

Лекция 1

Базовая теория

План курса

- 1. Классическое компьютерное зрение (лекции 1-3)
 - a. Базовая теория
 - b. Выделения признаков изображения и поиск
 - c. Сегментация и детекция

План курса

- 2. Применение нейросетей для задач CV (лекции 4-7)
 - a. Принцип работы нейронных сетей
 - b. Нейронные сети для обработки изображений
 - c. Особенности обучения нейронных сетей
 - d. Сегментация и детекция с помощью нейронных сетей

План курса

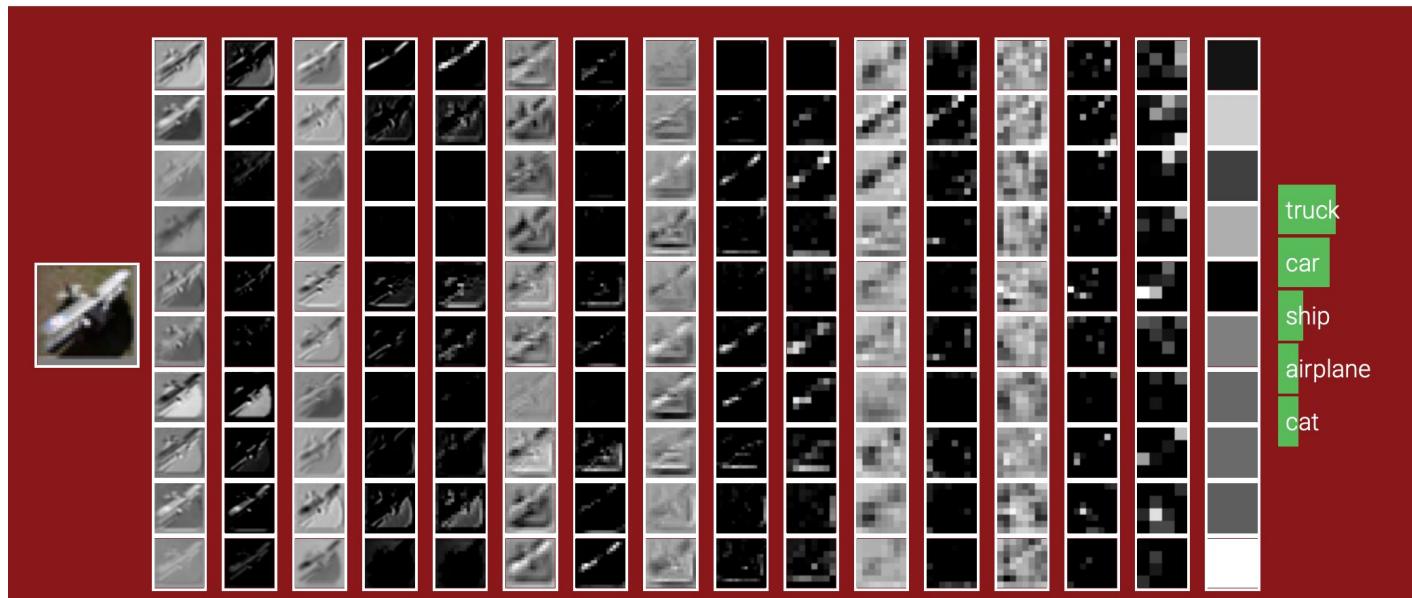
- 2. Применение нейросетей для задач CV (лекции 8-11)
 - a. Задача metric learning. Распознавание лиц.
 - b. Читаем вместе статью с конференции
 - c. Латентное представление изображений. Автоэнкодеры.
 - d. Генерация изображений. GAN, VAE.
 - e. Рекуррентные сети для изображений. Генерация описания по изображению.

Рекомендуемые материалы

CS231n: Convolutional Neural Networks for Visual Recognition

Spring 2019

Previous Years: [\[Winter 2015\]](#) [\[Winter 2016\]](#) [\[Spring 2017\]](#) [\[Spring 2018\]](#)



Рекомендуемые материалы

[ods.ai](#)

- #cv
- #deep_learning
- #article_essence
- #theory and practice

Рекомендуемые материалы

YouTube channels

- [ML Trainings](#)
- [Carnegie-Mellon University Deep Learning](#)
- [Kaggle](#)

Рекомендуемые материалы

Twitter

- [Yann LeCun](#)
- [Andrej Karpathy](#)
- [Pieter Abbeel](#)
- [Ruslan Salakhutdinov](#)
- [Ian Goodfellow](#)
- [Jeremy Howard](#)

Рекомендуемые материалы

The image shows a screenshot of a Twitter thread. The first tweet is from Andrej Karpathy (@karpathy) at 11:03 PM · Sep 20, 2019. It reads: "We see more significant improvements from training data distribution search (data splits + oversampling factor ratios) than neural architecture search. The latter is so overrated :)" This tweet has 317 Retweets and 1.7K Likes. Below it is a reply from Andrej Karpathy (@karpathy) at Sep 20, replying to @karpathy: "(likely an artifact of most of academia focused on finding models conditioned on standard datasets)" with 11 replies, 16 retweets, and 253 likes. The third tweet in the thread is from Aderemi Adeola (@21st_title) at Sep 20, replying to @karpathy: "Could you explain how to go about this(training data distribution search) or point to a paper that does?" with 5 replies, 1 retweet, and 15 likes. The final tweet in the thread is from Andrej Karpathy (@karpathy) at Sep 20: "There's no paper. That's exactly the problem." with 17 replies, 3 retweets, and 111 likes.

Thread

Andrej Karpathy ✅ @karpathy

We see more significant improvements from training data distribution search (data splits + oversampling factor ratios) than neural architecture search. The latter is so overrated :)

11:03 PM · Sep 20, 2019 · Twitter Web App

317 Retweets 1.7K Likes

Andrej Karpathy ✅ @karpathy · Sep 20
Replying to @karpathy
(likely an artifact of most of academia focused on finding models conditioned on standard datasets)

11 16 253

Aderemi Adeola. @21st_title · Sep 20
Replying to @karpathy
Could you explain how to go about this(training data distribution search) or point to a paper that does?

5 1 15

Andrej Karpathy ✅ @karpathy · Sep 20
There's no paper. That's exactly the problem.

17 3 111

Какие вопросы решает CV

1. Что это за объект?
2. Где находится объект?
3. Кто на изображении/видео?
4. Как далеко?
5. Что делает?
6. Какой формы объект?

Пример. Применение фильтров для обработки изображений

noisy lena



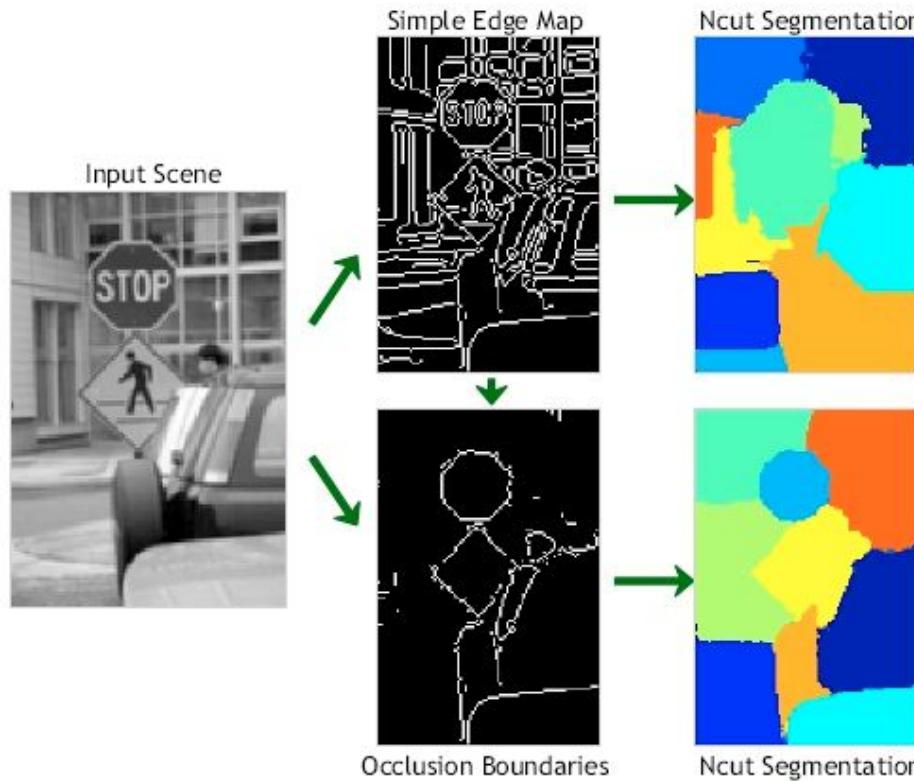
Gaussian filter



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)

