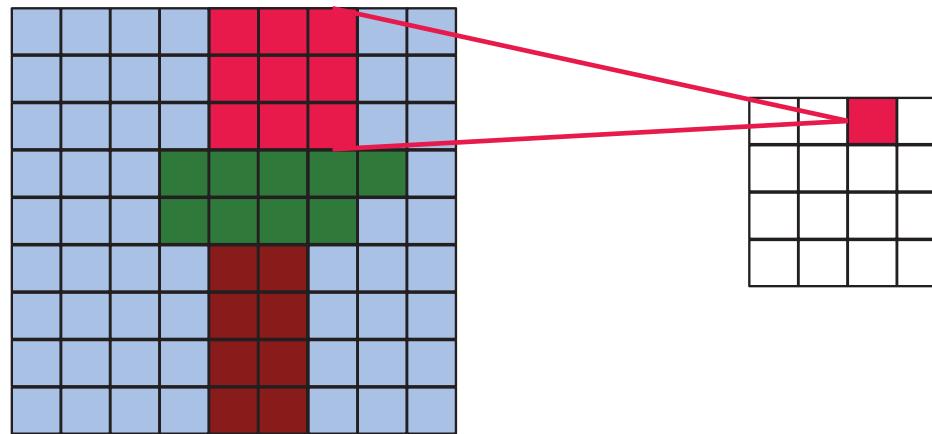
Варианти на прилагане на конволюцията



Разкрачена (Strided) конволюция: ядрото се транслира по изображението със стъпка > 1



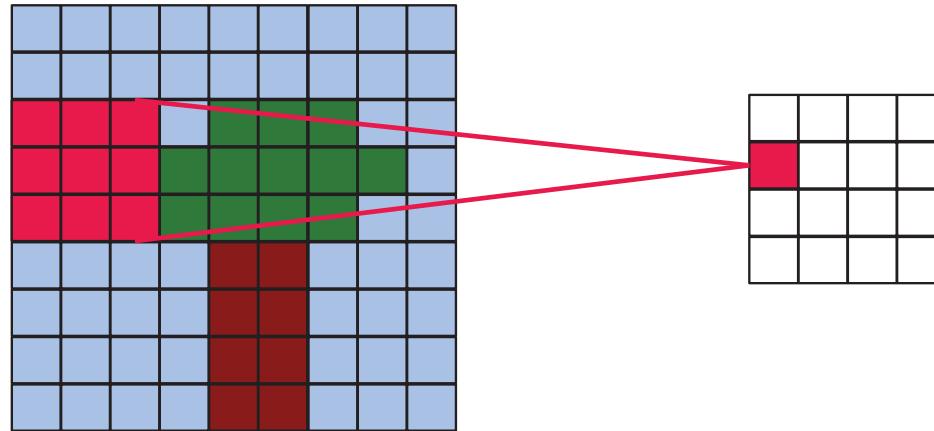
Варианти на прилагане на конволюцията



Разкрачена (Strided) конволюция: ядрото се транслира по изображението със стъпка > 1



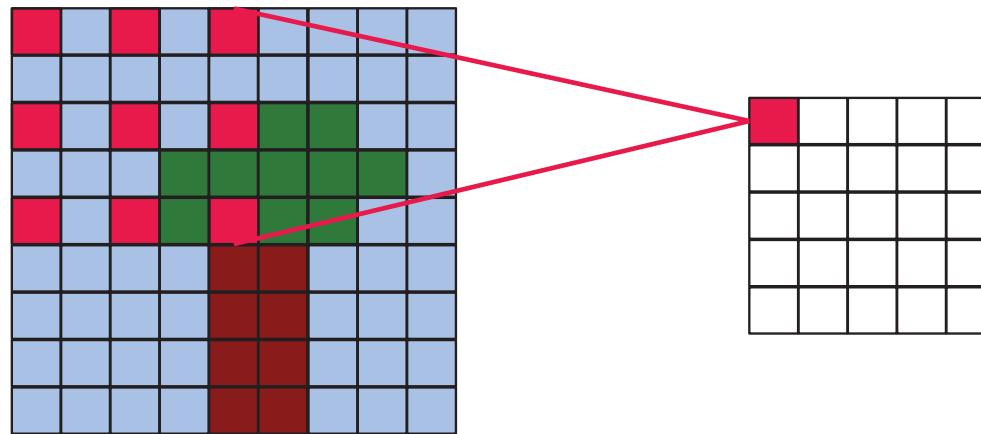
Варианти на прилагане на конволюцията



Разкрачена (Strided) конволюция: ядрото се транслира по изображението със стъпка > 1



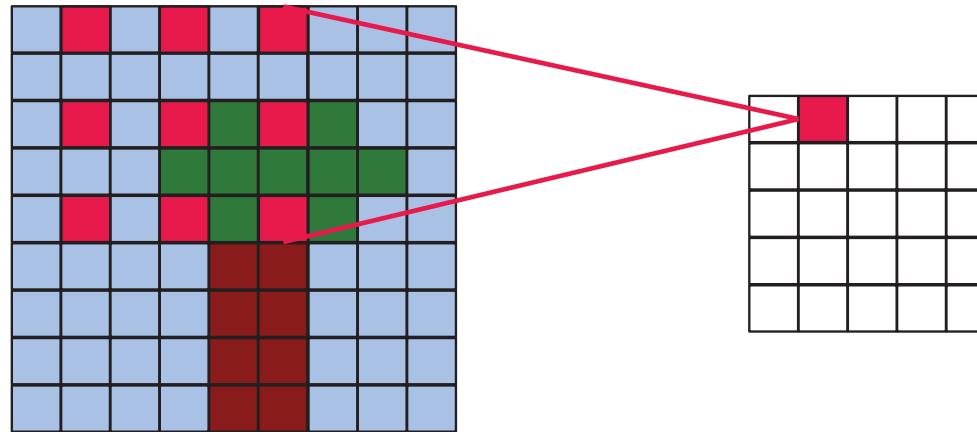
Варианти на прилагане на конволюцията



Разширена конволюция: ядрото се разпростира навън, стъпка > 1 между всички елементи тип ядро



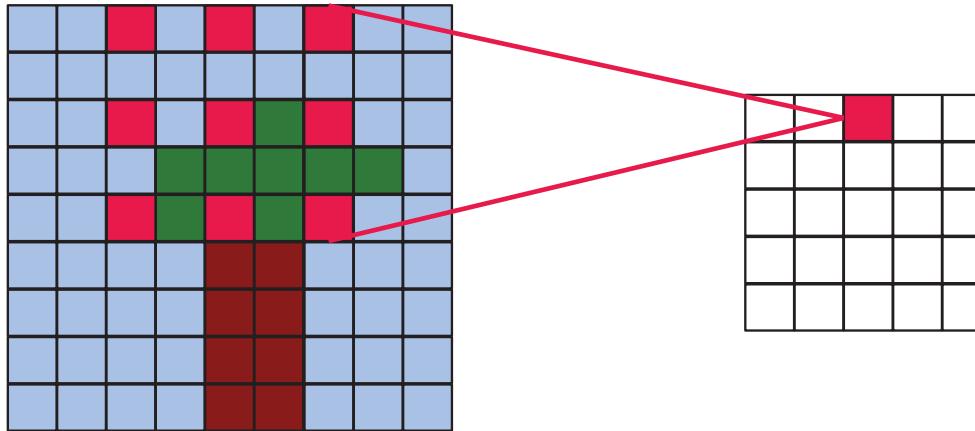
Варианти на прилагане на конволюцията



Разширена конволюция: ядрото се разпростира навън, стъпка > 1 между всички елементи тип ядро



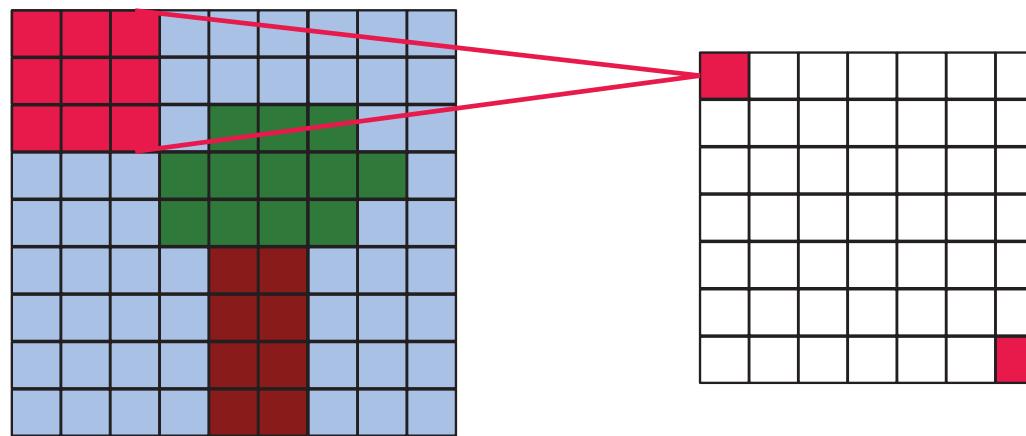
Варианти на прилагане на конволюцията



Разширена конволюция: ядрото се разпростира навън, стъпка > 1 между всички елементи тип ядро



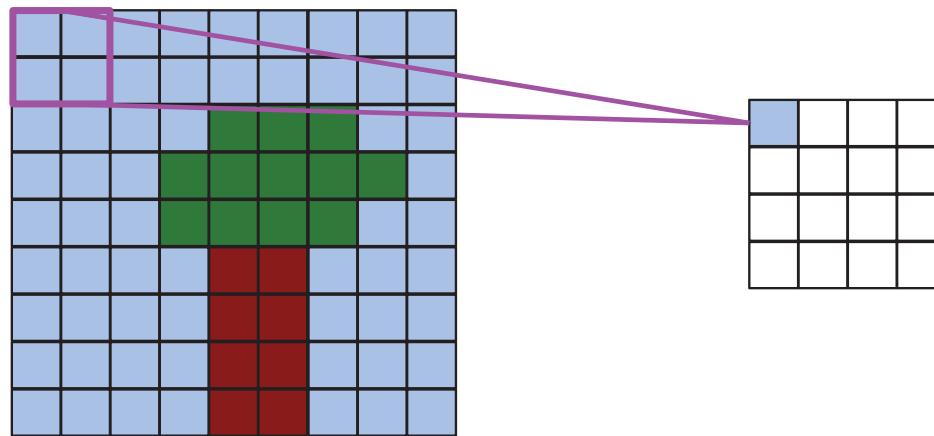
Варианти на прилагане на конволюцията



Конволюция в дълбочина: всеки изходен канал е свързан само с един канал от входния слой



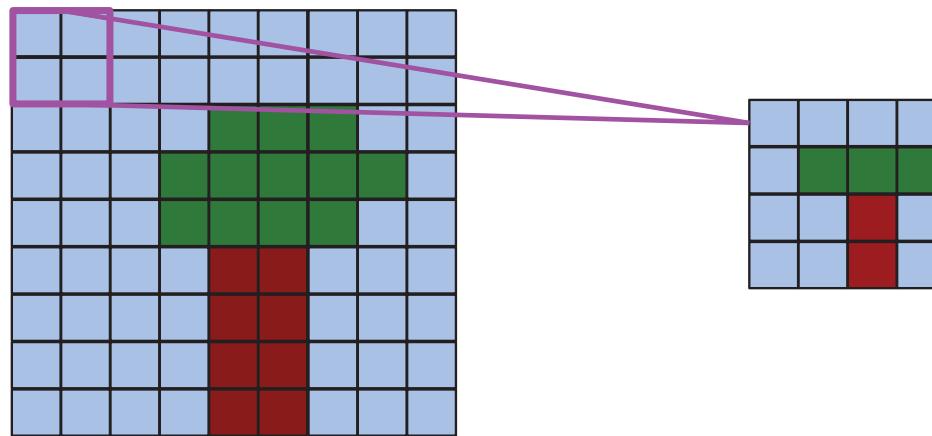
Обединяване



Обединяване: изчисляват се усредните или максималните стойности в малки групи прозорци с цел намаляване на разделителната способност (намалят се бройката пиксели чрез обединението им в един представителен), в примера 4 пиксела се обединяват в един



