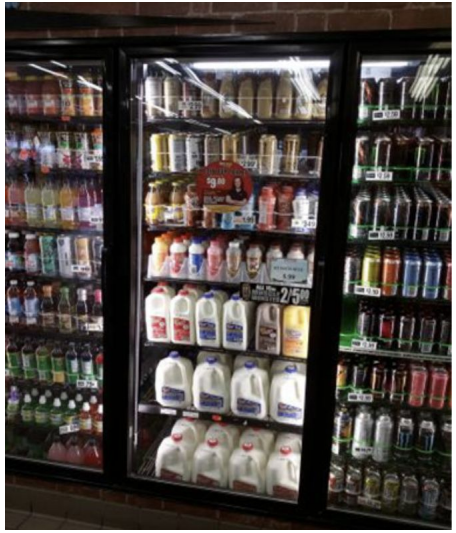
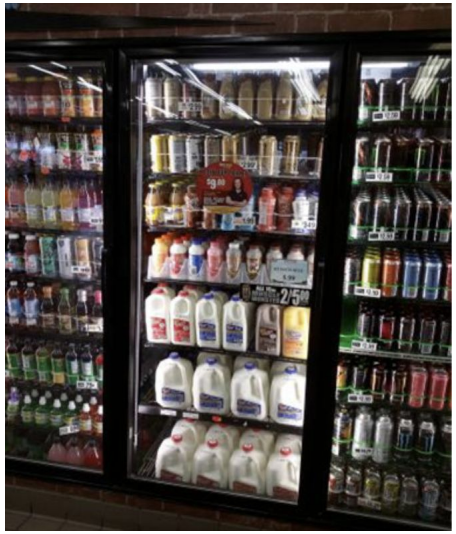
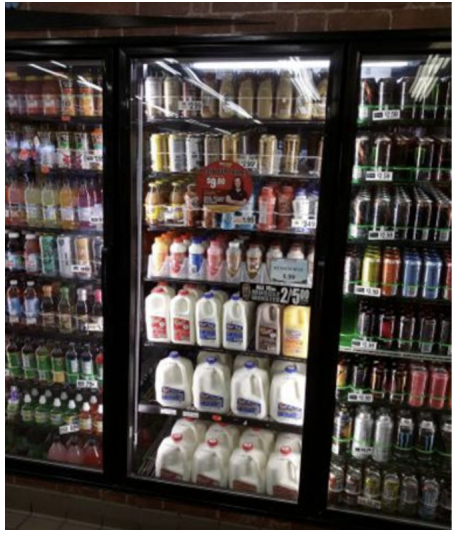
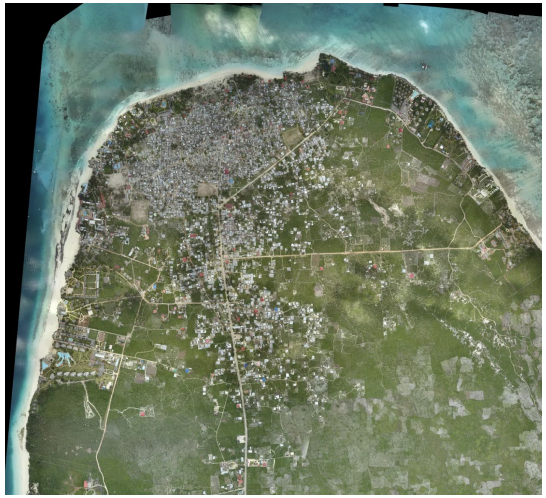
cv and ml

