Лекция 6. Компьютерное зрение

Денис Деркач, Дмитрий Тарасов

24 февраля 2025 года

Введение

Использование компьютерного зрения

Smartphones: QR codes, computational photography (Android Lens Blur, iPhone Portrait Mode), panorama construction (Google Photo Spheres), Night Sight (Pixel), iPhone Pro 3D scanning (LiDAR), body workout form detection, face filters, FaceID (iPhone)

Smartwatches: Heart rate detection, proximity detection

Security: Fingerprint/iris/face scanning (offices, airports), CCTV monitoring

Laptops/Desktops: Biometrics auto-login (face recognition, 3D)

Web: Image search, Google photos (face recognition, object recognition, scene recognition, geolocalization from vision), Facebook (image captioning), Google maps aerial imaging (image stitching), YouTube (content categorization), Photoshop, PowerPoint (captioning, design suggestions)

Virtual Worlds: VR/AR head tracking (Meta Quest, Apple Vision Pro), simultaneous localization and mapping, person tracking (Kinect), gesture recognition, virtual try-on, digital humans

Telepresence: Virtual backgrounds (Zoom, Google Meet), webcam person/face following

Media: Visual effects for film/TV, virtual sports replay, automatic camera control, semantics-based auto edits

Transportation: Assisted driving (cruise control, self-driving), face tracking/iris dilation for safety

Supermarkets: Cashier-less checkout, theft detection (Walmart), fruits/vegetables sorting, packaging + manufacture

Medical imaging: CAT / MRI reconstruction, assisted diagnosis, surgery planning, automatic pathology, connectomics

Space Exploration: Mar rovers, space telescopes (Hubble, James Webb)

Industry: Vision-based robotics (human+robot spaces in Amazon warehouses), online shopping (Amazon, Walmart), machine-assisted tools (routers, jigs), OCR (USPS), ANPR (number plates for tolls), drones

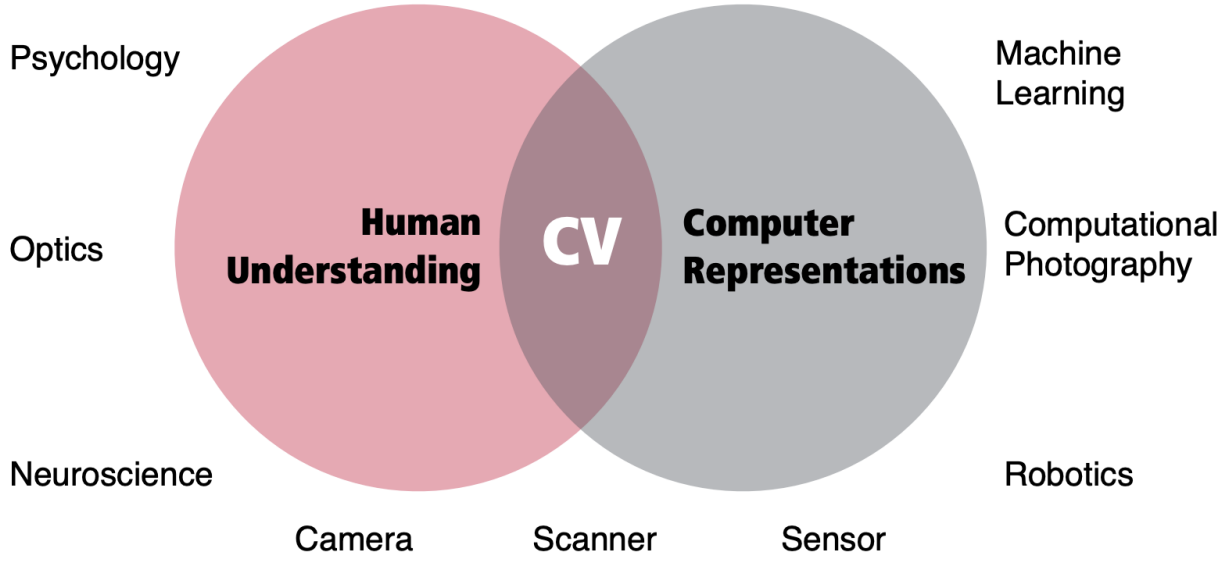
Creative: Photoshop, vision-language models for image and video generation (Dall-E, SORA), automatic video editors

https://browncsci1430.github.io/index.html

Определение

Компьютерное зрение связано с **автоматическим извлечением, анализом и пониманием полезной информации** из одного изображения или последовательности изображений. Оно включает в себя разработку **теоретической и алгоритмической основы** для достижения автоматического визуального понимания

British Machine Vision Association and Society for Pattern Recognition



Саб-домены

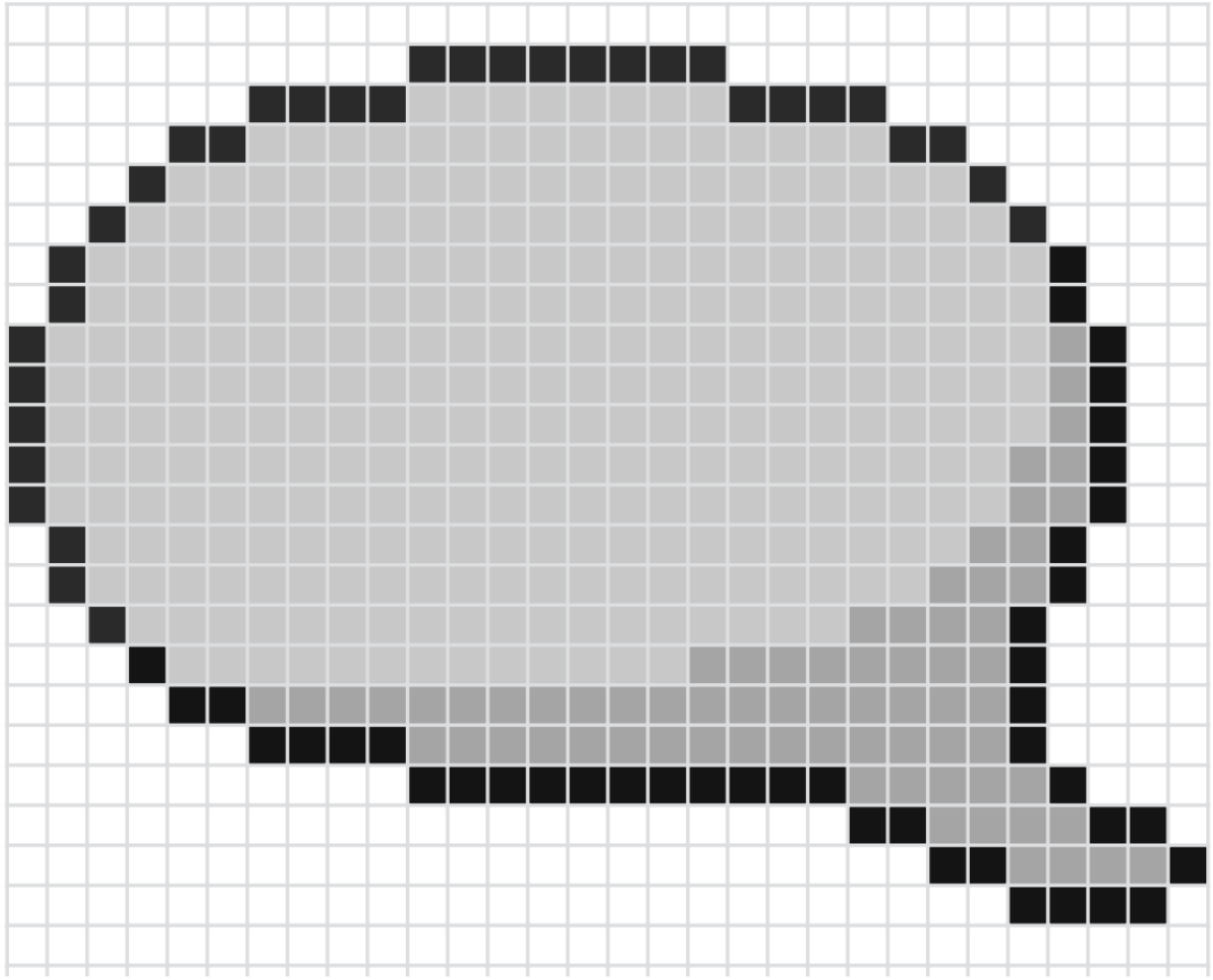
- (Предварительная) обработка изображений имеет дело с низкоуровневыми особенностями изображений.
- Обнаружение особенностей обеспечивает уточненное представление изображений.
- Сегментация обнаруживает части изображений.
- 3D-реконструкция создает 3D-модели объектов из 2D-изображений.
- Распознавание объектов маркирует то, что появляется на изображениях.
- Анализ движения имеет дело с движущимися объектами в видео.

