

Цифровая обработка изображений

1. Базовая теория

<https://github.com/a4tunado/lectures/tree/master/cv/001>

Мой опыт

- Более 6-ти лет занимаюсь задачами в области машинного обучения
- Проектами в области компьютерного зрения занимаюсь более трех лет
- Успешный опыт внедрения проектов обработки изображений в рамках Yandex Data Factory
- В настоящее время работаю в Yandex над проектом беспилотного автомобиля

План курса

- 1. Классическое компьютерное зрение
 - 1.1 Базовая теория
 - 1.2 Выделение признаков изображения и поиск
 - 1.3 Сегментация и детекция объектов на изображении

План курса

- 2. Применение нейросетей для задач компьютерного зрения
 - 2.1 Принцип работы нейронных сетей
 - 2.2 Нейронные сети для обработки изображений
 - 2.3 Особенности обучения нейронных сетей

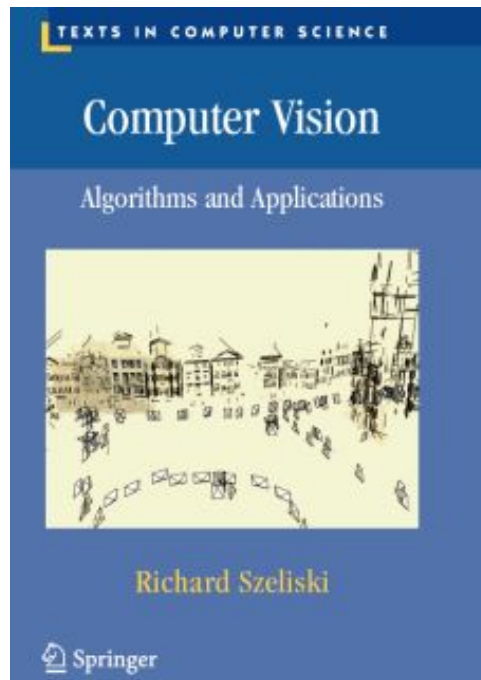
План курса

- 2. Применение нейросетей для задач компьютерного зрения
 - 2.4 Сегментация и детекция объектов на изображении
 - 2.5 Рекуррентные сети в задачах обработки изображений
 - 2.6 Генеративные модели

Лабораторные работы

1. Классификация рукописного ввода на данных MNIST
2. Реализация полносвязной нейронной сети
3. Классификация Cats vs Dogs с помощью сверточной нейронной сети

Рекомендуемые материалы по курсу



<http://szeliski.org/Book/>

Рекомендуемые материалы по курсу

CS231n Convolutional Neural Networks for Visual Recognition

These notes accompany the Stanford CS class [CS231n: Convolutional Neural Networks for Visual Recognition](#). For questions/concerns/bug reports contact [Justin Johnson](#) regarding the assignments, or contact [Andrej Karpathy](#) regarding the course notes. You can also submit a pull request directly to our [git repo](#). We encourage the use of the [hypothes.is](#) extension to annotate comments and discuss these notes inline.

Spring 2017 Assignments

Assignment #1: Image Classification, kNN, SVM, Softmax, Neural Network

Assignment #2: Fully-Connected Nets, Batch Normalization, Dropout, Convolutional Nets

Assignment #3: Image Captioning with Vanilla RNNs, Image Captioning with LSTMs, Network Visualization, Style Transfer, Generative Adversarial Networks

Module 0: Preparation

Python / Numpy Tutorial

IPython Notebook Tutorial

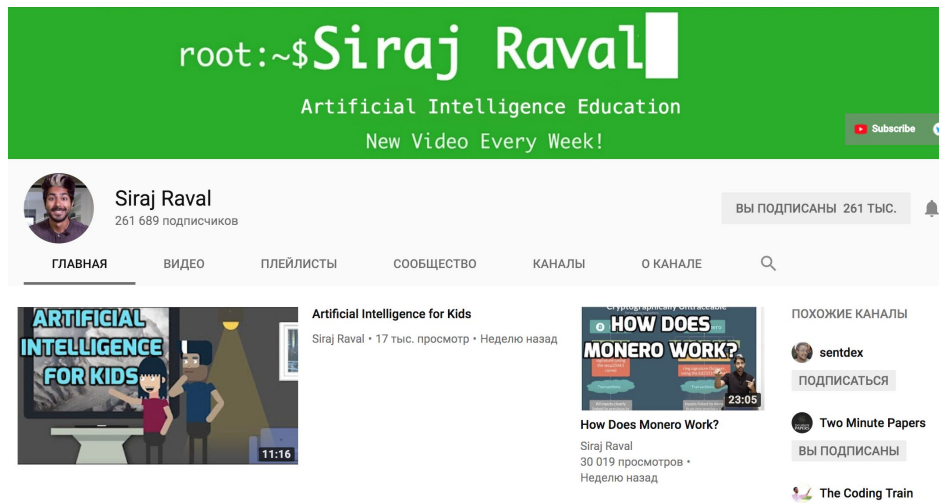
Google Cloud Tutorial

Google Cloud with GPUs Tutorial (for assignment 2 onwards)

AWS Tutorial

<http://cs231n.github.io/>

Рекомендуемые материалы по курсу



<https://www.youtube.com/channel/UCWN3xxRkmTPmbKwht9FuE5A>

Примеры задач компьютерного зрения

Пример. Применение фильтров для обработки изображений

noisy lena



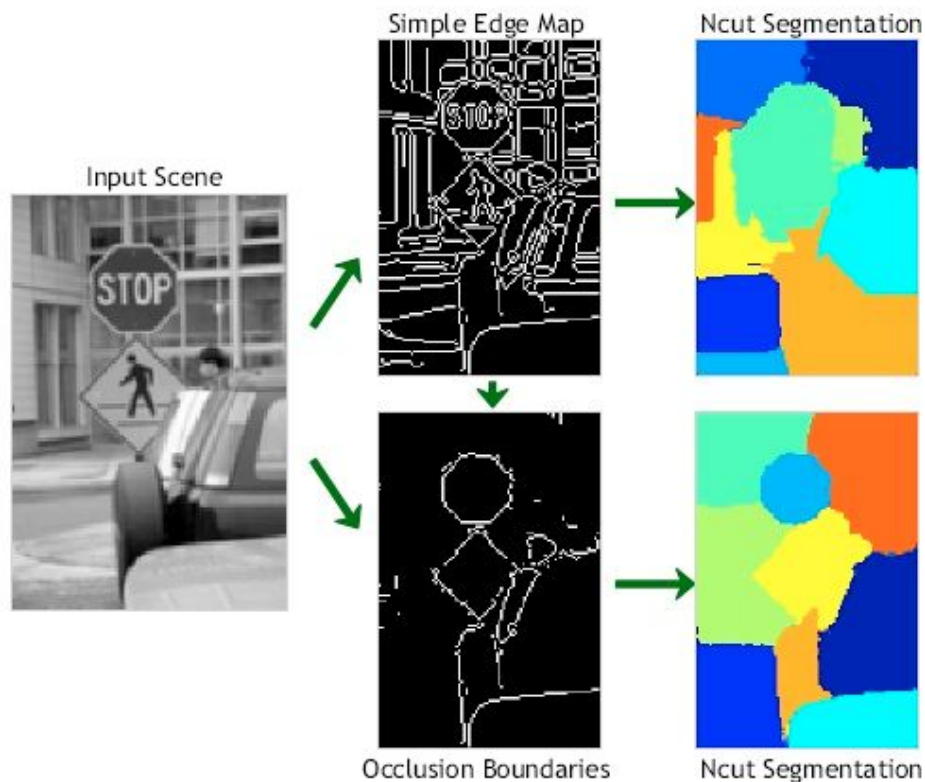
Gaussian filter



Пример. Удаление шума на изображении



Пример. Детектор границ объектов



План занятия

План занятия

1. Цифровое представление изображений
2. Растровые форматы, сжатие
3. Преобразование изображений с помощью линейных фильтров
4. Нелинейные фильтры для обработки изображения
5. Другие способы обработки изображений