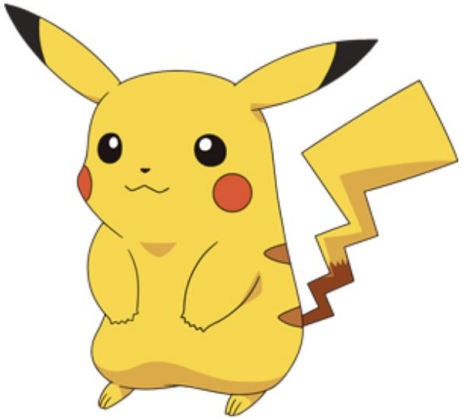
Владимир Глазачев
cv в rosebud.ai

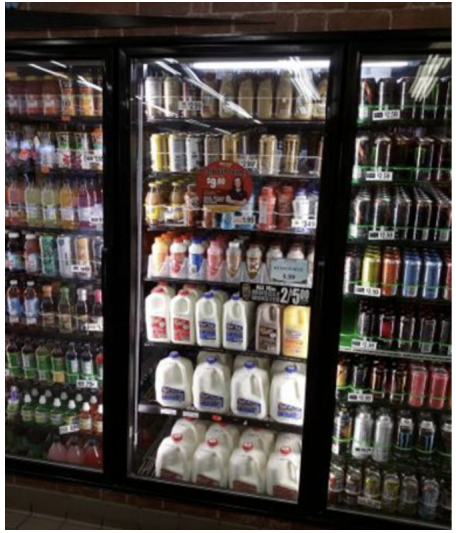


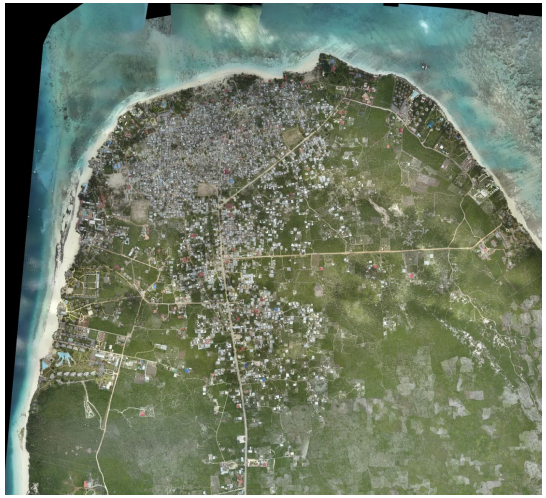


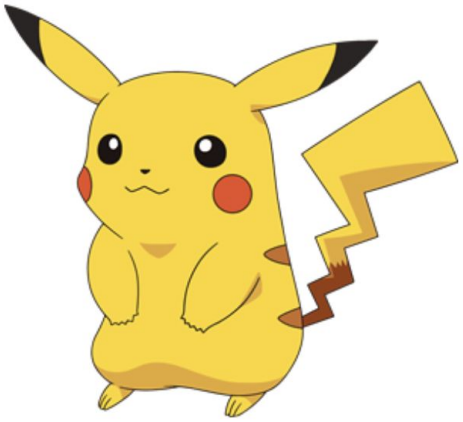


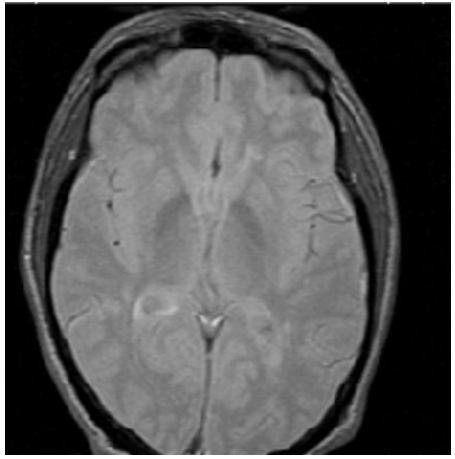


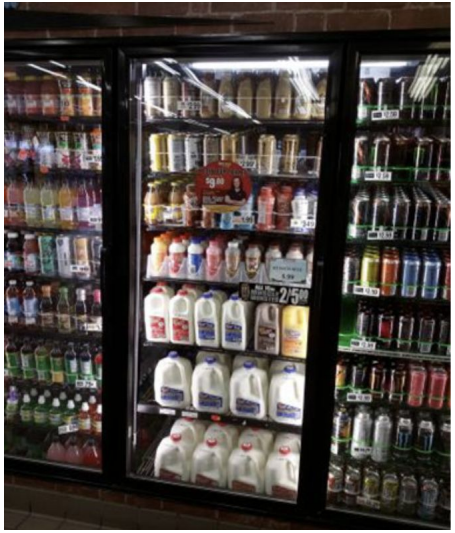


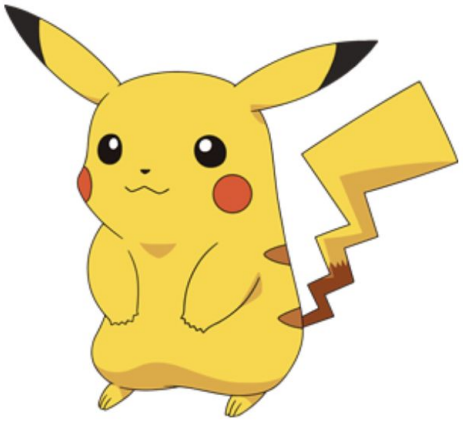


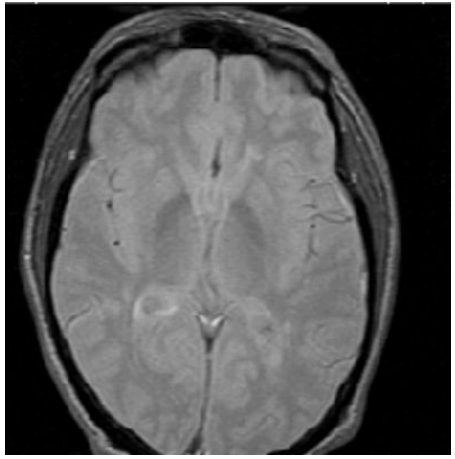


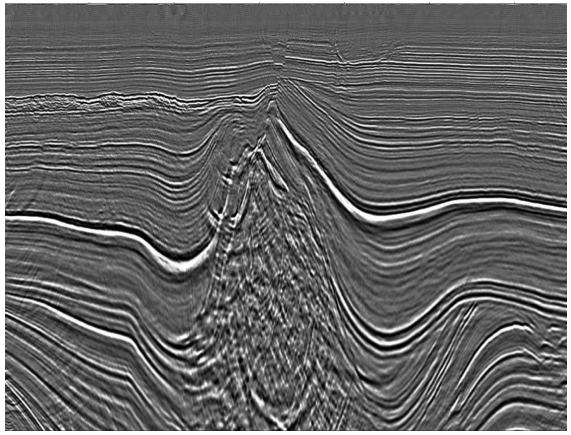


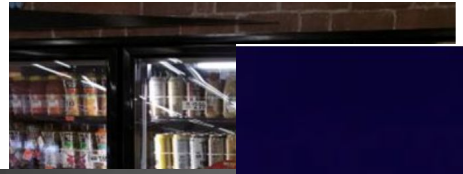


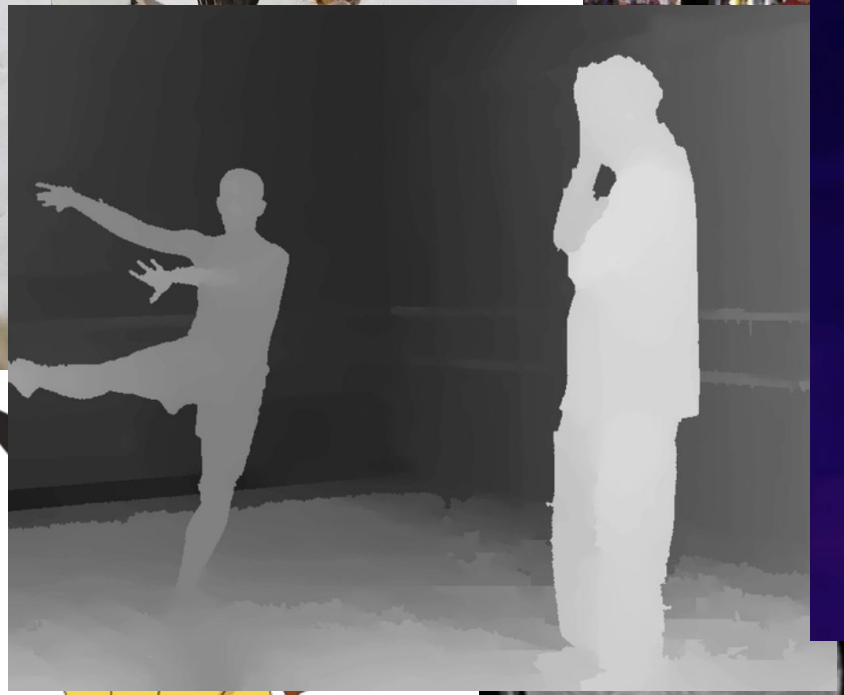


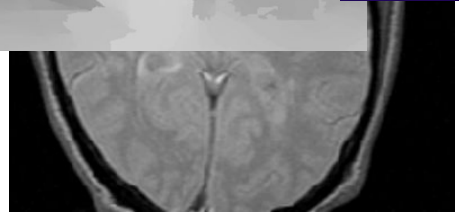


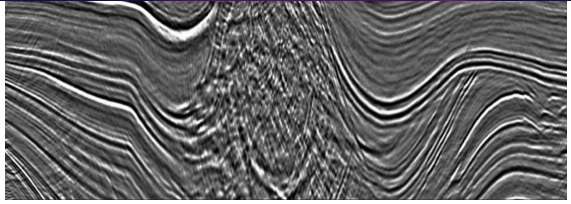




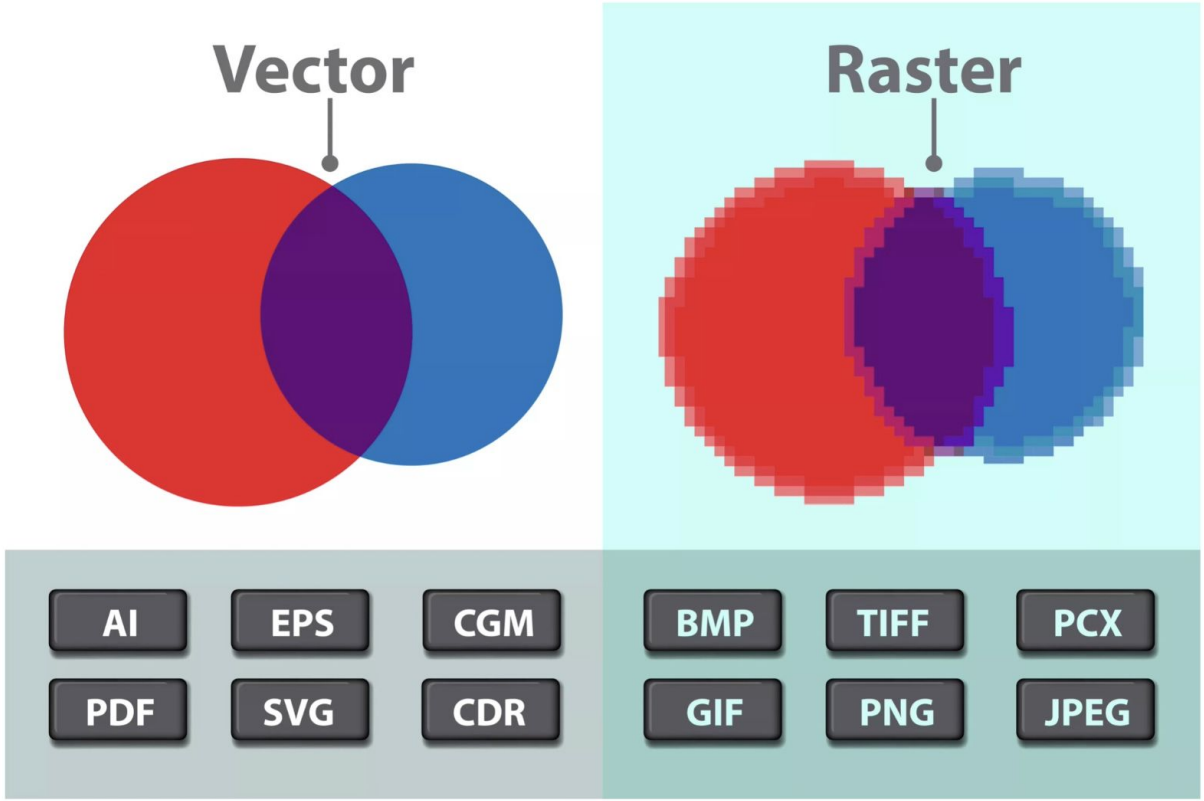


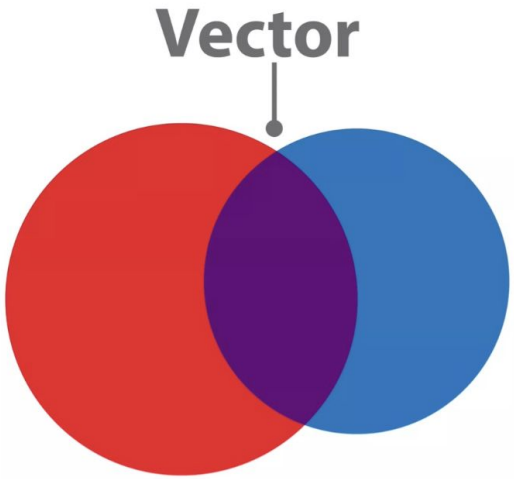


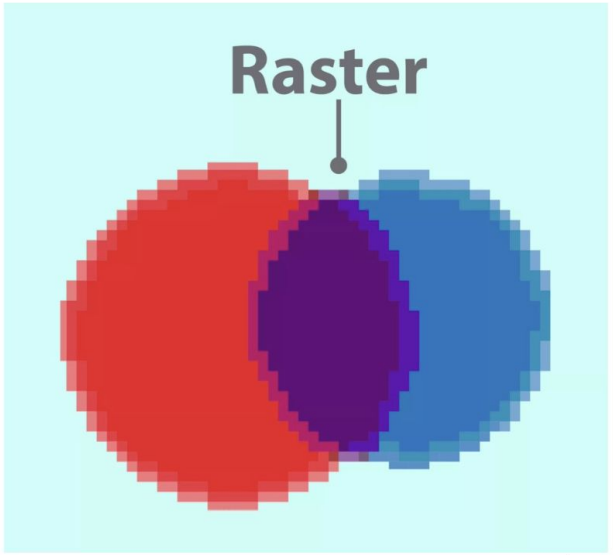




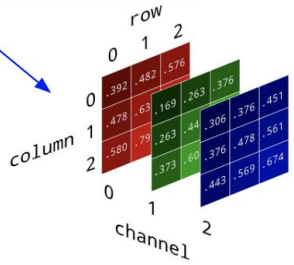
Ну и так далее


Vector
Raster
AI
EPS
CGM
PDF
SVG
CDR
BMP
TIFF
PCX
GIF
PNG
JPEG






```
1  circle of color red with radius r1 at position (x1, y1, z1)
2  circle of color blue with radius r2 at position (x2, y2, z2)
3
```



Что у нас?

