



НАЦИОНАЛЬНЫЙ ИССЛЕДОВАТЕЛЬСКИЙ
УНИВЕРСИТЕТ

Цифровая обработка изображений

Савченко А.В.

Д.т.н., проф. Каф. Информационные системы и технологии
avsavchenko@hse.ru

- 1. Информация о лекторе**
- 2. Основные задачи компьютерного зрения**
- 3. Фильтрация изображений**
- 4. Выделение краев (Edge detection)**
- 5. Выделение ключевых точек**
- 6. Дескрипторы**

Информация о лекторе

Персональная страница НИУ ВШЭ: <http://www.hse.ru/staff/avsavchenko>

Савченко Андрей Владимирович

Старший научный сотрудник: [НИУ ВШЭ в Нижнем Новгороде](#) / [Лаборатория алгоритмов и технологий анализа сетевых структур](#)

Профессор: [НИУ ВШЭ в Нижнем Новгороде](#) / [Факультет информатики, математики и компьютерных наук](#) / [Кафедра информационных систем и технологий](#)

Начал работать в НИУ ВШЭ в 2008 году.

Научно-педагогический стаж: 10 лет.

[🏠 Домашняя страница](#) [Преподавание](#) [Публикации и исследования](#) [Опыт работы](#) [Прочее](#)

[В новостях](#)



Владение языками

английский

Контакты

Телефон:
8(831) 4169800

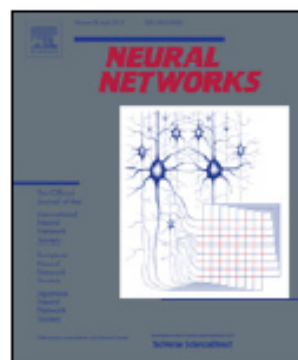
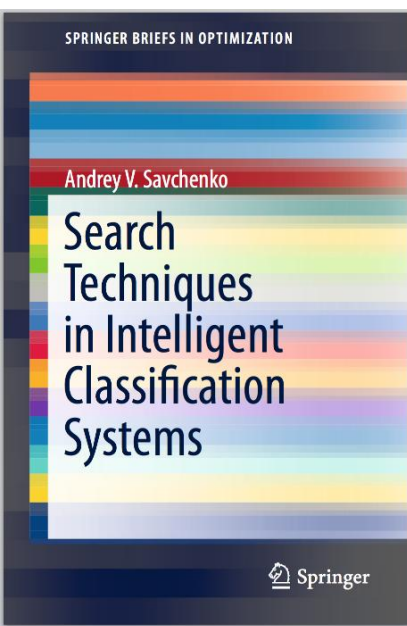
Электронная почта:
avsavchenko@hse.ru

Адрес: г. Нижний Новгород, Б.
Печерская ул., д. 25/12, каб. 301
Время присутствия: в
зависимости от расписания

Образование, учёные степени и учёные звания

- 2016 Доктор технических наук: ГОУ ВПО Нижегородский государственный технический университет им. Р. Е. Алексеева, специальность 05.13.01 «Системный анализ, управление и обработка информации», тема диссертации: Методы классификации аудиовизуальной информации на основе посегментного анализа однородности

- Распознавание лица по фотографиям/видео. Работа поддержана грантом Президента РФ для молодых докторов наук
- Распознавание речи в системах голосового управления. Проект вошел в число 10 лучших в рамках ФЦП «Исследования и разработки» (2007-2013)
- Большие данные. Эффективная классификация дескрипторов изображений и текстов (Пономаренко А.А.)
- Компьютерная лингвистика. Анализ текстов, opinion mining (Карпов Н.В.)



Deep Learning research at Steklov Institute, St. Petersburg, Russia

[Home](#) [Projects](#) [Publications](#) [Staff](#)

Staff



Sergey Nikolenko
Laboratory Head



Elena Tutubalina
Researcher



Valentin Malykh
Researcher



Anton Alekseev
Researcher



Andrey Savchenko
Senior Researcher



Alexandr Rassadin
Researcher



Ekaterina Arkhangelskaya
Researcher



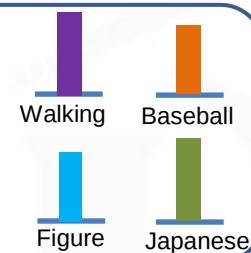
Ivan Grechikhin
Researcher

Проект User Preference Prediction in Visual Data

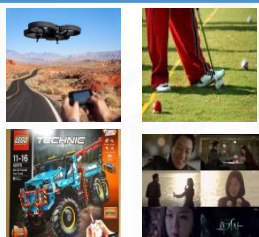
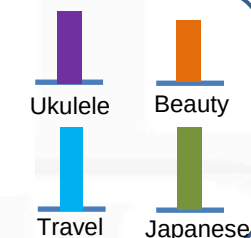
Генерация профиля пользователя



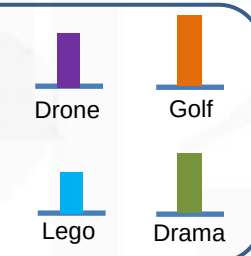
User 1



User 2



User 3



Основные задачи компьютерного зрения



<https://www.mathworks.com/help/wavelet/examples/denoising-signals-and-images.html>

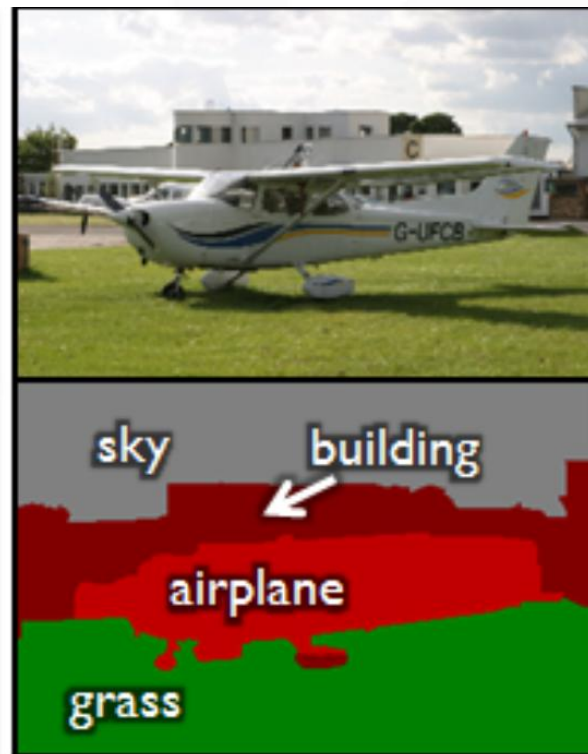


(a) Siberian husky



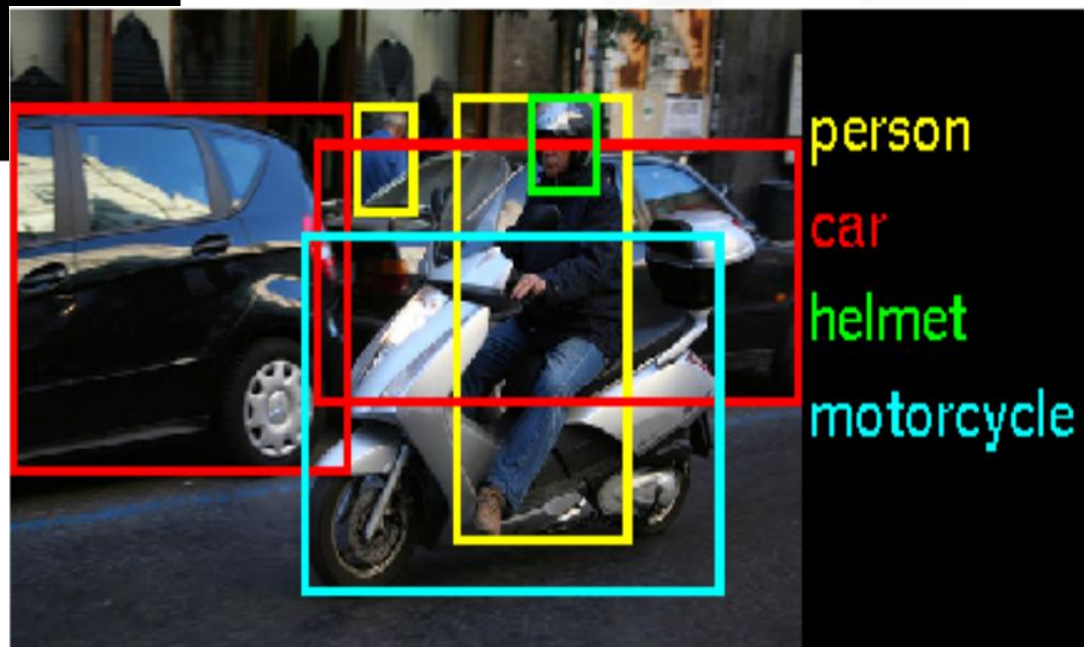
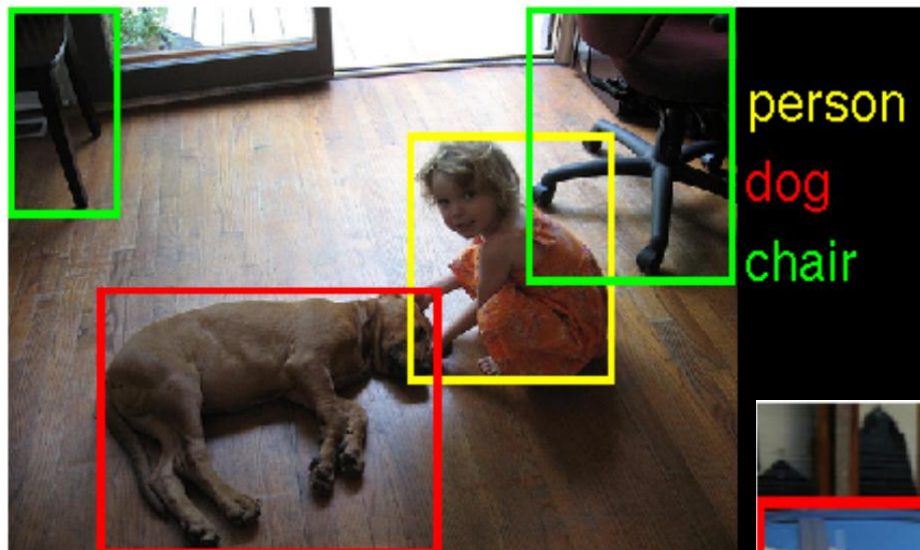
(b) Eskimo dog

Семантическая сегментация



object classes	building	grass	tree	cow	sheep	sky	airplane	water	face	car
bicycle	flower	sign	bird	book	chair	road	cat	dog	body	boat