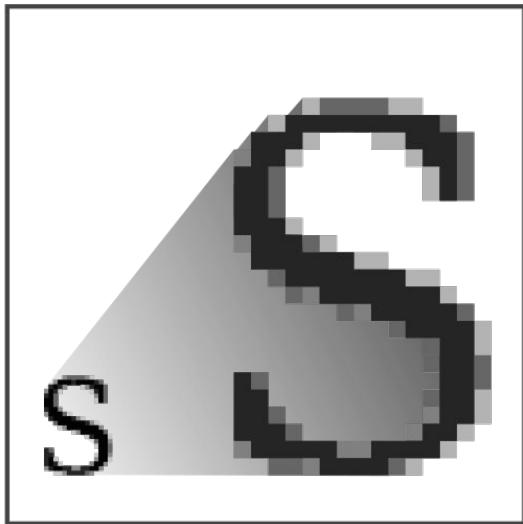
размер изображения ограничен числом пикселей

Вектор

описание изображения на уровне фигур и их свойств

размер изображения может быть произвольным

Растр vs Вектор

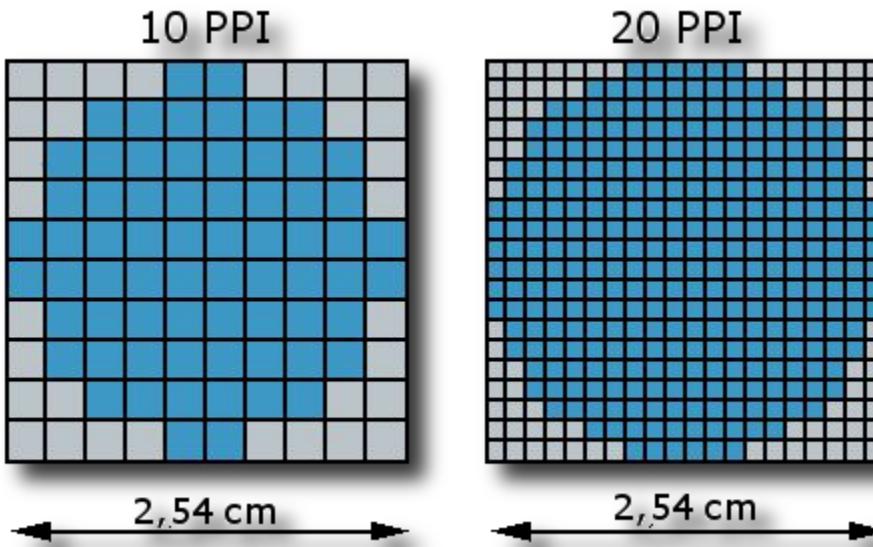


РАСТР
.jpeg .gif .png



ВЕКТОР
.svg

Разрешение



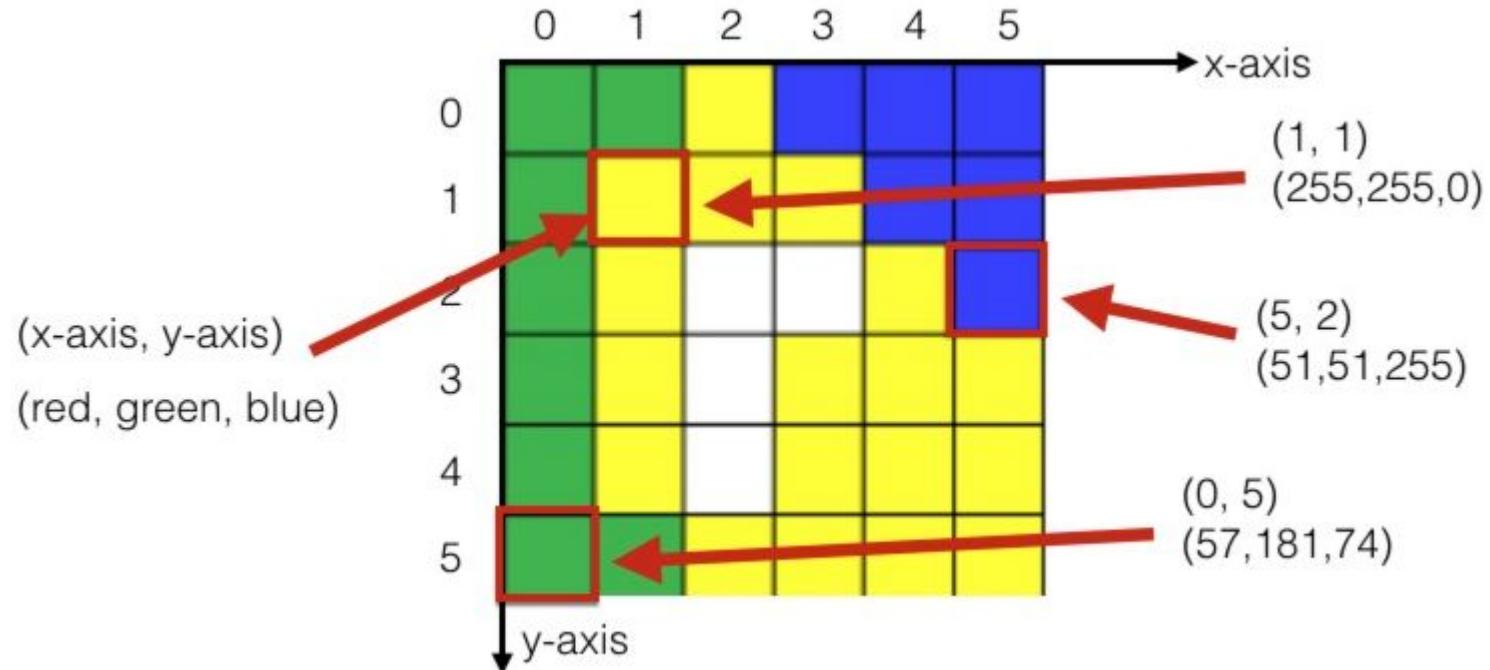
Каналы и динамический диапазон

- 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



Цвет - часть ЭМ волны

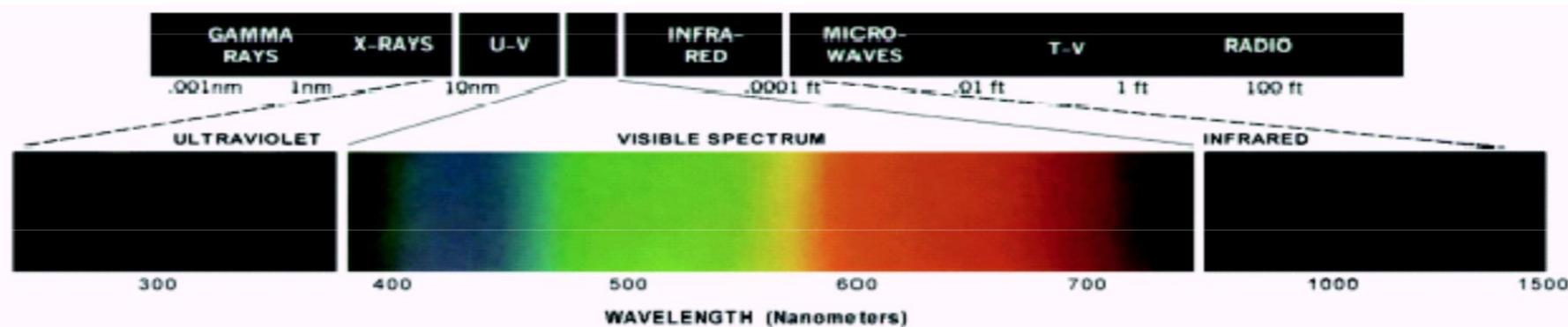


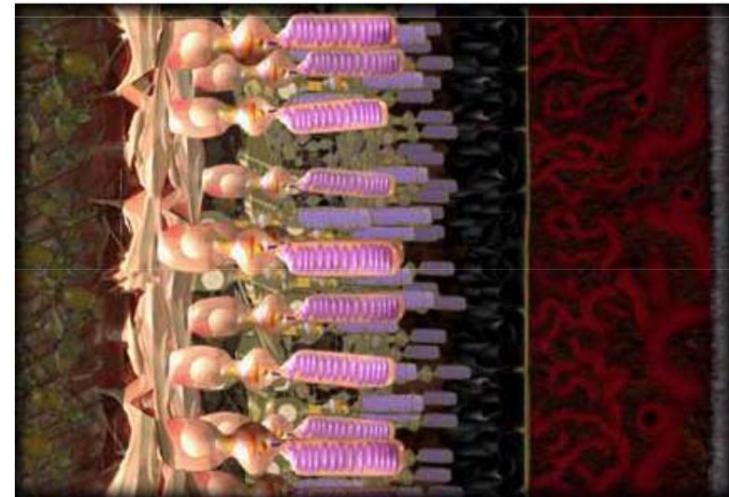
FIGURE 6.2 Wavelengths comprising the visible range of the electromagnetic spectrum.
(Courtesy of the General Electric Co., Lamp Business Division.)

Illuminating and Reflecting Light

- Излучающие источники (первичный свет):
 - Излучают свет (солнце, монитор)
 - Воспринимаемый цвет зависит от частоты ЭМ волны
 - Аддитивное правило: $R+G+B=White$
- Отражающие источники (вторичный свет):
 - Отражают поступающий свет (краска, одежда)