Под-домены

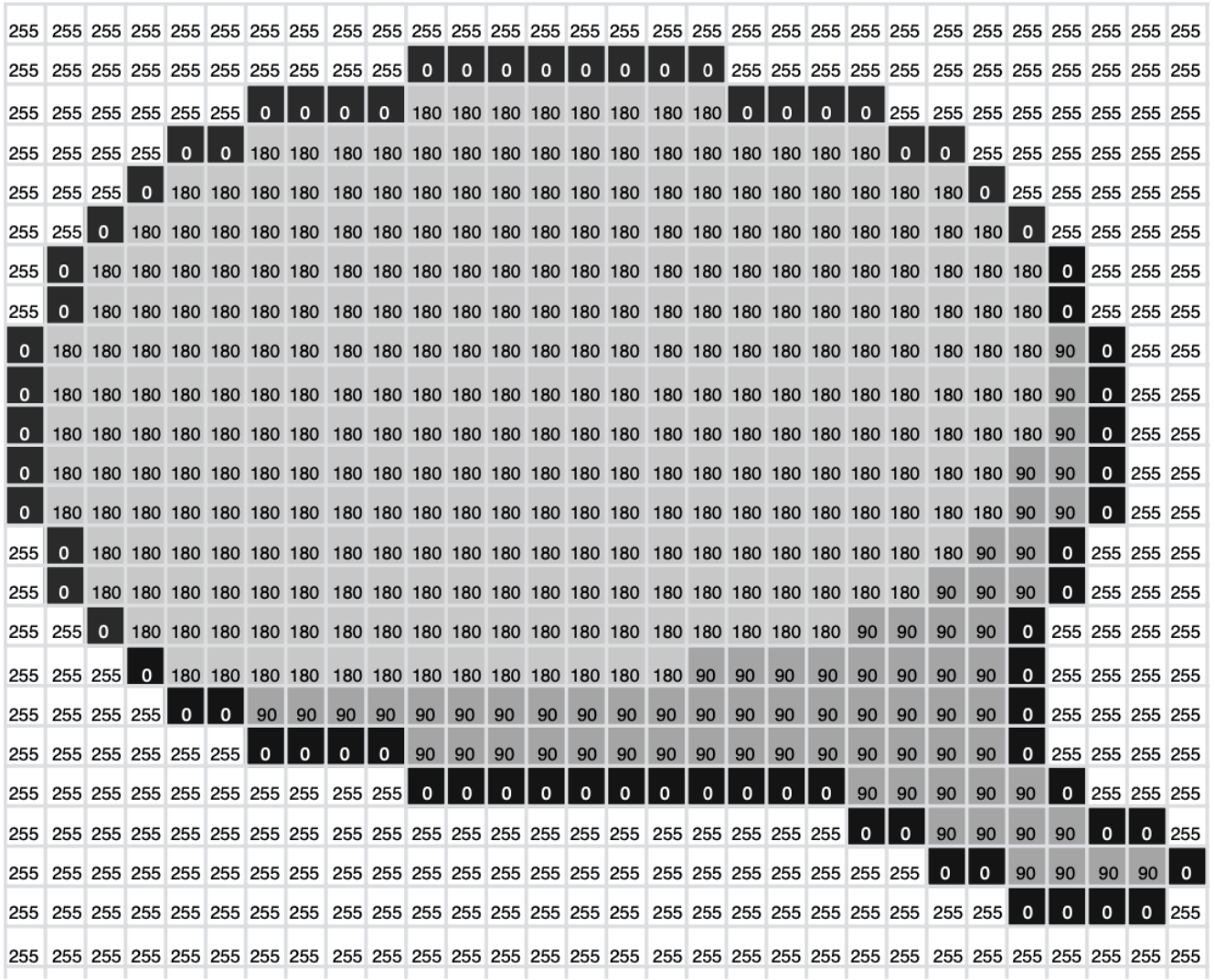
- (Предварительная) обработка изображений имеет дело с низкоуровневыми особенностями изображений.
- Обнаружение особенностей обеспечивает уточненное представление изображений.
- Сегментация обнаруживает части изображений.
- 3D-реконструкция создает 3D-модели объектов из 2D-изображений.
- Распознавание объектов маркирует то, что появляется на изображениях.
- Анализ движения имеет дело с движущимися объектами в видео.

Обработка изображений

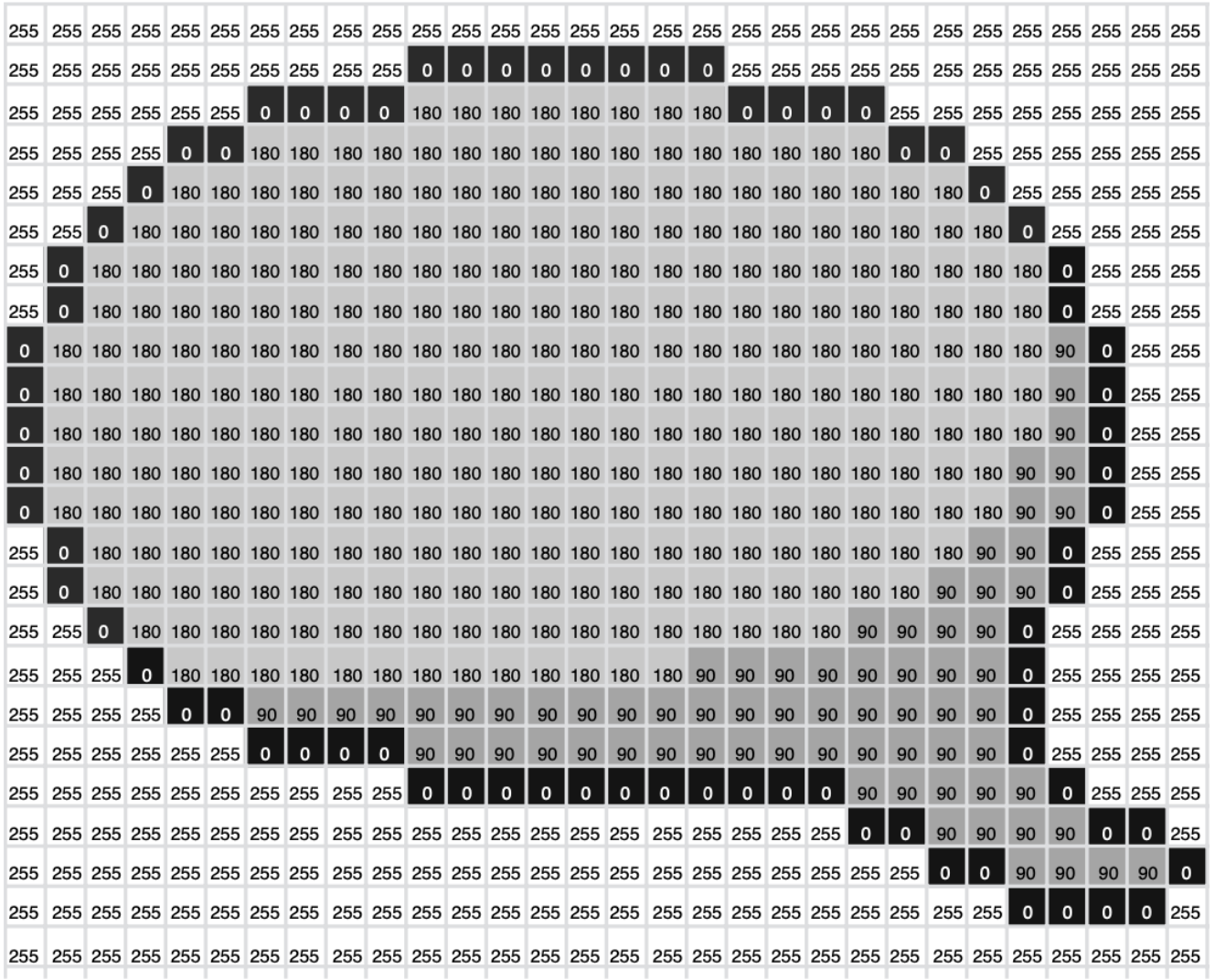
Pixel (picture element)



Pixel (picture element)



Pixel (picture element)