Детектирование объектов на изображениях



Classification



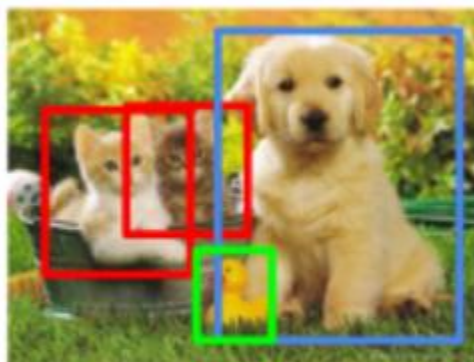
CAT

Classification + Localization



CAT

Object Detection



CAT, DOG, DUCK

Instance Segmentation



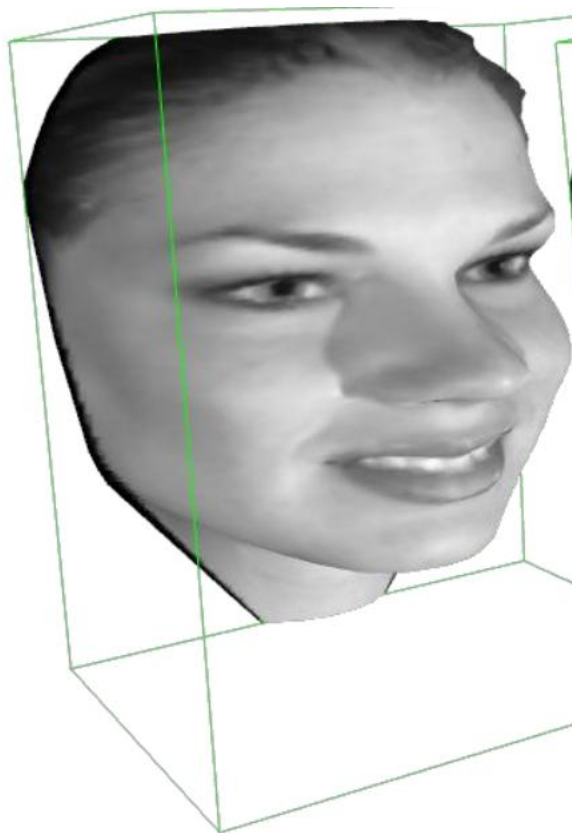
CAT, DOG, DUCK

Single object

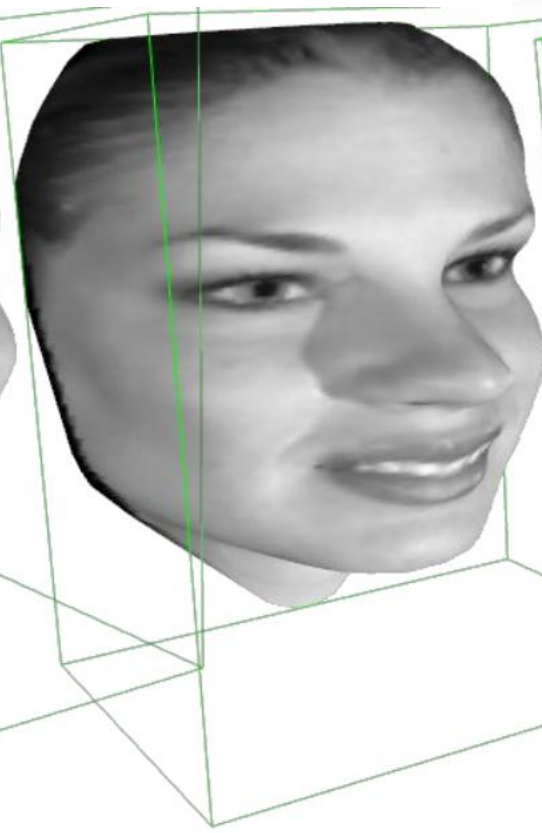
Multiple objects

https://leonardaraujosantos.gitbooks.io/artificial-intelligence/content/object_localization_and_detection.html

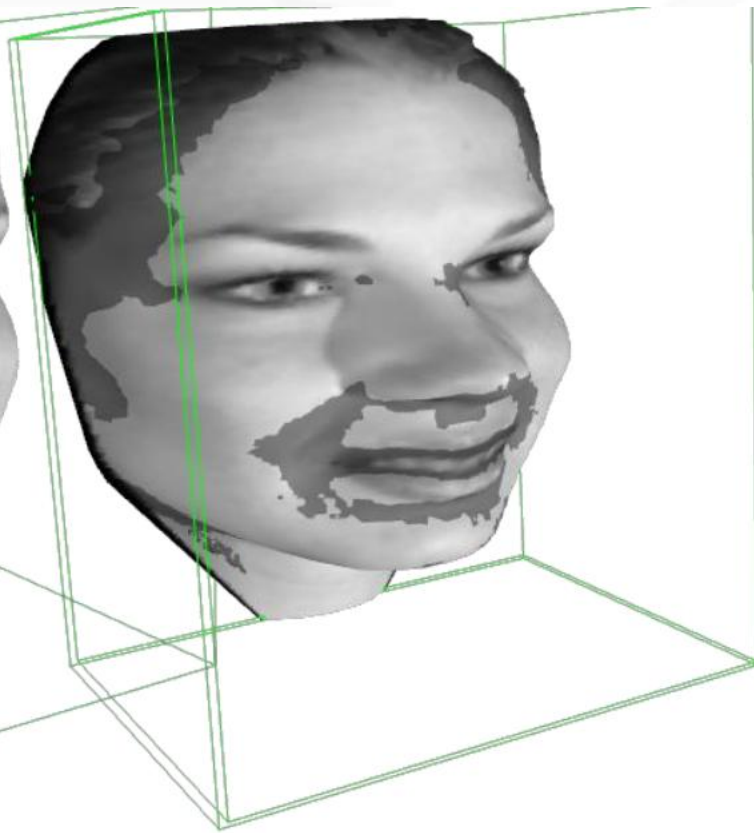
**Предсказание
нейросетевой модели**



**Оригинальная 3D
модель**



Ошибка



[Рассадин, 2017]

Текстовое описание изображений

Describes without errors



A person riding a motorcycle on a dirt road.



A group of young people playing a game of frisbee.



A herd of elephants walking across a dry grass field.

Describes with minor errors



Two dogs play in the grass.



Two hockey players are fighting over the puck.



A close up of a cat laying on a couch.

Somewhat related to the image



A skateboarder does a trick on a ramp.



A little girl in a pink hat is blowing bubbles.



A red motorcycle parked on the side of the road.

Unrelated to the image



A dog is jumping to catch a frisbee.



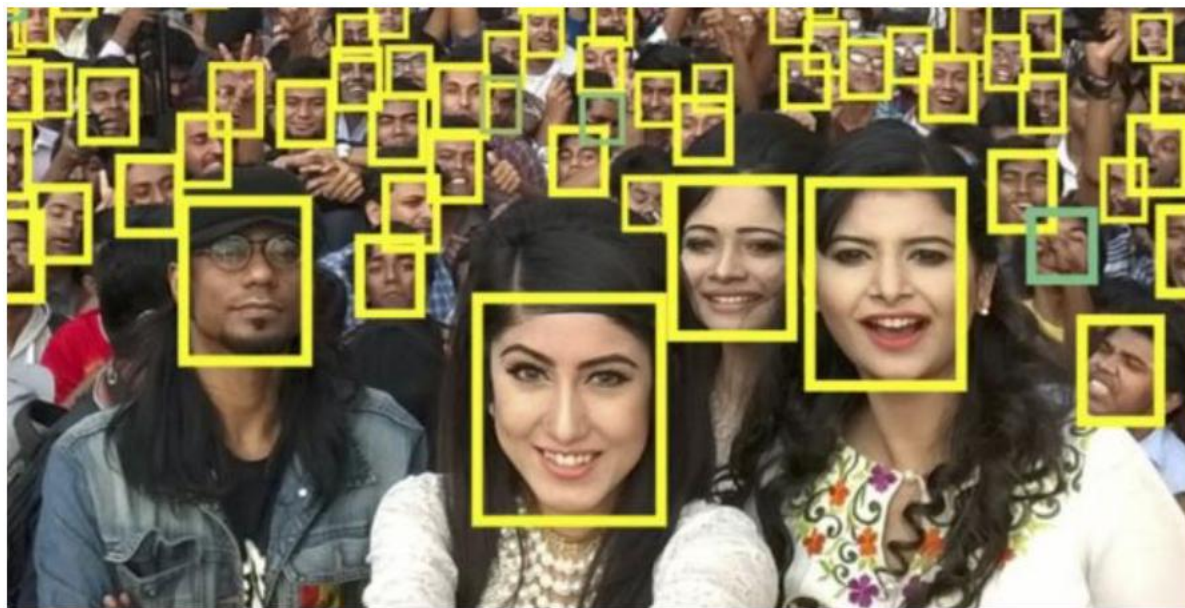
A refrigerator filled with lots of food and drinks.



A yellow school bus parked in a parking lot.

<http://arxiv.org/abs/1411.4555> "Show and Tell: A Neural Image Caption Generator"

Обработка изображений лиц (1). Детектирование



Обработка изображений лиц (2). Верификация



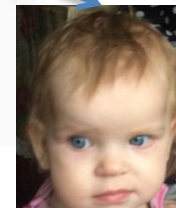
Kumar et al, TPAMI, 2011



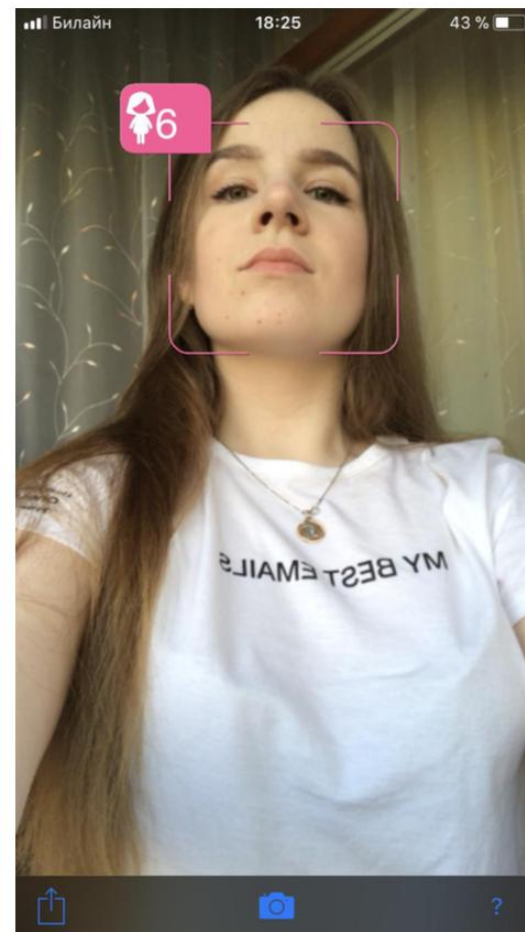
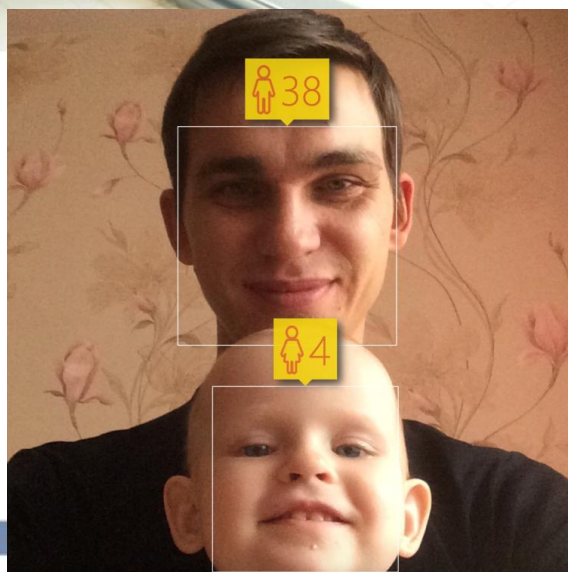
Входная видео последовательность



Эталонные
фотографии



Обработка изображений лиц (4). Распознавание пола и возраста



Обработка изображений лиц (5). Распознавание эмоций



“positive”



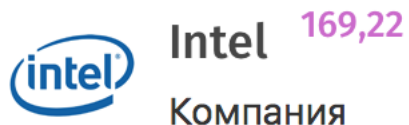
“neutral”



“negative”



Фильтрация изображений



9 ноября 2016 в 14:52

Itseez, дважды Intel Company

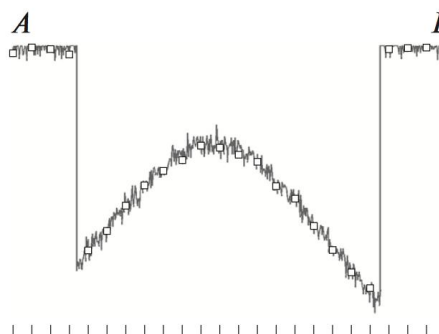
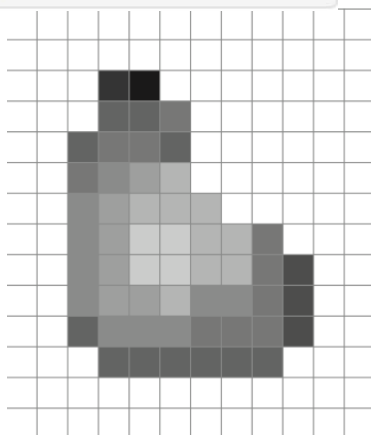
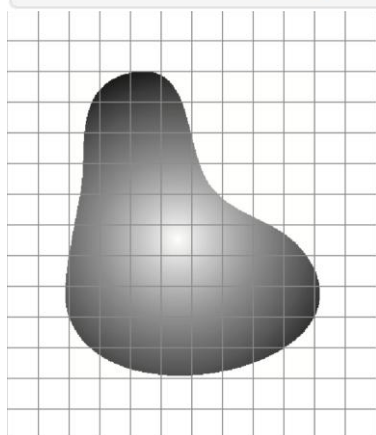
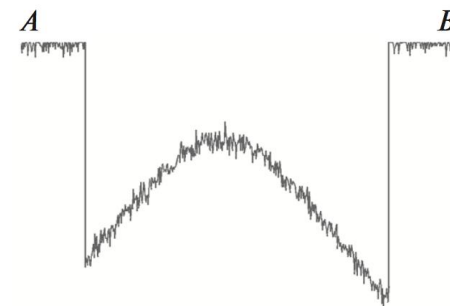
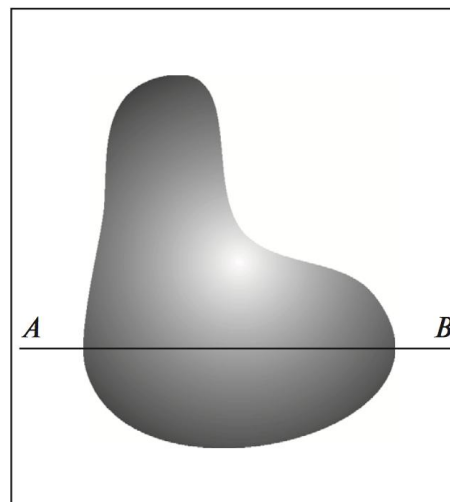
 Программирование*, Обработка изображений*, Блог компании Intel