Обединяване

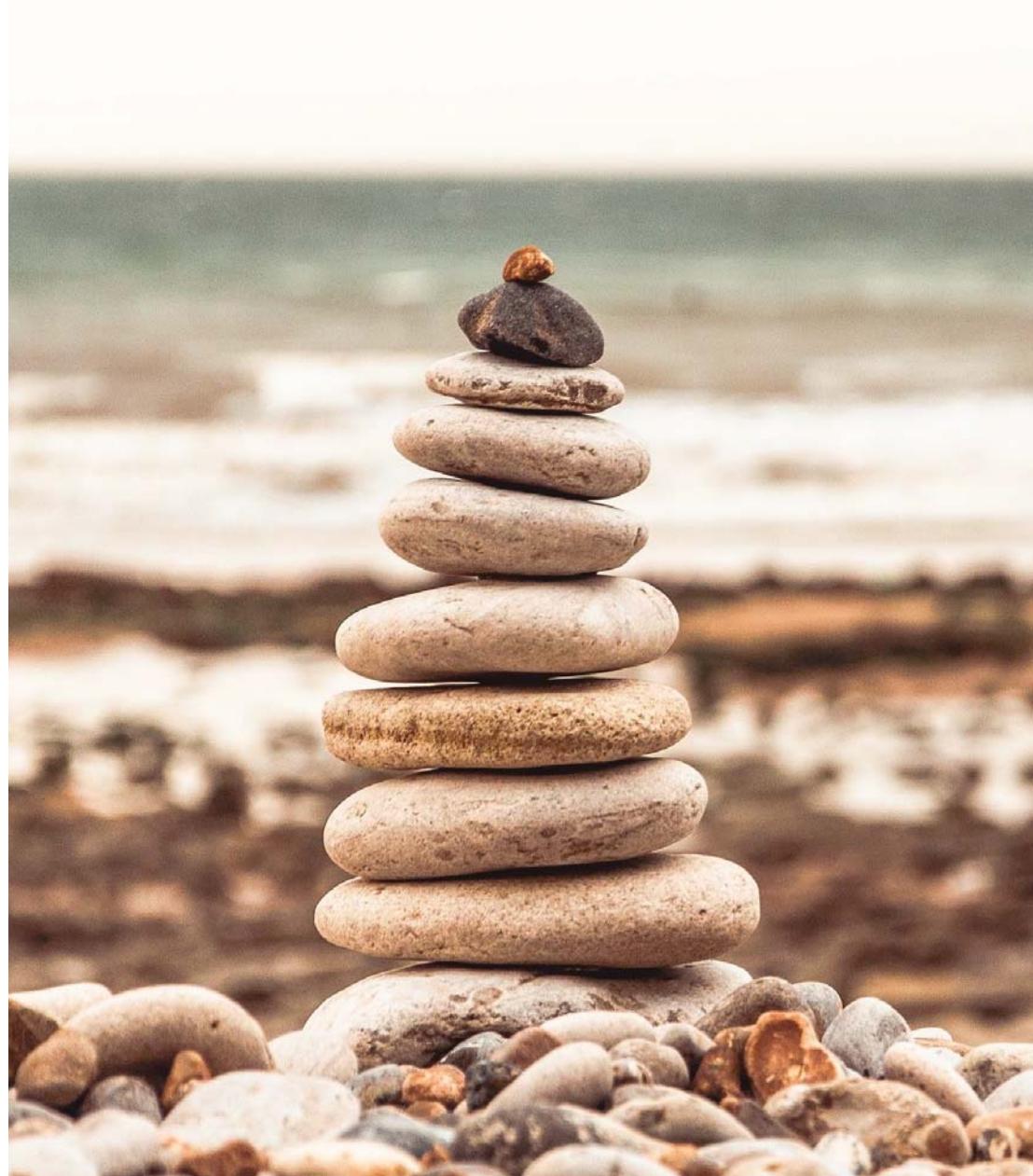


Обединяване: изчисляват се усредните или максималните стойности в малки групи прозорци с цел намаляване на разделителната способност (намалят се бройката пиксели чрез обединението им в един представителен), в примера 4 пиксела се обединяват в един

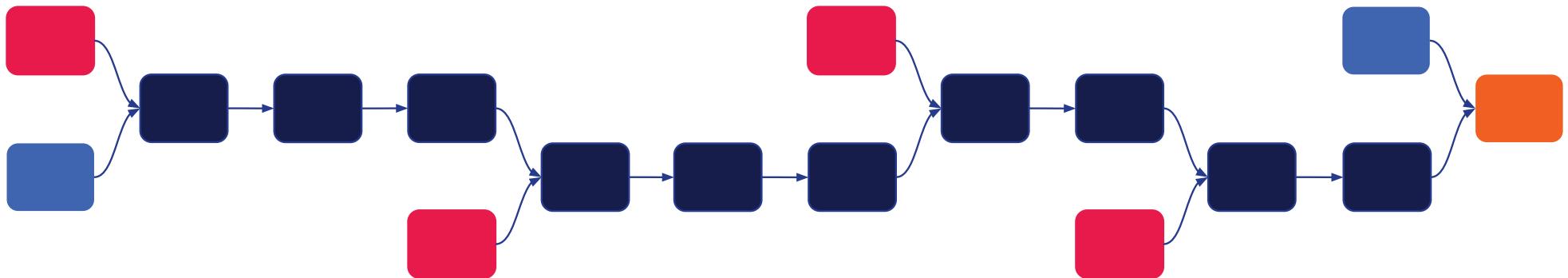


Главната цел на
Конволюционните невронни
мрежи (CNN) е пакетиране на
изграждащи блокове

- CNNs или “convnets”
- До или и над стотици слоеве
- Реализация на алтернативни
конволюции и обединявания за
създаване на определени
йерархични структури



За сравнение: класическите Невронни мрежи представени като граф на изчисленията



вход

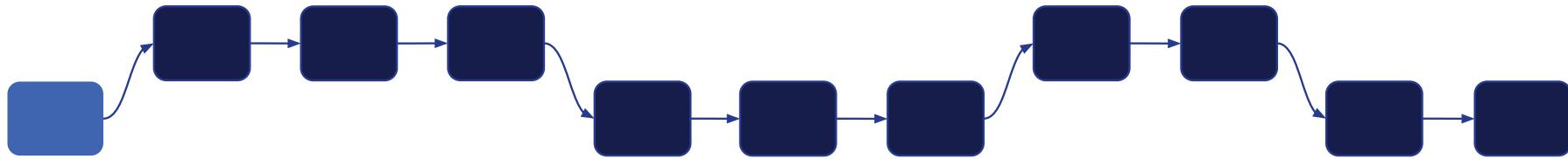
изчисления

загуби

параметри



Опростена диаграма: имплицитни параметри и загуби



вход



изчисления



Опростена диаграма: имплицитни параметри и загуби



вход



изчисления



Изчислителни изграждащи блокове на конволюционните НМ



вход



нелинейност



пълна свързаност



конволюция



обединяване



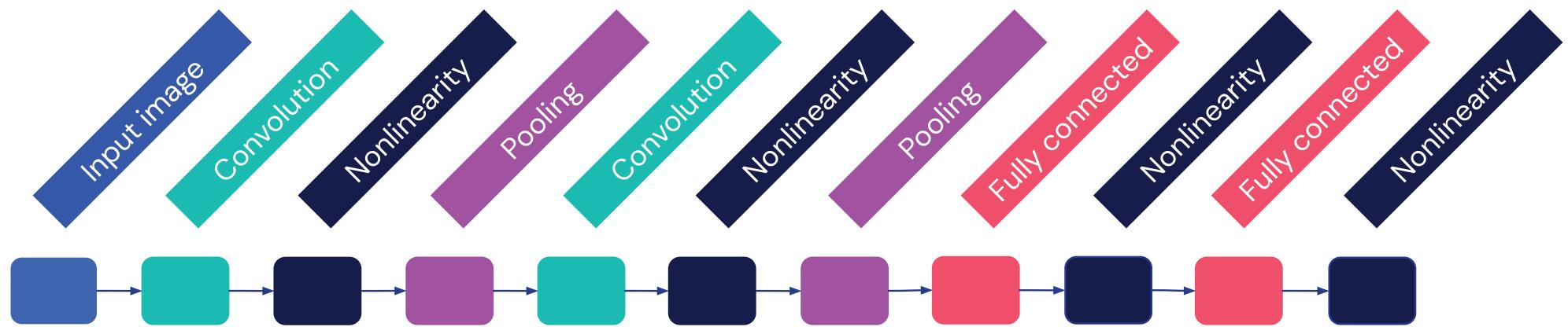
LeNet-5 (1998)



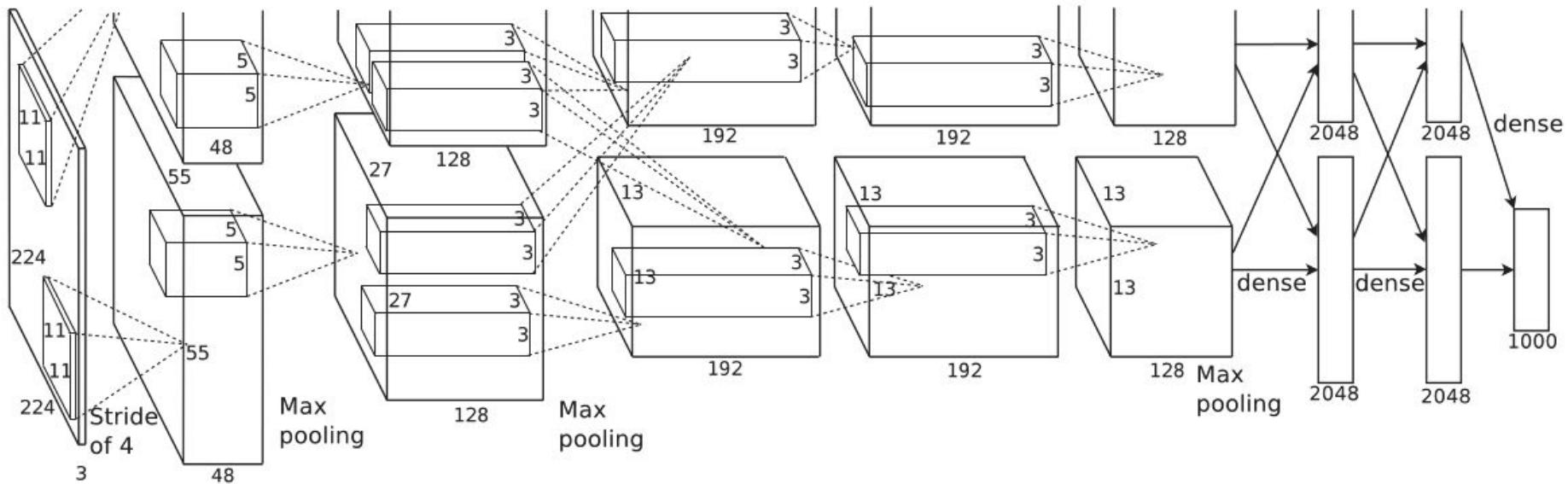
Архитектура на **LeNet-5**, convnet
изградена за разпознаване на
ръкопини числа



LeNet-5 (1998)



AlexNet (2012)