Цвета

Три цвета = три канала

Fig. 1.4 Original RGB colour image `Fountain` (*upper left*), showing a square in Guanajuato, and its decomposition into the three contributing channels: *Red* (*upper right*), *Green* (*lower left*), and *Blue* (*lower right*). For example, *red* is shown with high intensity in the *red channel*, but in low intensity in the *green* and *blue channel*

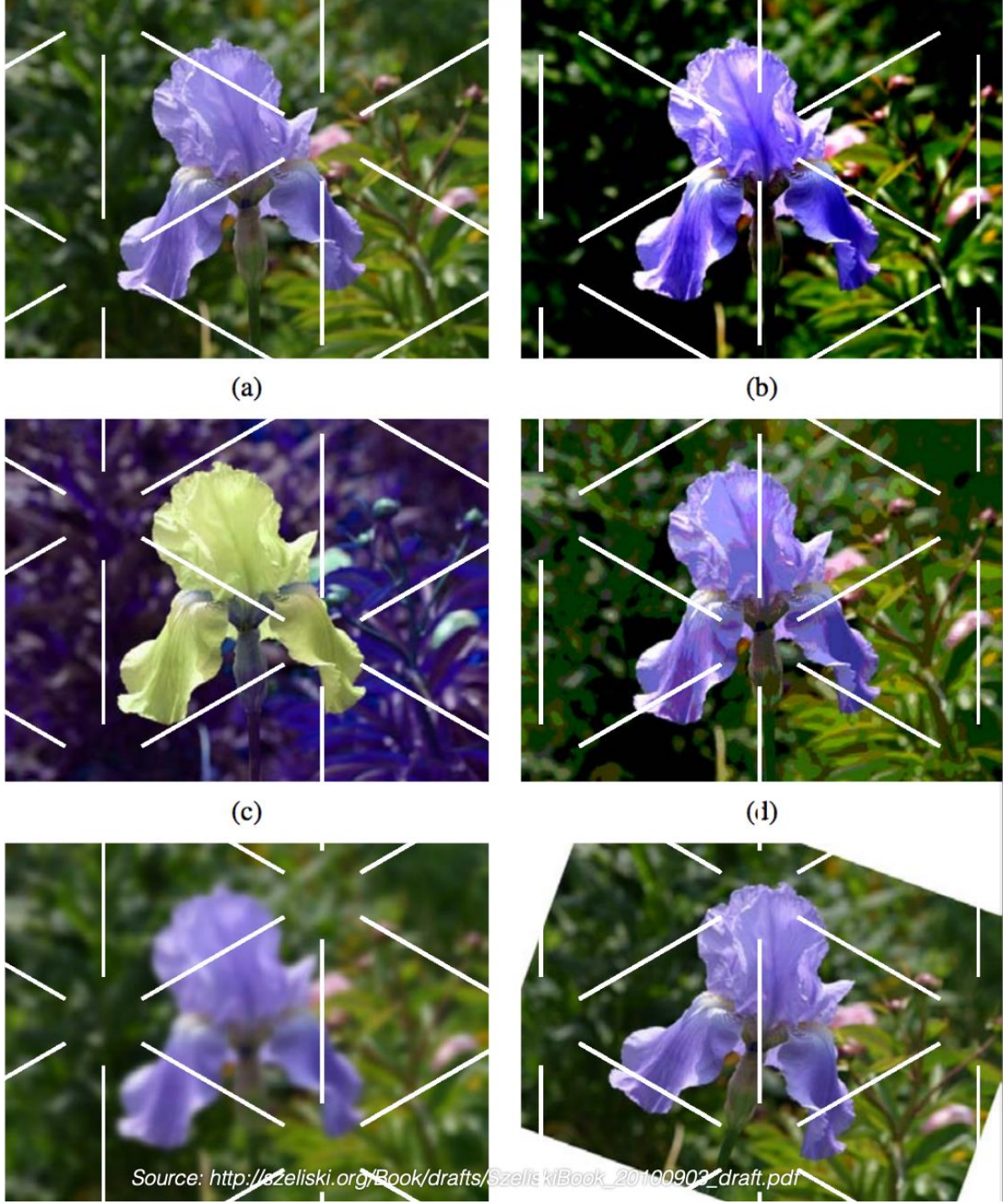
Source: Reinhard Klette, Concise Computer Vision, Springer

(Pre-)processing

Предварительная обработка изображения — это прямая манипуляция значениями пикселей.

Возможны различные операции:

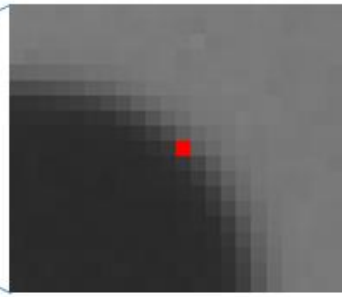
- Яркость, контраст
- Выравнивание гистограммы
- Нормализация цвета
- Фильтрация

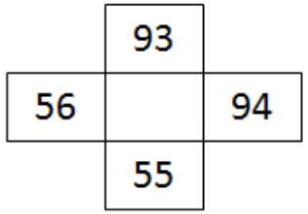


Computer Vision: Algorithms and Applications (Texts in Computer Science) [2022] Richard Szeliski

Яркость, контраст

https://mccormickml.com/2013/05/09/hog-person-detector-tutorial/

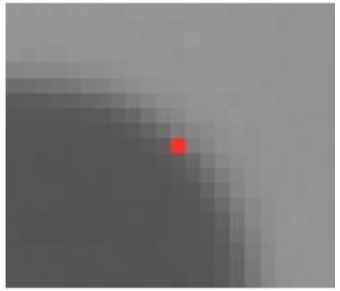


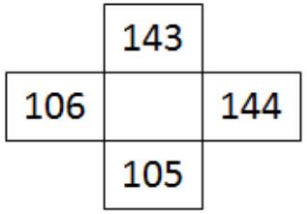


$$\nabla f = \begin{bmatrix} 38 \\ 38 \end{bmatrix}$$

$$|\nabla f| = \sqrt{(38)^2+(38)^2} = 53.74$$

Добавьте X ко всем значениям пикселей для яркости

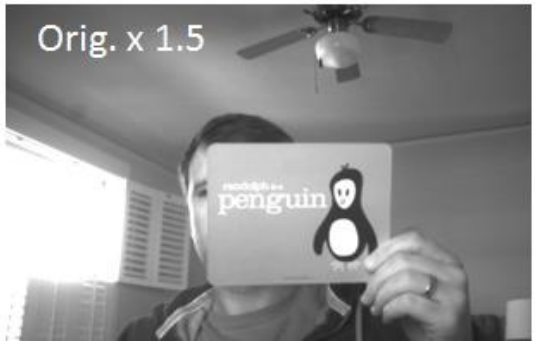


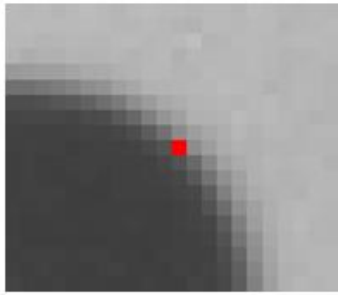


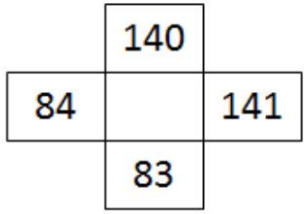
$$\nabla f = \begin{bmatrix} 38 \\ 38 \end{bmatrix}$$

$$|\nabla f| = \sqrt{(38)^2+(38)^2} = 53.74$$

Умножьте X на все значения пикселей для контраста







$$\nabla f = \begin{bmatrix} 57 \\ 57 \end{bmatrix}$$

$$|\nabla f| = \sqrt{(57)^2+(57)^2} = 80.61$$

Если разделить все три вектора на их соответствующие величины, то получится одинаковый результат для всех трех векторов: [0,71 0,71].