- Open Source Computer Vision Library
- BSD license
- C++, C, Python, Java
- Windows, Linux, Mac OS, iOS and Android

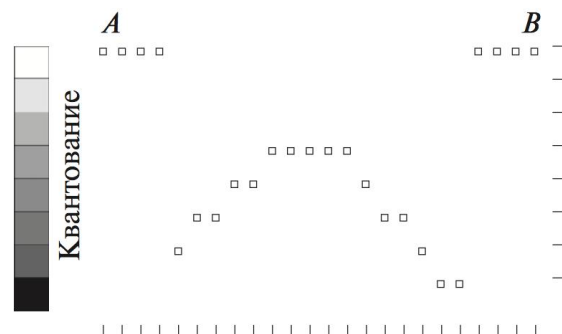
Представления цифровых изображений. Дискретизация и квантование



Image via Wikipedia



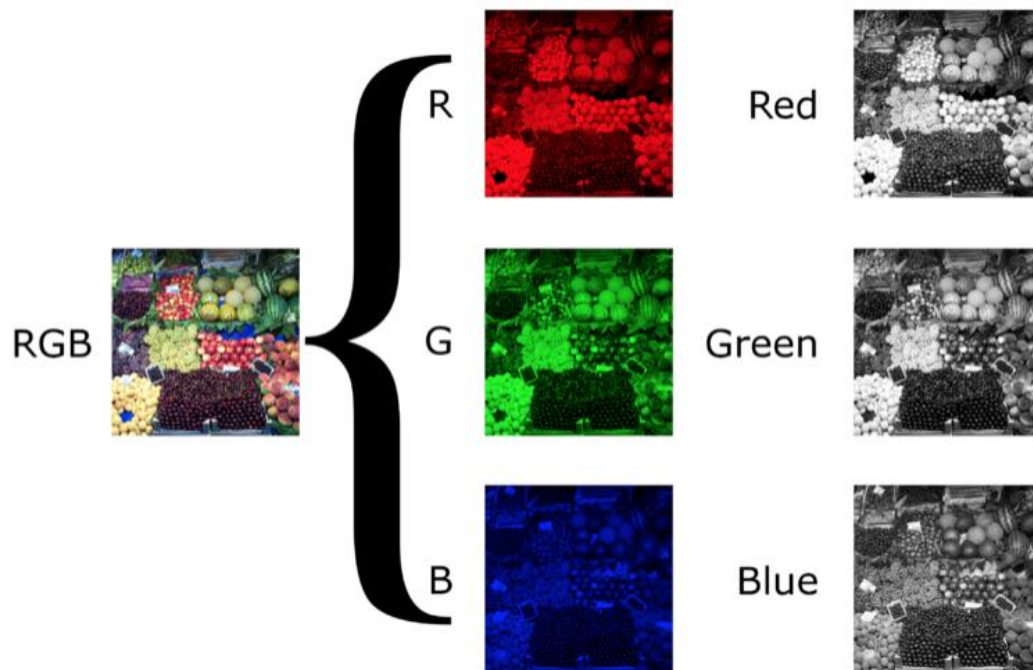
Дискретизация



Квантование

Гонсалес, Вудс «Цифровая обработка изображений»

Преобразование формата. Полутоновые изображения



Solem «Programming Computer Vision with Python»

$$Y' = 0.299R' + 0.587G' + 0.114B'$$

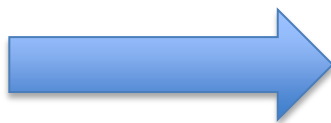
$$Y' = 0.2126R' + 0.7152G' + 0.0722B'$$

OpenCV: `cvtColor()`

Приведение всех пикселей к нулевому среднему и единичной дисперсии

$$\mu = \frac{\sum_{i=1}^I \sum_{j=1}^J p_{ij}}{IJ}$$

$$\sigma^2 = \frac{\sum_{i=1}^I \sum_{j=1}^J (p_{ij} - \mu)^2}{IJ}$$



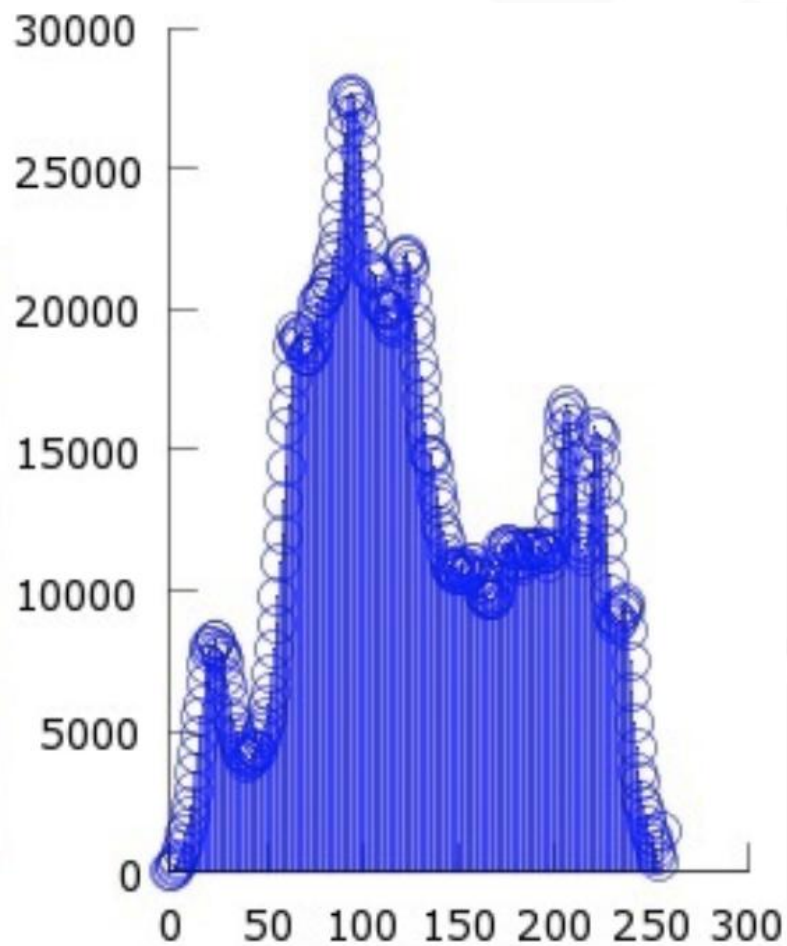
$$x_{ij} = \frac{p_{ij} - \mu}{\sigma}$$



Ponce «Computer vision: models, learning and inference»

OpenCV: `normalize()`

Гистограмма полутонового изображения



OpenCV: calcHist()

Muhammad «OpenCV Android Programming By Example»

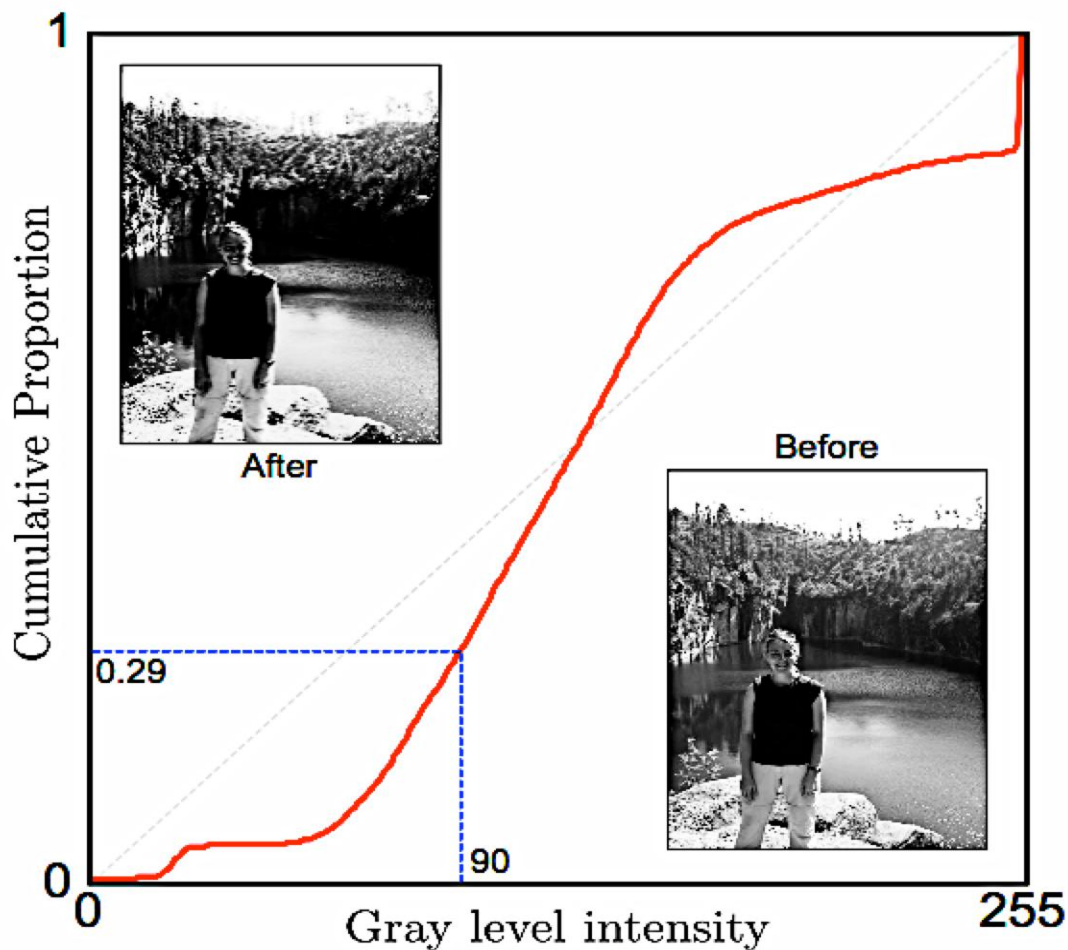
$$h_k = \sum_{i=1}^I \sum_{j=1}^J \delta[p_{ij} - k]$$