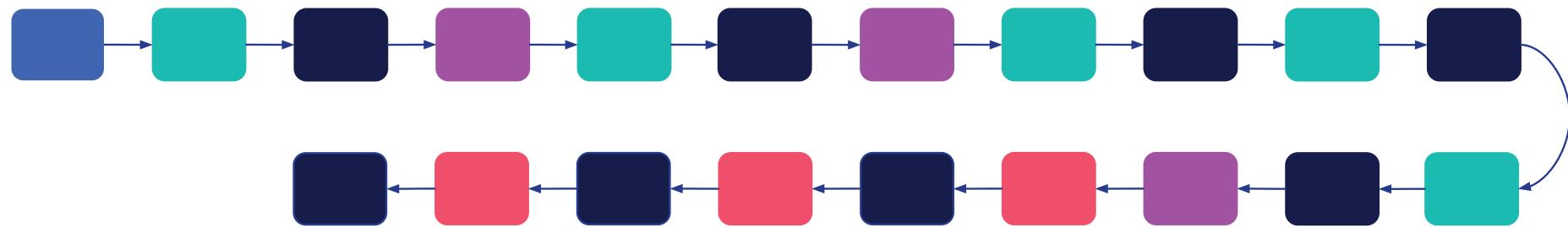
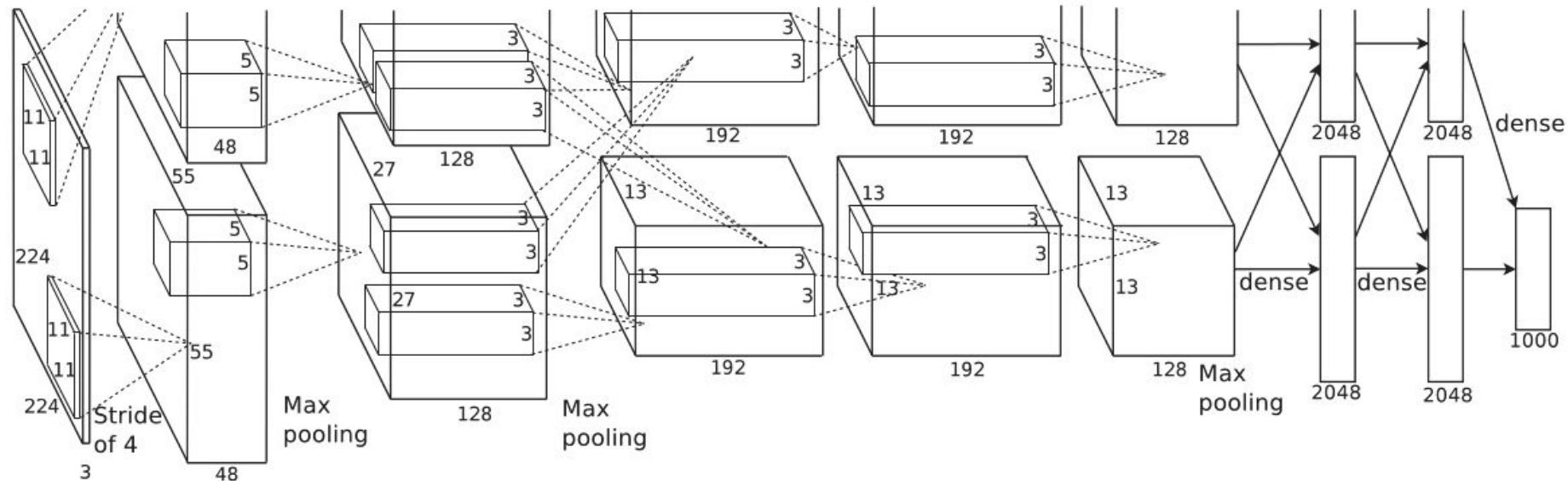


Архитектура: 8 слоя, Ректифициран линеен блок (Rectified Linear Unit - ReLU), отпадания (dropout), намаляване на теглата с въвеждане на умножение на сумите на квадратите с по-малко число (weight decay)

Инфраструктура: голяма база данни с обучение траещо една седмица при реализация на изчисленията в две видеокарти (2 GPUs)

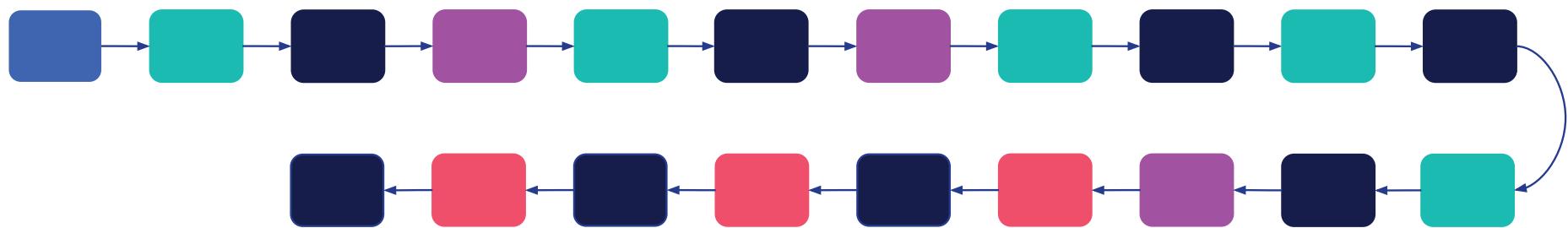


AlexNet (2012)



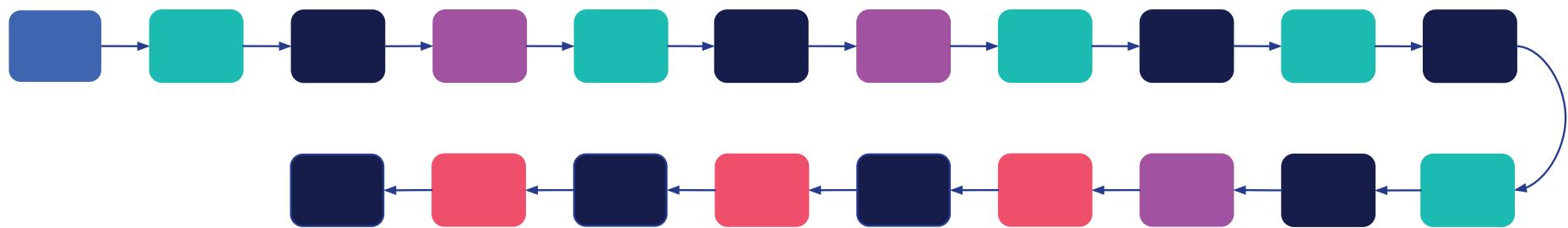
AlexNet (2012)

Входно изображение:
→ 224×224×3

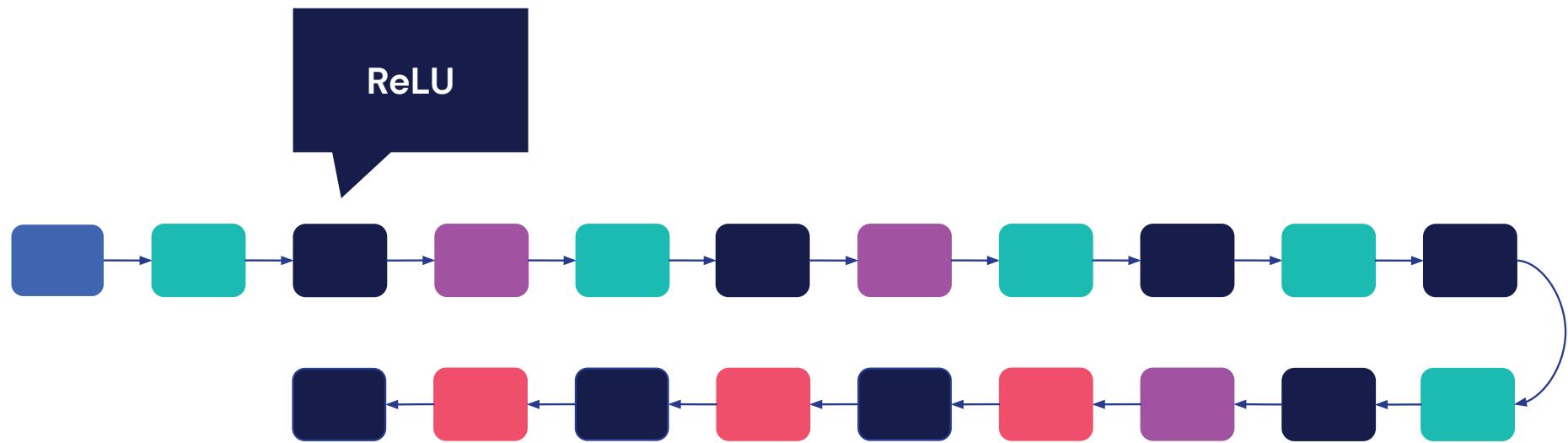


AlexNet (2012)

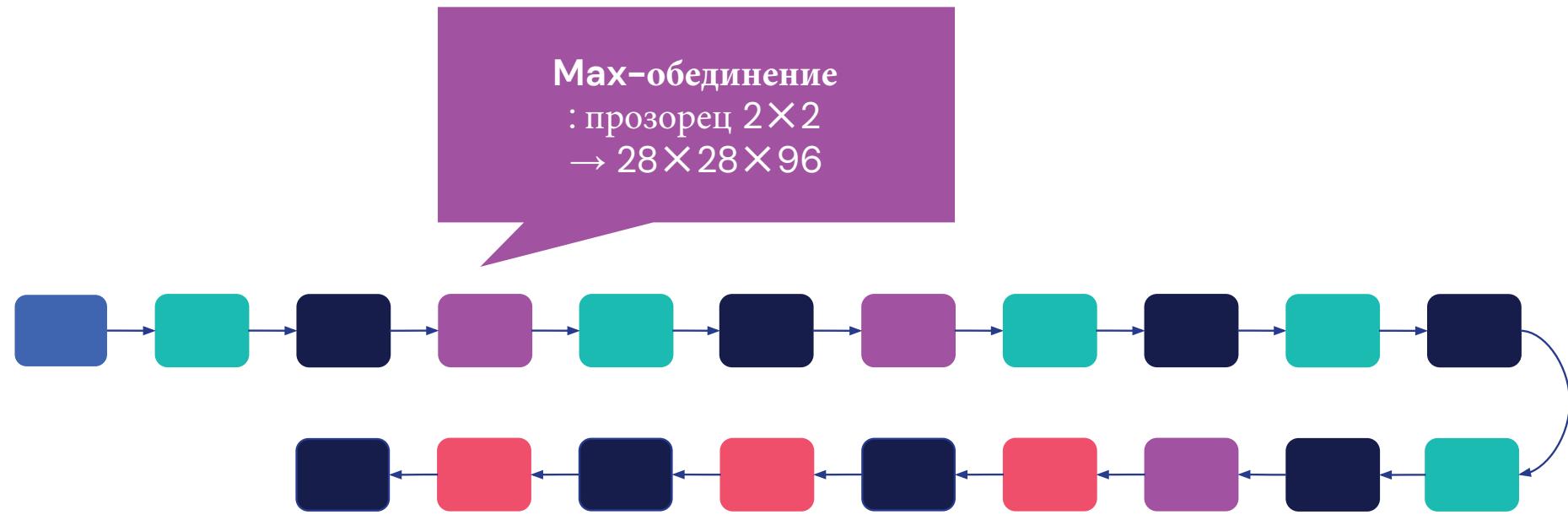
Слой 1 конволюция:
ядра 11×11 , 96 канала, крачка 4
 $\rightarrow 56 \times 56 \times 96$