1.1 Цифровое представление изображений

Растр vs Вектор

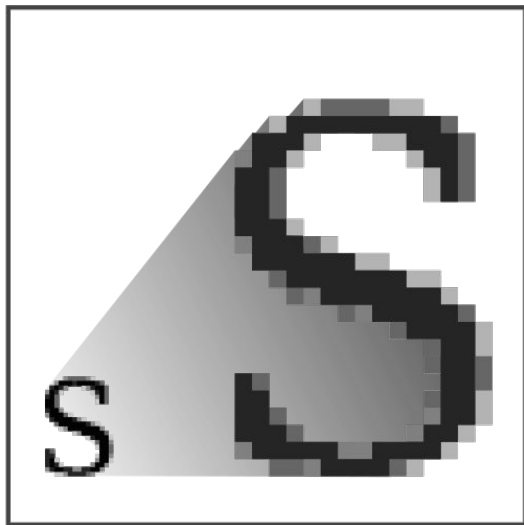
Растр

описание изображения на уровне точек (пикселей)
размер изображения ограничен числом пикселей

Вектор

описание изображения на уровне фигур и их свойств
размер изображения может быть произвольным

Растр vs Вектор

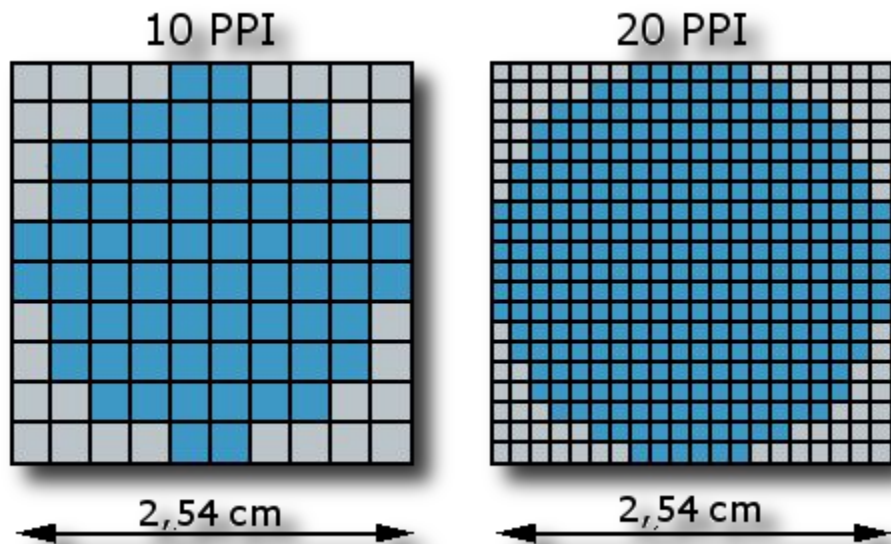


РАСТР
.jpeg .gif .png



ВЕКТОР
.svg

Разрешение



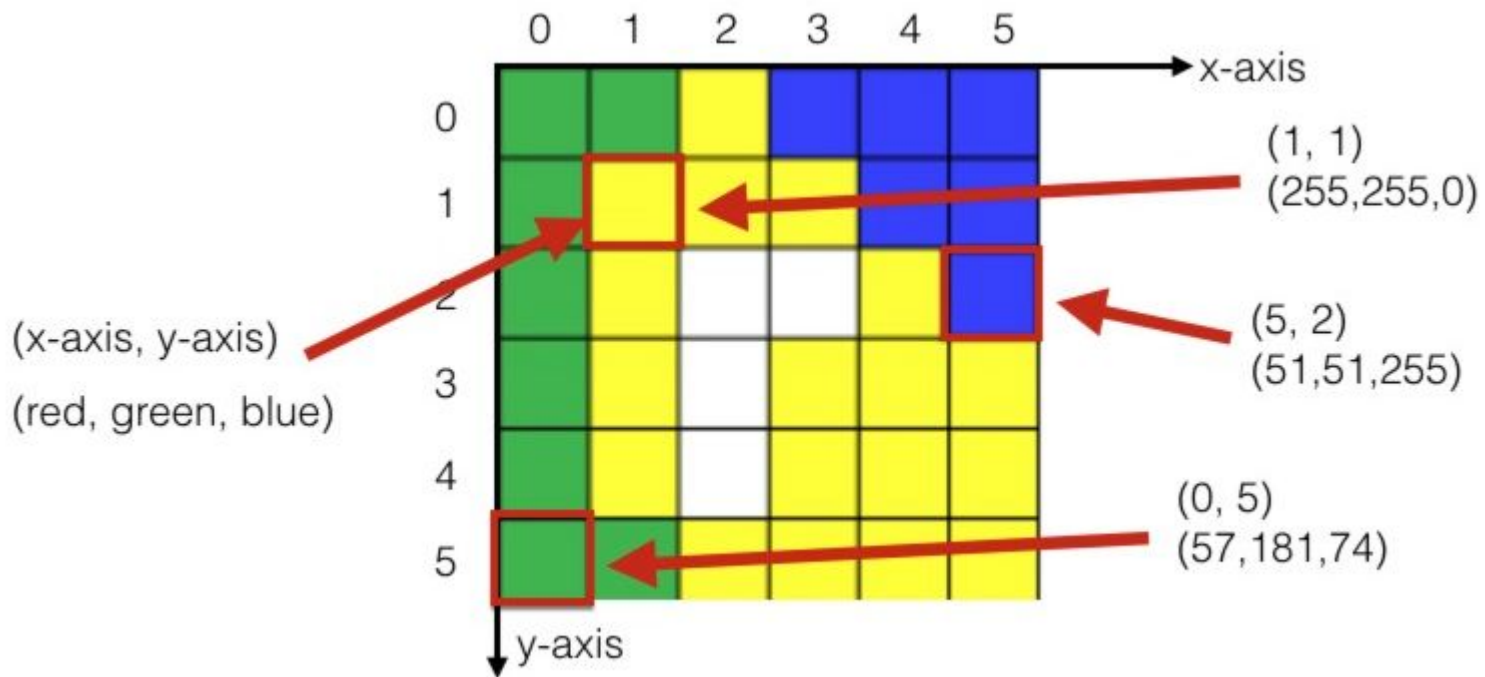
Каналы и динамический диапазон

- каждый пиксель изображения кодируется одним или несколькими значениями (каналами)
- стандартный диапазон значений в каждом канале: 0..255 (один байт или 8 бит)

Каналы и динамический диапазон

- для представления черно-белого изображения достаточно одного канала (передача яркости пикселя)
- цветные изображения, как правило, содержат 3 канала