$$c_k = \frac{\sum_{l=1}^k h_l}{IJ}$$

$$x_{ij} = Kc_{p_{ij}}$$

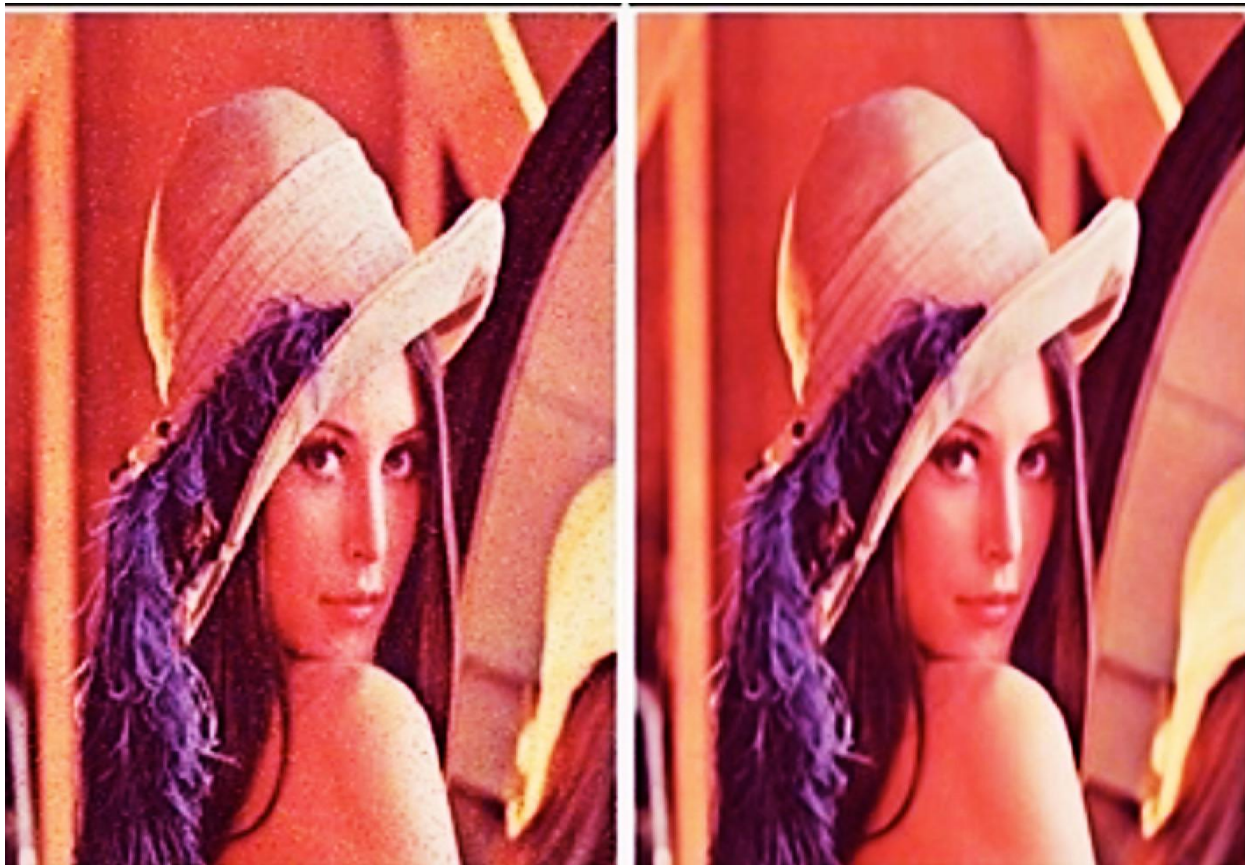
K – максимальное значение интенсивности (обычно 256)

OpenCV: `equalizeHist()`



Ponce «Computer vision: models, learning and inference»

Шум «Salt-and-pepper»



OpenCV: medianBlur()

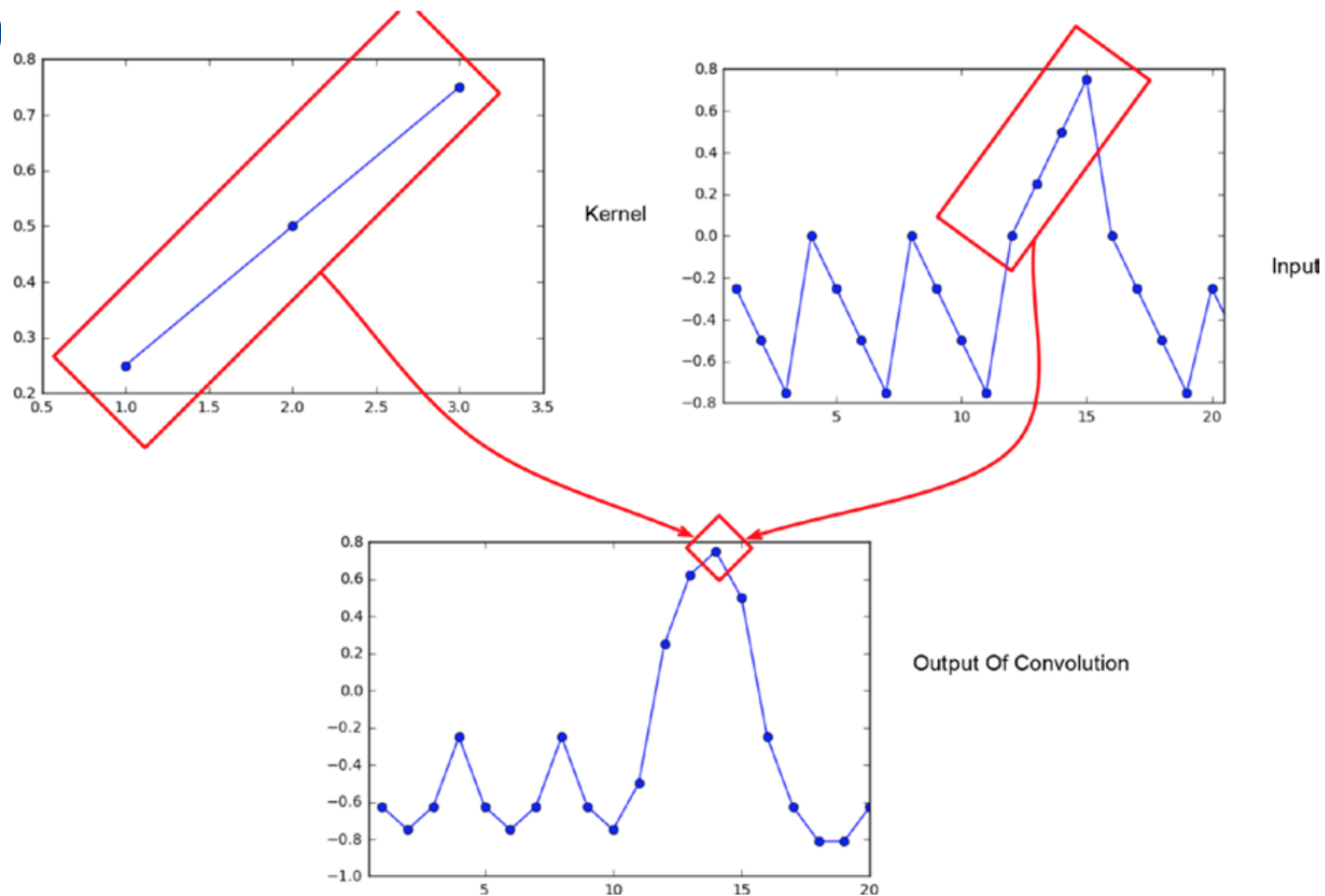
Muhammad «OpenCV Android Programming By Example»

Свертка (convolution)

$$s(t) = \sum_a I(t-a) \cdot K(a)$$

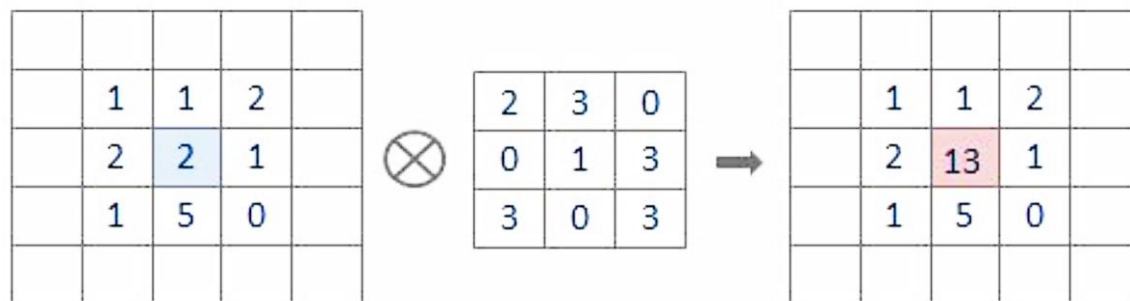
Кросс-корреляция

$$s(t) = \sum_a I(t+a) \cdot K(a)$$

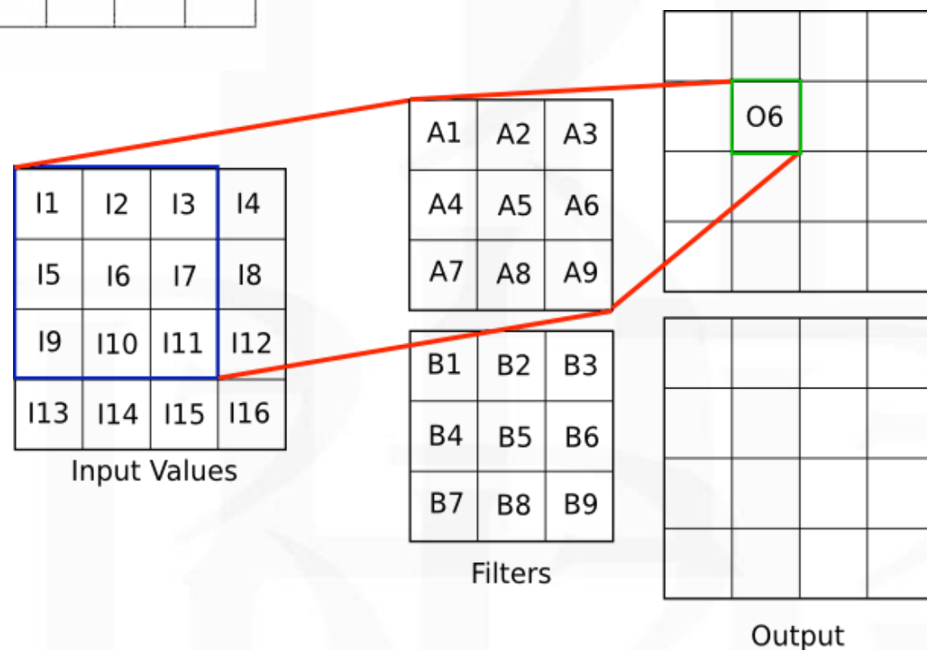
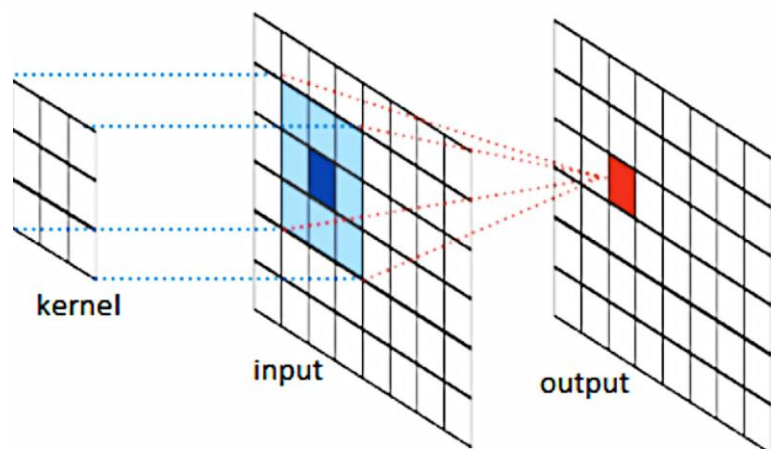


Ketkar "Deep Learning with Python"