Будет:

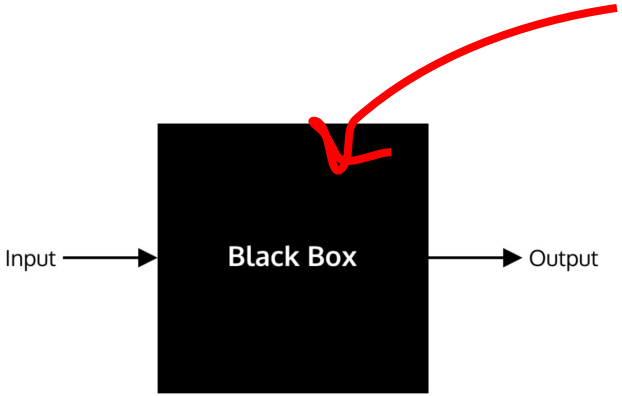
- Растровые картинки - тензор размера height x width x num_channels
- Data Driven - картинки к нам откуда то пришли в разумном количестве
- Хотим делать какой то ml вокруг этого

Не будет:

- traditional cv, восприятие, цветовые пространства, как работают камеры, как хранить картинки, исторические справки, физика света и цвета и т.д.
  - про это интересно можно послушать тут - https://www.youtube.com/watch?v=zNCvTcoM1I4

Dataset

ML algo

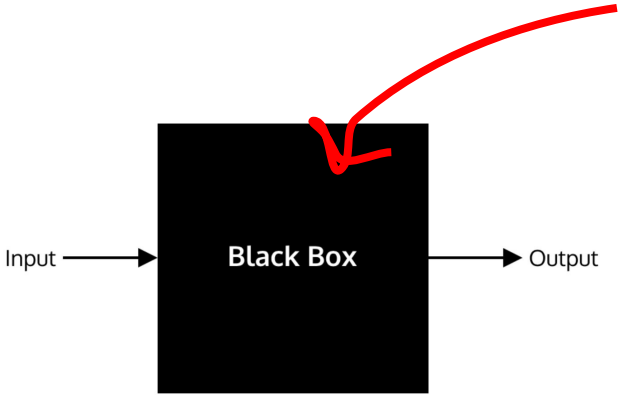


Classes:

dog or food or nothing?

Dataset

ML algo



Classes:

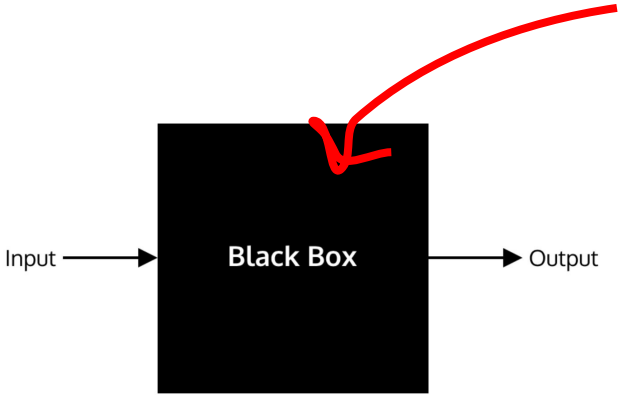
dog or food or nothing?

Что нам нужно?

- Данные - куча картинок

Dataset

ML algo



Classes:

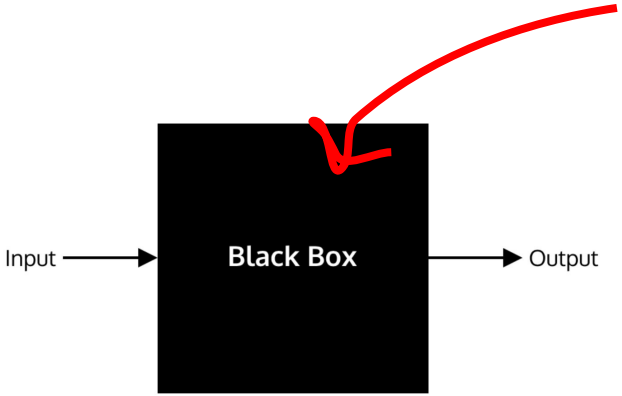
dog or food or nothing?

Что нам нужно?

- Данные - куча картинок
- Алгоритм - мл моделька

Dataset

ML algo



Classes:

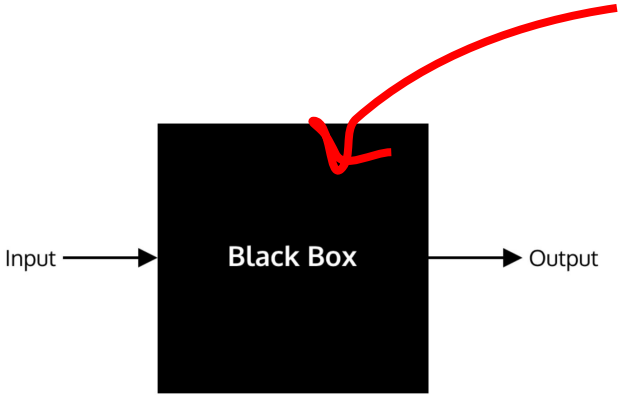
dog or food or nothing?

Что нам нужно?

- Данные - куча картинок
- Алгоритм - мл моделька
- Фичи - ???

Dataset

ML algo



Classes:

dog or food or nothing?

Что нам нужно?

- Данные - куча картинок
- Алгоритм - мл моделька
- Фичи - ???

Крафтим фичи руками

- Используем пиксели as is
- Можно если все картинки одинакового размера
- Если ML модель не использует доп связи - то пиксели не связаны

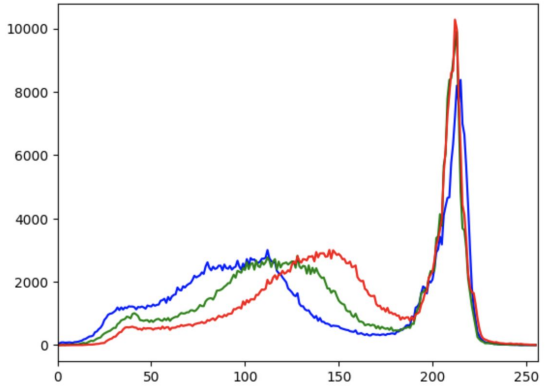
image 64x64x3

1d tensor (12288,)

Крафтим фичи руками

- Считаем гистограммы по цветам
- Работает с variable размерами

image 64x64x3

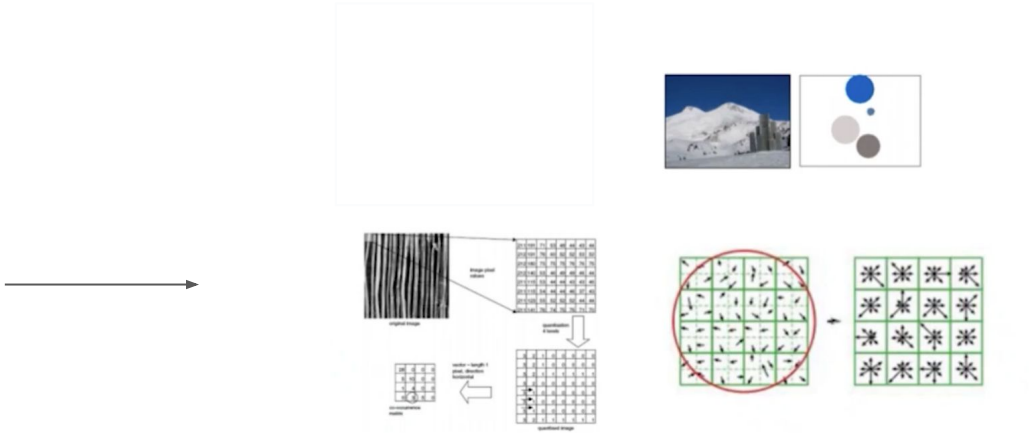


1d tensor (255*3,)