 - Воспринимаемый цвет зависит от отраженного цвета ($= emitted freq - absorbed freq$)
 - Вычитающие правила: $R+G+B=Black$

Человеческое восприятие цвета

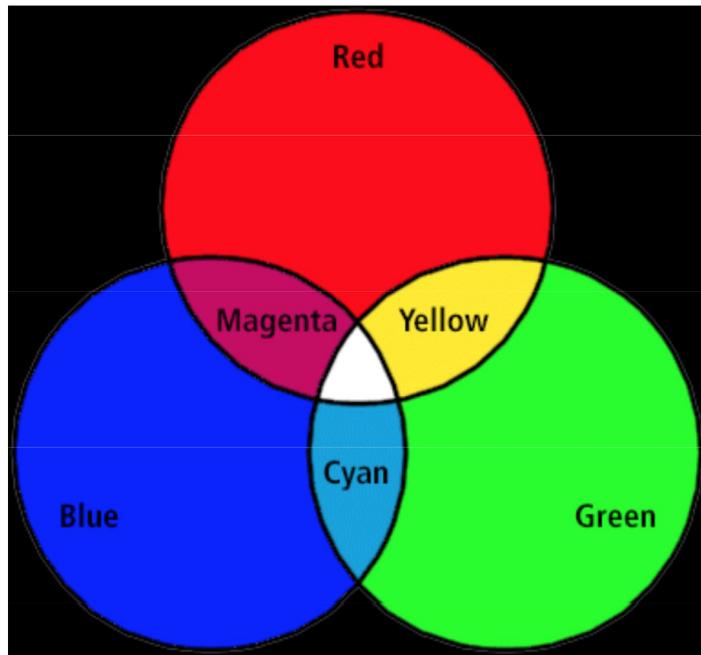
- Сетчатка содержит рецепторы
 - Колбочки
 - Дневной свет, оттенок цвета
 - R, G, B колбочки
 - Палочки
 - Ночной свет, яркость
- Чувствительность к свету
 - Luminance (brightness)
 - яркость
 - Chrominance
 - Hue (color tone)
 - Оттенок цвета, доминирующая длина волны
 - Saturation (color purity)
 - Относительная насыщенность цвета или количество белого света смешанного с оттенком



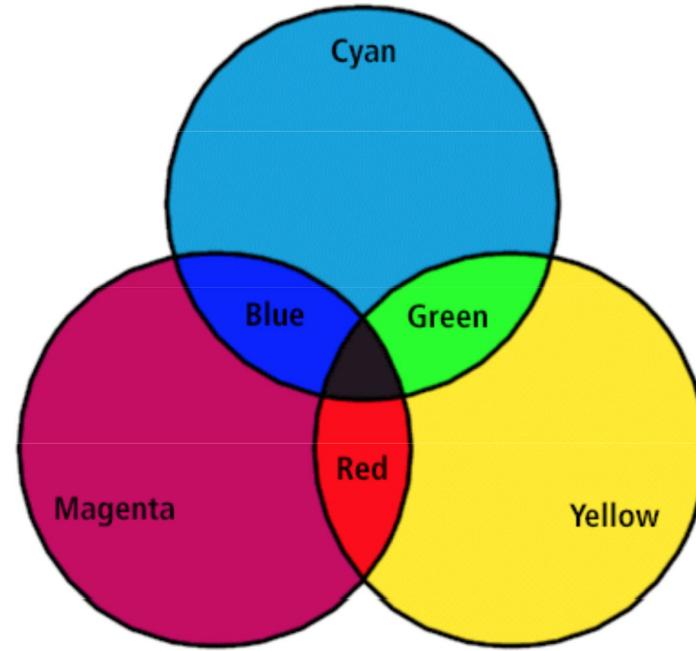
Трихроматическая теория цвета

- Любой цвет может быть получен смешиванием трех основных цветов в правильном отношении
- Основные цвета для освещивающих источников:
 - R, G, B
 - $R + G + B = \text{white}$
- Основные цвета для отражающих источников:
 - Cyan, Magenta, Yellow (CMY)
 - В принтерах CMYK (=CMY + black)
 - $R + G + B = \text{black}$