Линейная фильтрация (2). Двумерный сигнал (изображение)



$$x_{ij} = \sum_{m=-M}^M \sum_{n=-N}^N p_{i-m, j-n} f_{m,n}$$



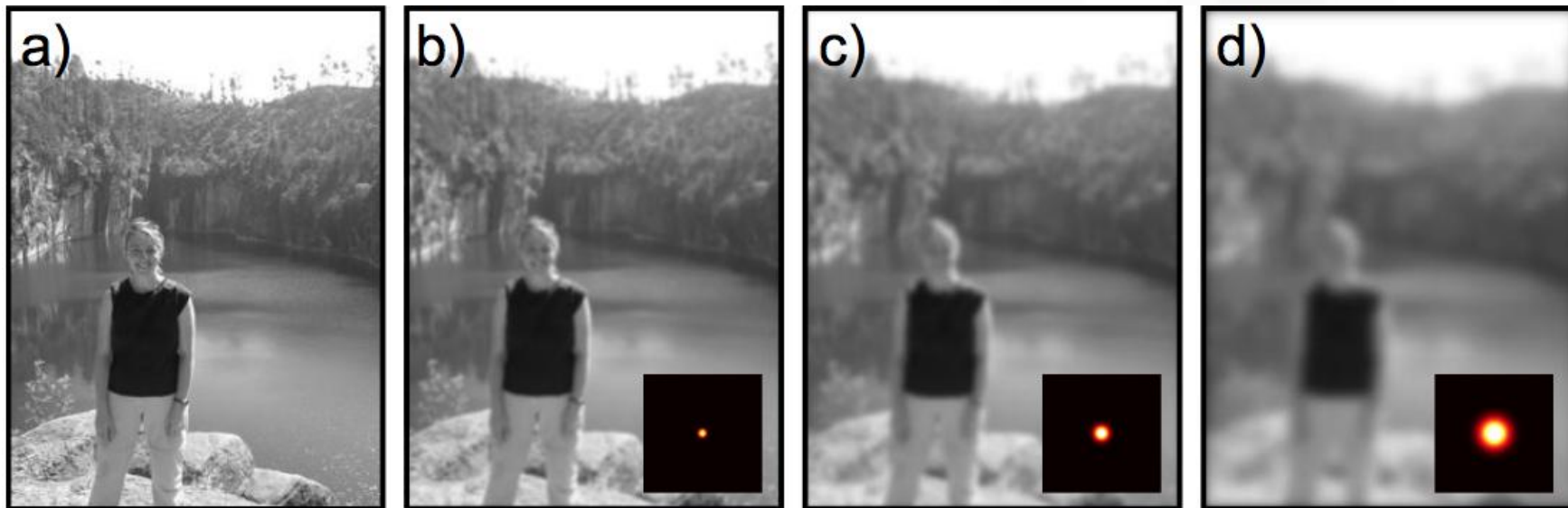
<http://intellabs.github.io/RiverTrail/tutorial/>

OpenCV: filter2D()

DLI-Teaching-Kit

Гауссовский фильтр (smoothing, blurring)

$$f(m, n) = \frac{1}{2\pi\sigma^2} \exp \left[-\frac{m^2 + n^2}{2\sigma^2} \right]$$



Ponce «Computer vision: models, learning and inference»

OpenCV: GaussianBlur()

Билатеральный фильтр

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

Domain kernel

Range kernel



OpenCV: `bilateralFilter()`

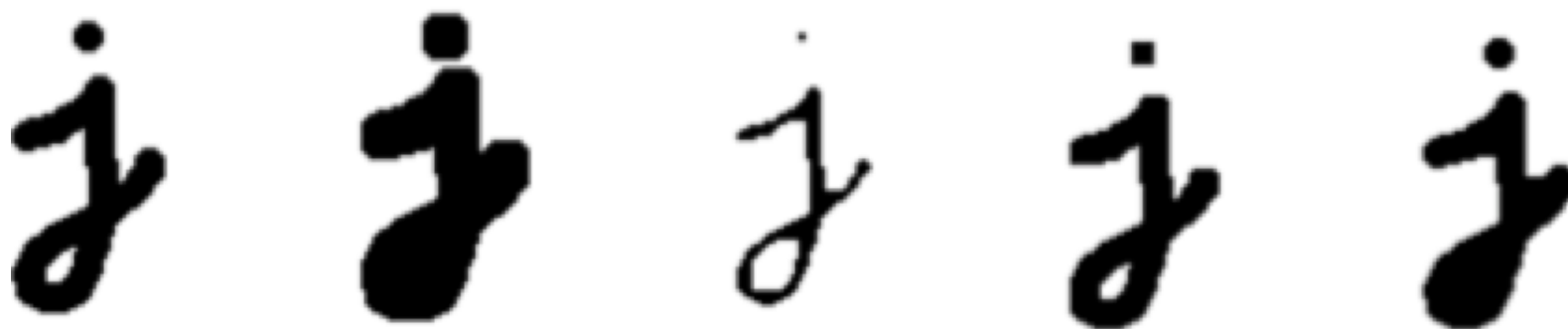
Wikipedia

**Расширение
(Dilation)**

**Сужение
(Erosion)**

**Открытие
(Opening)**

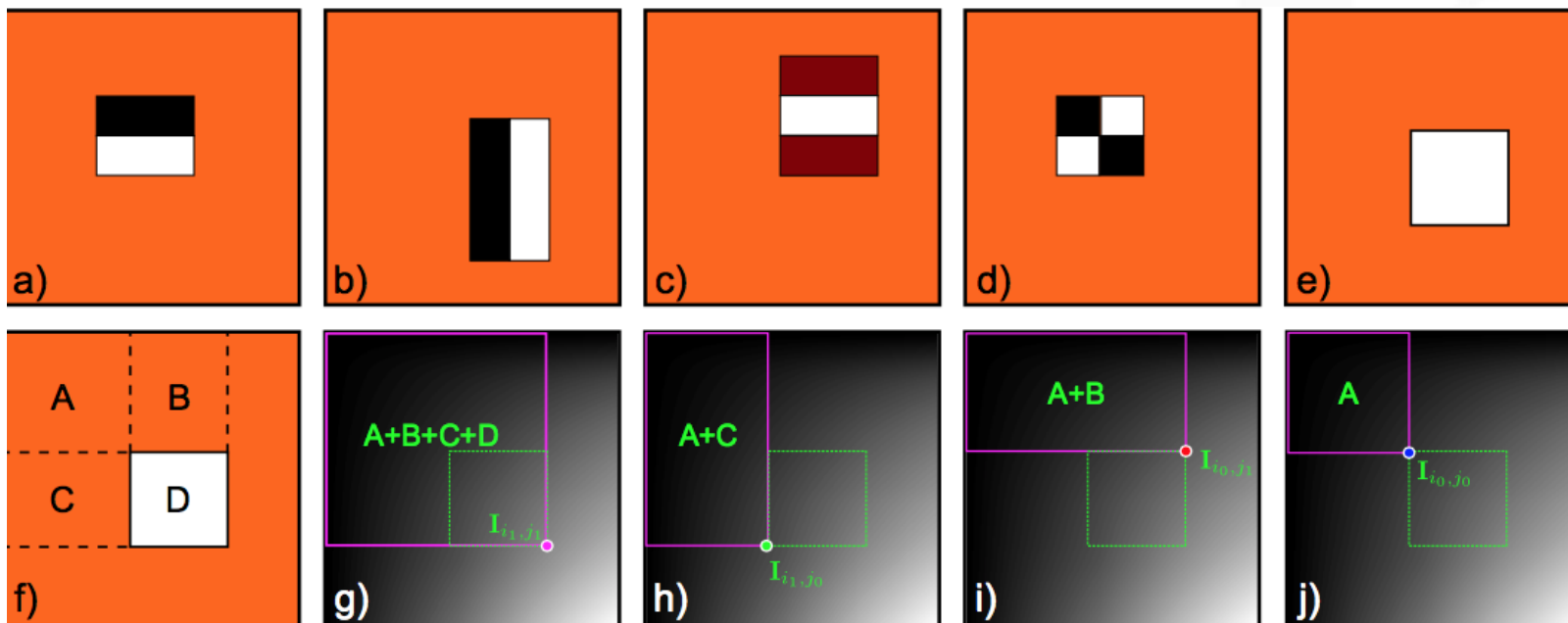
**Заккрытие
(Closing)**



Szeliski «Computer Vision: Algorithms and Applications»

OpenCV:

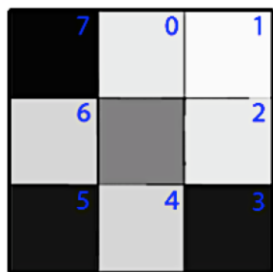
`erode(),
dilate(),
morphologyEx() //MORPH_OPEN, MORPH_CLOSE`



$$\mathbf{I}_{i_1, j_1} + \mathbf{I}_{i_0, j_0} - \mathbf{I}_{i_1, j_0} - \mathbf{I}_{i_0, j_1} = (A + B + C + D) + A - (A + C) - (A + B) = D$$

Ponce «Computer vision: models, learning and inference»

Локальные бинарные шаблоны (Local Binary Patterns, LBP)



42 ⁷	199 ⁰	234 ¹
177 ⁶	129	199 ²
65 ⁵	177 ⁴	65 ³

1 ⁷	0 ⁰	0 ¹
0 ⁶	X	0 ²
1 ⁵	0 ⁴	1 ³

Uniform pattern – число с 0, 1 или 2 переходами 0-1 или 1-0.
Примеры: «00111100», «11111111» и «01111111».

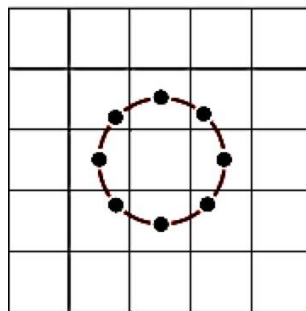
1 ⁷	0 ⁶	1 ⁵	0 ⁴	1 ³	0 ²	0 ¹	0 ⁰
----------------	----------------	----------------	----------------	----------------	----------------	----------------	----------------

$$128 + 32 + 8 = 168$$

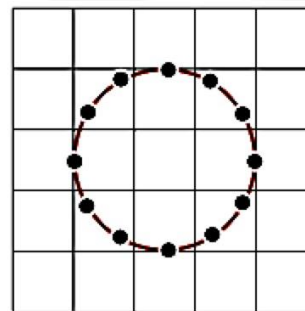
Non-uniforms patterns
объединяются в один:
«10100000» и «01010101»

<https://habrahabr.ru/post/280888/>

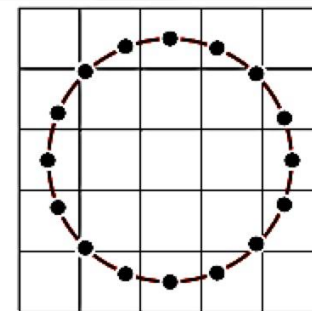
LBP_{P,R}



$$(P = 8, R = 1.0)$$



$$(P = 12, R = 1.5)$$



$$(P = 16, R = 2.0)$$

Ponce «Computer vision: models, learning and inference»

Выделение краев (Edge detection)

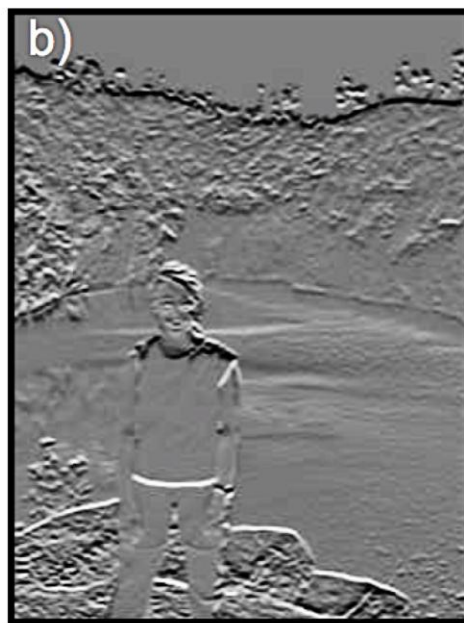
Фильтры Превитт

$$\mathbf{F}_x = \begin{bmatrix} 1 & 0 & -1 \\ 1 & 0 & -1 \\ 1 & 0 & -1 \end{bmatrix},$$

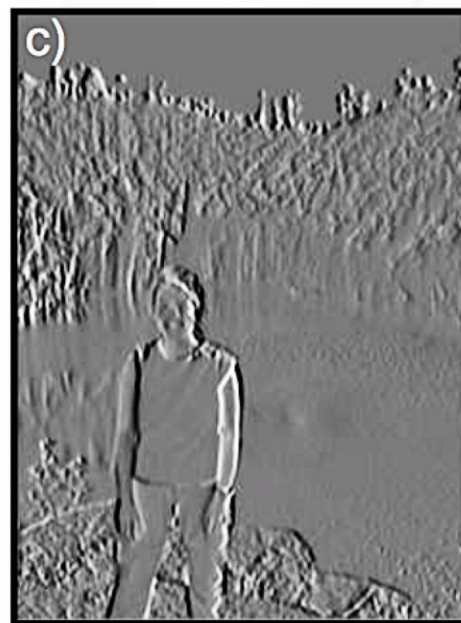
$$\mathbf{F}_y = \begin{bmatrix} 1 & 1 & 1 \\ 0 & 0 & 0 \\ -1 & -1 & -1 \end{bmatrix}$$



Original image



Prewitt (vertical)



Prewitt (horizontal)

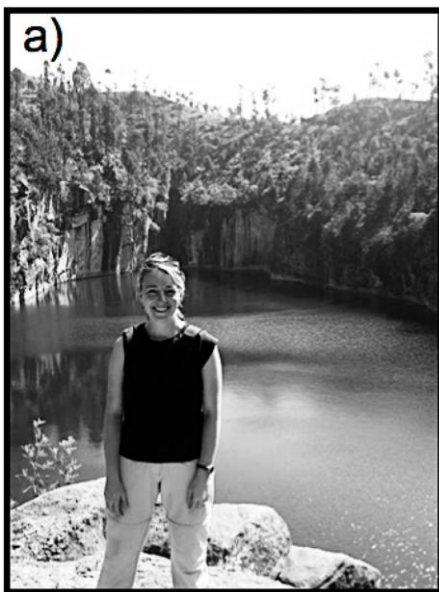
Фильтры Собеля

$$\mathbf{F}_x = \begin{bmatrix} 1 & 0 & -1 \\ 2 & 0 & -2 \\ 1 & 0 & -1 \end{bmatrix},$$

$$\mathbf{F}_y = \begin{bmatrix} 1 & 2 & 1 \\ 0 & 0 & 0 \\ -1 & -2 & -1 \end{bmatrix}$$

Ponce «Computer vision: models, learning and inference»

Вторые производные. Фильтр Лапласа.