Крафтим фичи руками - локальные

- Histogram of Oriented Gradients (HOG)
- Scale-Invariant Feature Transform (SIFT)
- Speeded Up Robust Feature (SURF)
- … и так далее

image 64x64x3

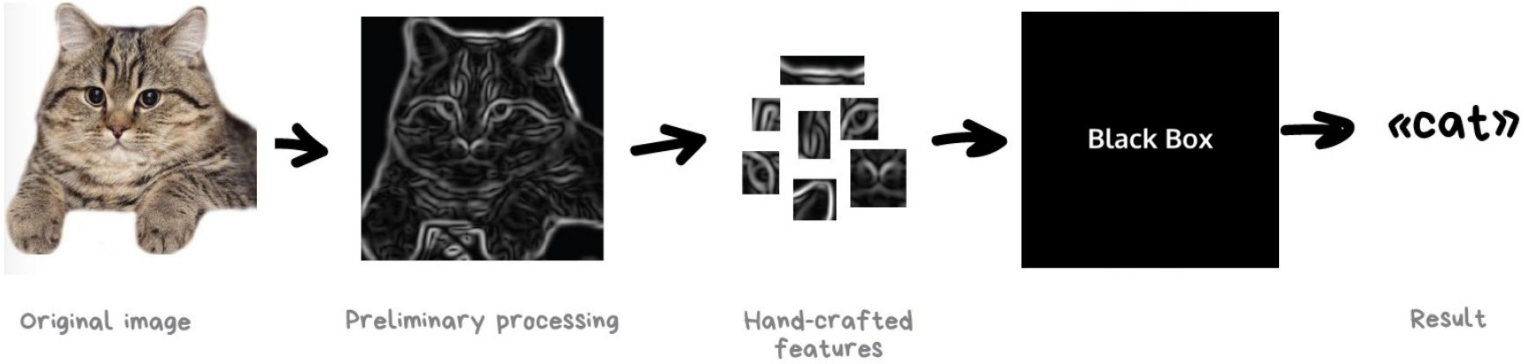


Крафтим фичи руками - bow

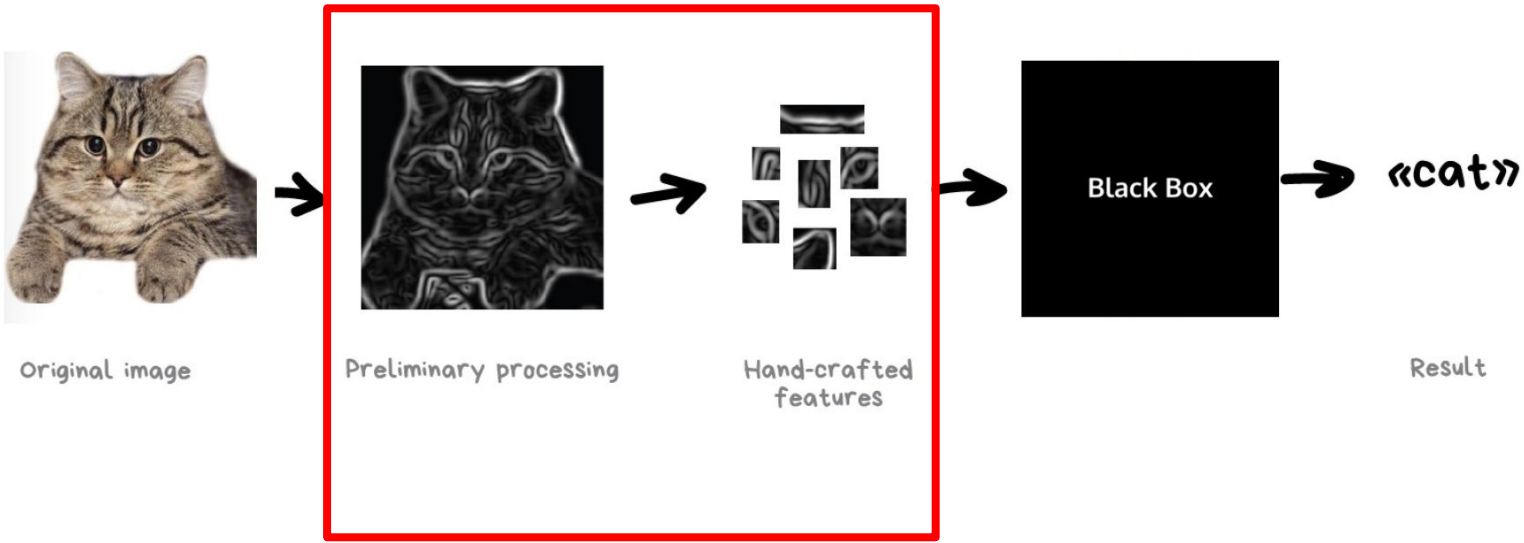
Из пикселей или локальных фич можем взять точки интереса, нарезать картинку оттуда и использовать их в мешке слов

image 64x64x3

Итого

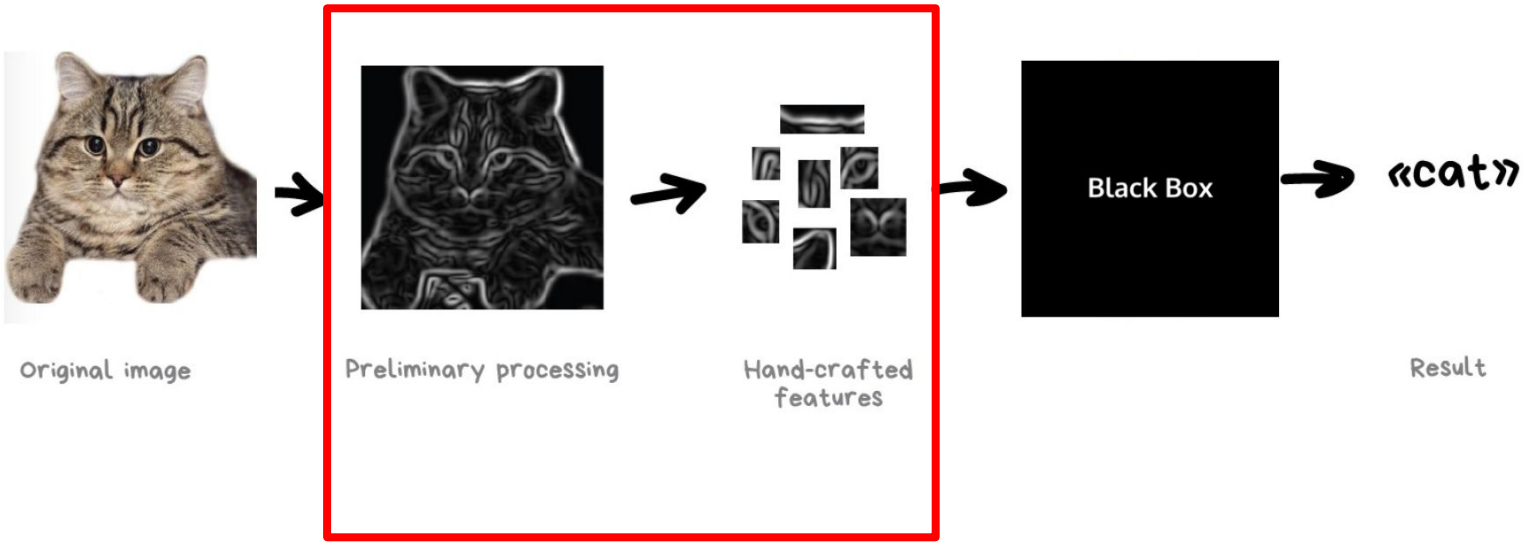

Black Box
«cat»
Original image
Preliminary processing
Hand-crafted features
Result

Итого



Хотим избавиться от ручных фич

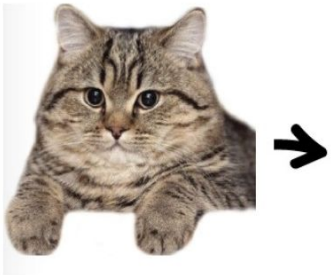
Итого



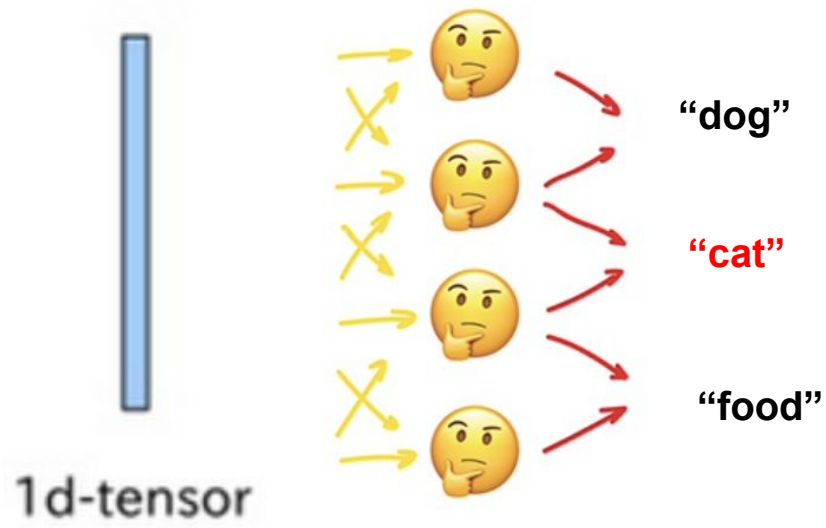
Хотим избавиться от ручных фич

У нас было это свойство в пиксель фичах, но не учитывались связи между пикселями ; через локальные можно добавить - но все еще делаем это руками