RGB vs CMY



Magenta = Red + Blue
Cyan = Blue + Green
Yellow = Green + Red



Magenta = White - Green
Cyan = White - Red
Yellow = White - Blue

Цветное изображение



Red



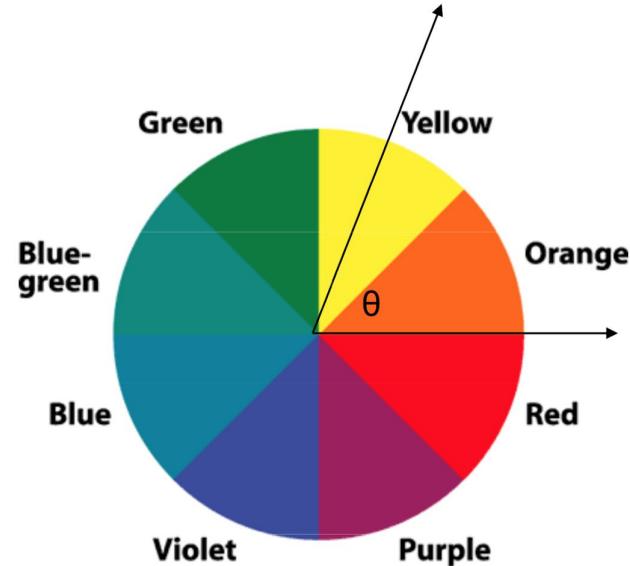
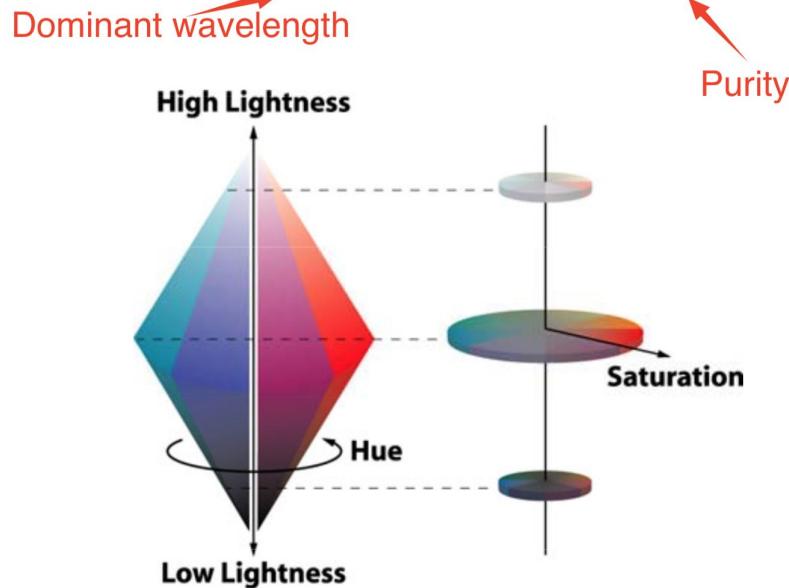
Green



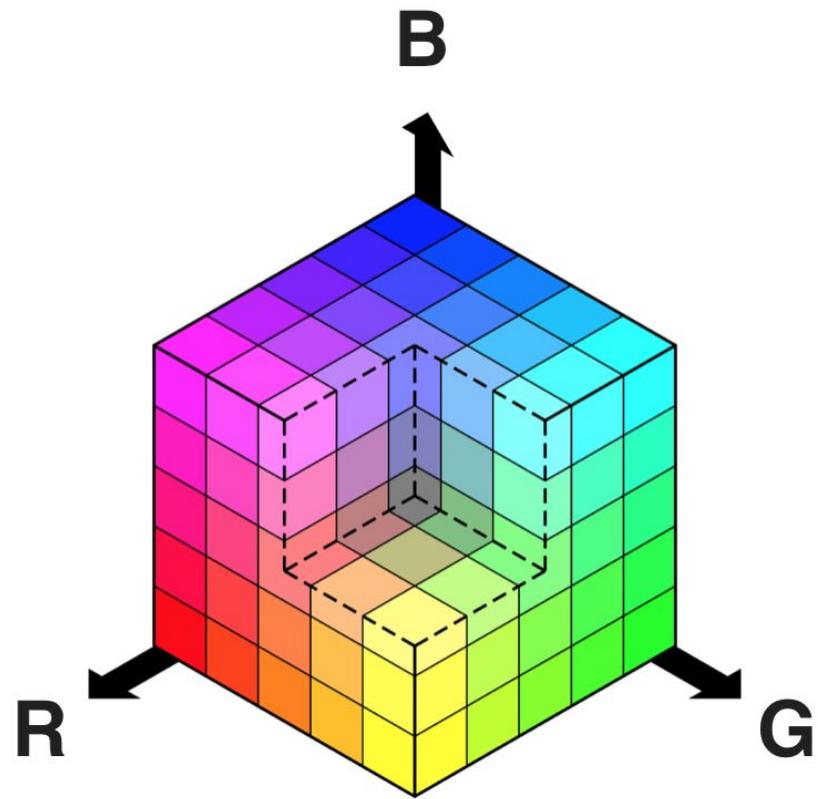
Blue

Три атрибута цвета

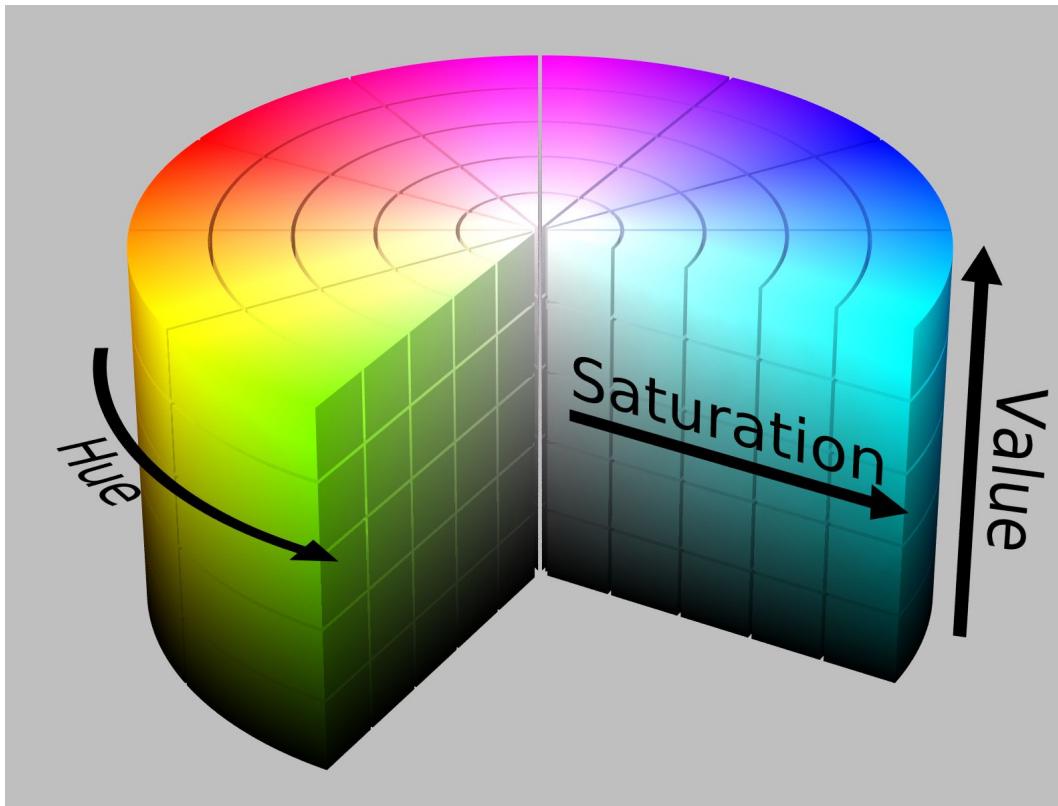
- Luminance (brightness) ← Intensity
- Chrominance
 - Hue (color tone) and Saturation