Original image

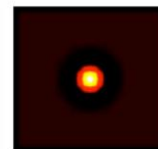


Laplacian

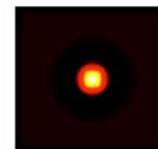
$$\begin{bmatrix} 0 & -1 & 0 \\ -1 & 4 & -1 \\ 0 & -1 & 0 \end{bmatrix}$$



Laplacian of Gaussian



Difference of Gaussians

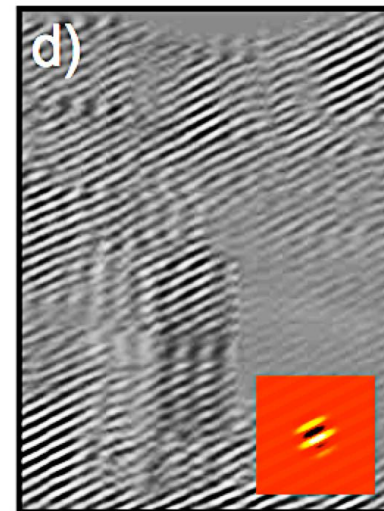
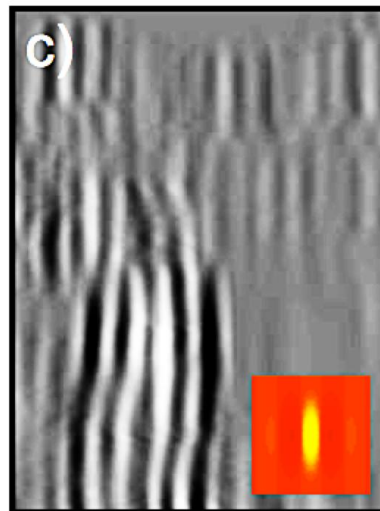
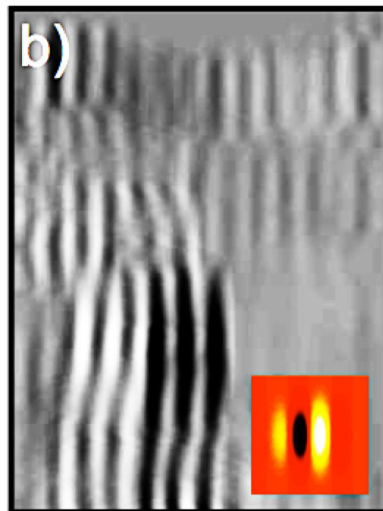
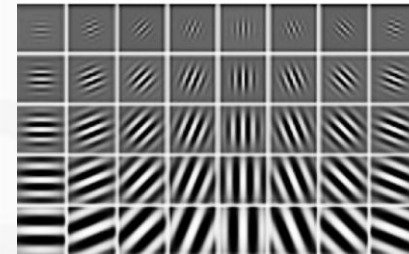


**OpenCV: `Laplacian()`,
`filter2D`**

$$LoG(x; \sigma) = \left(\frac{x^2}{\sigma^4} - \frac{1}{\sigma^2} \right) G(x; \sigma)$$

Ponce «Computer vision: models, learning and inference»

$$f_{mn} = \frac{1}{2\pi\sigma^2} \exp\left[-\frac{m^2 + n^2}{2\sigma^2}\right] \sin\left[\frac{2\pi(\cos[\omega]m + \sin[\omega]n)}{\lambda} + \phi\right]$$



OpenCV: `getGaborFilter()`

Ponce «Computer vision: models, learning and inference»

б)Вертикальный фильтр

Превитт

с)Горизонтальный фильтр

Превитт

д)Квантование ориентации

$$\theta_{ij} = \arctan[v_{ij}/h_{ij}]$$

е)Амплитуда градиента

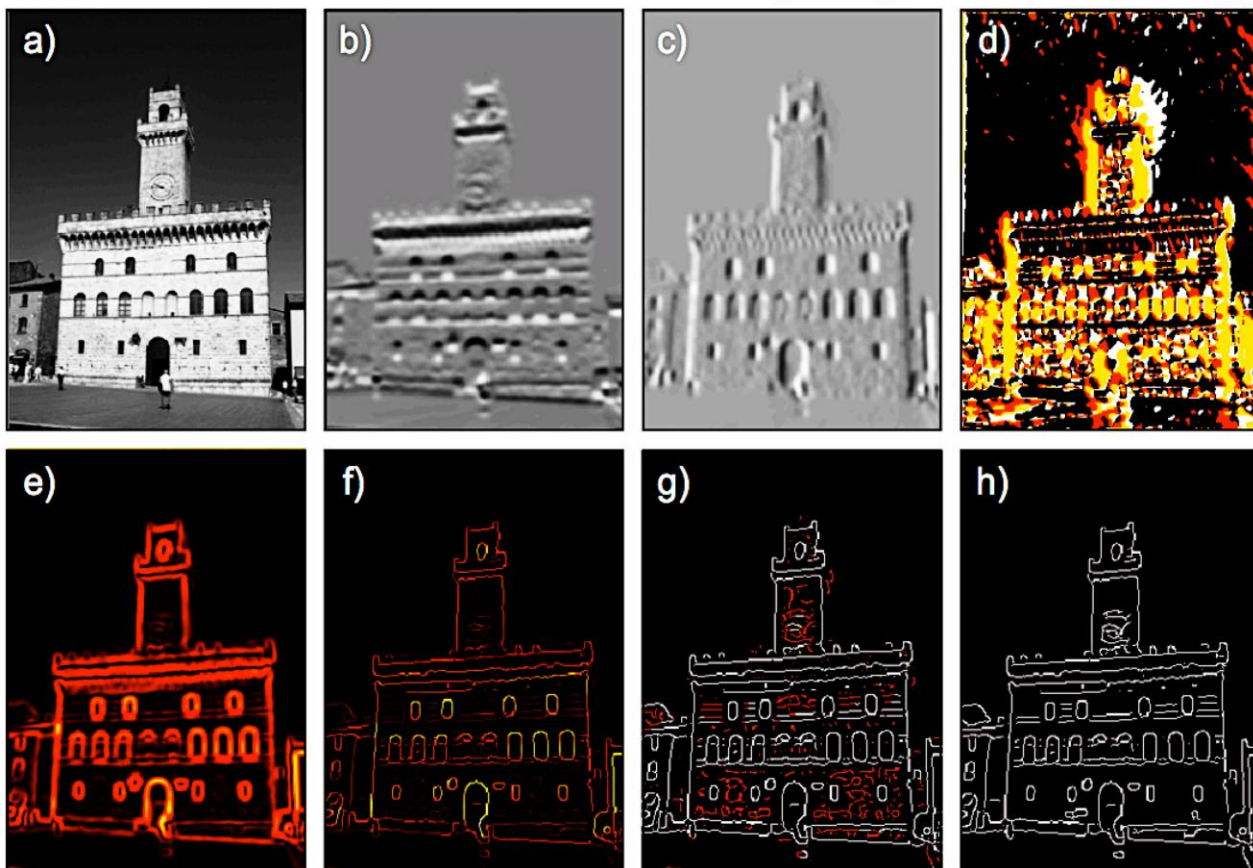
$$a_{ij} = \sqrt{h_{ij}^2 + v_{ij}^2}$$

ф)Non-maximal suppression:

угол квантуется ($\{0^\circ, 45^\circ, 90^\circ, 135^\circ\}$), амплитуда:=0, если меньше хотя бы 1 из 2 пикселей перпендикулярных градиенту

г)Два порога: белые пиксели выше максимального порога, красные – больше минимального

h) Hysteresis thresholding – все белые пиксели и связанные с ними красные

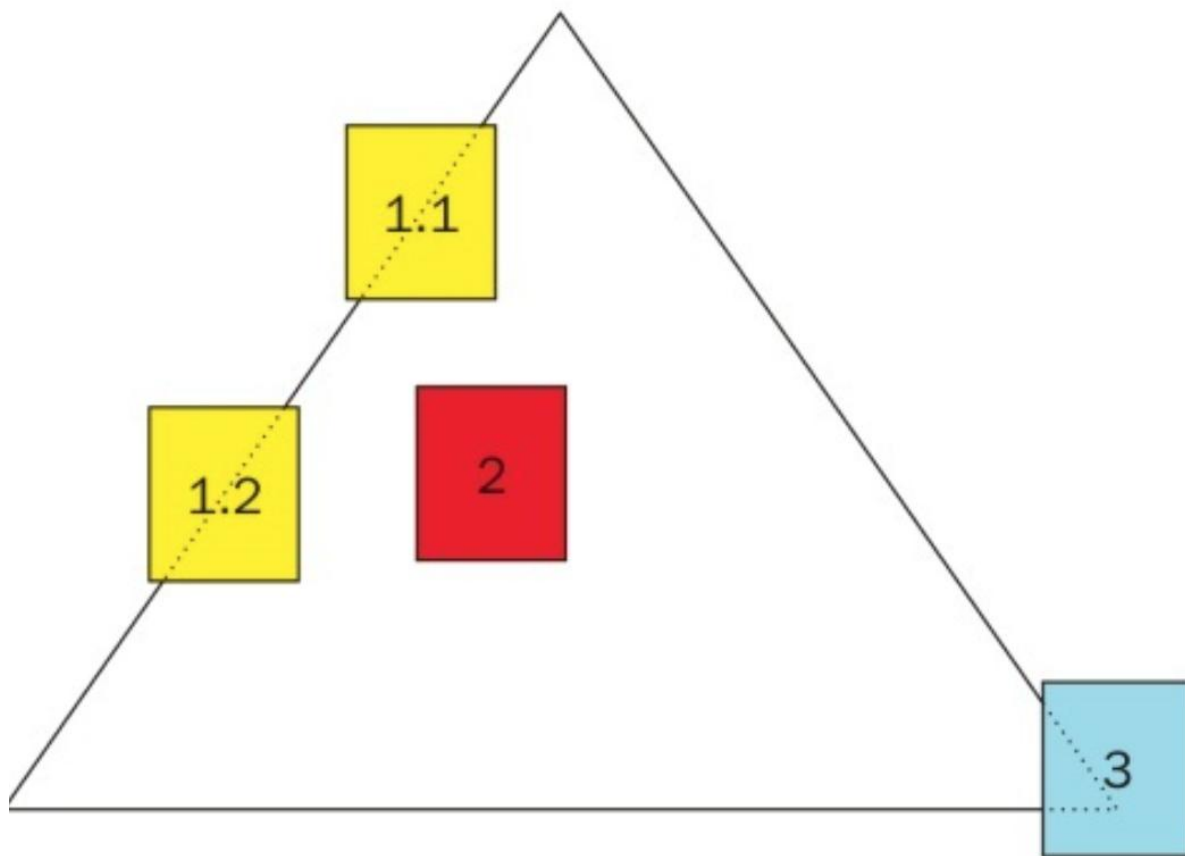


OpenCV: Canny()