Яркость, контраст

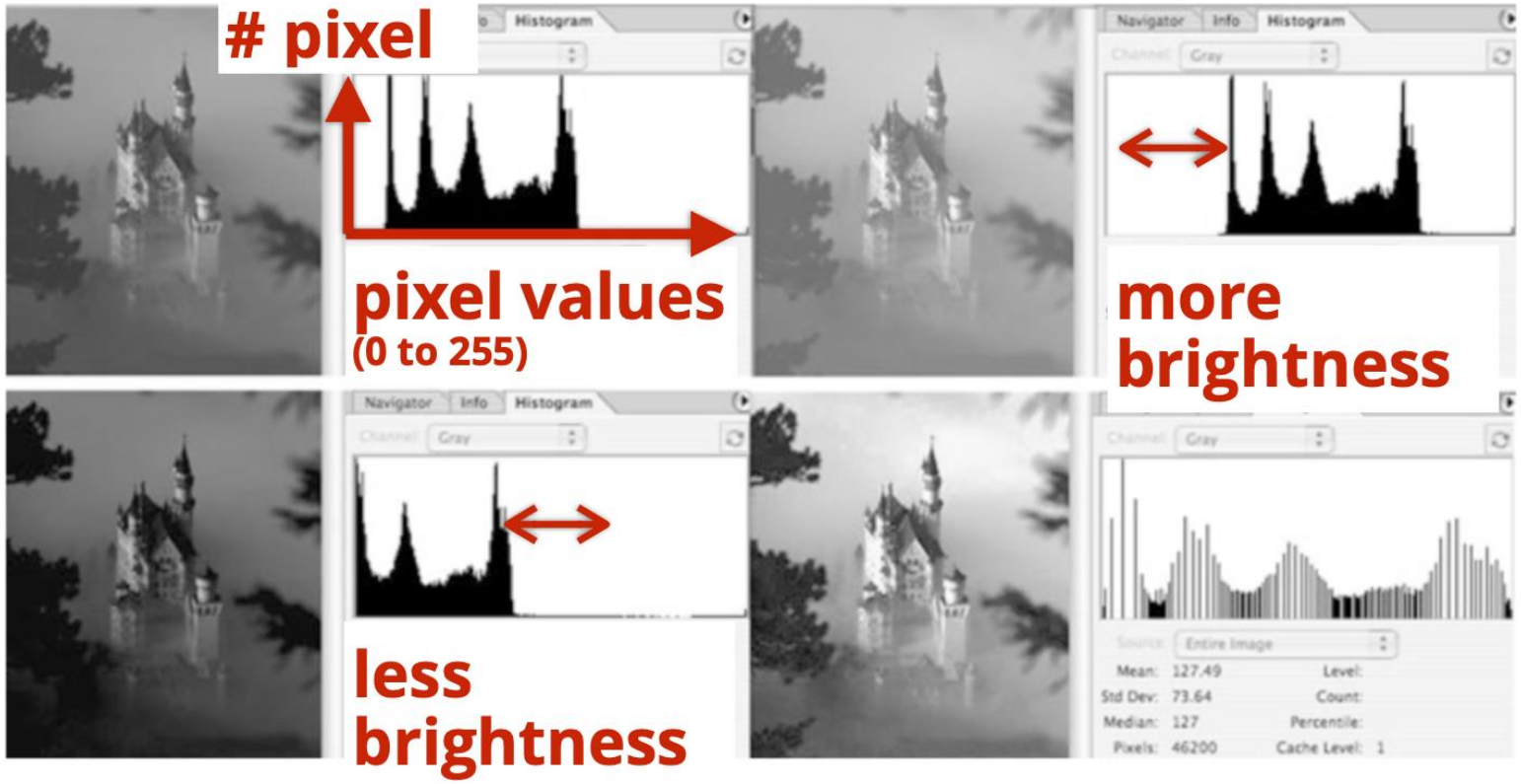
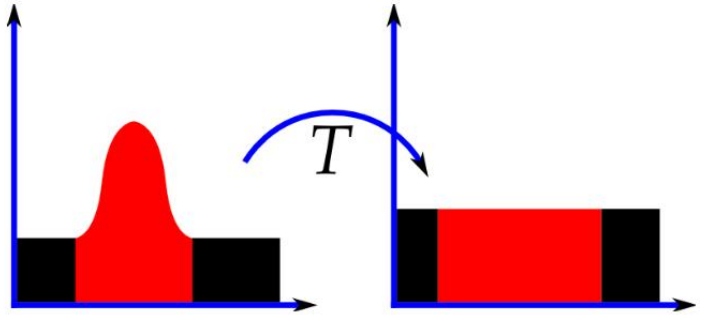
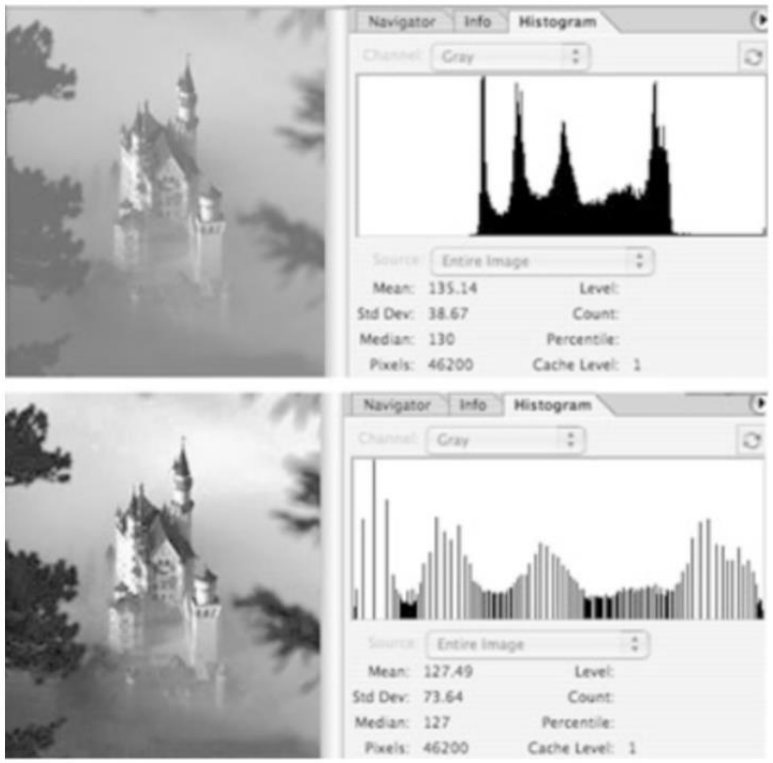


Fig. 1.5 Histograms for the 200 × 231 image `Neuschwanstein`. *Upper left*: Original image. *Upper right*: Brighter version. *Lower left*: Darker version. *Lower right*: After histogram equalization (will be defined later)

Source: Reinhard Klette, Concise Computer Vision, Springer

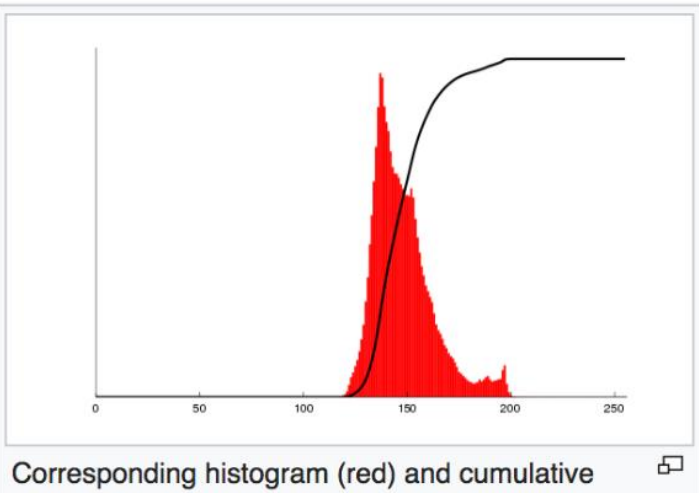
Выравнивание гистограммы





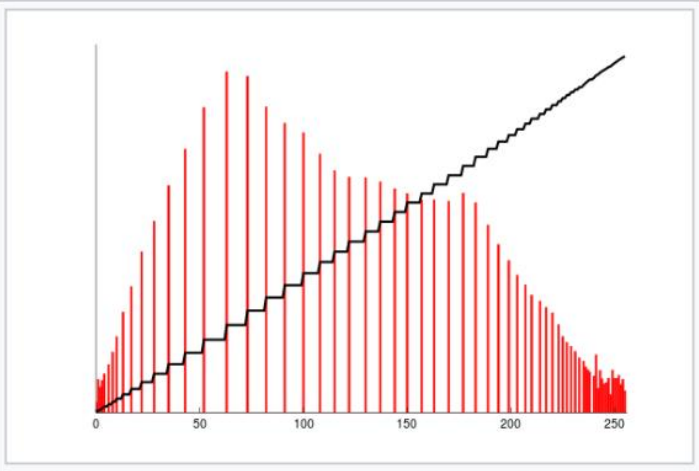
Выравнивание гистограммы

Before Histogram Equalization



Corresponding histogram (red) and cumulative histogram (black)