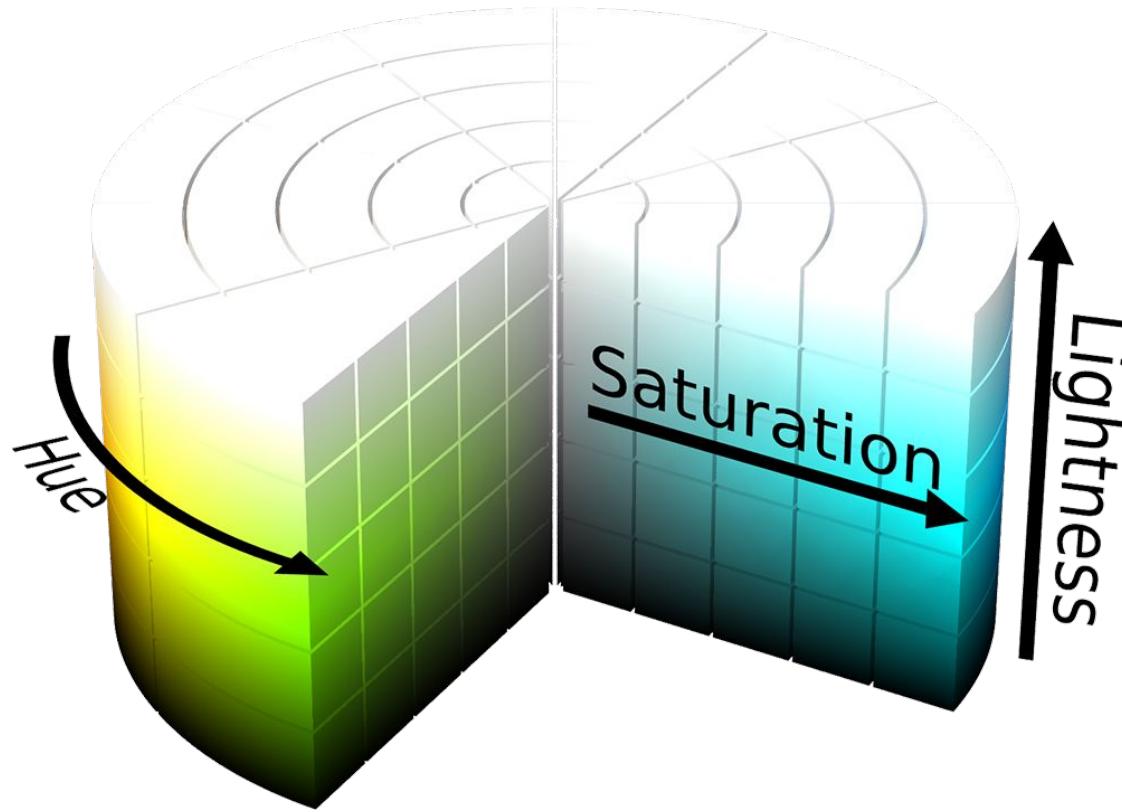
RGB



HSV



HSL



Модель цвета

- Определите 3 первичных или вторичных цвета
 - RGB
 - CMY
- Определите luminance и chrominance
 - HSB(=HSV) or HSL = hue, saturation, value
 - YIQ~YUV (аналоговое ТВ)
 - YCbCr (цифровое ТВ)
- Выберите амплитуду
 - 8 битов на компоненту или 24 бита на пиксель

RGB <-> CMY

$$\begin{bmatrix} C \\ M \\ Y \end{bmatrix} = \begin{bmatrix} 1 \\ 1 \\ 1 \end{bmatrix} - \begin{bmatrix} R \\ G \\ B \end{bmatrix}, \quad \begin{bmatrix} R \\ G \\ B \end{bmatrix} = \begin{bmatrix} 1 \\ 1 \\ 1 \end{bmatrix} - \begin{bmatrix} C \\ M \\ Y \end{bmatrix}.$$

RGB -> HSL

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

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

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

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

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

RGB -> HSV

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

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

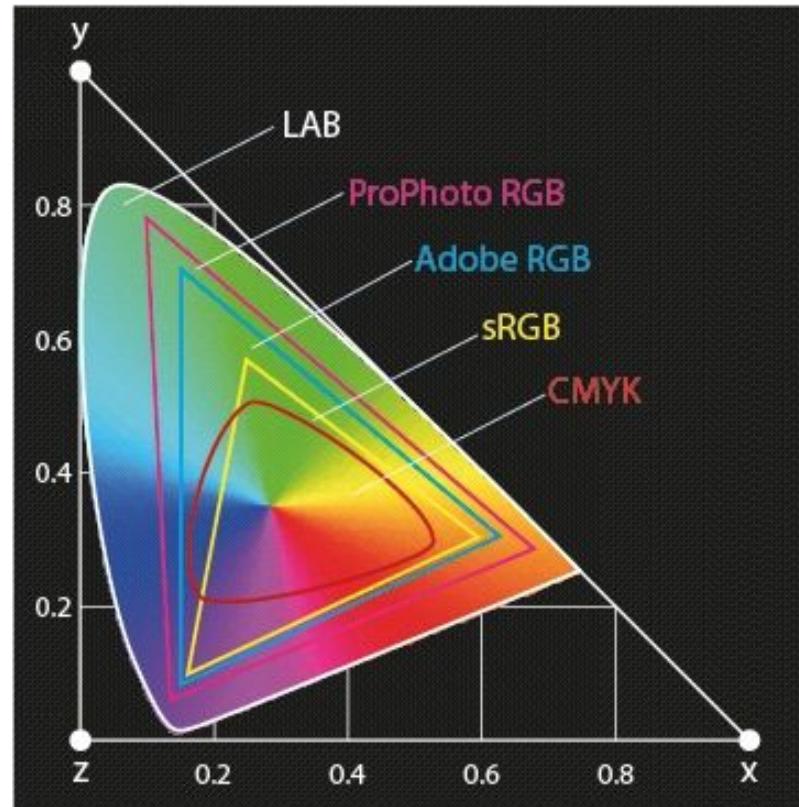
$$H \leftarrow \begin{cases} 60(G - B)/(V - \min(R, G, B)) & \text{if } V = R \\ 120 + 60(B - R)/(V - \min(R, G, B)) & \text{if } V = G \\ 240 + 60(R - G)/(V - \min(R, G, B)) & \text{if } V = B \end{cases}$$

Квантизация света

24 bits -> 8 bits



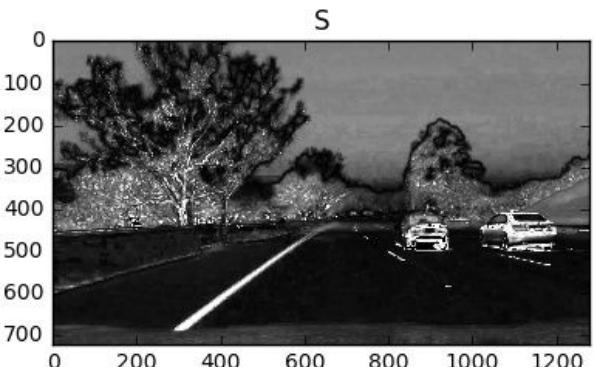
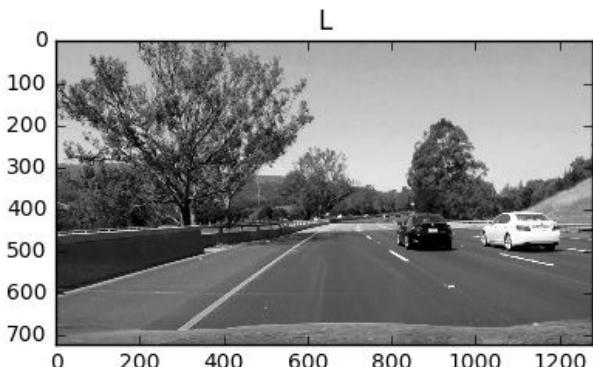
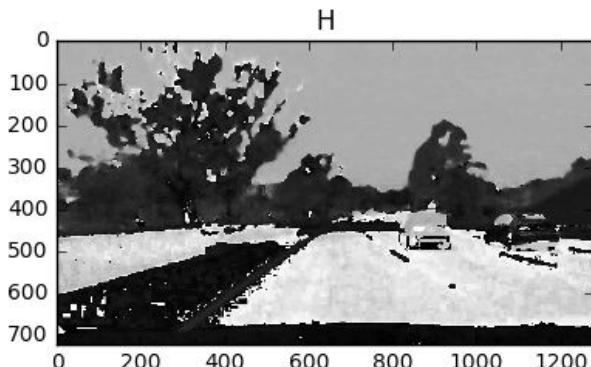
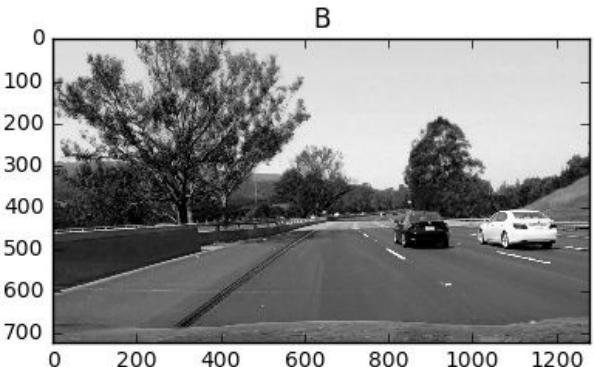
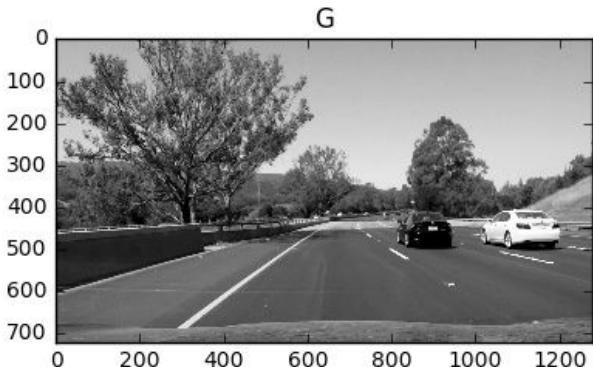
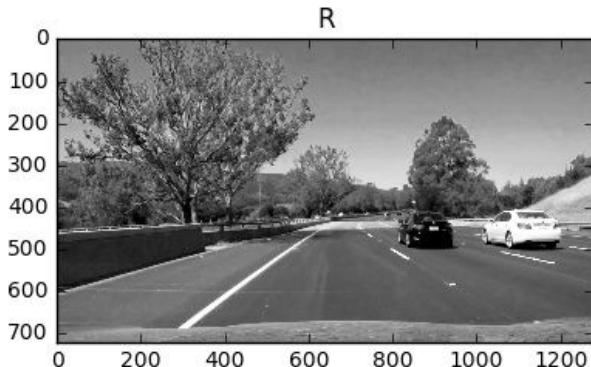
CMYK vs RGB