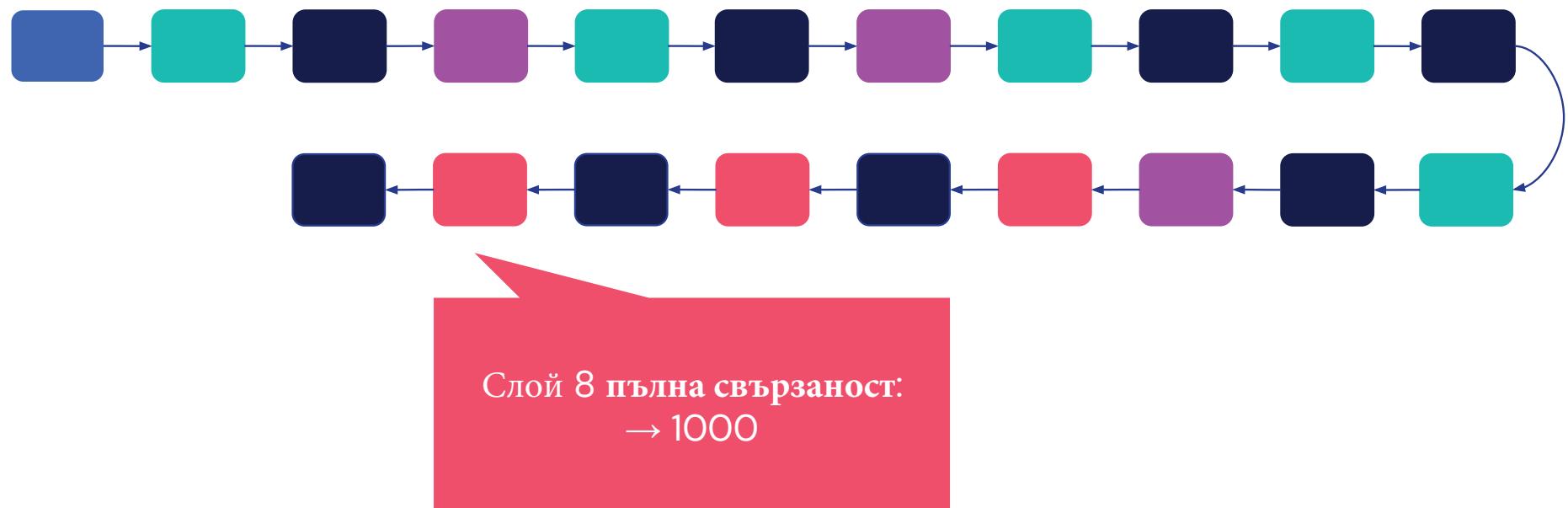
AlexNet (2012)



AlexNet (2012)



AlexNet (2012)



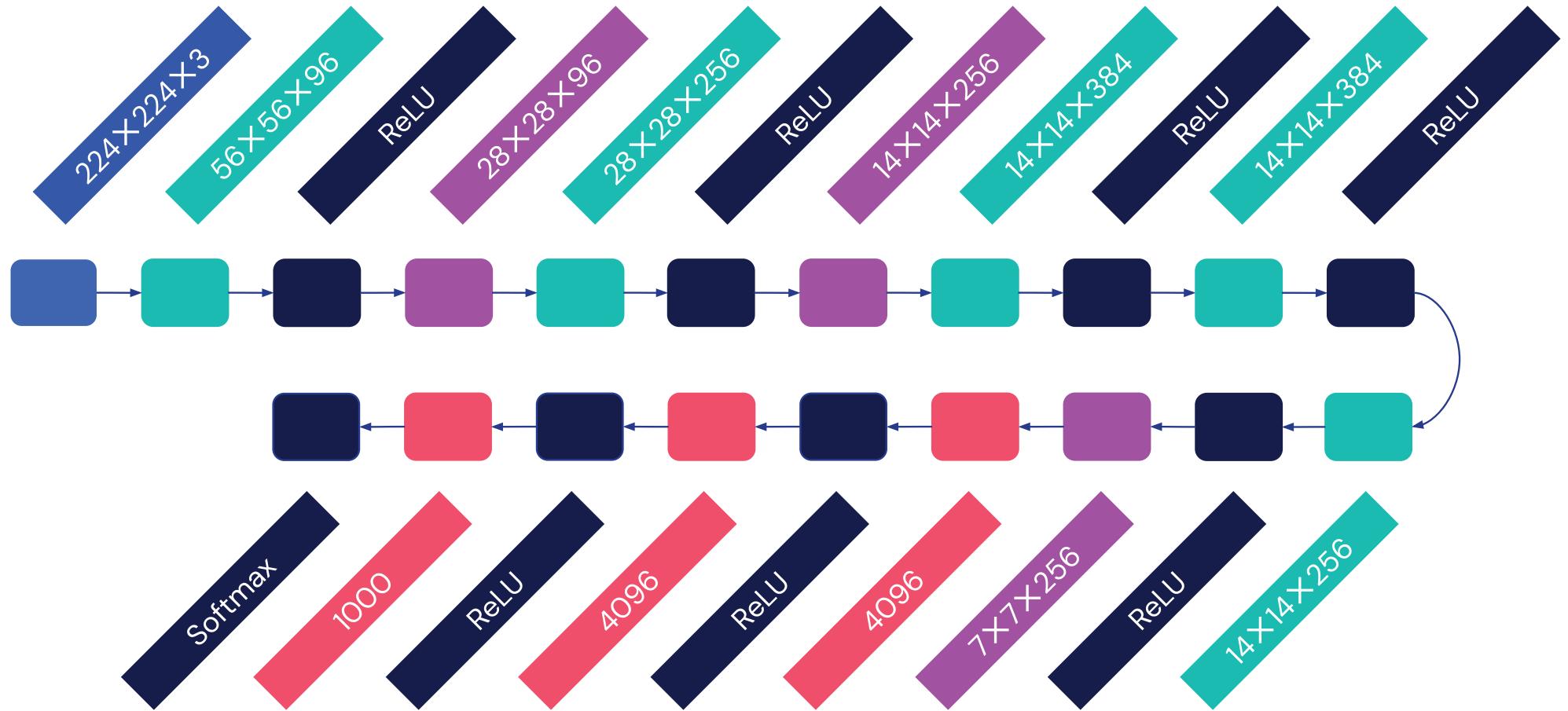
AlexNet (2012)



Функцията softmax, известна още и като softargmax или нормализирана експоненциална функция е обобщение на логистичната функция за множество измерения.

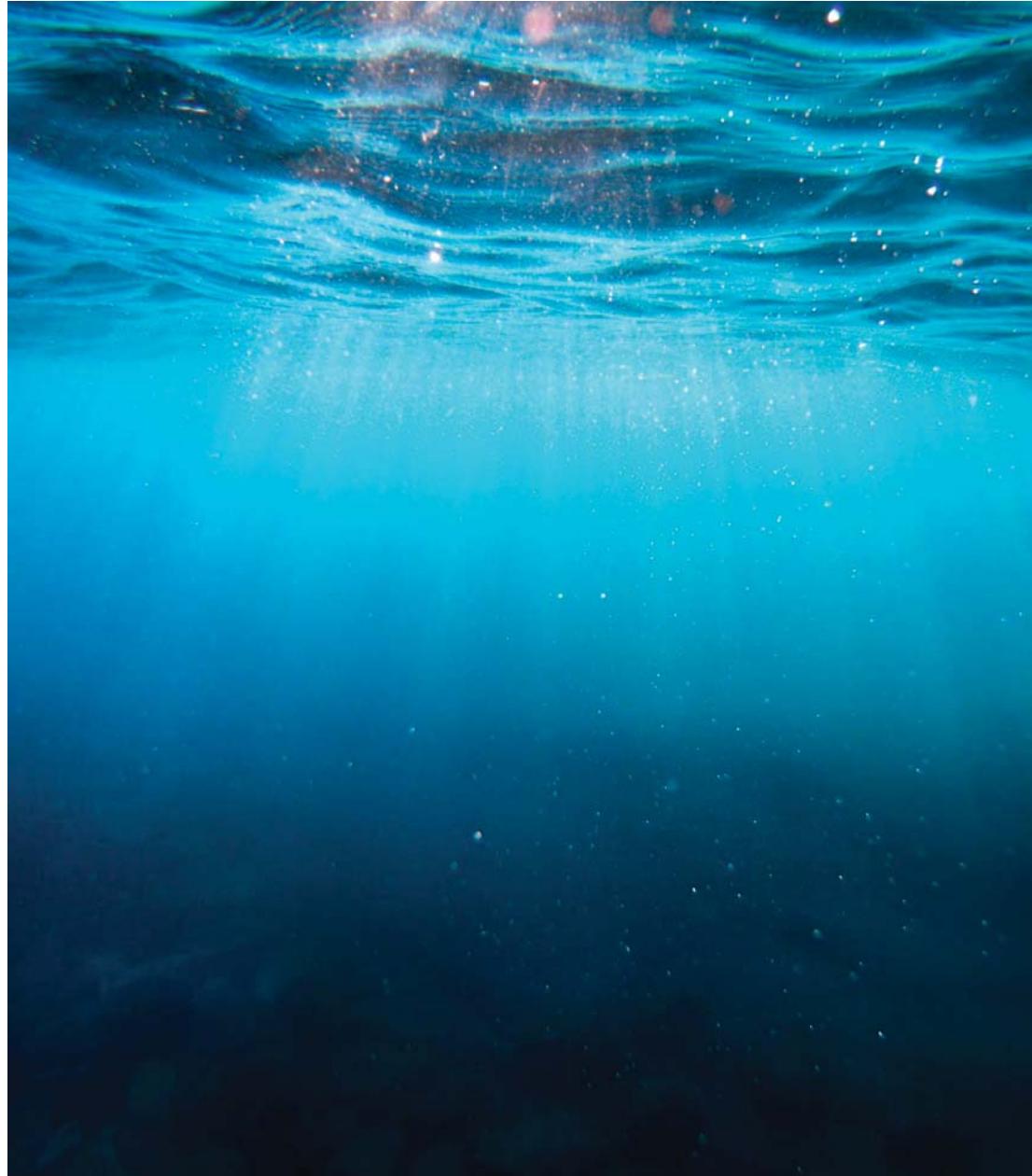


AlexNet (2012)

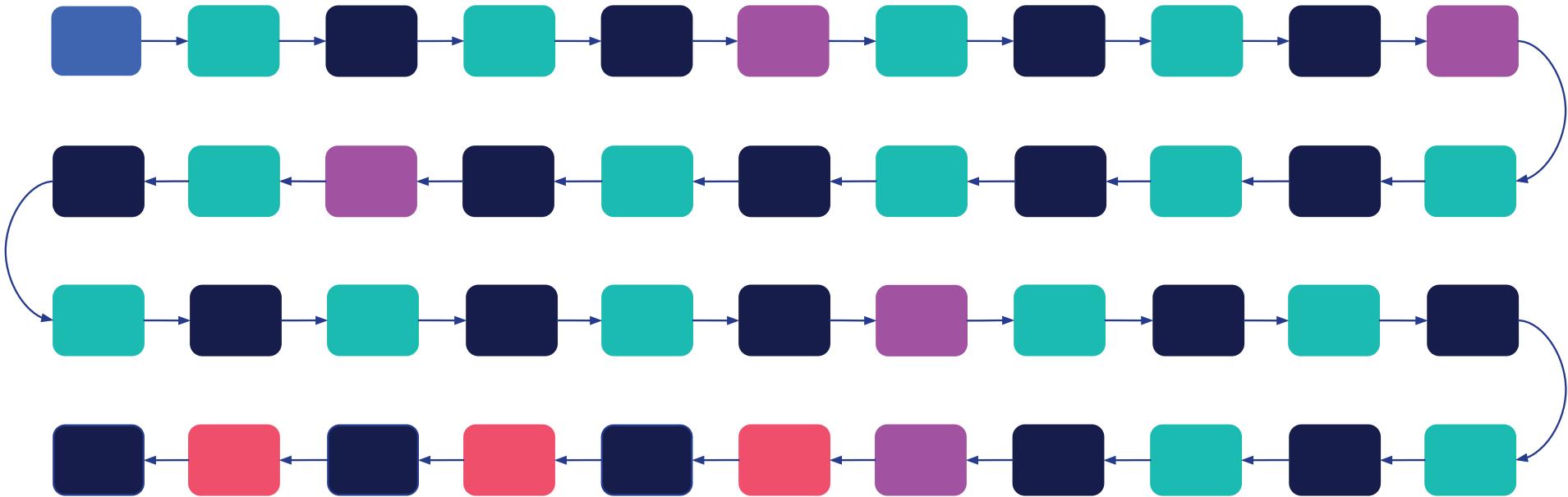


По-голямата дълбочина води до по-добри резултати

- Всеки слой по същество е линеен класификатор
- Повече слоеве – по-голяма неллинейност
- Каква е границата за краен брой слоеве на convnet конволовационните НМ?