Deep Learning quick intro - linear

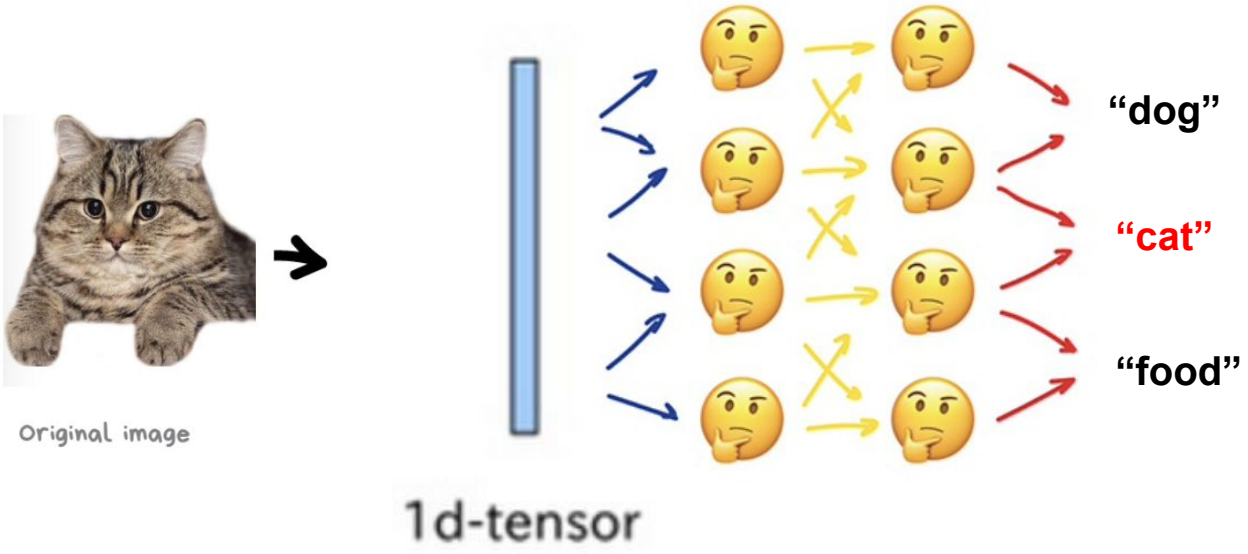


Original image



$y = Wx$

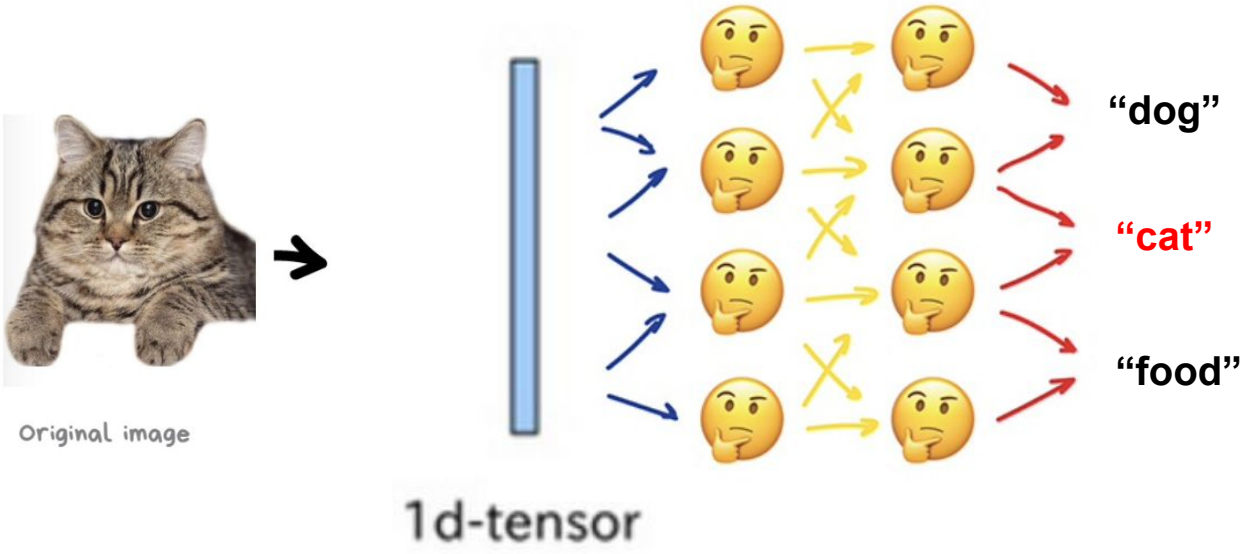
Deep Learning quick intro - linear



$y = W_2 (W_1 x)$

чего то не хватает

Deep Learning quick intro - linear



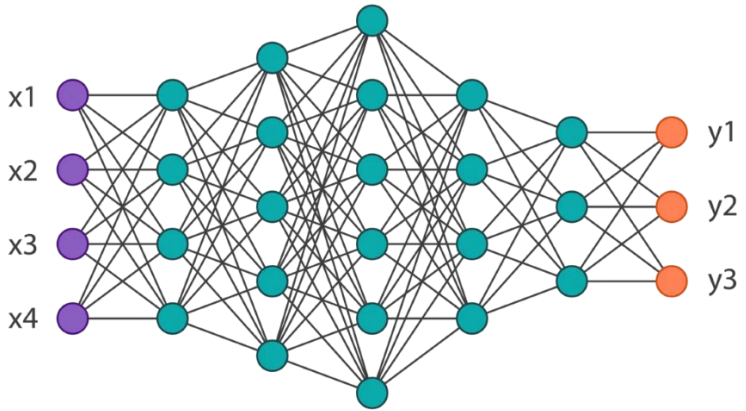
нелинейность - relu=max(0, x), sigmoid=1/(1+e^(-x)) и т д

y = W_2 relu(W_1 x)

Deep Learning quick intro

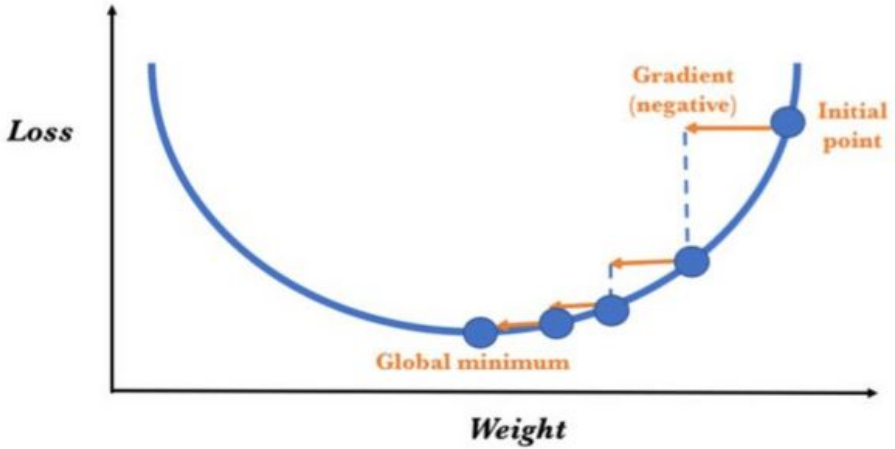
y = W_2 nonlinearity(W_1 x) - двухслойная нейронная сеть

- комбинация линейных слоев и нелинейностей - универсальный апроксиматор
- представляется как вычислительный граф



Deep Learning quick intro - оптимизируем

- для линейных моделей у нас были аналитические решения
- сейчас у нас произвольный граф, в общем случае нейронная сеть - невыпуклая функция
- оптимизируем градиентным спуском, над уметь считать градиенты
- оптимизируем какую нибудь loss функцию, для классификации - кроссентропию например