After Histogram Equalization

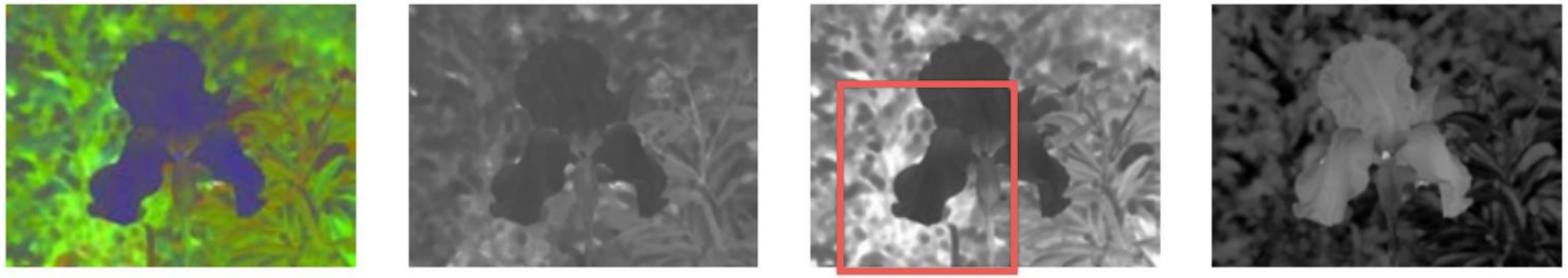


Corresponding histogram (red) and cumulative histogram (black)

https://en.wikipedia.org/wiki/Histogram_equalization

Нормализация цвета

(a) RGB (b) R (c) G (d) B



(e) rgb (f) r (g) g (h) b

$$r = \frac{R}{R+G+B}, \quad g = \frac{G}{R+G+B}, \quad b = \frac{B}{R+G+B}$$

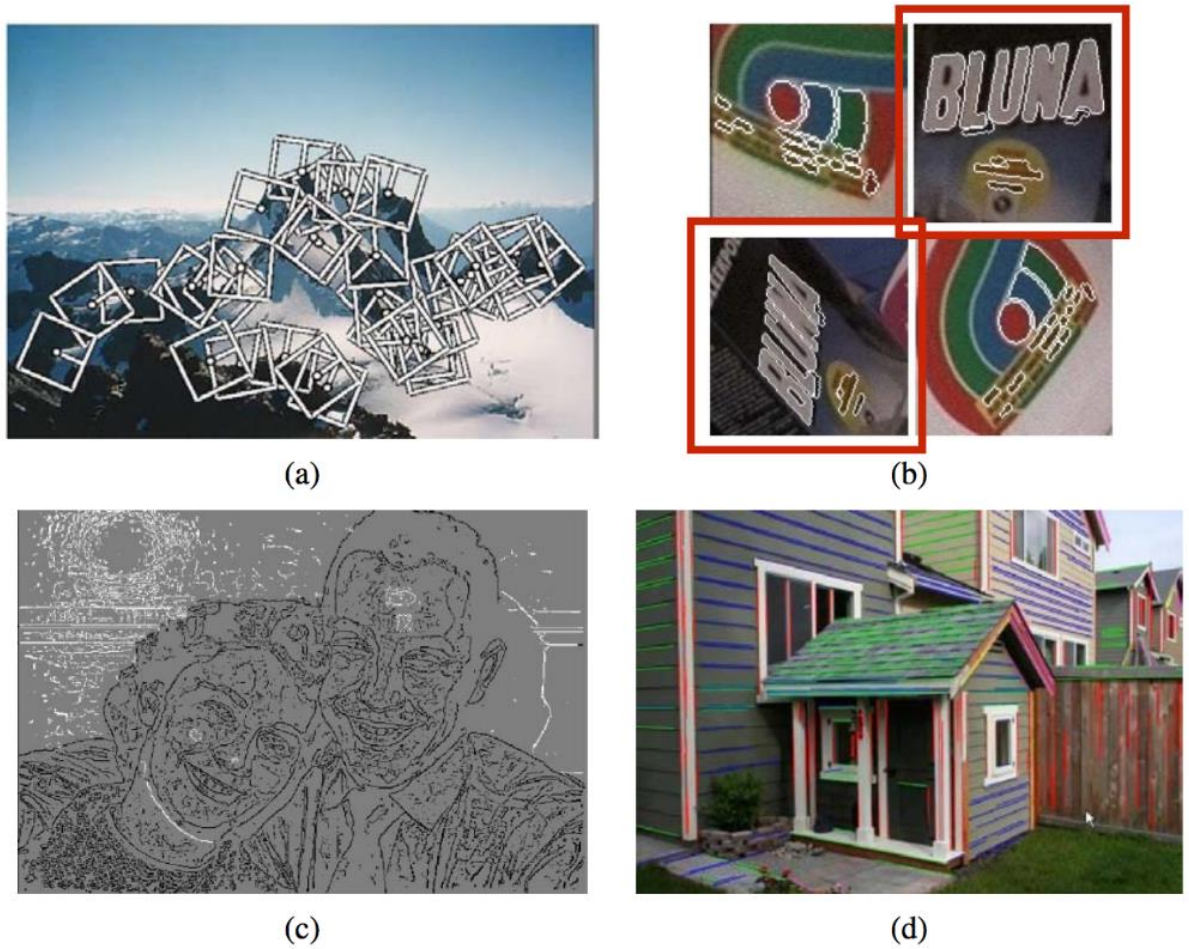
https://en.wikipedia.org/wiki/Histogram_equalization

Feature detection

Точки интереса

К интересным точкам относятся края, углы, пятна, участки, выступы, текстуры.

Основная цель – сравнить картинки.

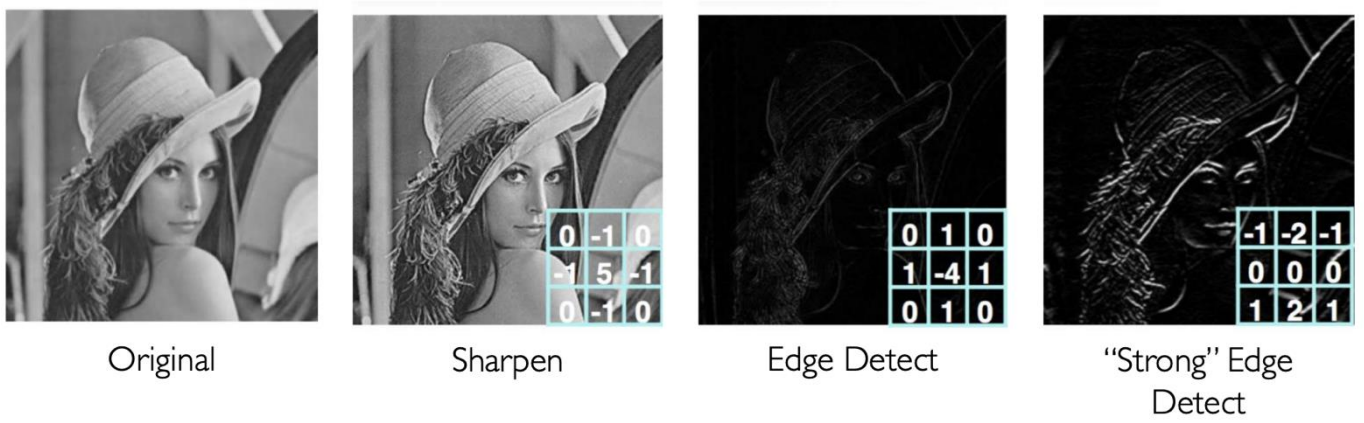


http://szeliski.org/Book/drafts/SzeliskiBook_20100903_draft.pdf

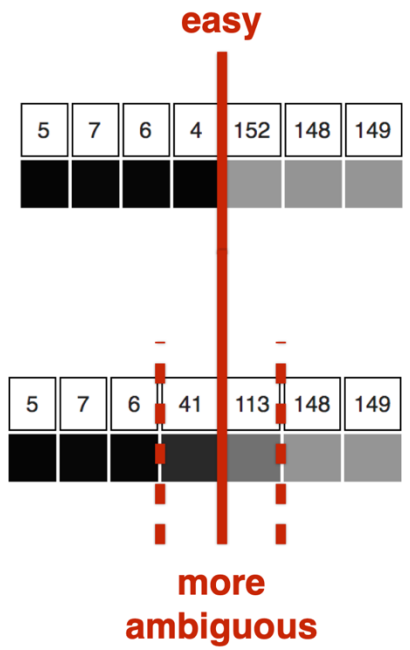
Точки интереса

К интересным точкам относятся края, углы, пятна, участки, выступы, текстуры.