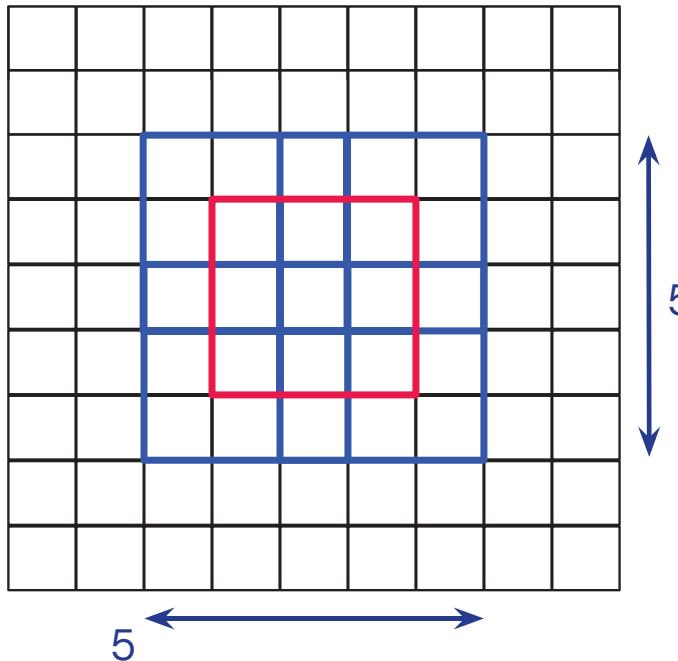
VGGNet (2014): изграждане на много дълбоки convnets



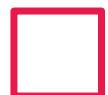
Групиране на множество конволюционни слоеве преди фазата на обединяването. Използване на 'еднакви' конволюции с цел избягване намаляване на разделителната способност.



VGGNet (2014): стековане на 3×3 ядра



1-ви 3×3 конволюционен слой



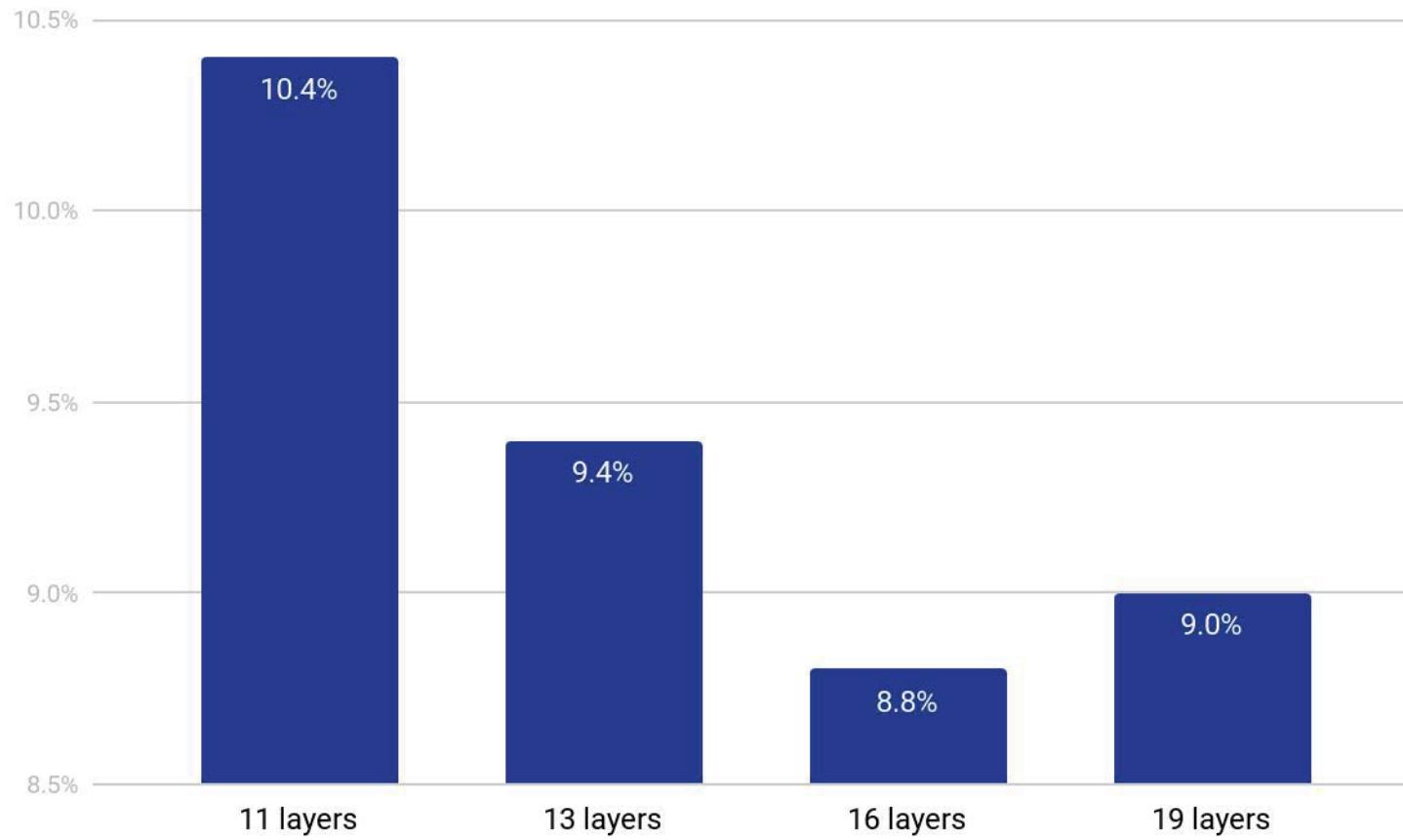
2-ри 3×3 конволюционен слой

Архитектура: до 19 слоя, само 3×3 ядра, "еднакви" конволюции

Инфраструктура: обучение траещо 2-3 седмици на 4 GPU (изчислителна паралелност)



VGGNet (2014): грешката не се подобрява след 16 слоя



Предизвикателства пред използването на дълбоки невронни мрежи

- Повишена изчислителна сложност
- Трудна оптимизация

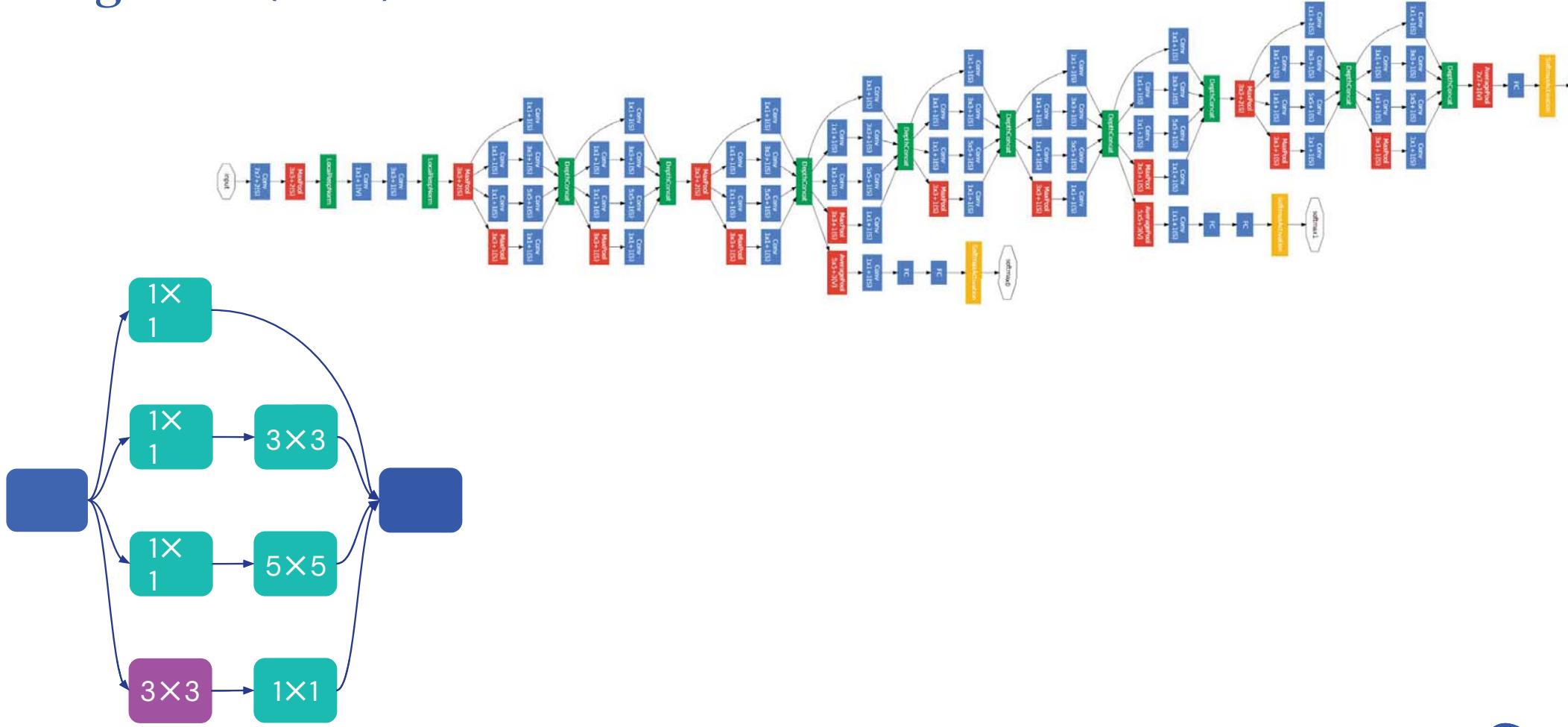


Как може да се подобри оптимизацията

- С внимателна инициализация
- Сложни оптимизатори
- Нормализационни слоеве
- Дизайн на архитектурата



GoogLeNet (2014)



Пакетно нормализиране

Input: Values of x over a mini-batch: $\mathcal{B} = \{x_1 \dots m\}$;

Parameters to be learned: γ, β

Output: $\{y_i = \text{BN}_{\gamma, \beta}(x_i)\}$

$$\mu_{\mathcal{B}} \leftarrow \frac{1}{m} \sum_{i=1}^m x_i \quad // \text{mini-batch mean}$$

$$\sigma_{\mathcal{B}}^2 \leftarrow \frac{1}{m} \sum_{i=1}^m (x_i - \mu_{\mathcal{B}})^2 \quad // \text{mini-batch variance}$$

$$\hat{x}_i \leftarrow \frac{x_i - \mu_{\mathcal{B}}}{\sqrt{\sigma_{\mathcal{B}}^2 + \epsilon}} \quad // \text{normalize}$$

$$y_i \leftarrow \gamma \hat{x}_i + \beta \equiv \text{BN}_{\gamma, \beta}(x_i) \quad // \text{scale and shift}$$