Ponce «Computer vision: models, learning and inference»

Выделение ключевых точек

Interest points



Muhammad «OpenCV Android Programming By Example»

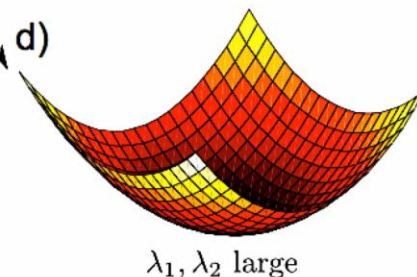
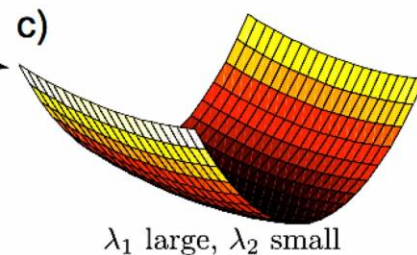
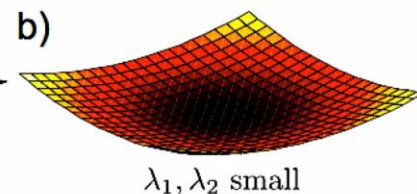
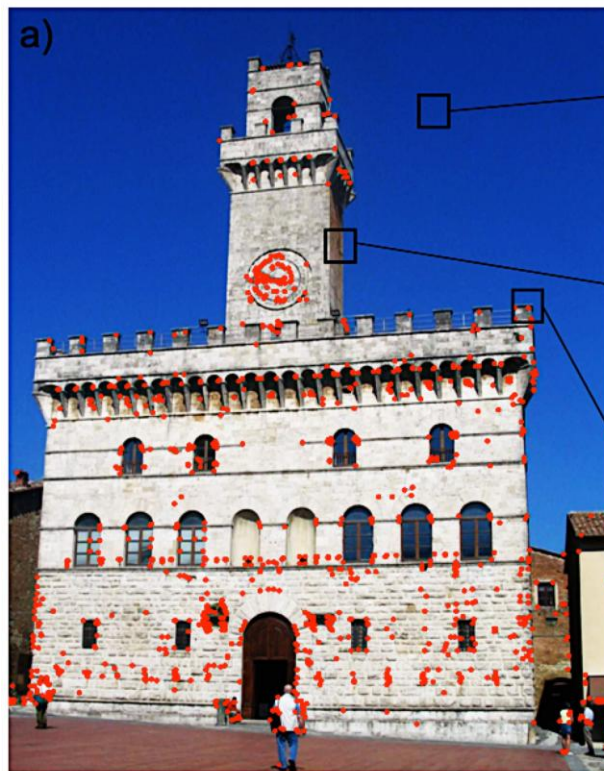
Image structure tensor

$$\mathbf{S}_{ij} = \sum_{m=i-D}^{i+D} \sum_{n=j-D}^{j+D} w_{mn} \begin{bmatrix} h_{mn}^2 & h_{mn}v_{mn} \\ h_{mn}v_{mn} & v_{mn}^2 \end{bmatrix}$$

h_{mn}, v_{mn} – отклики
фильтров
горизонтальных/
вертикальных частных
производных

$$c_{ij} = \lambda_1 \lambda_2 - \kappa(\lambda_1^2 + \lambda_2^2) = \det[\mathbf{S}_{ij}] - \kappa \cdot \text{trace}[\mathbf{S}_{ij}]$$

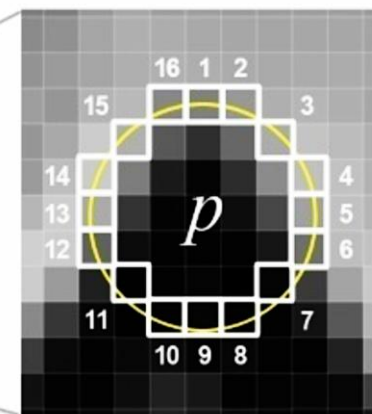
OpenCV: `cornerHarris()`



Ponce «Computer vision: models, learning and inference»

Тестирование 16 пикселей в окрестности, детектирование угла, если 12 из них ярче/темнее центрального пикселя

В начале проверяются только 4 пикселя (1, 9, 5, 13)



OpenCV:
`FastFeatureDetector ()`

Muhammad «OpenCV Android Programming By Example»

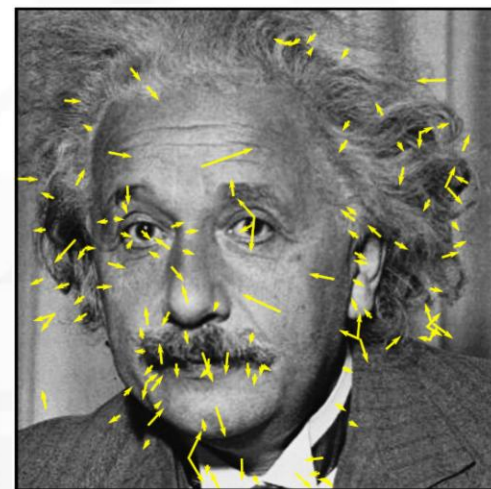
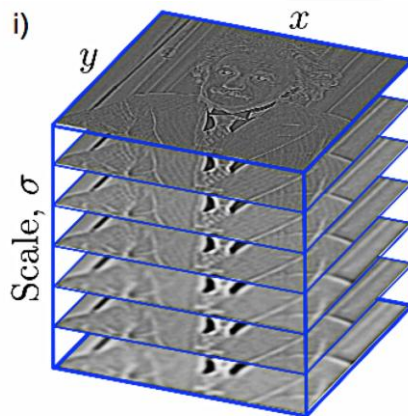
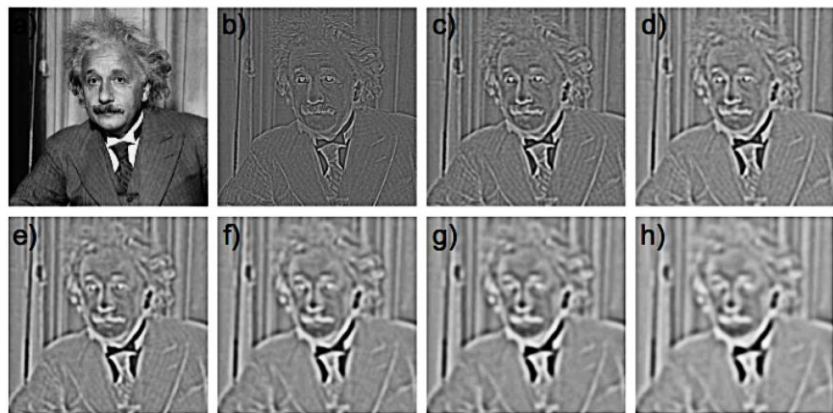
Инвариантность к масштабу. Пирамида изображений



Muhammad «OpenCV Android Programming By Example»

SIFT детектор (Scale-Invariant Feature Transform)

1. Поиск в пирамиде экстремальных точек – интенсивность больше/меньше всех ($3 \times 3 \times 3 - 1 = 26$ соседей)
2. Проверка условий детектора Харриса
3. Построение гистограммы ориентации градиента в локальной окрестности (36 блоков на 360 градусов) и выбор одного или двух пиков в гистограмме

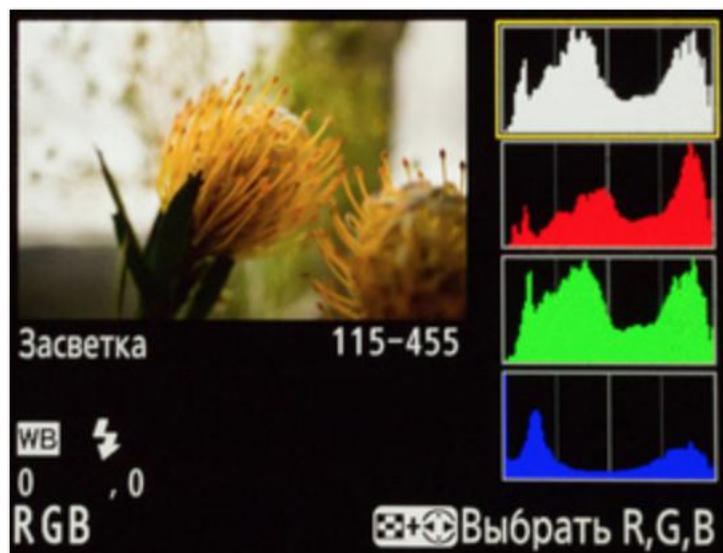


OpenCV:
`SiftFeatureDetector ()`

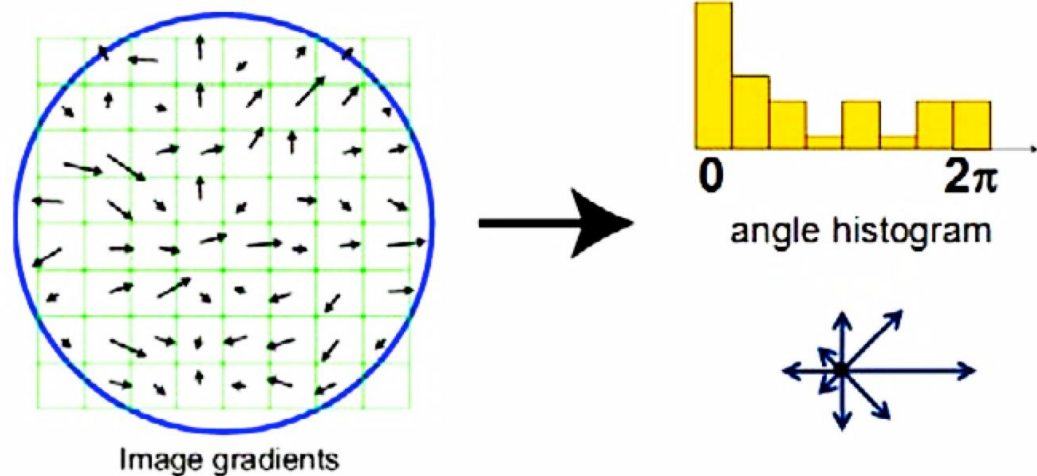
Ponce «Computer vision: models, learning and inference»

Дескрипторы

Гистограмма цвета



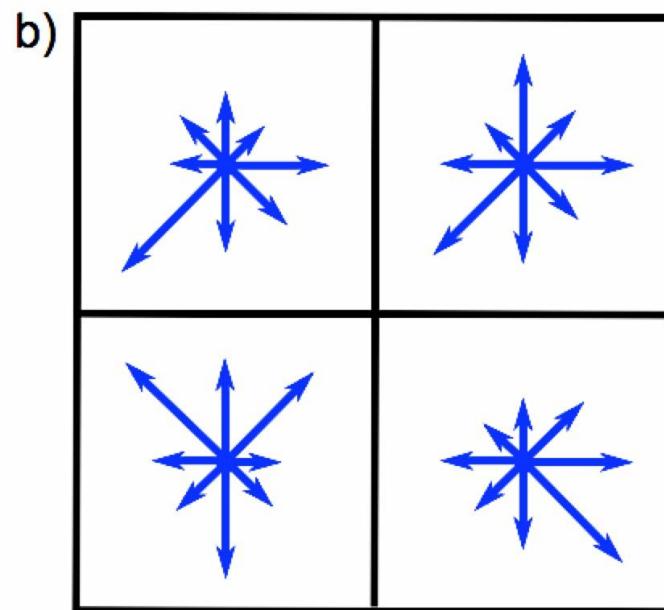
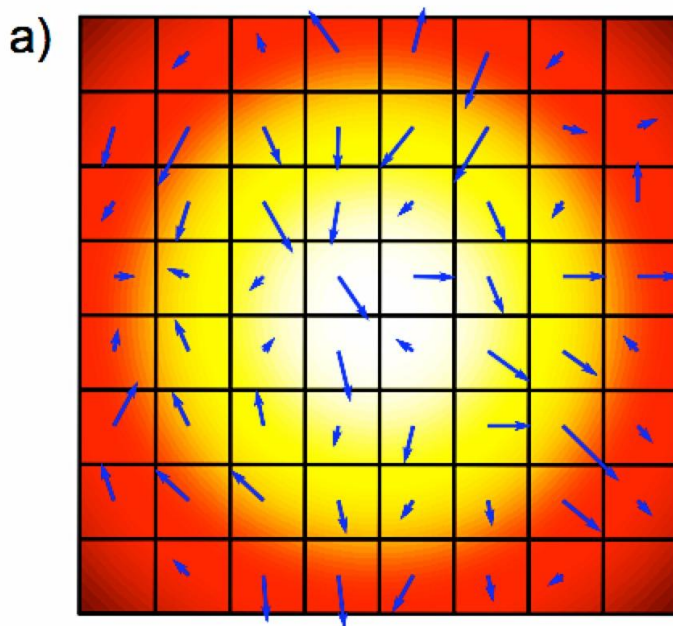
Гистограмма ориентаций градиента