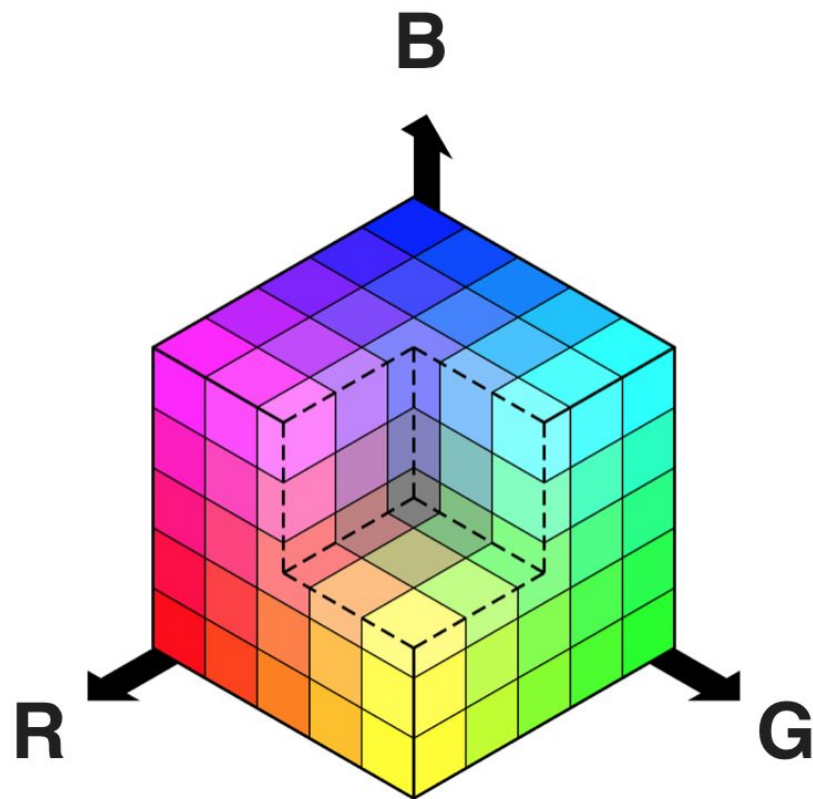
Растровое представление изображения



RGB

- RGB - **R**ed **G**reen **B**lue
 - наиболее распространенное представление цветного изображения
 - выбор основных цветов обусловлен восприятием цвета сетчатки глаза
 - 3 канала

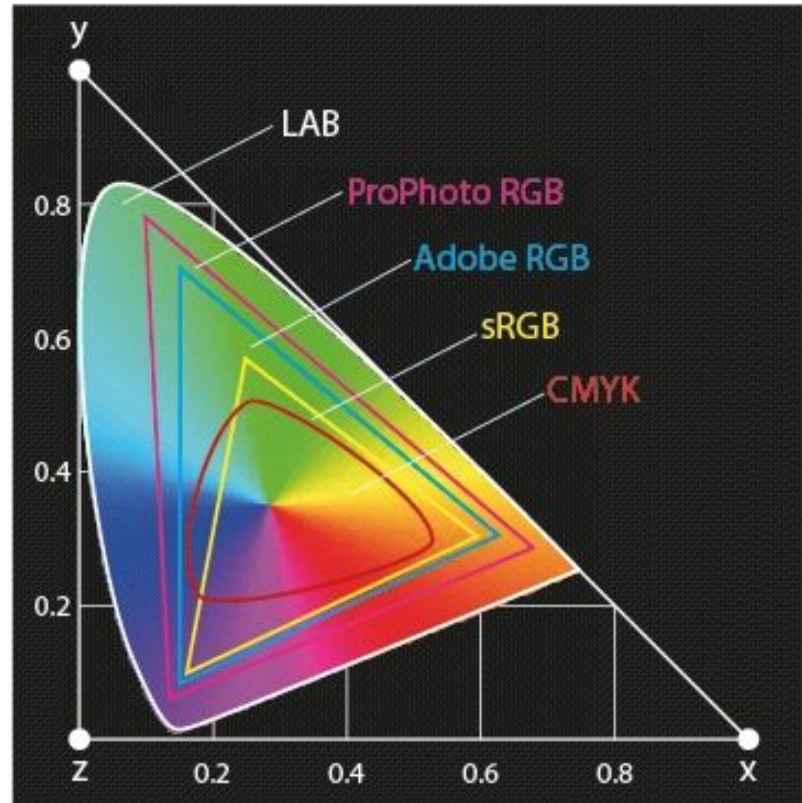
RGB



CMYK

- CMYK - **C**yan **M**agenta **Y**ellow **B**lack
 - 4 канала
 - в основном используется в полиграфии
 - цветовой охват меньше в сравнении с RGB

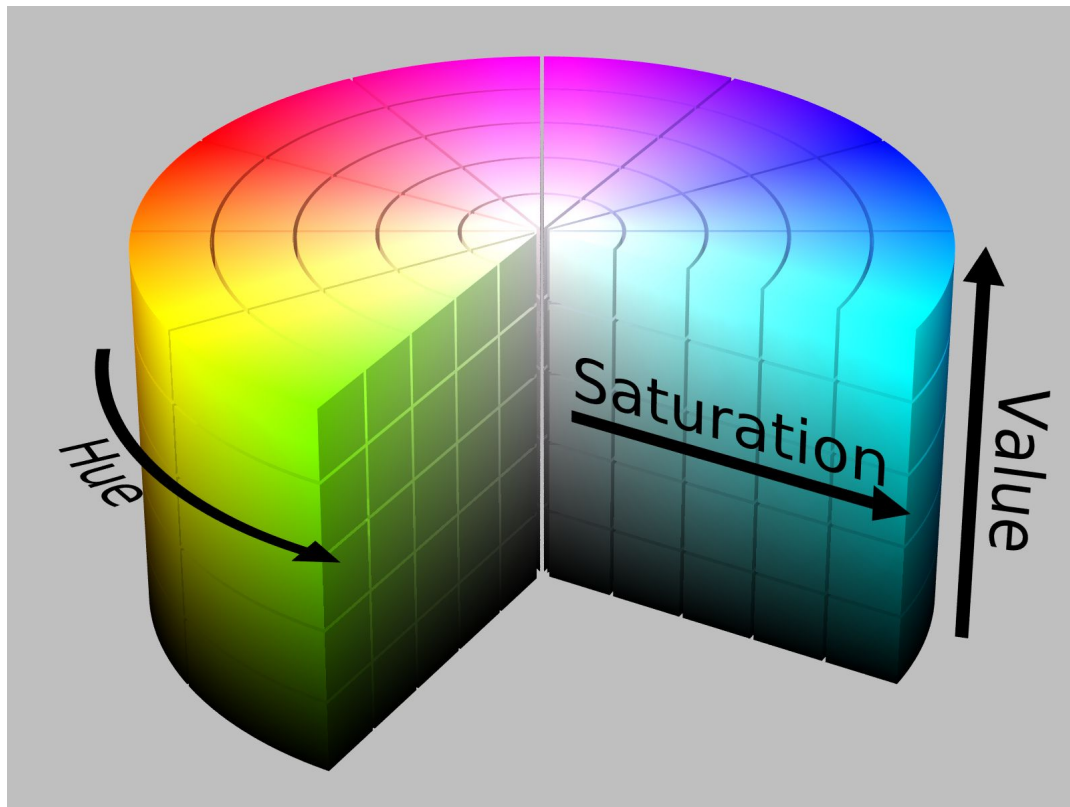
CMYK vs RGB



HSV

- HSV - **H**ue **S**aturation **V**alue
 - Hue - цветовой тон
 - Saturation - насыщенность цвета
 - Value - интенсивность