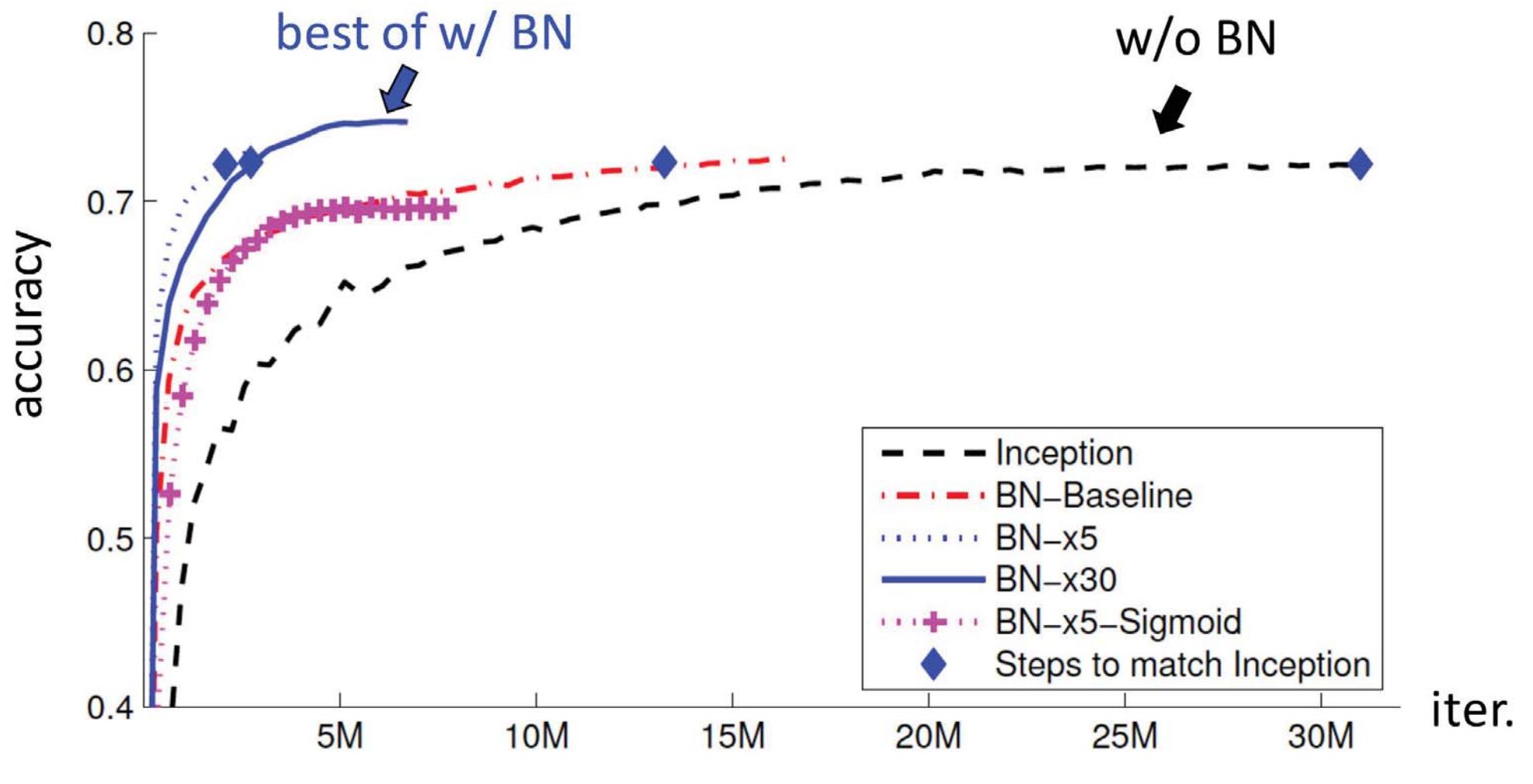
Намалява се чувствителността по отношение

на **инициализацията**

Въвежда се стохастичност и действа като **регулатор**



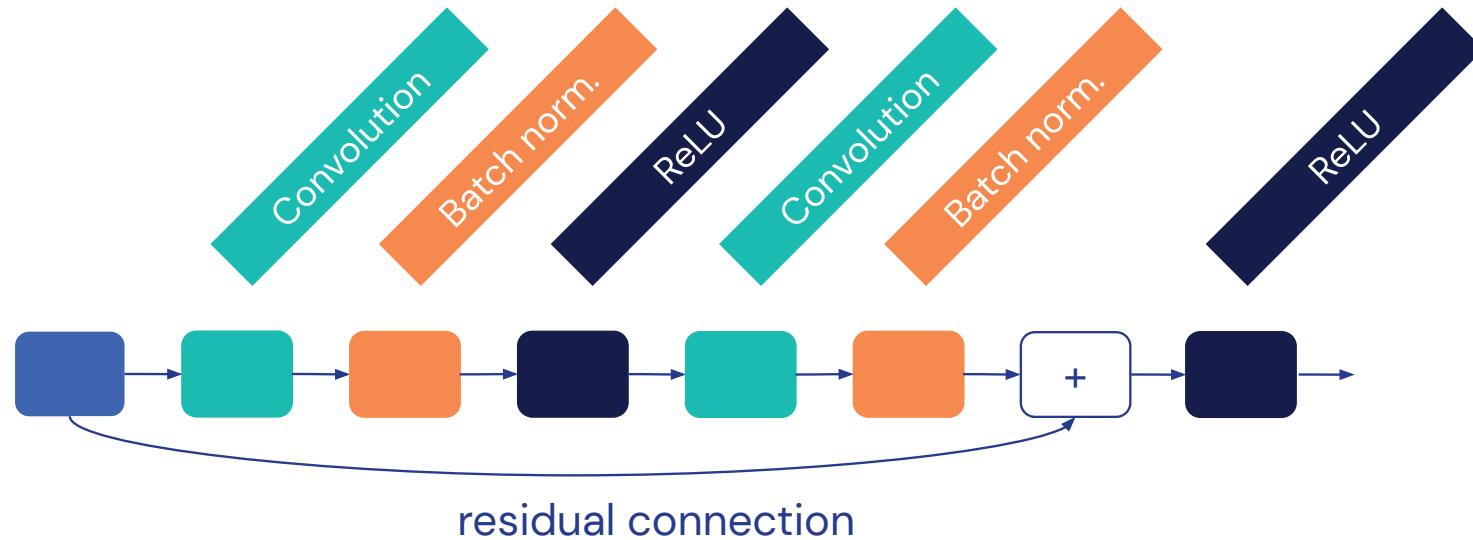
Пакетно нормализиране



iter.



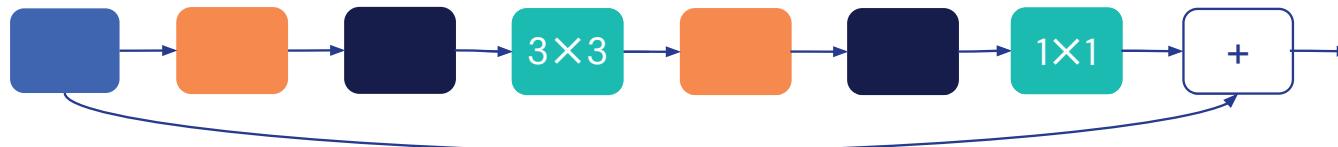
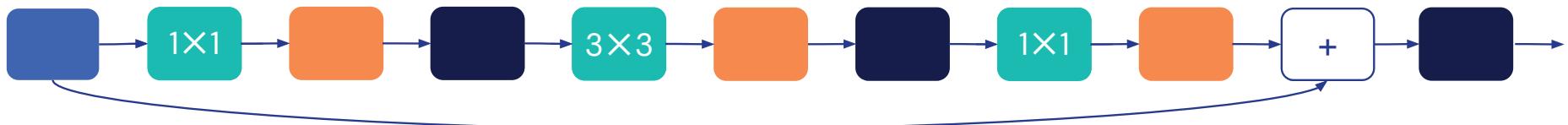
ResNet (2015): резидуумни връзки



С резидуумни връзки се улеснява обучението на дълбоки НМ



ResNet (2015): различни подвидове



ResNet V2 (отдолу) избягва всякакви
нелинейности в резидиумната част

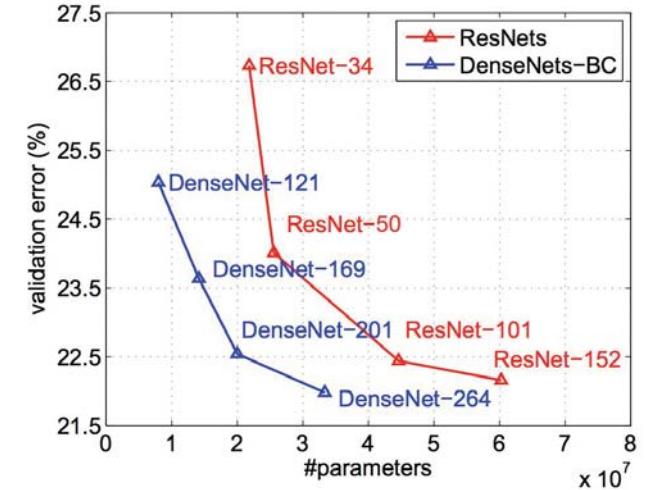
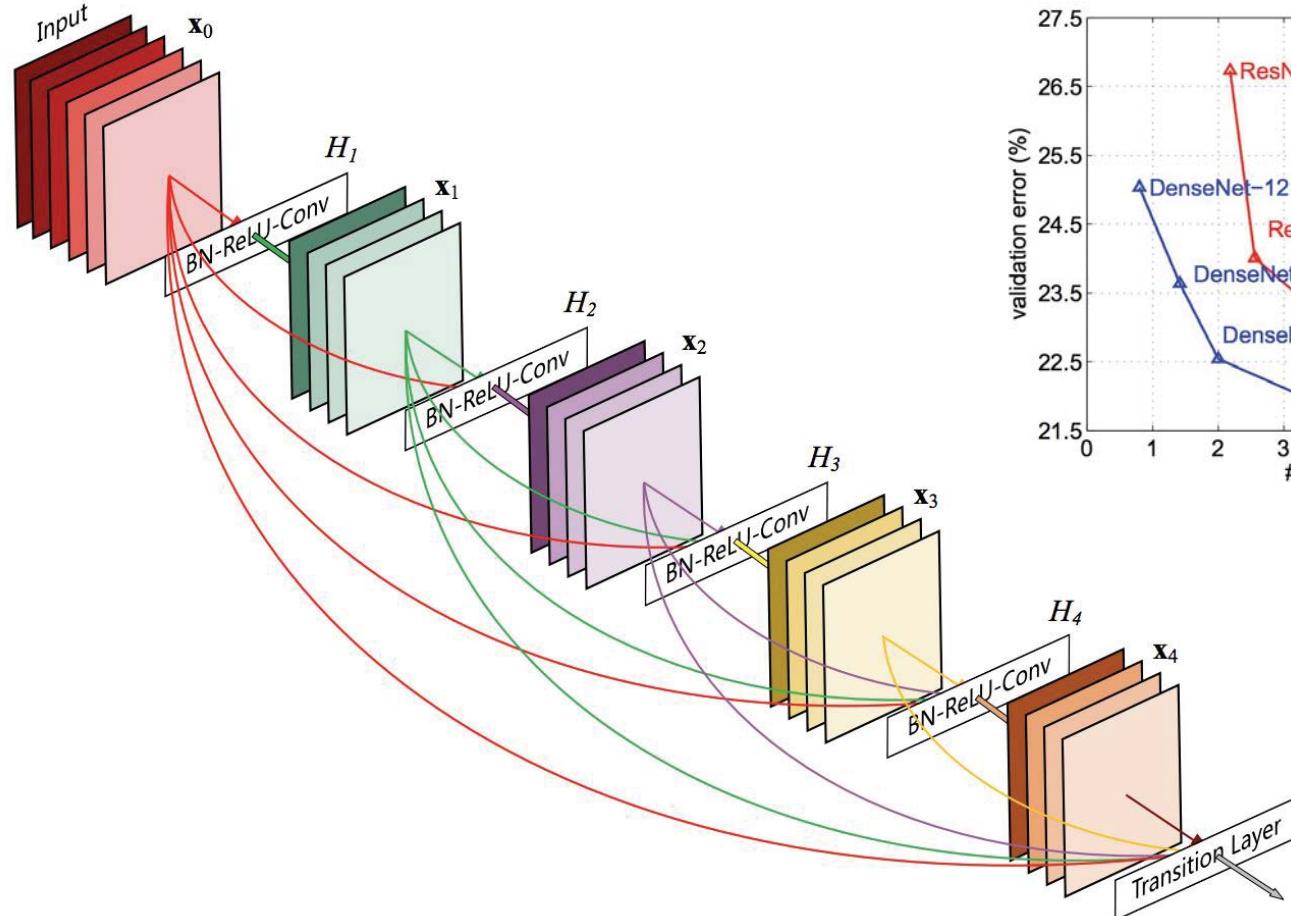