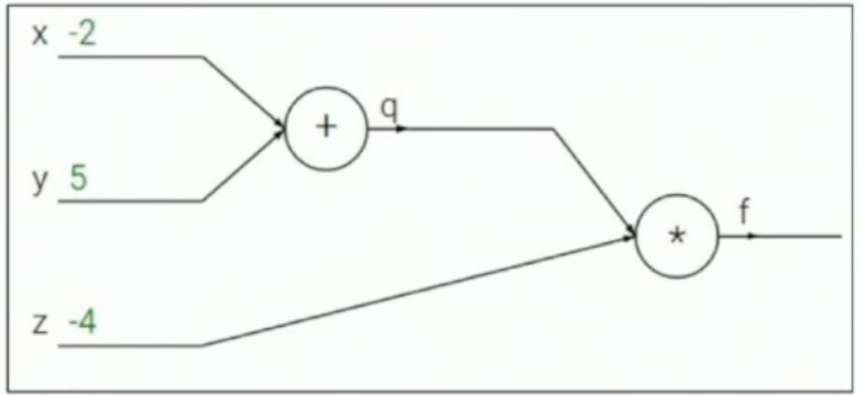


Deep Learning quick intro - оптимизируем

Backpropagation: Simple Example

$$f(x, y, z) = (x + y)z$$

e.g. x = -2, y = 5, z = -4



1. Forward pass: Compute outputs

$$q = x + y \qquad f = qz$$

2. Backward pass: Compute derivatives

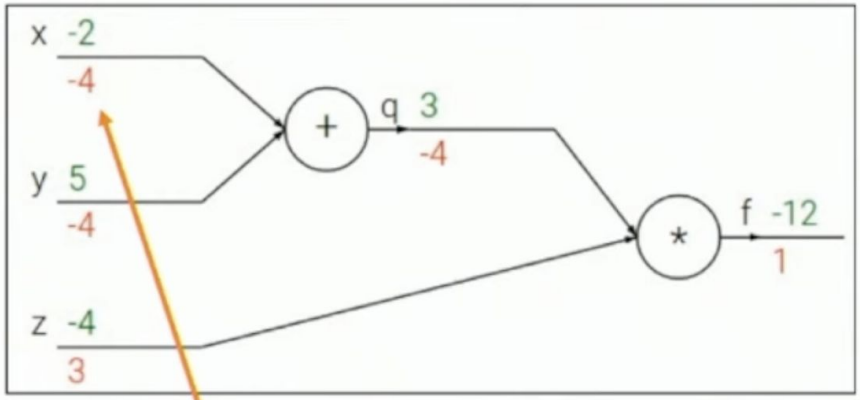
Want: $\frac{\partial f}{\partial x}, \frac{\partial f}{\partial y}, \frac{\partial f}{\partial z}$

Deep Learning quick intro - оптимизируем

Backpropagation: Simple Example

$$f(x, y, z) = (x + y)z$$

e.g. x = -2, y = 5, z = -4



1. Forward pass: Compute outputs

$$q = x + y \qquad f = qz$$

2. Backward pass: Compute derivatives

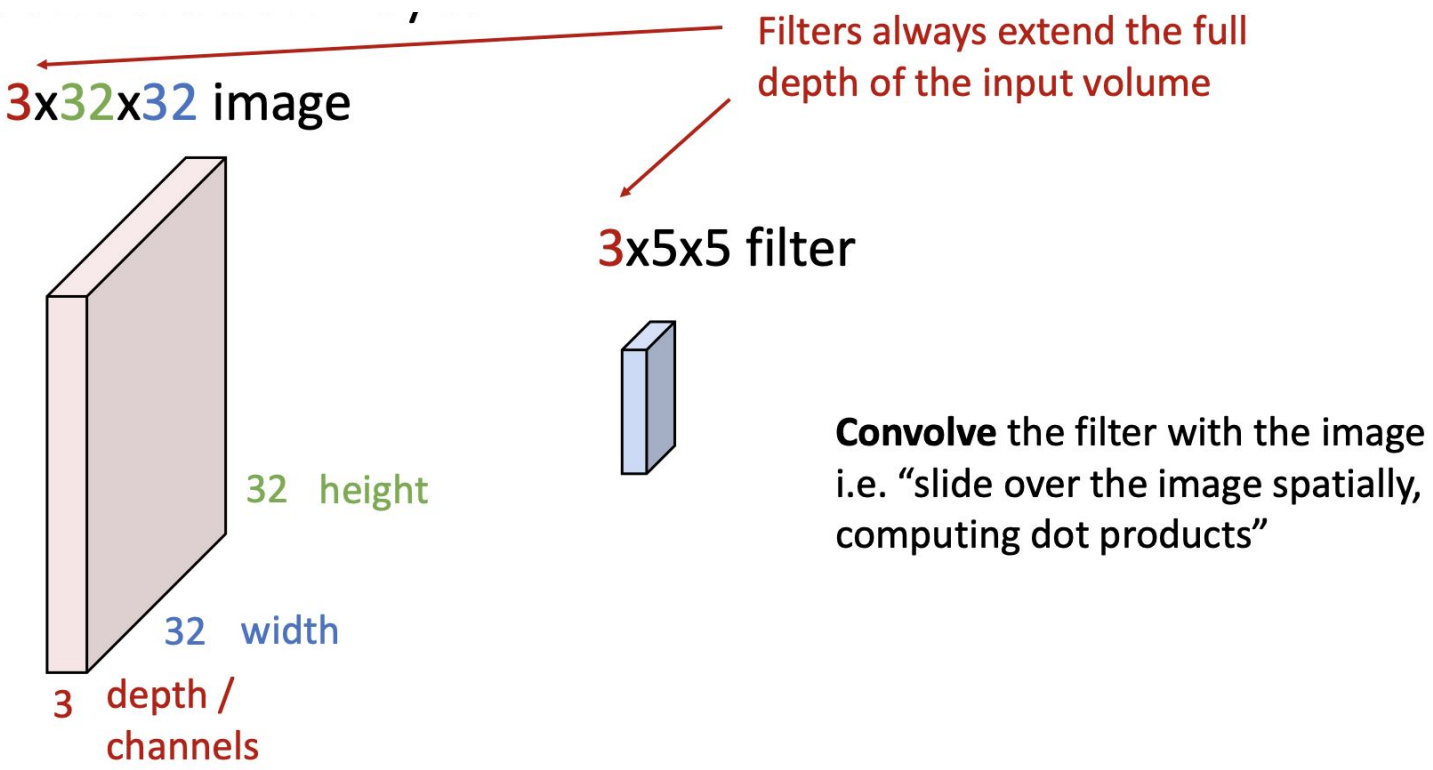
Want: $\frac{\partial f}{\partial x}, \frac{\partial f}{\partial y}, \frac{\partial f}{\partial z}$

Chain Rule

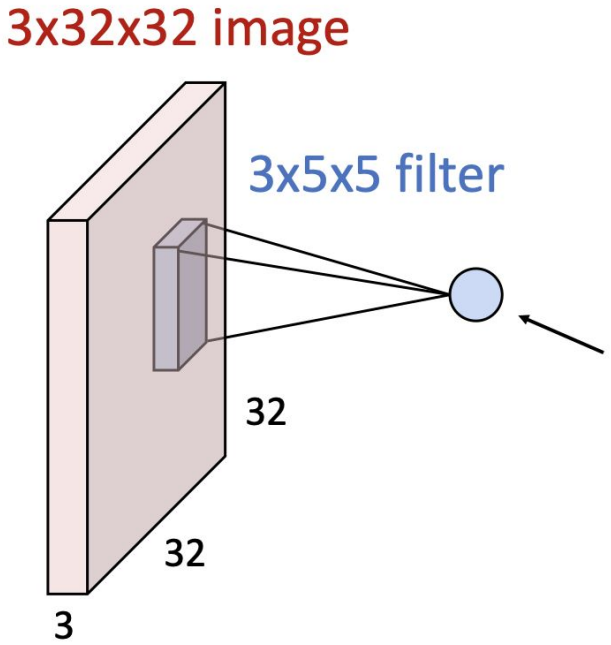
$$\frac{\partial f}{\partial x} = \frac{\partial q}{\partial x}\frac{\partial f}{\partial q}$$