HSV



RGB -> HSV

$$V \leftarrow \max(R, G, B)$$

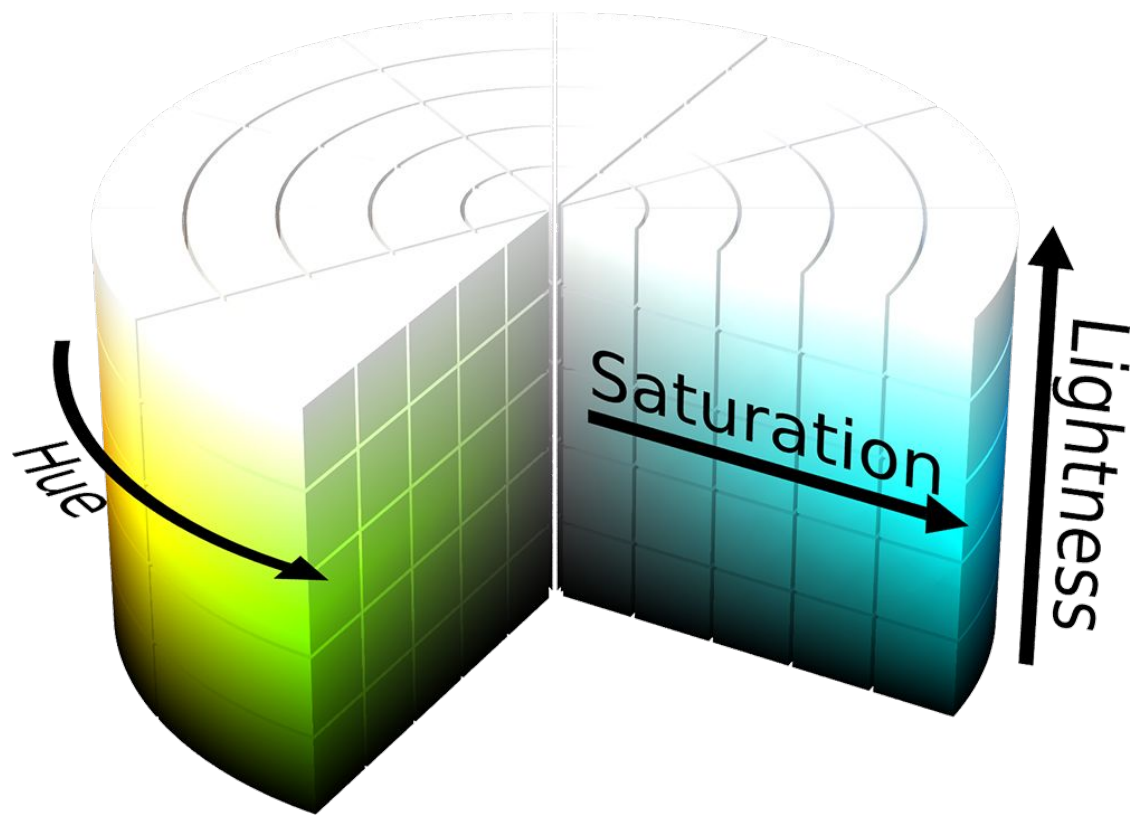
$$S \leftarrow \begin{cases} \frac{V - \min(R, G, B)}{V} & \text{if } V \neq 0 \\ 0 & \text{otherwise} \end{cases}$$

$$H \leftarrow \begin{cases} 60(G - B)/(V - \min(R, G, B)) & \text{if } V = R \\ 120 + 60(B - R)/(V - \min(R, G, B)) & \text{if } V = G \\ 240 + 60(R - G)/(V - \min(R, G, B)) & \text{if } V = B \end{cases}$$

HSL

- HLS - **H**ue **S**aturation **L**ightness
 - Hue - цветовой тон
 - Saturation - насыщенность цвета
 - Lightness - яркость

HSL



RGB -> HSL

$$V_{max} \leftarrow \max(R, G, B)$$

$$V_{min} \leftarrow \min(R, G, B)$$

$$L \leftarrow \frac{V_{max} + V_{min}}{2}$$

$$S \leftarrow \begin{cases} \frac{V_{max}-V_{min}}{V_{max}+V_{min}} & \text{if } L < 0.5 \\ \frac{V_{max}-V_{min}}{2-(V_{max}+V_{min})} & \text{if } L \geq 0.5 \end{cases}$$

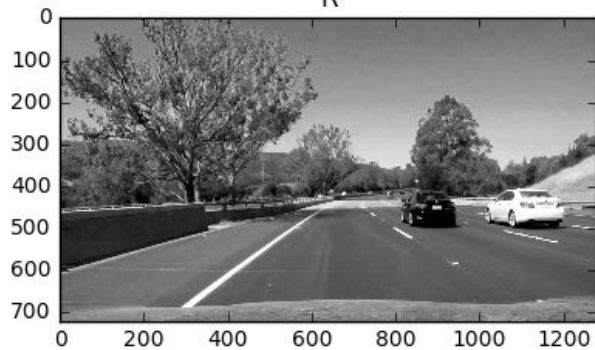
$$H \leftarrow \begin{cases} 60(G - B)/(V_{max} - V_{min}) & \text{if } V_{max} = R \\ 120 + 60(B - R)/(V_{max} - V_{min}) & \text{if } V_{max} = G \\ 240 + 60(R - G)/(V_{max} - V_{min}) & \text{if } V_{max} = B \end{cases}$$

HSL vs RGB

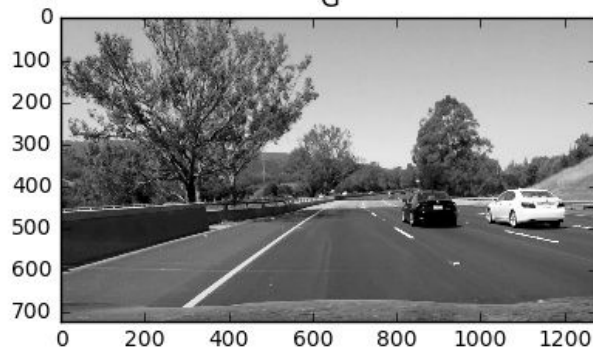


HSL vs RGB

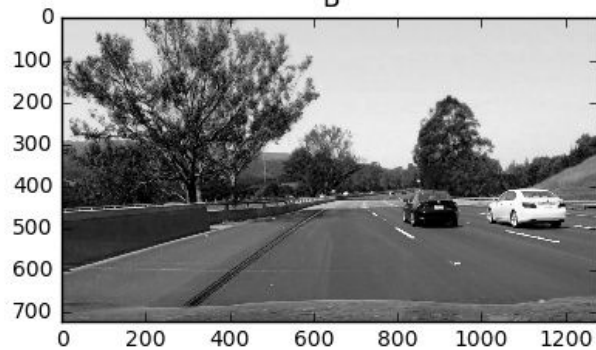
R



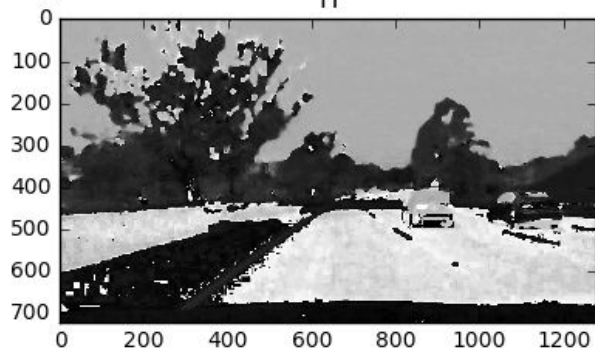
G



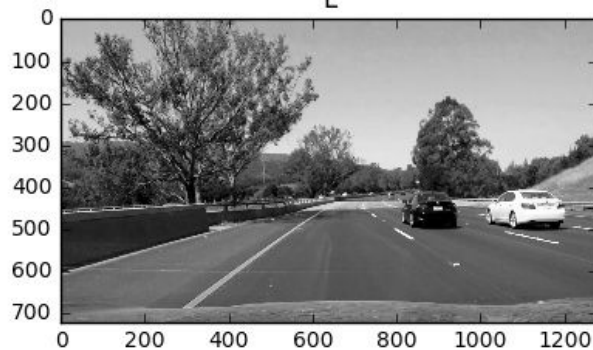
B



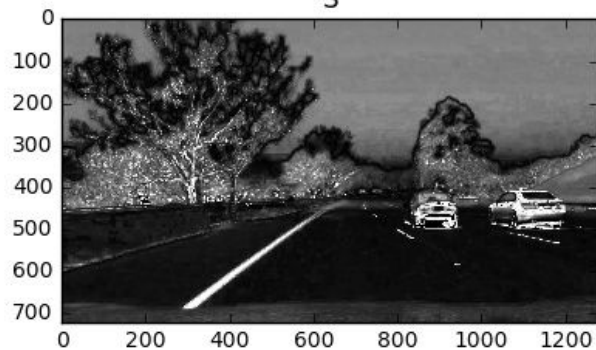
H



L



S



1.2 Растровые форматы. Сжатие

Сжатие изображений

- Матричное представление RGB требовательно к ресурсам памяти
- Каждый пиксель занимает $3 \times 8\text{bit} = 24\text{bit}$ памяти
- Изображение 1024×768 занимает 2,4Mb памяти