HSL vs RGB



HSL vs RGB



1.2 Растроевые форматы. Сжатие

Сжатие изображений

- Матричное представление RGB требовательно к ресурсам памяти
- Каждый пиксель занимает $3 \times 8\text{bit} = 24\text{bit}$ памяти
- Изображение 1024x768 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

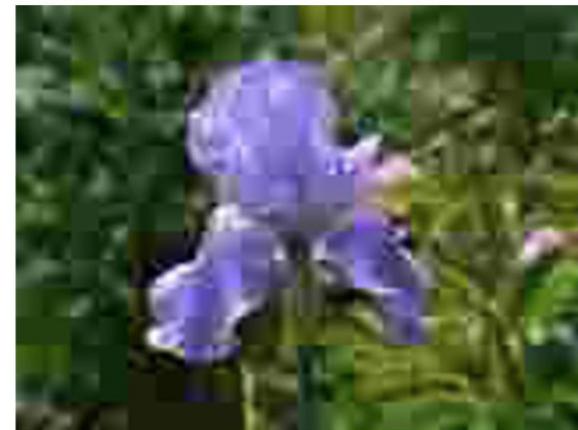
Алгоритмы сжатия изображений

- 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

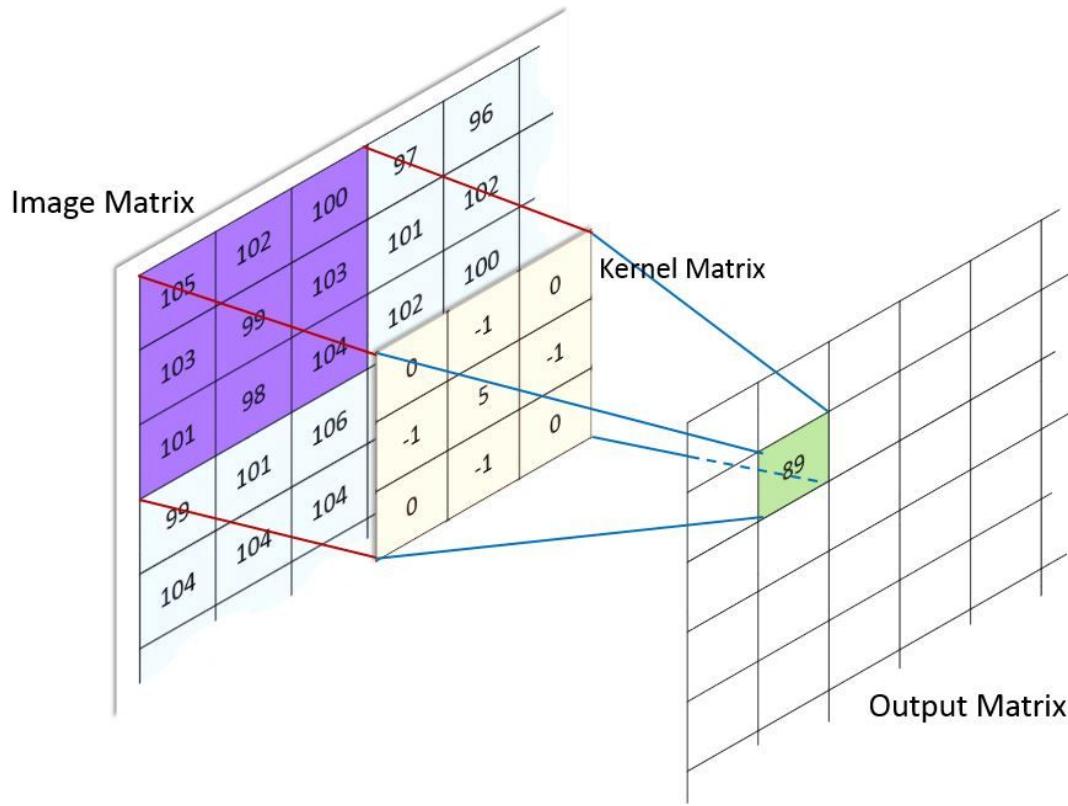
- PNG Portable Network Graphic
 - сжатие без потерь
 - палитра цветов изображения хранятся в таблице
 - для каждого пикселя указывается индекс цвета из палитры
 - ориентировочный коэффициент сжатия цветного изображения: 2,5:1

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 пикселя 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

=

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

$h(x,y)$

$g(x,y)$

Пример фильтра: идентичное отображение



*

0	0	0
0	1	0
0	0	0

=



Original

Identical image

Пример фильтра: смещение



*

0	0	0
1	0	0
0	0	0

=



Original

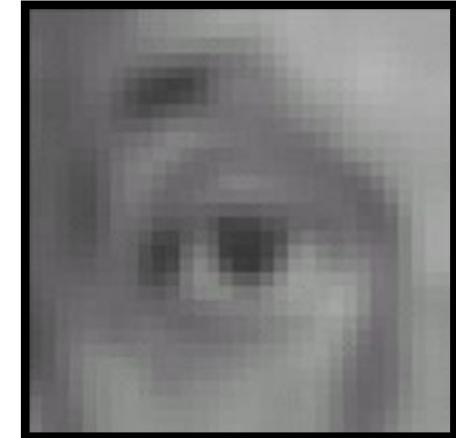
Shifted left
By 1 pixel

Пример фильтра: размытие



$$\frac{1}{9}$$

1	1	1
1	1	1
1	1	1

 $=$ 

Original

Blur (with a mean filter)

Пример фильтра: выделение границ (sharpen)