$$\frac{\partial q}{\partial x} = 1$$

Deep Learning - conv networks



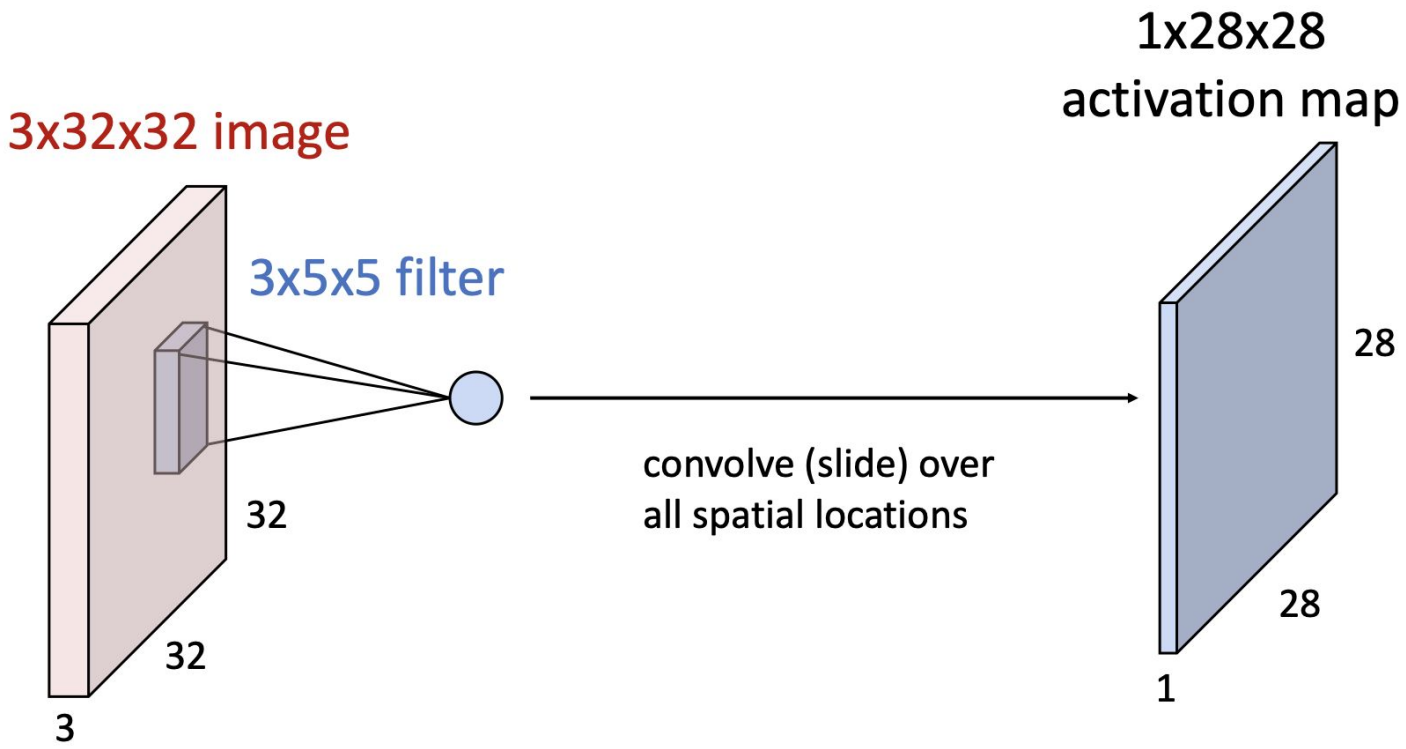
Deep Learning - conv networks



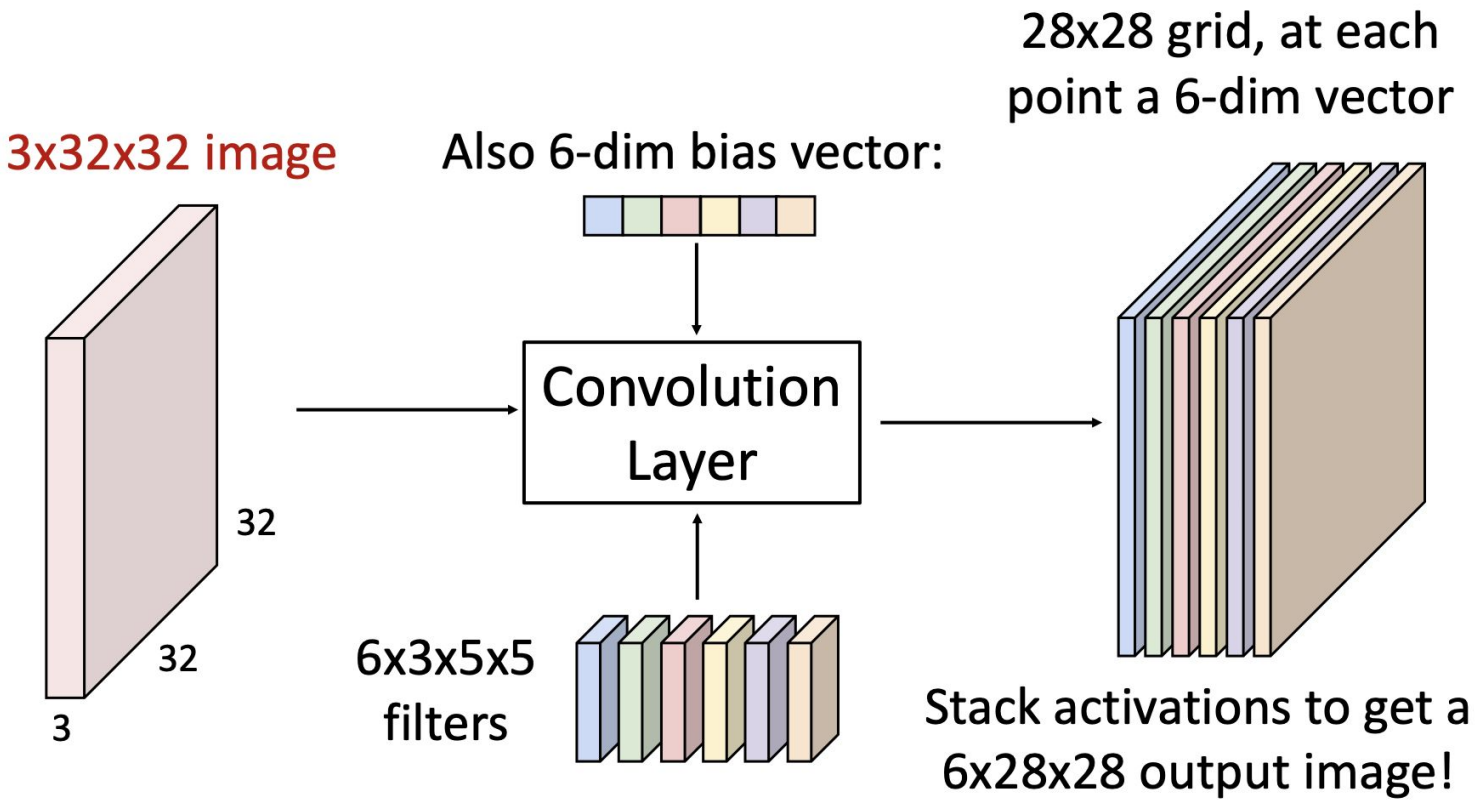
1 number:
the result of taking a dot product between the filter and a small 3x5x5 chunk of the image
(i.e. 3*5*5 = 75-dimensional dot product + bias)

$$w^T x + b$$

Deep Learning - conv networks

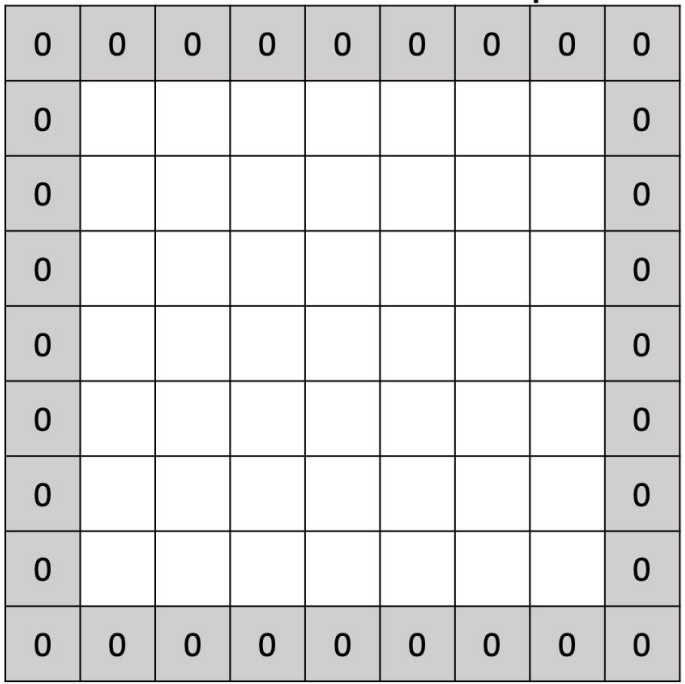


Deep Learning - conv networks



Deep Learning - conv networks

A closer look at spatial dimensions



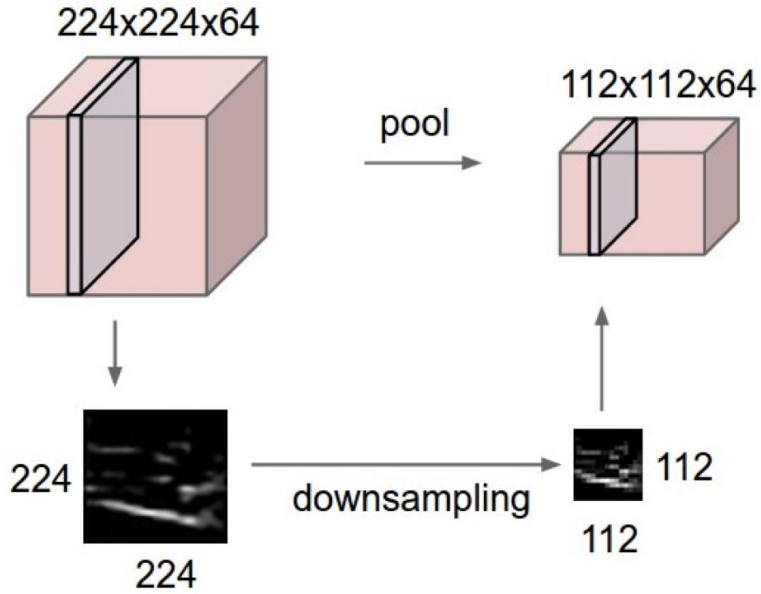
Input: 7x7
Filter: 3x3
Output: 5x5

In general:
Input: W
Filter: K
Padding: P
Output: W – K + 1 + 2P

Very common:
Set P = (K – 1) / 2 to make output have same size as input!