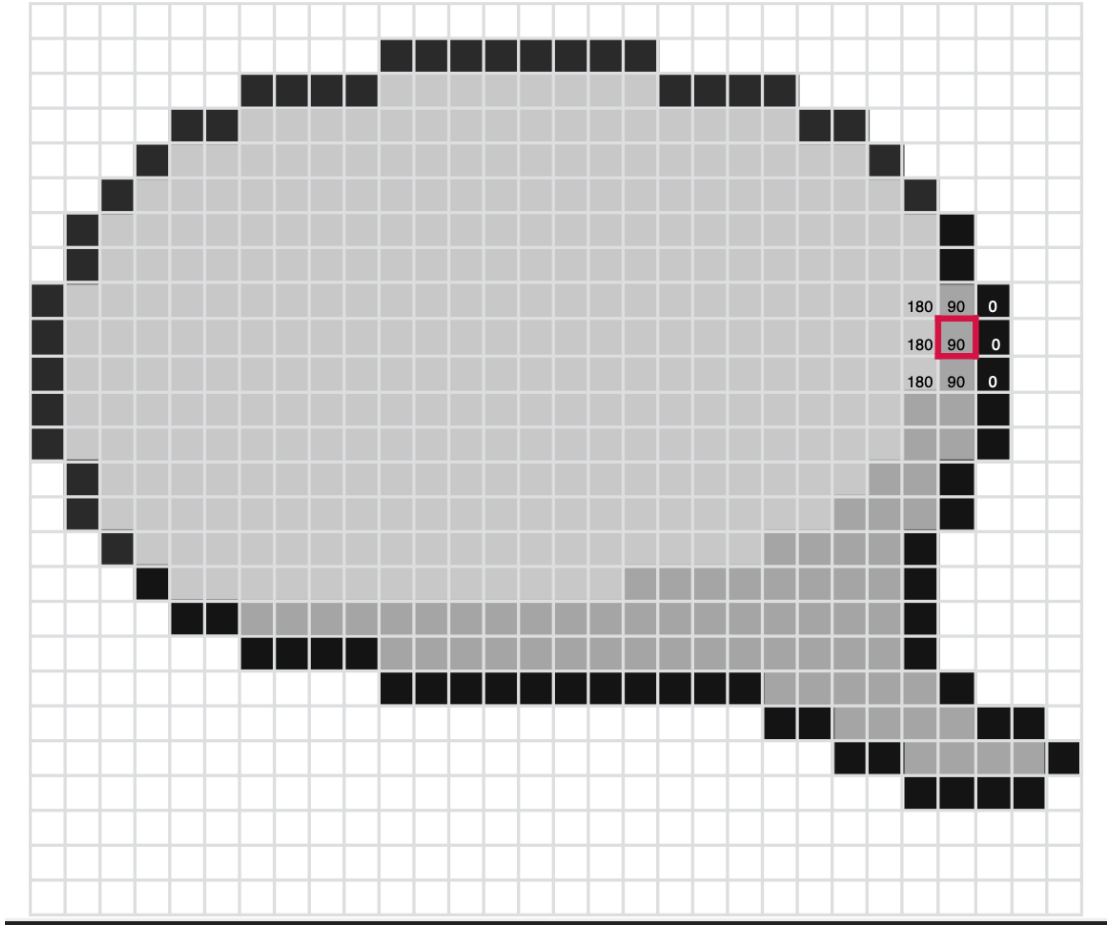
Основная цель – сравнить картинки.