Сжатие изображений

- Представление изображения в виде матриц RGB, как правило, содержит избыточную информацию
- Сжатие изображений осуществляется за счет уменьшения объема избыточной информации

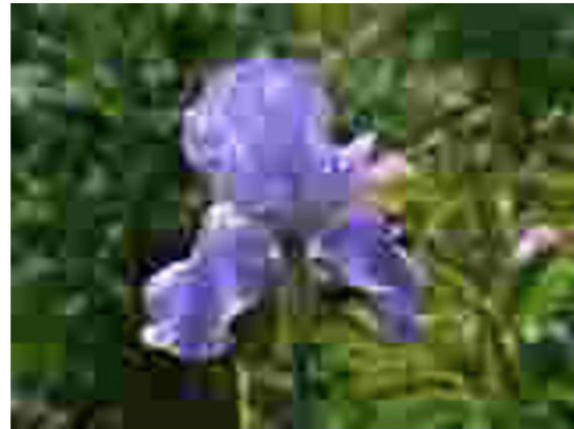
Алгоритмы сжатия изображений

- Сжатие с потерями
 - восстановленное после сжатия изображение может отличаться от исходного
- Сжатие без потерь
 - гарантируется что восстановленное после сжатия изображение совпадает с исходным

Форматы изображений: JPEG

- JPEG [Joint Photographic Experts Group](#)
 - сжатие с потерями - восстановленное изображение не является точной копией исходного
 - уровень сжатия является параметром алгоритма
 - ориентировочный коэффициент сжатия цветного изображения: 10:1 - 20:1
 - использует особенность восприятия изображения человеческим глазом, связанную с большей чувствительностью к изменению яркости пикселей и меньшей чувствительностью к небольшому изменению цвета

Форматы изображений: JPEG



Форматы изображений: PNG

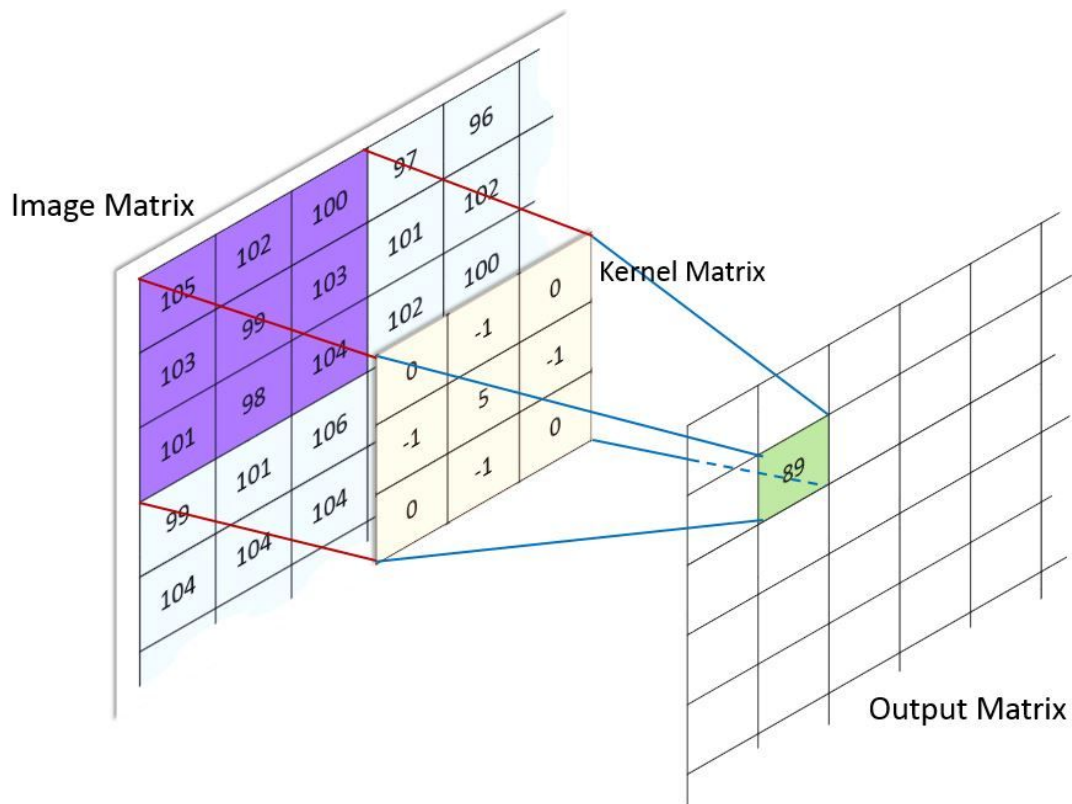
- PNG [Portable Network Graphic](#)
 - сжатие без потерь
 - палитра цветов изображения хранятся в таблице
 - для каждого пикселя указывается индекс цвета из палитры
 - ориентировочный коэффициент сжатия цветного изображения: 2,5:1

1.3 Преобразование изображений с помощью фильтров

Фильтрация изображений

- Свертка
- Размытие (blur)
- Выделение границ (sharpen)
- Удаление шума (denoise)

Свертка



Свертка

$$G[i, j] = \sum_{u=-k}^k \sum_{v=-k}^k H[u, v] F[i - u, j - v]$$

F - исходно изображение

H - фильтр размера k x k

G - результат на выходе свертки

i, j - координаты пиксела в области которого применяется операция свертки

Свертка

45	60	98	127	132	133	137	133
46	65	98	123	126	128	131	133
47	65	96	115	119	123	135	137
47	63	91	107	113	122	138	134
50	59	80	97	110	123	133	134
49	53	68	83	97	113	128	133
50	50	58	70	84	102	116	126
50	50	52	58	69	86	101	120

$f(x,y)$

*

0.1	0.1	0.1
0.1	0.2	0.1
0.1	0.1	0.1

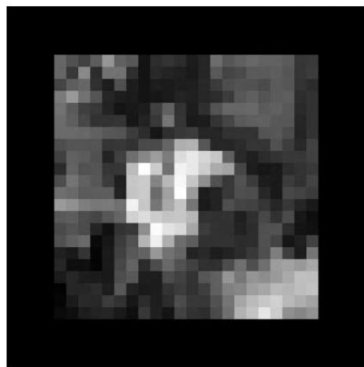
$h(x,y)$

=

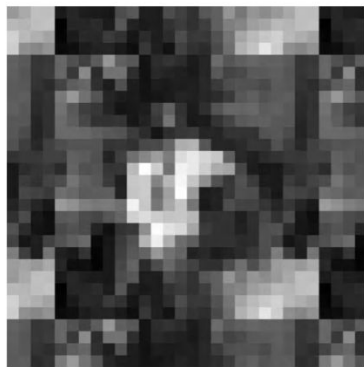
69	95	116	125	129	132
68	92	110	120	126	132
66	86	104	114	124	132
62	78	94	108	120	129
57	69	83	98	112	124
53	60	71	85	100	114

$g(x,y)$

Отступы (Padding)