Deep Learning - conv networks

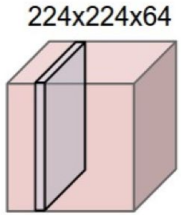
Стакая конволюции друг за другом мы не уменьшаем размерность



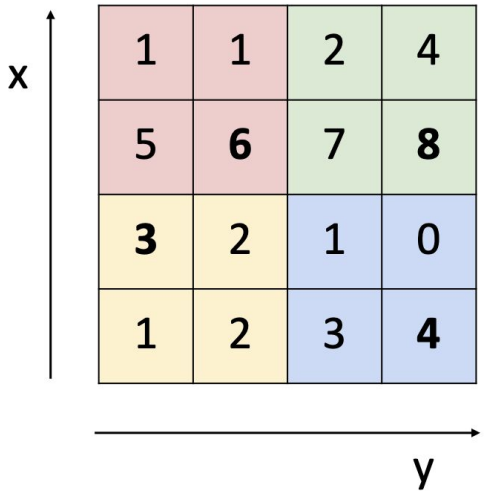
Hyperparameters:
Kernel Size
Stride
Pooling function

Deep Learning - conv networks

Max Pooling



Single depth slice



Max pooling with 2x2 kernel size and stride 2

6	8
3	4

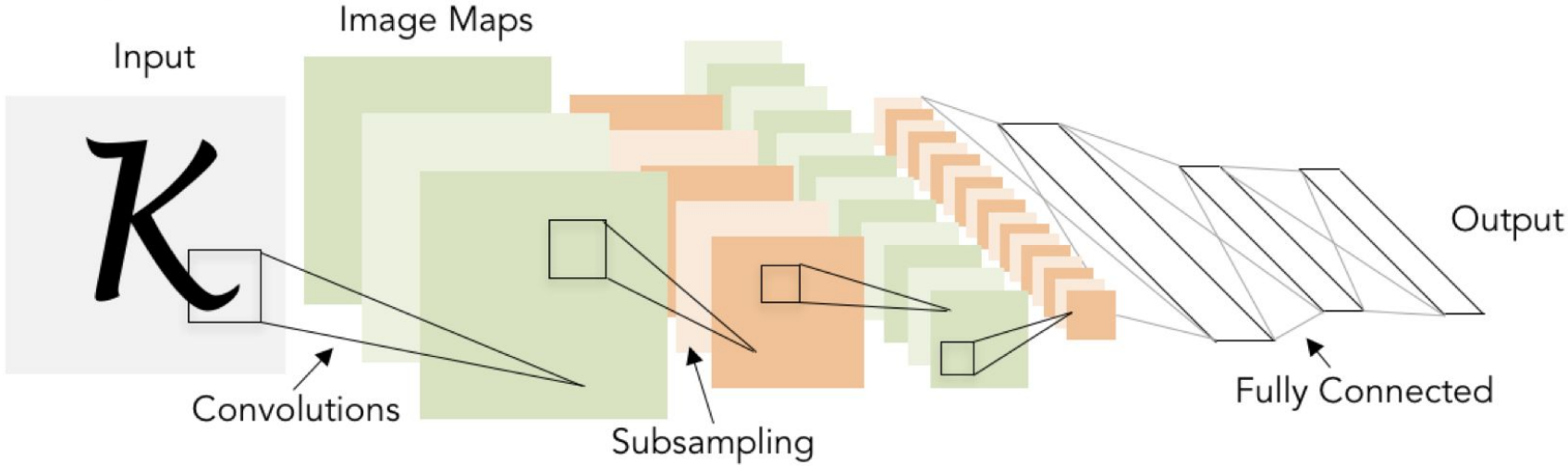
Introduces **invariance** to small spatial shifts

No learnable parameters!

Deep Learning - conv networks

Classic architecture: [Conv, ReLU, Pool] x N, flatten, [FC, ReLU] x N, FC

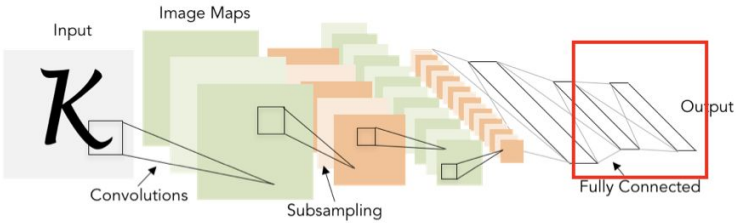
Example: LeNet-5



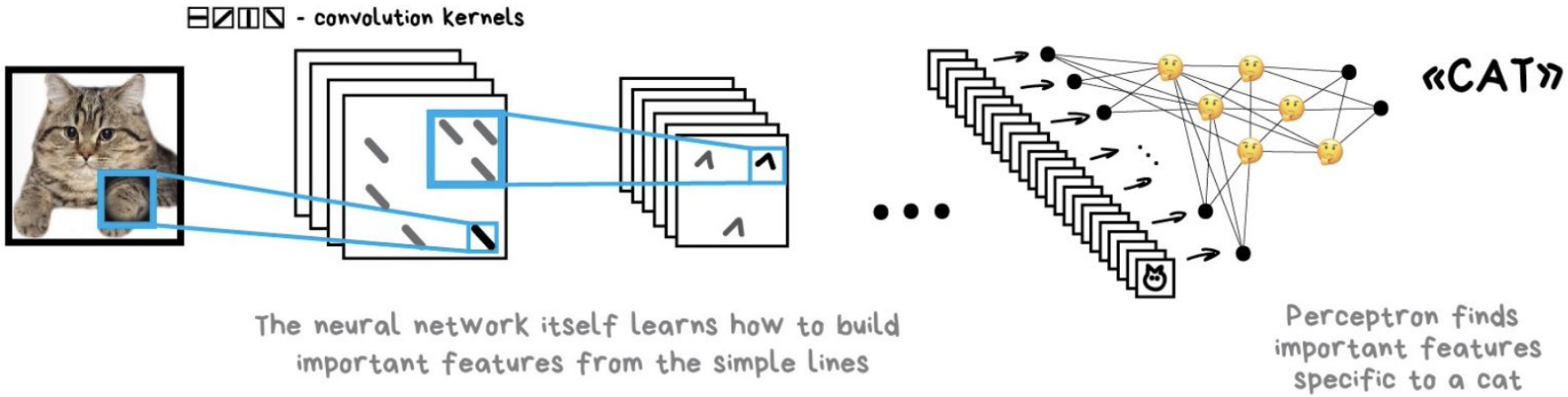
Deep Learning - conv networks

Example: LeNet-5

Layer	Output Size	Weight Size
Input	1 x 28 x 28	
Conv (C_{out}=20, K=5, P=2, S=1)	20 x 28 x 28	20 x 1 x 5 x 5
ReLU	20 x 28 x 28	
MaxPool(K=2, S=2)	20 x 14 x 14	
Conv (C_{out}=50, K=5, P=2, S=1)	50 x 14 x 14	50 x 20 x 5 x 5
ReLU	50 x 14 x 14	
MaxPool(K=2, S=2)	50 x 7 x 7	
Flatten	2450	
Linear (2450 -> 500)	500	2450 x 500
ReLU	500	
Linear (500 -> 10)	10	500 x 10



Что получилось?



deep learning во многом про feature (representation) learning ; мы не хотим крафтить фичи руками а хотим просто засыпать модель данными

Что дальше?

- Глубже смотрим в CNN
- Задачи зрения - детекция, сегментация, мультимодальные задачи
- Современные архитектуры
- Генерация изображений
- Как это все запускать и использовать

Ссылки

- куча изображений и отсылок из https://vas3k.com/blog/machine_learning/
- примеры и картинки из курса csc по compvision https://www.youtube.com/watch?v=zNCvTcoM1I4
- примеры и картинки из курса по compvision michigan university https://www.youtube.com/watch?v=ANyxBVxmdZ0
- если есть вопросы можно мне написать в телеграмме @vladgl
-