ResNet (2015): до 152 слоя

layer name	output size	18-layer	34-layer	50-layer	101-layer	152-layer
conv1	112×112			7×7, 64, stride 2		
conv2_x	56×56	$\begin{bmatrix} 3\times3, 64 \\ 3\times3, 64 \end{bmatrix} \times 2$	$\begin{bmatrix} 3\times3, 64 \\ 3\times3, 64 \end{bmatrix} \times 3$	$\begin{bmatrix} 1\times1, 64 \\ 3\times3, 64 \\ 1\times1, 256 \end{bmatrix} \times 3$	$\begin{bmatrix} 1\times1, 64 \\ 3\times3, 64 \\ 1\times1, 256 \end{bmatrix} \times 3$	$\begin{bmatrix} 1\times1, 64 \\ 3\times3, 64 \\ 1\times1, 256 \end{bmatrix} \times 3$
conv3_x	28×28	$\begin{bmatrix} 3\times3, 128 \\ 3\times3, 128 \end{bmatrix} \times 2$	$\begin{bmatrix} 3\times3, 128 \\ 3\times3, 128 \end{bmatrix} \times 4$	$\begin{bmatrix} 1\times1, 128 \\ 3\times3, 128 \\ 1\times1, 512 \end{bmatrix} \times 4$	$\begin{bmatrix} 1\times1, 128 \\ 3\times3, 128 \\ 1\times1, 512 \end{bmatrix} \times 4$	$\begin{bmatrix} 1\times1, 128 \\ 3\times3, 128 \\ 1\times1, 512 \end{bmatrix} \times 8$
conv4_x	14×14	$\begin{bmatrix} 3\times3, 256 \\ 3\times3, 256 \end{bmatrix} \times 2$	$\begin{bmatrix} 3\times3, 256 \\ 3\times3, 256 \end{bmatrix} \times 6$	$\begin{bmatrix} 1\times1, 256 \\ 3\times3, 256 \\ 1\times1, 1024 \end{bmatrix} \times 6$	$\begin{bmatrix} 1\times1, 256 \\ 3\times3, 256 \\ 1\times1, 1024 \end{bmatrix} \times 23$	$\begin{bmatrix} 1\times1, 256 \\ 3\times3, 256 \\ 1\times1, 1024 \end{bmatrix} \times 36$
conv5_x	7×7	$\begin{bmatrix} 3\times3, 512 \\ 3\times3, 512 \end{bmatrix} \times 2$	$\begin{bmatrix} 3\times3, 512 \\ 3\times3, 512 \end{bmatrix} \times 3$	$\begin{bmatrix} 1\times1, 512 \\ 3\times3, 512 \\ 1\times1, 2048 \end{bmatrix} \times 3$	$\begin{bmatrix} 1\times1, 512 \\ 3\times3, 512 \\ 1\times1, 2048 \end{bmatrix} \times 3$	$\begin{bmatrix} 1\times1, 512 \\ 3\times3, 512 \\ 1\times1, 2048 \end{bmatrix} \times 3$
	1×1	average pool, 1000-d fc, softmax				
FLOPs		1.8×10^9	3.6×10^9	3.8×10^9	7.6×10^9	11.3×10^9



DenseNet (2016): свързва слоевете с всички предходни слоеве



Squeeze-and-excitation мрежи (2017)

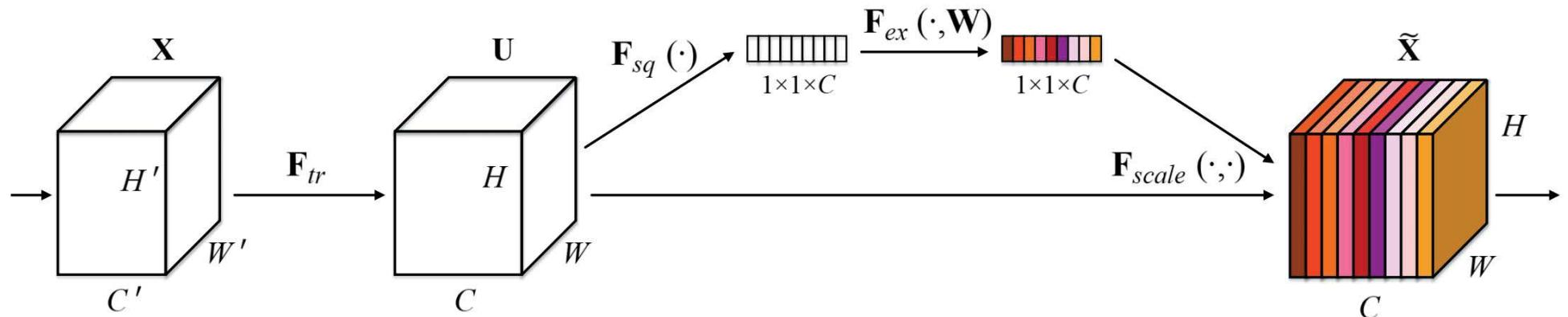


Figure from Hu et al. (2018)

Признаците могат да съдържат глобален контекст



AmoebaNet (2018): Търсене на невронна архитектура

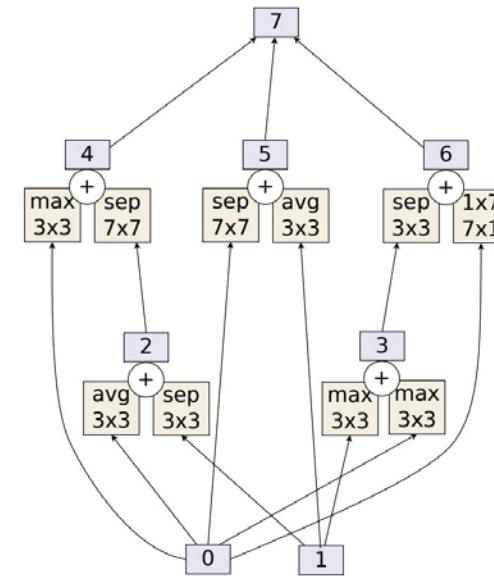
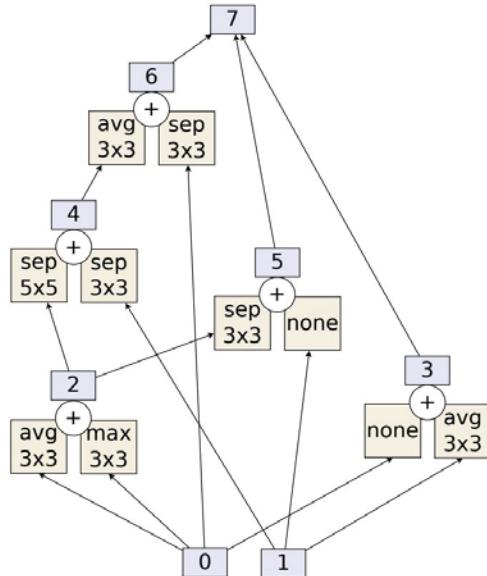
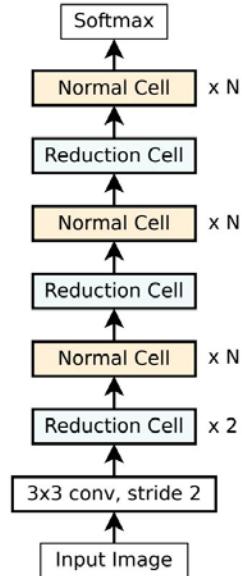


Figure from Real et al. (2019)

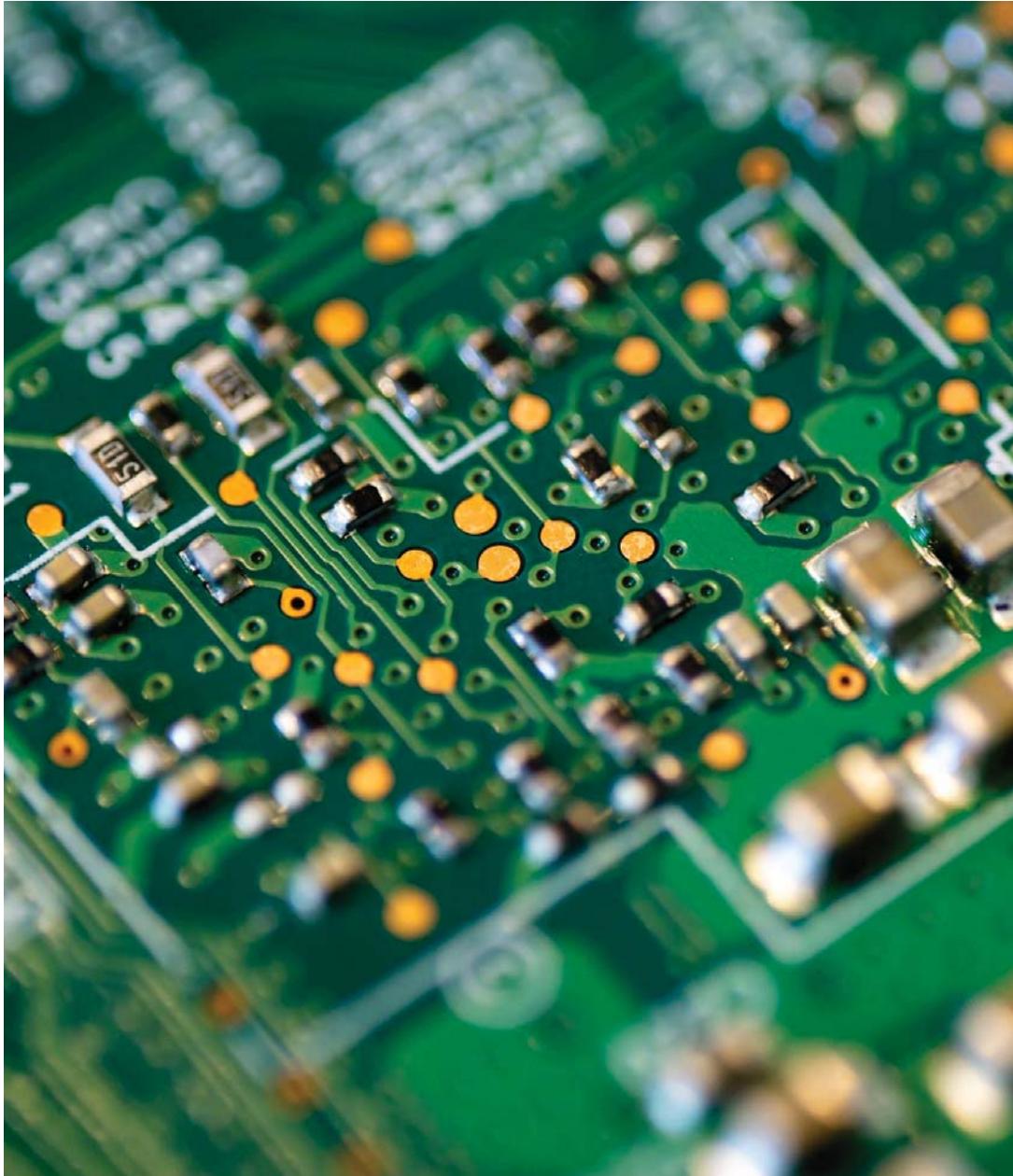
Архитектурата се задава след процес на **еволюция**

Търсене на нециклични графи съставени
от предварително зададени слоеве



Намаляване на сложността

- Конволюции в дълбочина
- Отделими конволюции
- Обърнати 'тапи' (bottleneck)
(MobileNetV2, MNasNet,
EfficientNet)



Подобряване на данните

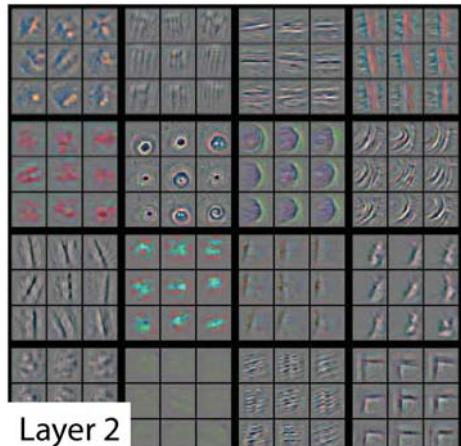


По дизайн, convnets са робастни и устойчиви само за транслации.