Original | Sharpen | Edge Detect | "Strong" Edge Detect

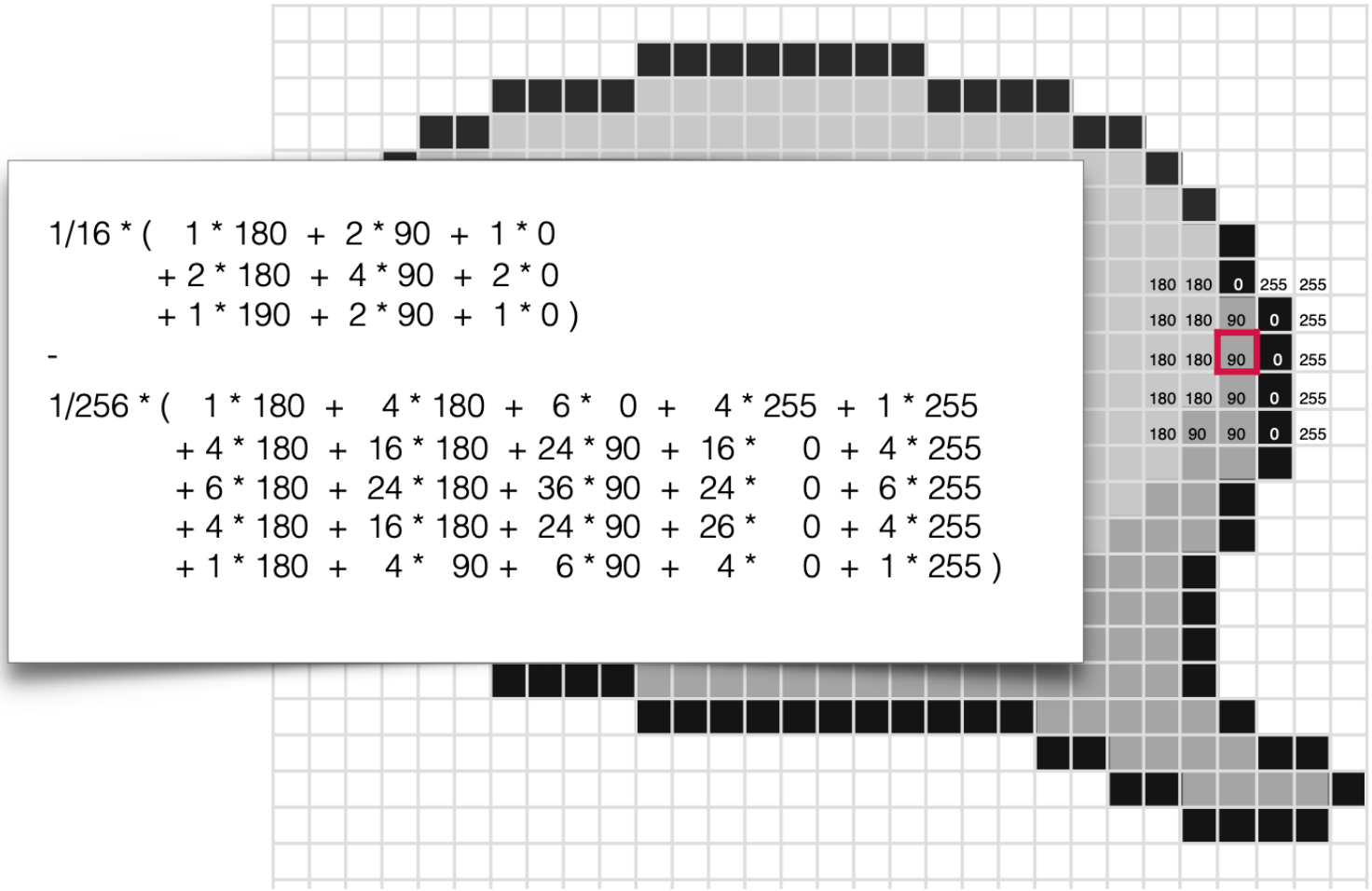


Difference of a Gaussian



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

Difference of a Gaussian

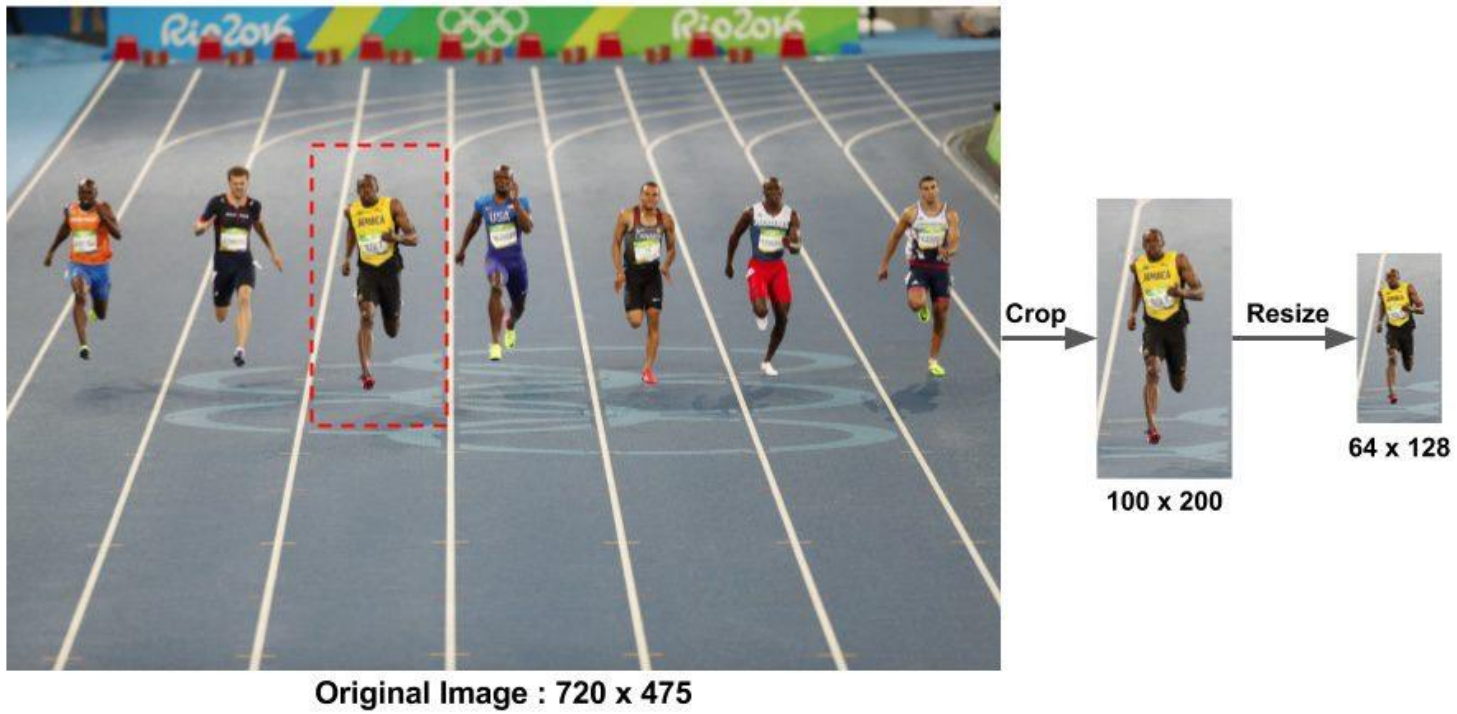


$$\frac{1}{16}\begin{bmatrix} 1 & 2 & 1 \\ 2 & 4 & 2 \\ 1 & 2 & 1 \end{bmatrix}$$

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

Difference of a Gaussian

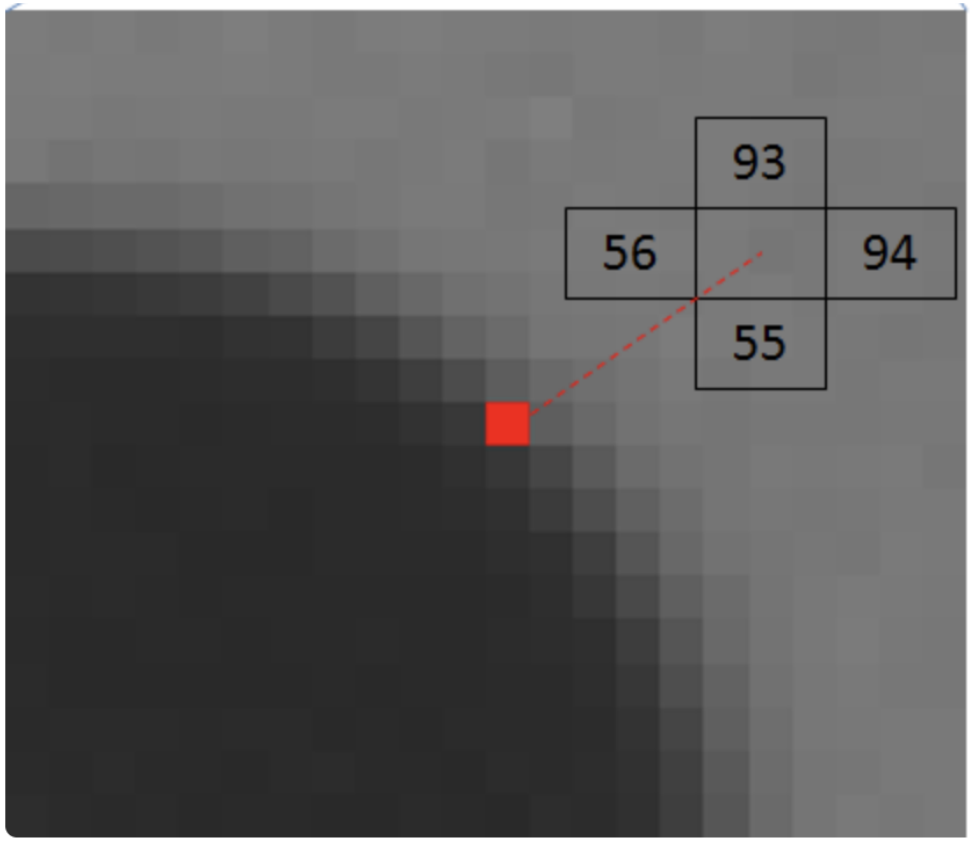
HOG (Histogram of Oriented Gradients)



1. Предобработка

https://learnopencv.com/histogram-of-oriented-gradients/

HOG (Histogram of Oriented Gradients)



93 - 55 = 38 in the y-direction.

2. Вычислить вектор градиента каждого пикселя..

-1	0	1

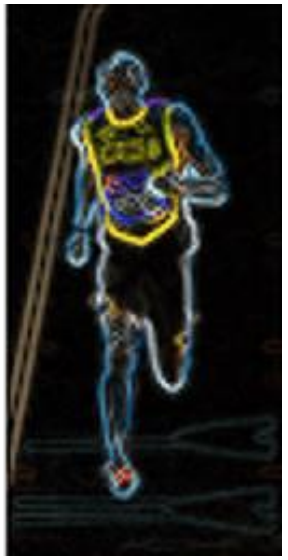
-1
0
1

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

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

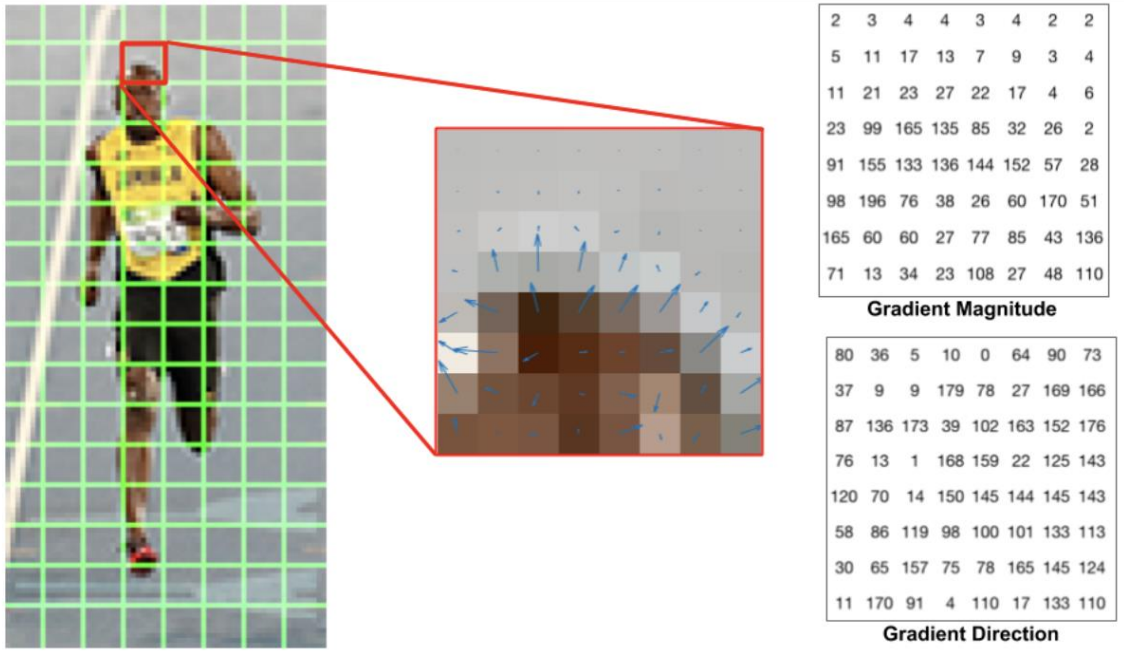
Горизонтальные градиенты

Вертикальные градиенты



Абсолютное значение

HOG (Histogram of Oriented Gradients)