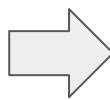
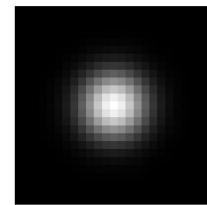
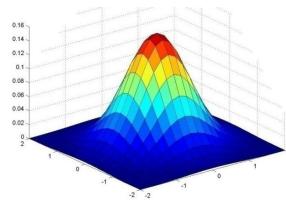
Original

$$\ast \left(\begin{array}{ccc} 0 & 0 & 0 \\ 0 & 2 & 0 \\ 0 & 0 & 0 \end{array} - \frac{1}{9} \begin{array}{ccc} 1 & 1 & 1 \\ 1 & 1 & 1 \\ 1 & 1 & 1 \end{array} \right) =$$



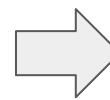
Sharpening filter
(accentuates edges)

Размытие - фильтр Гаусса

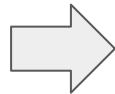
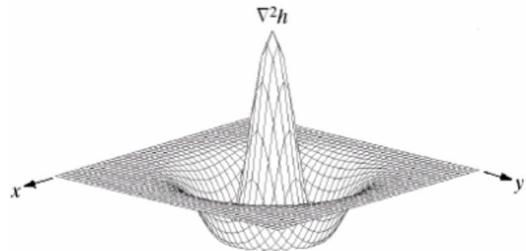


$$\frac{1}{273}$$

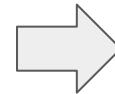
1	4	7	4	1
4	16	26	16	4
7	26	41	26	7
4	16	26	16	4
1	4	7	4	1



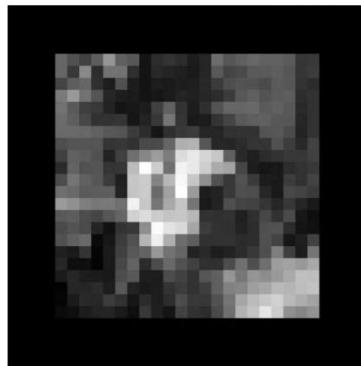
Выделение границ - оператор Лапласа



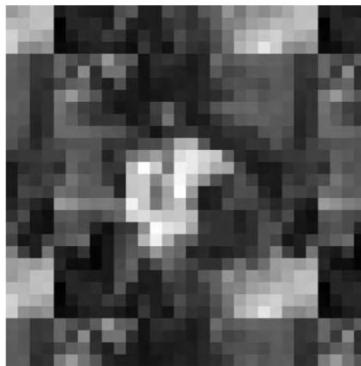
-1	-1	-1
-1	8	-1
-1	-1	-1



Отступы (Padding)



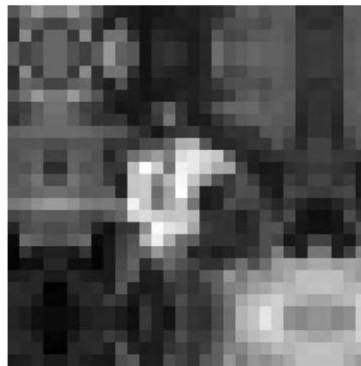
zero



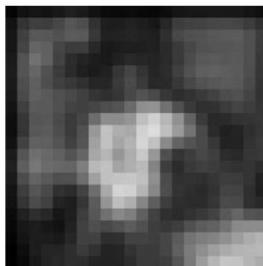
wrap



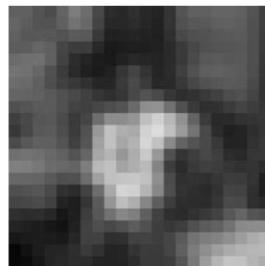
clamp



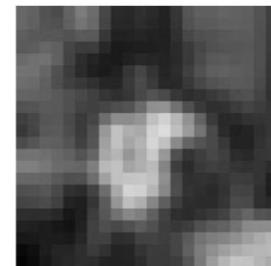
mirror



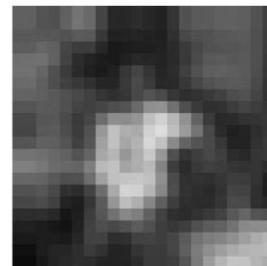
blurred zero



normalized zero



blurred clamp



blurred mirror

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



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
theta - угол наклона градиента в полярной системе координат

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)

$$\begin{array}{|c|c|c|} \hline 1 & 2 & 1 \\ \hline 2 & 4 & 2 \\ \hline 1 & 2 & 1 \\ \hline \end{array} = ?$$

Сепарабельные фильтры (separable)

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

Сепарабельные фильтры (separable)

$$\begin{array}{|c|c|c|} \hline 1 & 2 & 1 \\ \hline 2 & 4 & 2 \\ \hline 1 & 2 & 1 \\ \hline \end{array} * \begin{array}{|c|c|c|} \hline 2 & 3 & 3 \\ \hline 3 & 5 & 5 \\ \hline 4 & 4 & 6 \\ \hline \end{array} = \begin{array}{l} 2 + 6 + 3 = 11 \\ 6 + 20 + 10 = 36 \\ 4 + 8 + 6 = 18 \\ \hline 65 \end{array}$$

$$\begin{array}{|c|c|c|} \hline 1 & 2 & 1 \\ \hline 2 & 4 & 2 \\ \hline 1 & 2 & 1 \\ \hline \end{array} = \begin{array}{|c|} \hline 1 \\ \hline 2 \\ \hline 1 \\ \hline \end{array} \times \begin{array}{|c|c|c|} \hline 1 & 2 & 1 \\ \hline \end{array}$$

$$\begin{array}{|c|c|c|} \hline 1 & 2 & 1 \\ \hline \end{array} * \begin{array}{|c|c|c|} \hline 2 & 3 & 3 \\ \hline 3 & 5 & 5 \\ \hline 4 & 4 & 6 \\ \hline \end{array} = \begin{array}{|c|c|c|} \hline 11 & & \\ \hline 18 & & \\ \hline 18 & & \\ \hline \end{array}$$

$$\begin{array}{|c|} \hline 1 \\ \hline 2 \\ \hline 1 \\ \hline \end{array} * \begin{array}{|c|c|c|} \hline 11 & & \\ \hline 18 & & \\ \hline 18 & & \\ \hline \end{array} = \begin{array}{|c|c|c|} \hline & & \\ \hline & & \\ \hline & & \\ \hline \end{array}$$

$$= 65$$

Сепарабельные фильтры (separable)

$$\frac{1}{K^2} \begin{bmatrix} 1 & 1 & \cdots & 1 \\ 1 & 1 & \cdots & 1 \\ \vdots & \vdots & 1 & \vdots \\ 1 & 1 & \cdots & 1 \end{bmatrix}$$

$$\frac{1}{16} \begin{bmatrix} 1 & 2 & 1 \\ 2 & 4 & 2 \\ 1 & 2 & 1 \end{bmatrix}$$

$$\frac{1}{256} \begin{bmatrix} 1 & 4 & 6 & 4 & 1 \\ 4 & 16 & 24 & 16 & 4 \\ 6 & 24 & 36 & 24 & 6 \\ 4 & 16 & 24 & 16 & 4 \\ 1 & 4 & 6 & 4 & 1 \end{bmatrix}$$

$$\frac{1}{8} \begin{bmatrix} -1 & 0 & 1 \\ -2 & 0 & 2 \\ -1 & 0 & 1 \end{bmatrix}$$

$$\frac{1}{4} \begin{bmatrix} 1 & -2 & 1 \\ -2 & 4 & -2 \\ 1 & -2 & 1 \end{bmatrix}$$