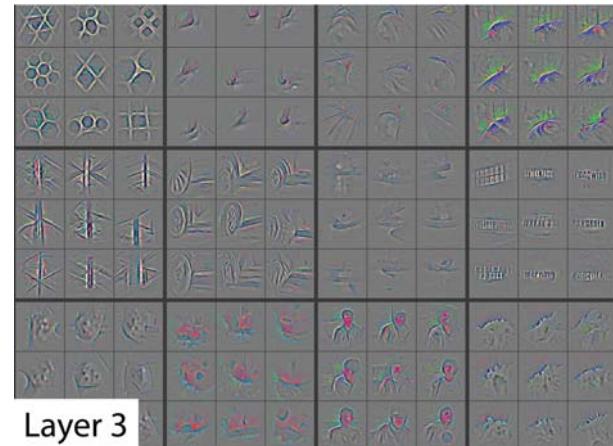
С подобряването на данните convnets стават робастни и устойчиви и за други трансформации като завъртания, различни мащабирания, отрязвания, изкривявания ...



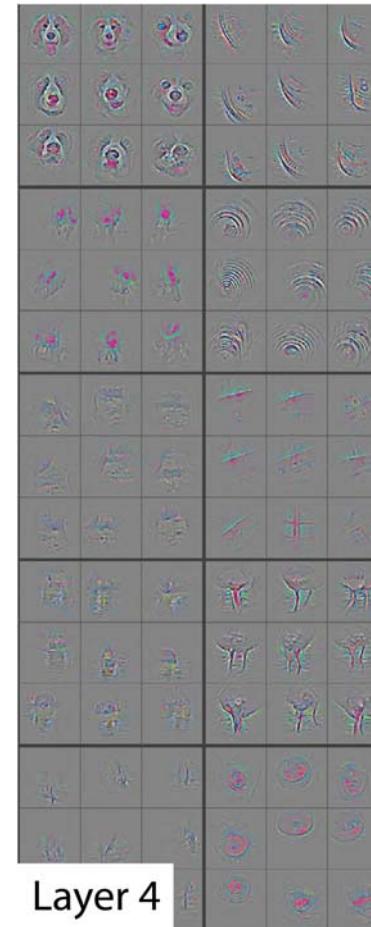
Визуализация на това като една **convnet** научава



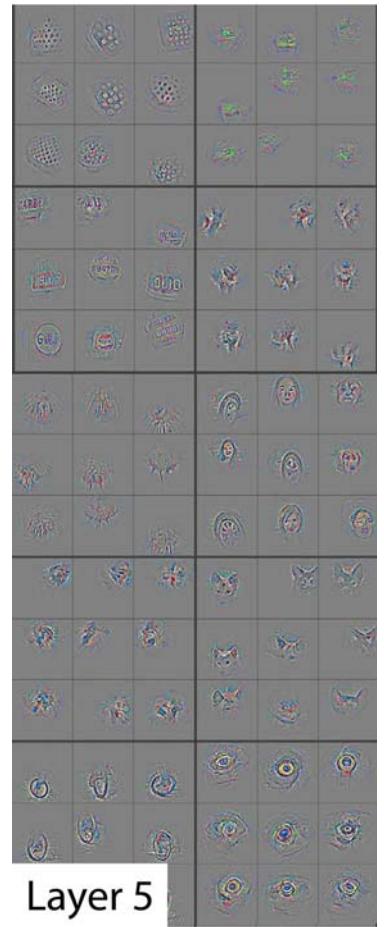
Layer 2



Layer 3



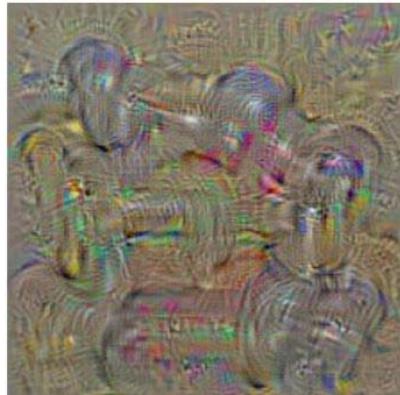
Layer 4



Layer 5



Визуализация на това като една **convnet** научава



dumbbell



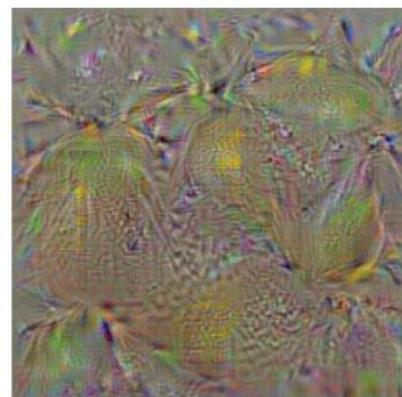
cup



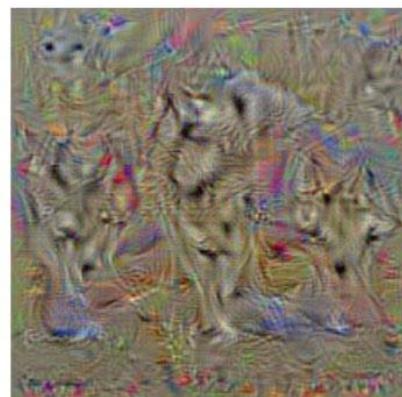
dalmatian



bell pepper



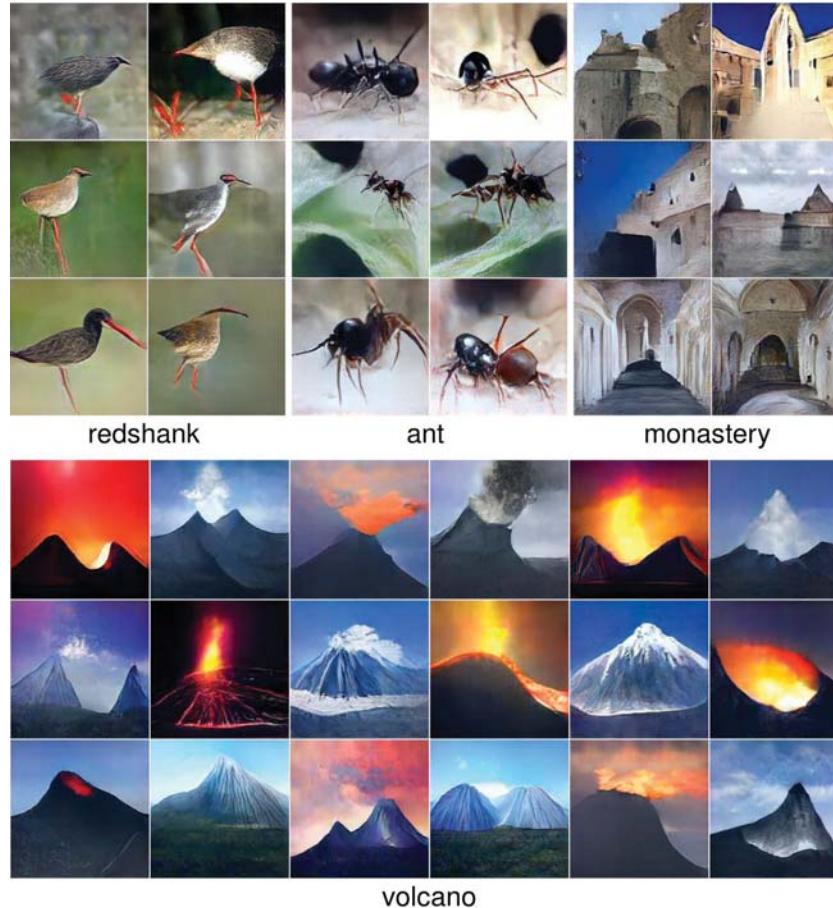
lemon



husky



Визуализация на това като една **convnet** научава



Feature Visualization

How neural networks build up their understanding of images



Feature visualization allows us to see how GoogLeNet^[1], trained on the ImageNet^[2] dataset, builds up its understanding of images over many layers. Visualizations of all channels are available in the [appendix](#).

AUTHORS

Chris Olah
Alexander Mordvintsev
Ludwig Schubert

AFFILIATIONS

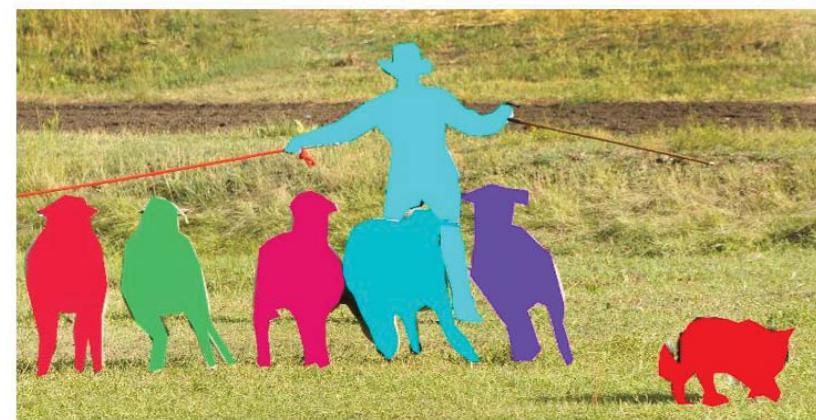
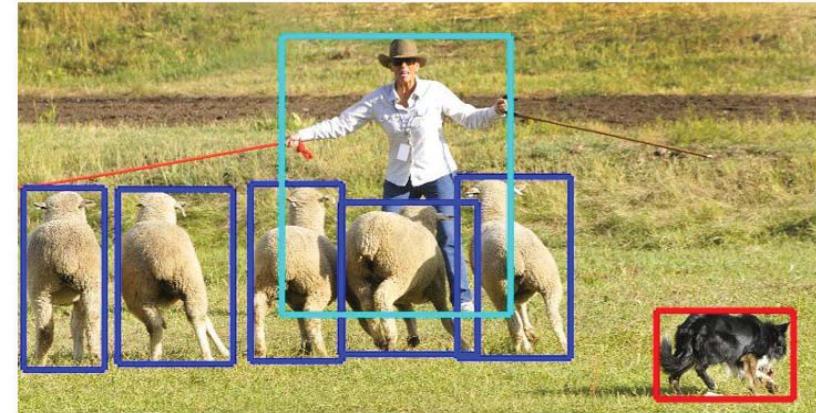
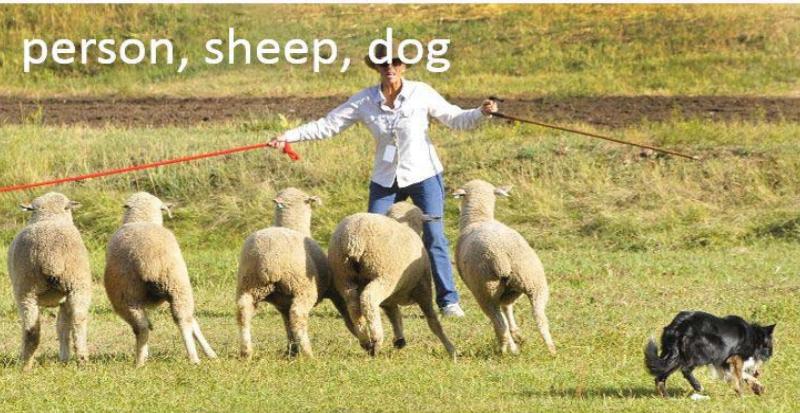
Google Brain Team
Google Research
Google Brain Team

PUBLISHED

Nov. 7, 2017

DOI

10.23915/distill.00007



Изводи и особености на convnets

- Конволюционните невронни мрежи заменят създаваните на ръка информативни признания със създадени на ръка архитектури.
- Предварителното познание за обектите все още не е ненужно: то просто се интегрира и представя в абстракция от по-високо ниво.