<https://prophotos.ru/lessons/17601-gistogramma-i-obrabotka-izobrazheniya>

[Lowe, 2004]

1. 16x16 окрестность точки разбивается сеткой 4x4 ячейки
2. В каждой ячейке вычисляется взвешенная гистограмма ориентации градиента (8 блоков)
3. Гистограммы всех ячеек нормируются по ориентации ключевой точки и объединяются в единый SIFT дескриптор (размерность $4 \times 4 \times 8 = 128$)

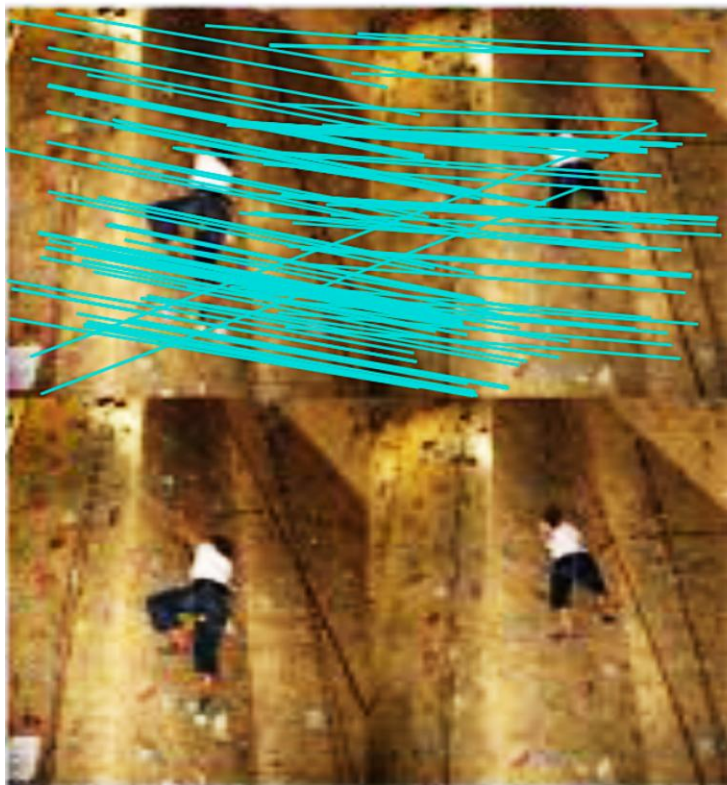


OpenCV: SIFT class

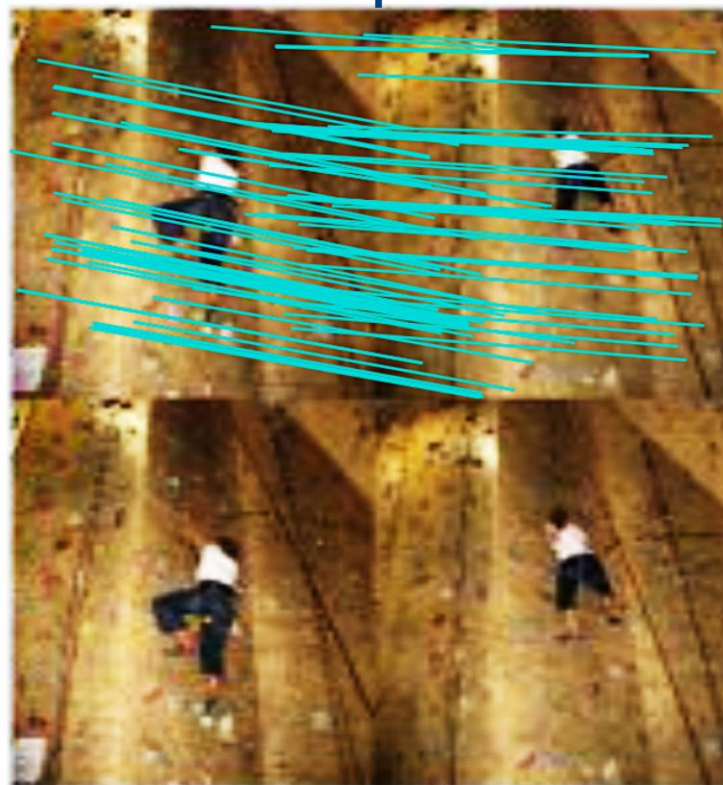
Ponce «Computer vision: models, learning and inference»

**Дескрипторы совпадают, если расстояние до 1NN (nearest neighbor)
/расстояние до 2NN меньше определенного порога (0,6-0,8)**

**Matching дескрипторов
левого изображения**



**Matching дескрипторов
обоих изображений**



Solem «Programming Computer Vision with Python»

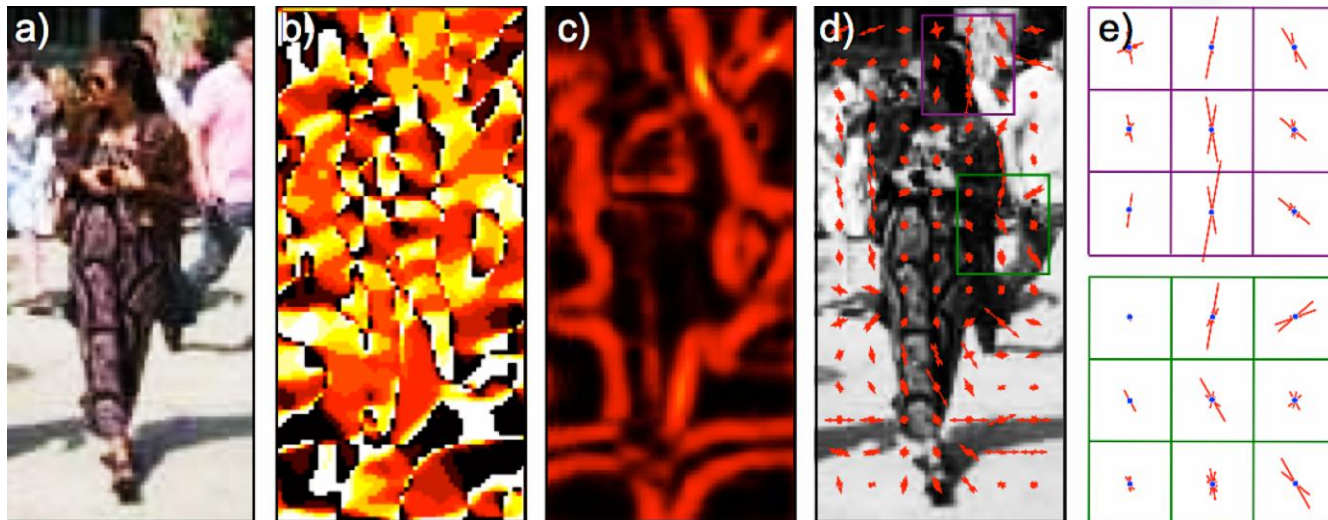
b) Ориентация градиента (квантованная на 9 блоков для 180 градусов)

c) Амплитуда градиента

d) Дескрипторы ячеек 6x6 пикселей – взвешенные гистограммы ориентаций

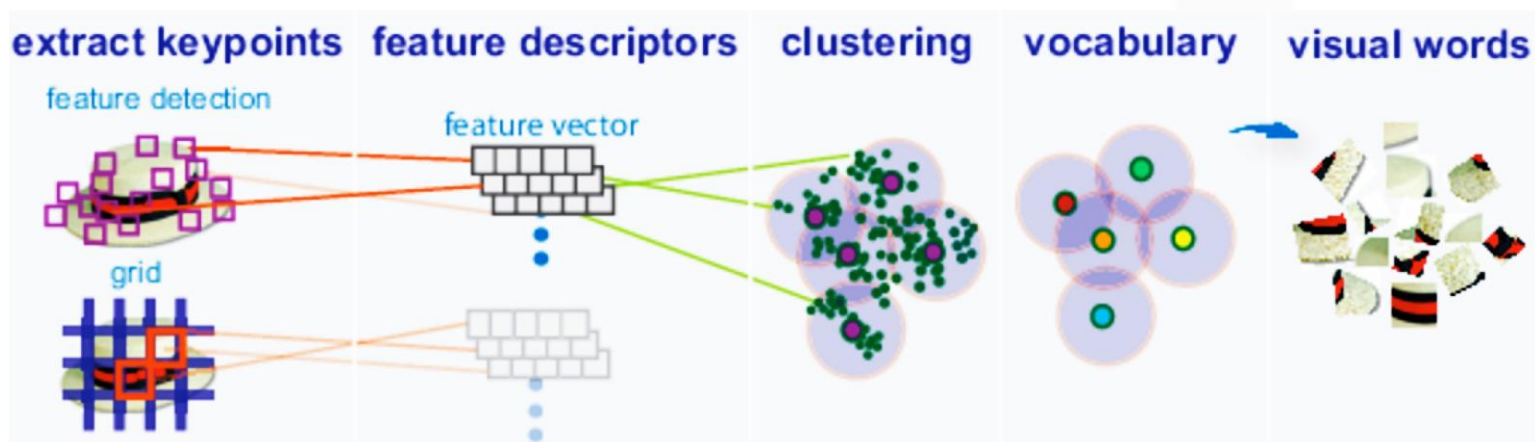
e) Дескриптор блока – нормированный объединенный вектор дескрипторов смежных 3x3 ячеек

Итоговый дескриптор HOG для изображения 64x128 – объединение дескрипторов блоков

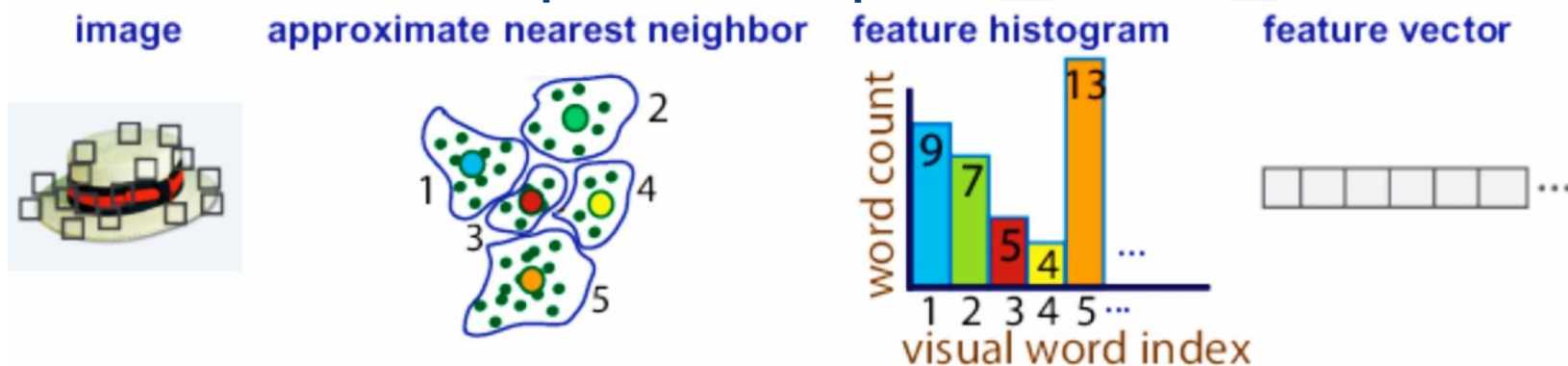


Ponce «Computer vision: models, learning and inference»

Формирование «словаря»



Кодирование изображения



<https://www.mathworks.com/help/vision/ug/image-classification-with-bag-of-visual-words.html>



НАЦИОНАЛЬНЫЙ ИССЛЕДОВАТЕЛЬСКИЙ
УНИВЕРСИТЕТ

Спасибо
за внимание!