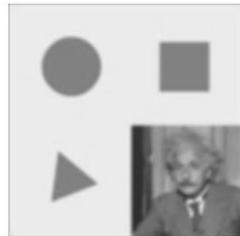
$$\frac{1}{K} \begin{bmatrix} 1 & 1 & \cdots & 1 \end{bmatrix}$$

$$\frac{1}{4} \begin{bmatrix} 1 & 2 & 1 \end{bmatrix}$$

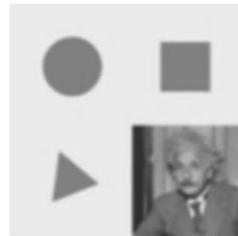
$$\frac{1}{16} \begin{bmatrix} 1 & 4 & 6 & 4 & 1 \end{bmatrix}$$

$$\frac{1}{2} \begin{bmatrix} -1 & 0 & 1 \end{bmatrix}$$

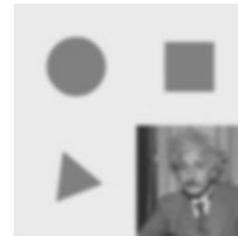
$$\frac{1}{2} \begin{bmatrix} 1 & -2 & 1 \end{bmatrix}$$



(a) box, $K = 5$



(b) bilinear



(c) “Gaussian”



(d) Sobel



(e) corner

Сепарабельные фильтры (separable)

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

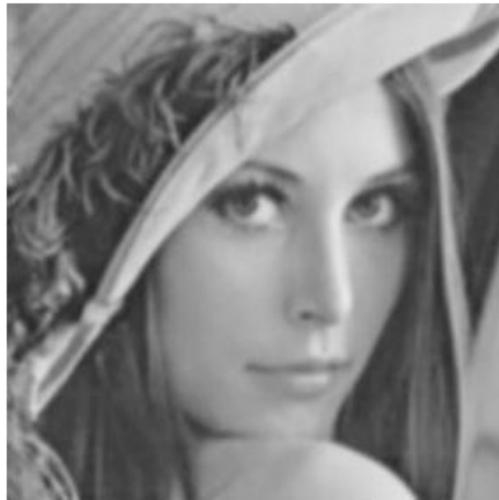
- **src** - исходное изображение
- **ddepth** - тип данных выходного изображения
- **kernelX** - горизонтальная компонента ядра
- **kernelY** - вертикальная компонента ядра

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

sigma - 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. Exploring the many waterways deep enough to enjoy the beauty and fun of the ocean, river, stream, or lake is a great way to make campsite trips with a family a real adventure. Just look at our collection of durable models. Whether you're interested in options like screens, private sections, or even a porch, we have the features that these rooms and your family will need. The unique designs, elegant colors, and spacious layouts are designed to make a difference in what you'll feel when you stay at the campsite with us and will make the vacation time for the family a success.

Camping can be Fun and Adventurous in Our Sturdy Tents!		
Campout is an exciting and adventurous activity for the entire family. Exploring the scenic views there are moments in enjoying the sun, and lots of fun, new camping experience to make memorable with a right eight all-weather tent from our collection of domestic models. There are a variety of options and colors, practical comfort and privacy. These elements for these houses and houses are MIL certified. Our unique high quality materials and technology. Other features include the door-tilted or side windows. You may stay in the most secure sites and need and provide a comfortable and safe family vacation.		
Our tents	SLX Series	SLX+Dome Series
• Standard	2 person 3 person 4 person 5 person 6 person 7 person 8 person 9 person 10 person 11 person 12 person 13 person 14 person 15 person 16 person 17 person 18 person 19 person 20 person	2 person 3 person 4 person 5 person 6 person 7 person 8 person 9 person 10 person 11 person 12 person 13 person 14 person 15 person 16 person 17 person 18 person 19 person 20 person
• Customized		
• Photo of the		
• Special Offer		
• Guarantee		
• Get it soon		

Camping can be Fun and Adventurous in Our Sturdy Tents!

(Caption) An exciting and adventurous activity for the entire family. Exploring the woods makes camping one of the most unique outdoor activities you can imagine. It's also extremely safe if the proper equipment and techniques are used. Our selection of durable models, from tents to pop-ups, will provide you with a comfortable and protective place for the outdoors. Whether you're looking for a simple tent or a more elaborate one, we have what you need. Our friendly sales staff and expert advice are available via the phone, fax, or via the Internet. The day after the rain comes again, you'll find yourself wanting to get back outside.

Бинаризация изображения

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

theta - 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

ж

ж

ж

ж

ж

ж

Морфологические преобразования

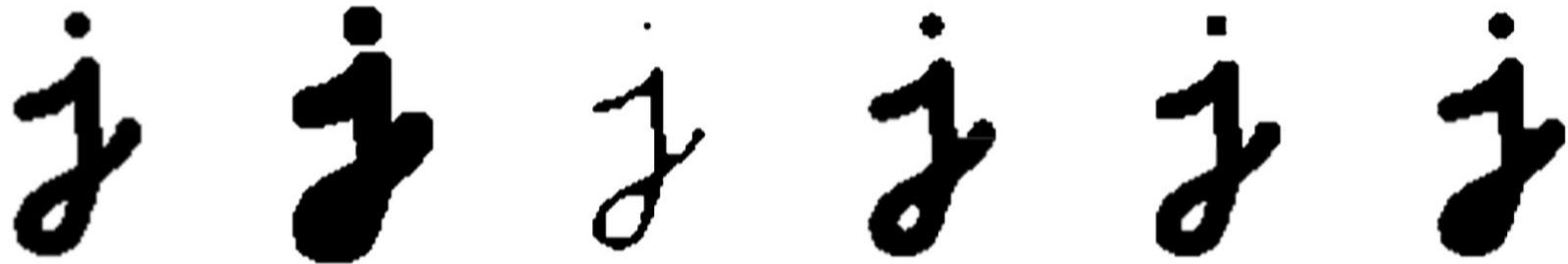
$$c = f \otimes s$$

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

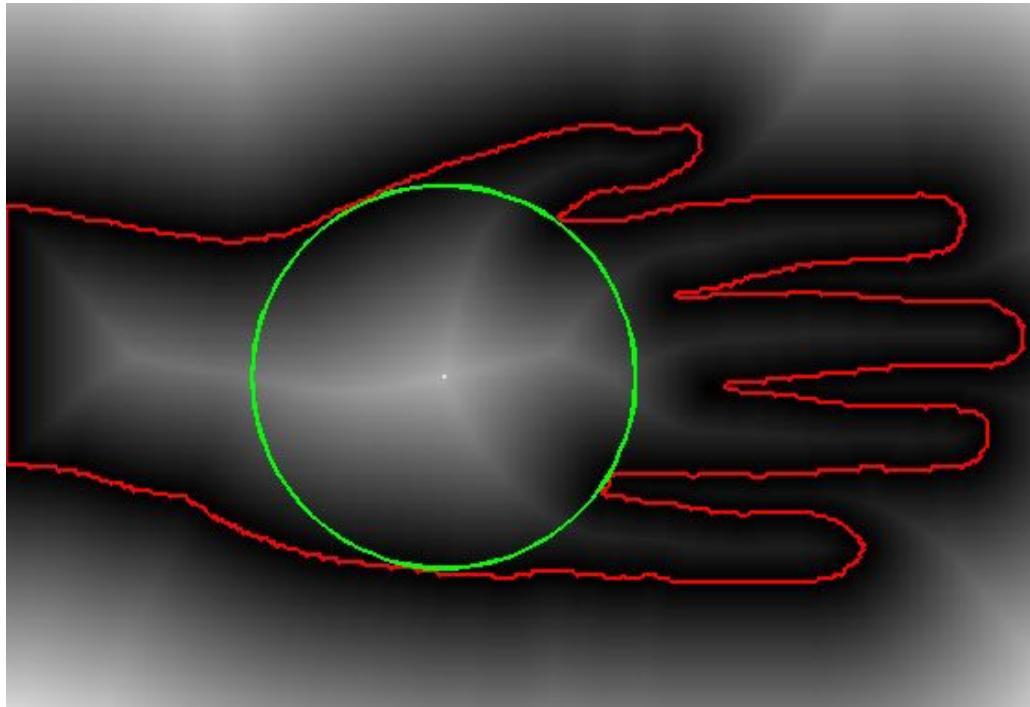
e) $\text{open}(f, s) = \text{dilate}(\text{erode}(f, s), s)$

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](#)