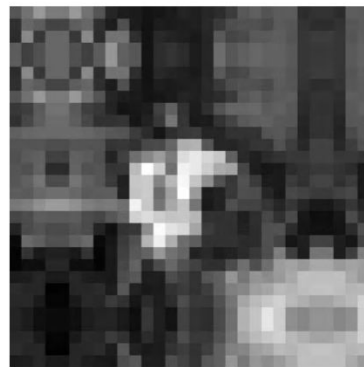
zero



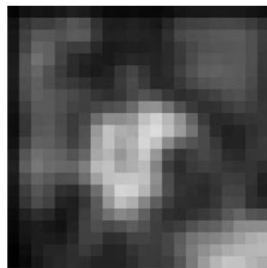
wrap



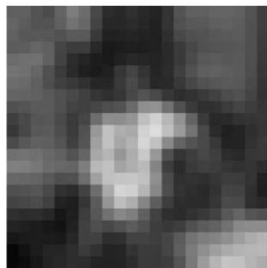
clamp



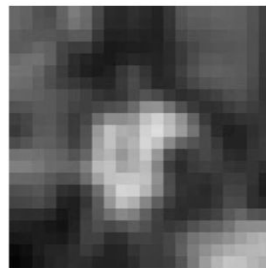
mirror



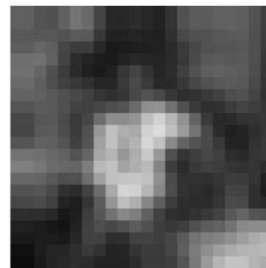
blurred zero



normalized zero



blurred clamp



blurred mirror

Пример фильтра: идентичное отображение



Original



0	0	0
0	1	0
0	0	0



Identical image

Пример фильтра: смещение



Original



0	0	0
1	0	0
0	0	0



Shifted left
By 1 pixel

Пример фильтра: размытие



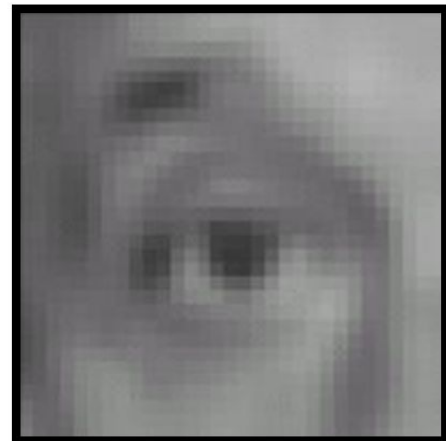
Original



$\frac{1}{9}$

1	1	1
1	1	1
1	1	1

=



Blur (with a mean filter)

Пример фильтра: выделение границ (sharpen)



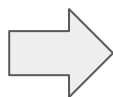
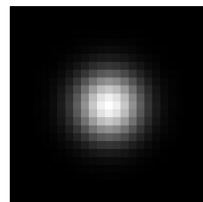
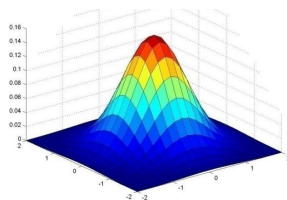
Original

$$* \left(\begin{array}{|c|c|c|} \hline 0 & 0 & 0 \\ \hline 0 & 2 & 0 \\ \hline 0 & 0 & 0 \\ \hline \end{array} - \frac{1}{9} \begin{array}{|c|c|c|} \hline 1 & 1 & 1 \\ \hline 1 & 1 & 1 \\ \hline 1 & 1 & 1 \\ \hline \end{array} \right) =$$



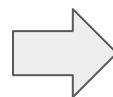
Sharpening filter
(accentuates edges)

Размытие - фильтр Гаусса

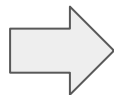
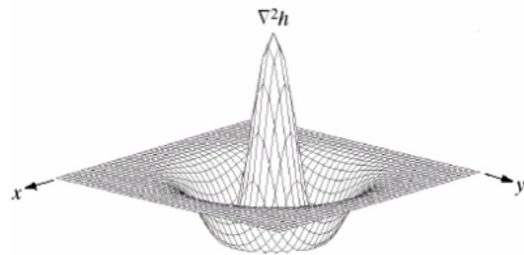


$$\frac{1}{273}$$

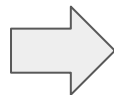
1	4	7	4	1
4	16	26	16	4
7	26	41	26	7
4	16	26	16	4
1	4	7	4	1



Выделение границ - оператор Лапласа



-1	-1	-1
-1	8	-1
-1	-1	-1



Выделение границ - оператор Собеля

-1	0	+1
-2	0	+2
-1	0	+1

x filter

+1	+2	+1
0	0	0
-1	-2	-1

y filter

Выделение границ - оператор Собеля

Original



Sobel X



Sobel Y



Оператор Собеля - градиент

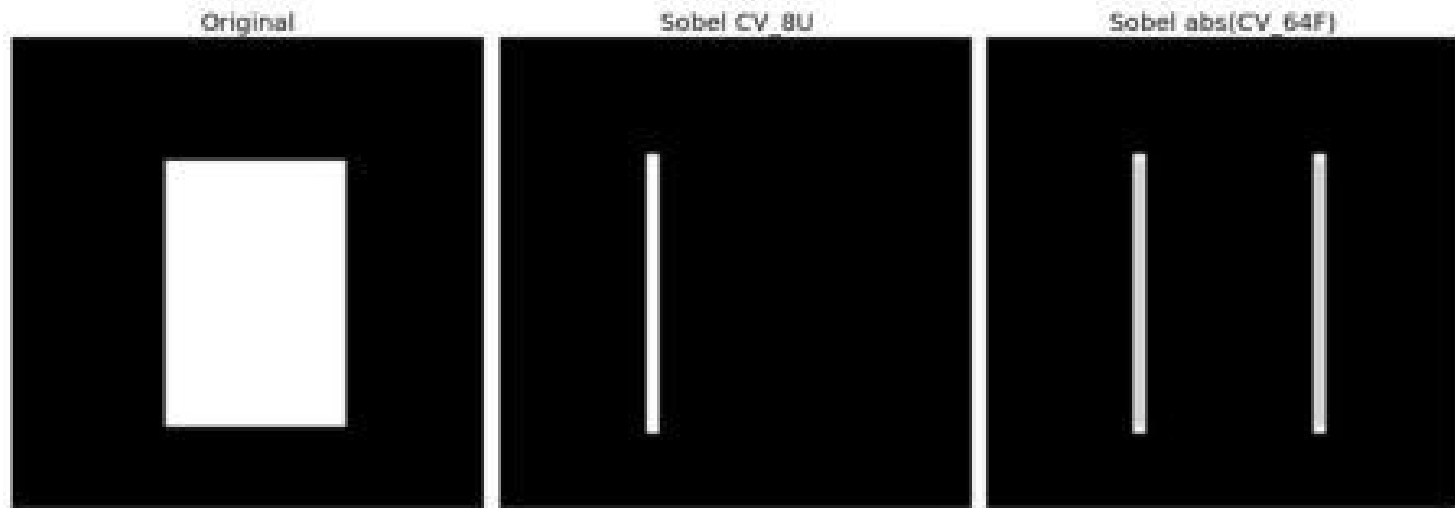
$$g = \sqrt{g_x^2 + g_y^2}$$

$$\theta = \arctan \frac{g_y}{g_x}$$

g, g_x, g_y - длина вектора градиента и его составляющих

θ - угол наклона градиента в полярной системе координат

Оператор Собеля - важен тип данных



Сепарабельные фильтры (separable)

- сложность работы фильтра - K^2 на каждый пиксель изображения (K - размер ядра)
- сложность можно сократить до $2K$, в случае, если ядро фильтра можно представить в виде разложения на вертикальную и горизонтальную составляющие: $K=vh^T$

Сепарабельные фильтры (separable)

1	2	1
2	4	2
1	2	1

 = ?

Сепарабельные фильтры (separable)

$$\begin{bmatrix} 1 & 2 & 1 \\ 2 & 4 & 2 \\ 1 & 2 & 1 \end{bmatrix} = \begin{bmatrix} 1 \\ 2 \\ 1 \end{bmatrix} \times \begin{bmatrix} 1 & 2 & 1 \end{bmatrix}$$

Сепарабельные фильтры (separable)

1	2	1
2	4	2
1	2	1

 *

2	3	3
3	5	5
4	4	6

$= 2 + 6 + 3 = 11$
 $= 6 + 20 + 10 = 36$
 $= 4 + 8 + 6 = 18$

65

1	2	1
2	4	2
1	2	1

 =

1
2
1

 x

1	2	1
---	---	---

1	2	1
---	---	---

 *

2	3	3
3	5	5
4	4	6

 =

	11	
	18	
	18	

1
2
1

 *

	11	
	18	
	18	

 =

	65	

Сепарабельные фильтры (separable)

$$\frac{1}{K^2}$$

1	1	...	1
1	1	...	1
⋮	⋮	1	⋮
1	1	...	1

$$\frac{1}{16}$$

1	2	1
2	4	2
1	2	1

$$\frac{1}{256}$$

1	4	6	4	1
4	16	24	16	4
6	24	36	24	6
4	16	24	16	4
1	4	6	4	1

$$\frac{1}{8}$$

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

$$\frac{1}{4}$$

1	-2	1
-2	4	-2
1	-2	1

$$\frac{1}{K}$$

1	1	...	1
---	---	-----	---

$$\frac{1}{4}$$

1	2	1
---	---	---

$$\frac{1}{16}$$

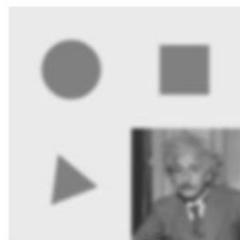
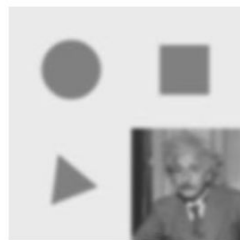
1	4	6	4	1
---	---	---	---	---

$$\frac{1}{2}$$

-1	0	1
----	---	---

$$\frac{1}{2}$$

1	-2	1
---	----	---



(a) box, $K = 5$

(b) bilinear

(c) “Gaussian”

(d) Sobel

(e) corner

Сепарабельные фильтры (separable)

[cv2.sepFilter2D](#)(src, ddepth, kernelX, kernelY) → dst

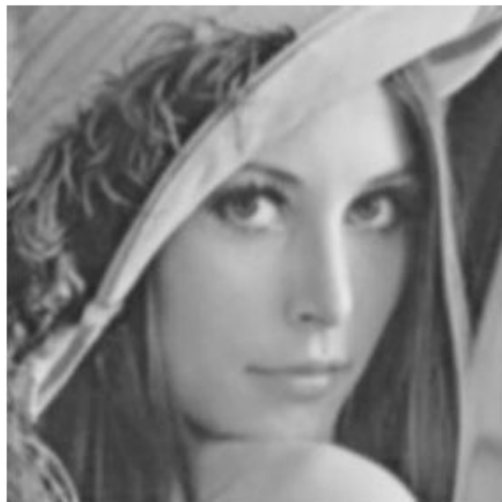
- **src** - исходное изображение
- **ddepth** - тип данных выходного изображения
- **kernelX** - горизонтальная компонента ядра
- **kernelY** - вертикальная компонента ядра

1.4 Нелинейные фильтры для обработки изображений

Медианный фильтр



Билатеральный фильтр



Gaussian filter



Bilateral filter

Билатеральный фильтр

$$g(i, j) = \frac{\sum_{k,l} f(k, l)w(i, j, k, l)}{\sum_{k,l} w(i, j, k, l)}$$

$g(i,j)$ - результат применения фильтра в точке i,j
 $f(k,l)$ - значение исходного изображения в точке k,l
 $w(i,j,k,l)$ - вес, с которым учитывается значение $f(k,l)$

Билатеральный фильтр

$$d(i, j, k, l) = \exp \left(-\frac{(i - k)^2 + (j - l)^2}{2\sigma_d^2} \right)$$

$$r(i, j, k, l) = \exp \left(-\frac{\|f(i, j) - f(k, l)\|^2}{2\sigma_r^2} \right)$$

d - фактор расстояния от точки i, j до k, l

r - “похожесть” значений исходного изображения в точке i, j и k, l

σ - параметры фильтра

Билатеральный фильтр

$$g(i, j) = \frac{\sum_{k,l} f(k, l) w(i, j, k, l)}{\sum_{k,l} w(i, j, k, l)}$$

$$w(i, j, k, l) = \exp \left(-\frac{(i - k)^2 + (j - l)^2}{2\sigma_d^2} - \frac{\|f(i, j) - f(k, l)\|^2}{2\sigma_r^2} \right)$$

w - учитывает как расстояние между точками (i,j) и (k,l) , так и “похожесть” значений исходного изображения в точках (i,j) и (k,l)

1.5 Другие способы обработки изображений изображений

Бинаризация изображения

Camping can be Fun and Adventurous in Our Sturdy Tents!

Camping is an exciting and adventurous activity for the entire family. Enjoying the beauty while doing any recreation is enjoying the view and fun of the scene. Your camping experience is more enjoyable with a lightweight all-weather tent from our selection of sturdy models. These tents, constructed of nylon and canvas, provide shelter and protection from the elements for those sunny and sunny to fall and snow. Our unique light design does not tear during use or vibration with the door closed or open. Our design is the most secure and safe and can be used in any weather to feel comfortable.



Our tents are the most secure and canvas materials available and are guaranteed for five years. All models are tested to our superior tent standards and all designs are made available. Check our tent size for pricing. Details are on our web page. Contact us for a list of frequently asked questions.

Select a model that meets your family's needs:



Our Tents are:

• Sturdy

• Durable

• Weatherproof

• Made of the Best Materials

• Guaranteed for 5 Years

	Sunny	Moonlight	Star
Sturdy	10' x 10'	10' x 10'	10' x 10'
Durable	10' x 10'	10' x 10'	10' x 10'
Weatherproof	10' x 10'	10' x 10'	10' x 10'
Made of the Best Materials	10' x 10'	10' x 10'	10' x 10'
Guaranteed for 5 Years	10' x 10'	10' x 10'	10' x 10'

For more information, visit our website at www.tent.com

Camping can be Fun and Adventurous in Our Sturdy Tents!

Camping is an exciting and adventurous activity for the entire family. Enjoying the beauty while doing any recreation is enjoying the view and fun of the scene. Your camping experience is more enjoyable with a lightweight all-weather tent from our selection of sturdy models. These tents, constructed of nylon and canvas, provide shelter and protection from the elements for those sunny and sunny to fall and snow. Our unique light design does not tear during use or vibration with the door closed or open. Our design is the most secure and safe and can be used in any weather to feel comfortable.



Our tents are the most secure and canvas materials available and are guaranteed for five years. All models are tested to our superior tent standards and all designs are made available. Check our tent size for pricing. Details are on our web page. Contact us for a list of frequently asked questions.

Select a model that meets your family's needs:



Our Tents are:

• Sturdy

• Durable

• Weatherproof

• Made of the Best Materials

• Guaranteed for 5 Years

	Sunny	Moonlight	Star
Sturdy	10' x 10'	10' x 10'	10' x 10'
Durable	10' x 10'	10' x 10'	10' x 10'
Weatherproof	10' x 10'	10' x 10'	10' x 10'
Made of the Best Materials	10' x 10'	10' x 10'	10' x 10'
Guaranteed for 5 Years	10' x 10'	10' x 10'	10' x 10'

For more information, visit our website at www.tent.com

Camping can be Fun and Adventurous in Our Sturdy Tents!

Camping is an exciting and adventurous activity for the entire family. Enjoying the beauty while doing any recreation is enjoying the view and fun of the scene. Your camping experience is more enjoyable with a lightweight all-weather tent from our selection of sturdy models. These tents, constructed of nylon and canvas, provide shelter and protection from the elements for those sunny and sunny to fall and snow. Our unique light design does not tear during use or vibration with the door closed or open. Our design is the most secure and safe and can be used in any weather to feel comfortable.



Select a model that meets your family's needs:



Our Tents are:

• Sturdy

• Durable

• Weatherproof

• Made of the Best Materials

• Guaranteed for 5 Years

	Sunny	Moonlight	Star
Sturdy	10' x 10'	10' x 10'	10' x 10'
Durable	10' x 10'	10' x 10'	10' x 10'
Weatherproof	10' x 10'	10' x 10'	10' x 10'
Made of the Best Materials	10' x 10'	10' x 10'	10' x 10'
Guaranteed for 5 Years	10' x 10'	10' x 10'	10' x 10'

For more information, visit our website at www.tent.com

Бинаризация изображения

$$\theta(f, t) = \begin{cases} 1 & \text{if } f \geq t, \\ 0 & \text{else,} \end{cases}$$

θ - оператор бинаризации

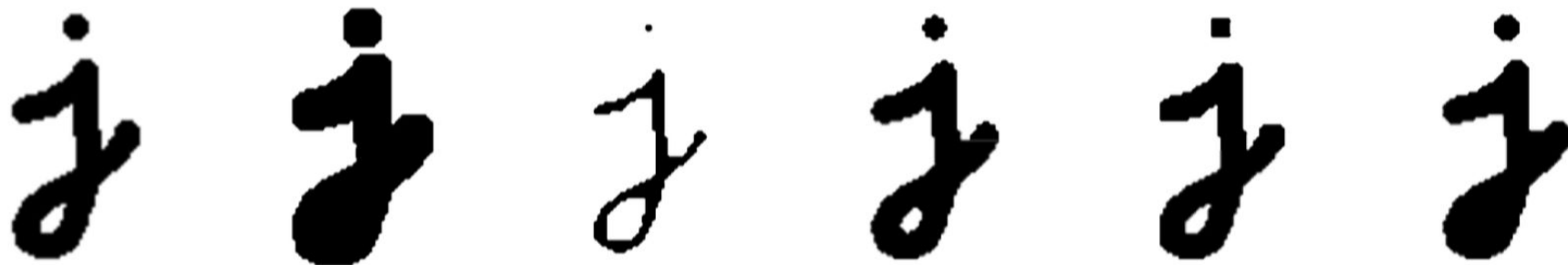
f - исходное изображение

t - значение порога

Морфологические преобразования

- применяются к бинаризованным изображениям
- изменяют форму объекта бинаризованного изображения
- морфологическое преобразование позволяет убрать шум после бинаризации изображения

Морфологические преобразования



Морфологические преобразования

$$c = f \otimes s$$

f - исходное бинаризованное изображение

s - оператор морфологического преобразования, как правило, box filter размера 3x3

c - результат свертки бинаризованного изображения и оператора

Морфологические преобразования



(a)



(b)



(c)



(d)



(e)



(f)

a) исходное изображение

b) $\text{dilate}(f, s) = \theta(c, 1)$

c) $\text{erode}(f, s) = \theta(c, S)$

d) $\text{maj}(f, s) = \theta(c, S/2)$

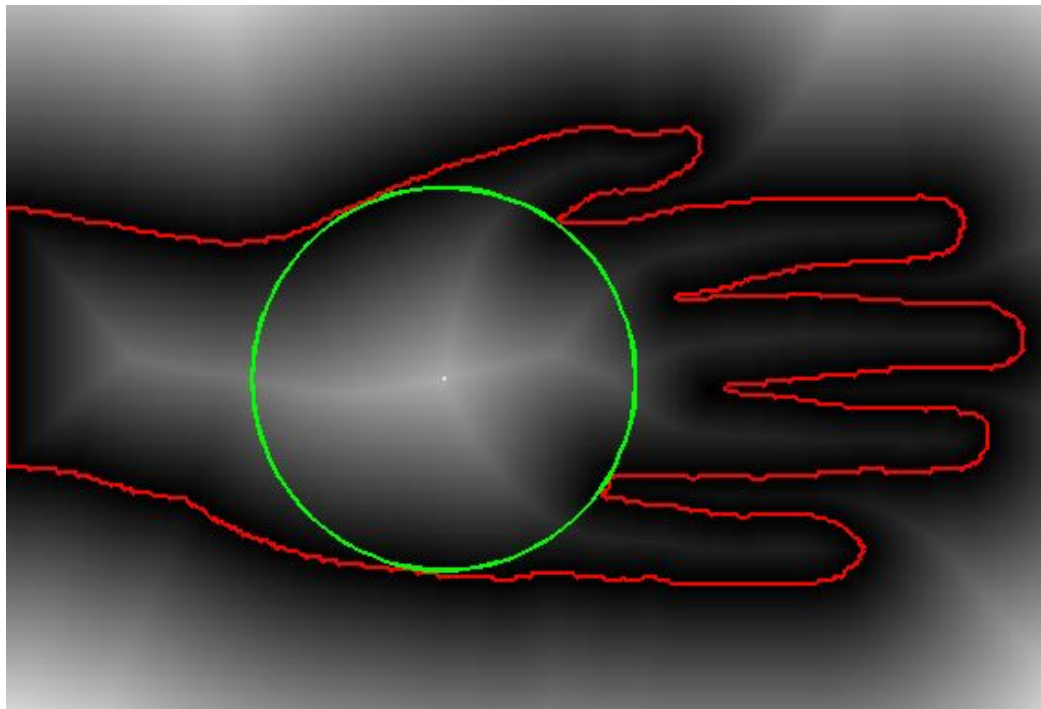
e) $\text{open}(f, s) = \text{dilate}(\text{erode}(f, s), s)$

f) $\text{close}(f, s) = \text{erode}(\text{dilate}(f, s), s)$

Distance Transform

- применяются к бинаризованным изображениям
- для каждой закрашенной точки изображения вычисляется расстояние до границы объекта
- применяется в задачах склейки изображений (stitching)

Distance Transform



Distance transform

$$d_1(k, l) = |k| + |l|$$

$$d_2(k, l) = \sqrt{k^2 + l^2}$$

$$D(i, j) = \min_{k, l: b(k, l) = 0} d(i - k, j - l)$$

$d(k, l)$ - мера расстояния

k, l - расстояние до границы объекта по координатам x и y соответственно

$D(i, j)$ - результат применения операции distance transform в точке (i, j)

Резюме

- изображения в цифровом виде представляются в виде матриц
- каждый пиксель матрицы цветного изображения содержит несколько значения
- значения пикселя зависят от пространства цветов RGB, HSV, HSL

Резюме

- Сжатие изображений может быть как с потерями в качестве, так и без потерь
- В случае сжатия с потерями размер цветного изображения может быть уменьшен в 10 - 20 раз

Резюме

- фильтрация изображений осуществляется с помощью сверточной операции
- фильтр Гаусса усредняет значение пикселя и размывает изображение
- оператор Лапласа вычисляет производную и выделяет грани на изображении
- медианный фильтр позволяет убрать шум на изображении

Резюме

- если фильтр обладает свойством сепарабельности, то время вычисления свертки можно сократить с K^2 до $2K$
- фильтры бывают как линейные, так и нелинейные
- примеры нелинейных фильтров: медианный фильтр и билатеральный фильтр

Резюме

- бинаризация изображений осуществляется сравнением значения в точке с заданным порогом
- для бинаризованных изображений доступны морфологические преобразования и преобразование distance transform

Полезные материалы

- [Digital Image Processing](#)
- [Computer Vision: Algorithms and Applications \(Chapter 2 and 3\)](#)
- [Frequently Asked Questions about Color](#)
- [HSL and HSV](#)
- [Image File Formats](#)
- [Miscellaneous Image Transformations](#)