2	3	4	4	3	4	2	2
5	11	17	13	7	9	3	4
11	21	23	27	22	17	4	6
23	99	165	135	85	32	26	2
91	155	133	136	144	152	57	28
98	196	76	38	26	60	170	51
165	60	60	27	77	85	43	136
71	13	34	23	108	27	48	110

Gradient Magnitude

80	36	5	10	0	64	90	73
37	9	9	179	78	27	169	166
87	136	173	39	102	163	152	176
76	13	1	168	159	22	125	143
120	70	14	150	145	144	145	143
58	86	119	98	100	101	133	113
30	65	157	75	78	165	145	124
11	170	91	4	110	17	133	110

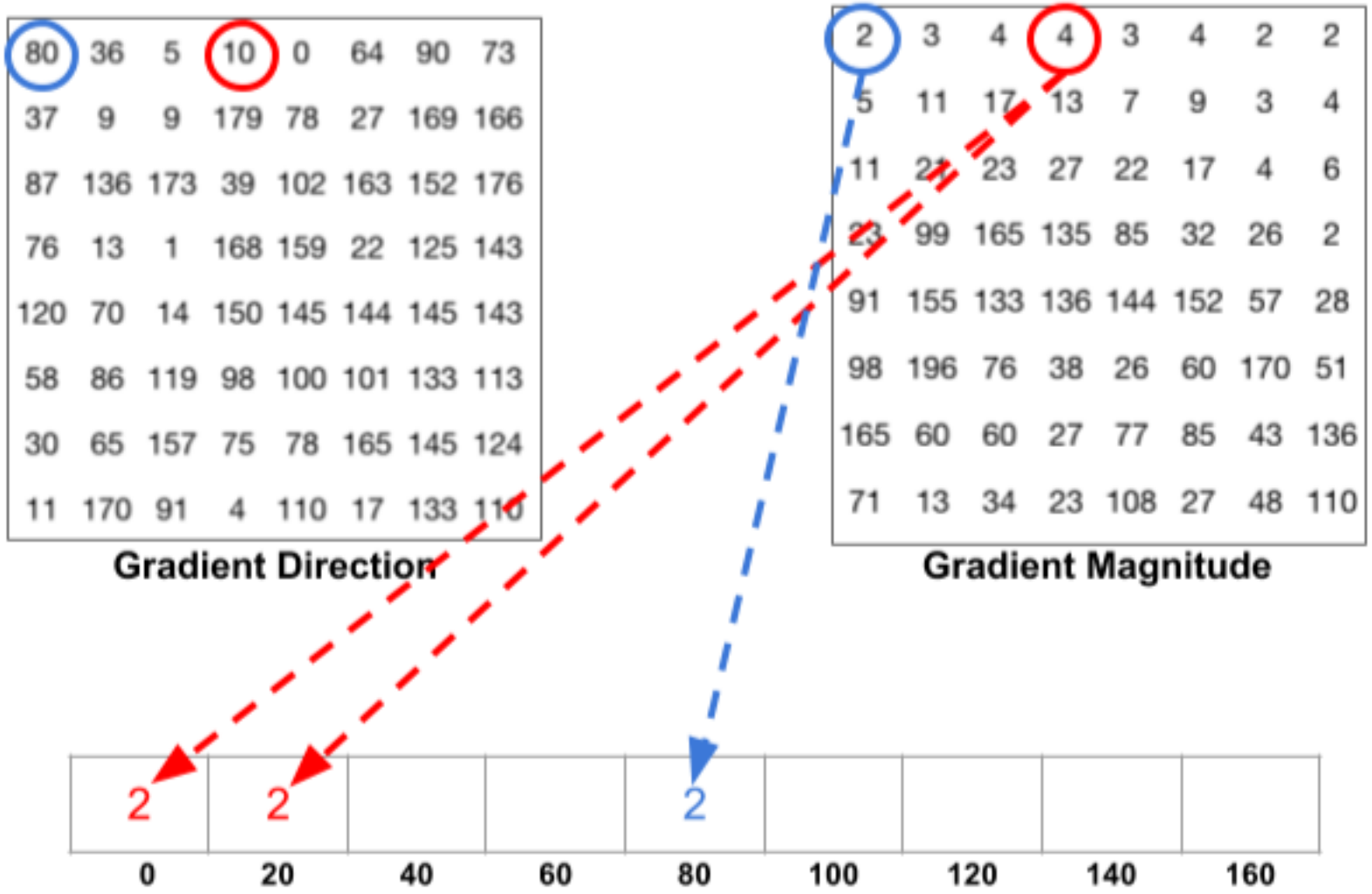
Gradient Direction

2. Вычислить вектор градиента каждого пикселя..
...и гистограммы градиента участков 8*8

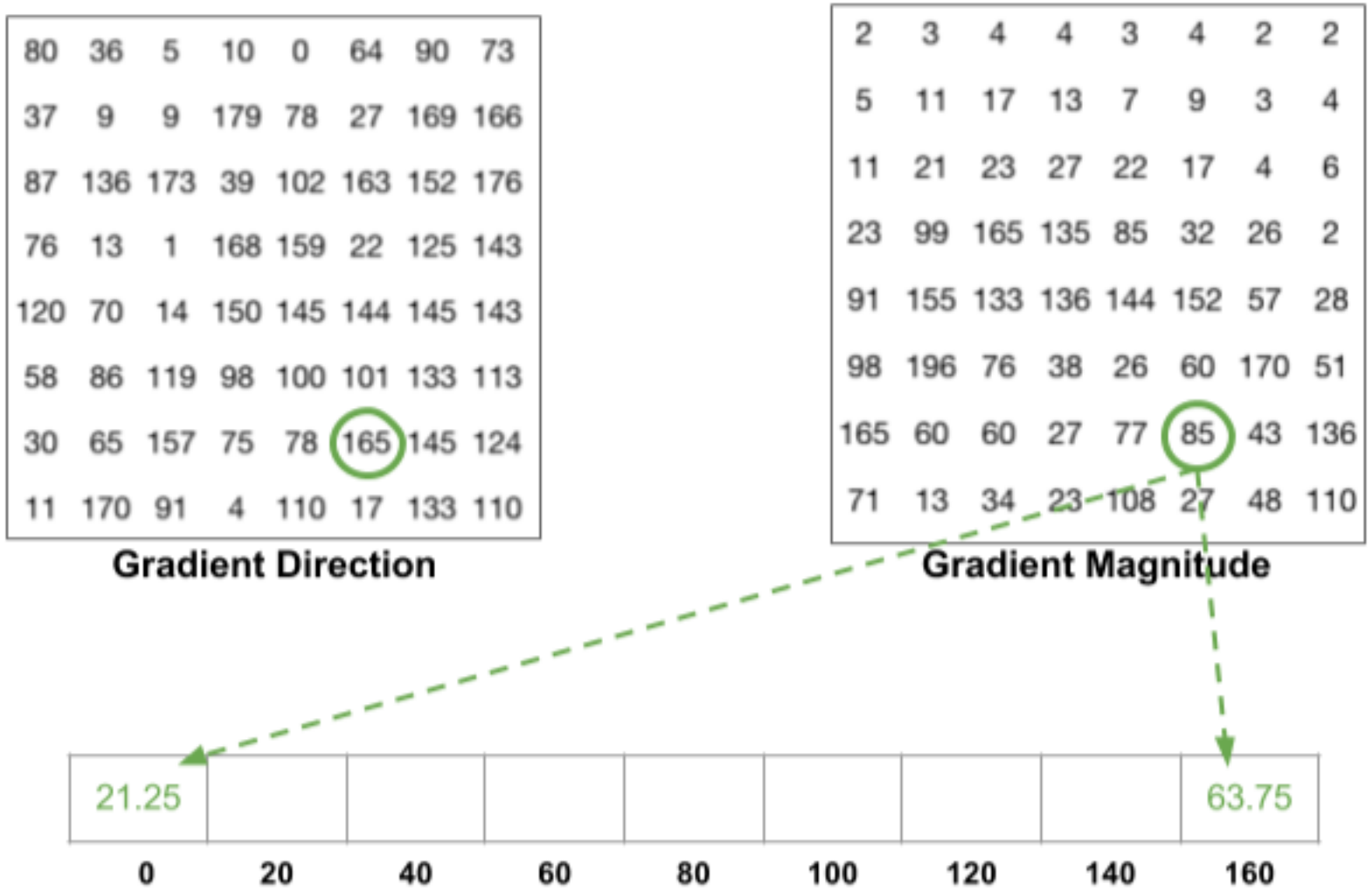
8*8 гиперпараметр, взят исходя из предыдущего опыта анализа картинок

Направления градиентов [0;180]

HOG (Histogram of Oriented Gradients)

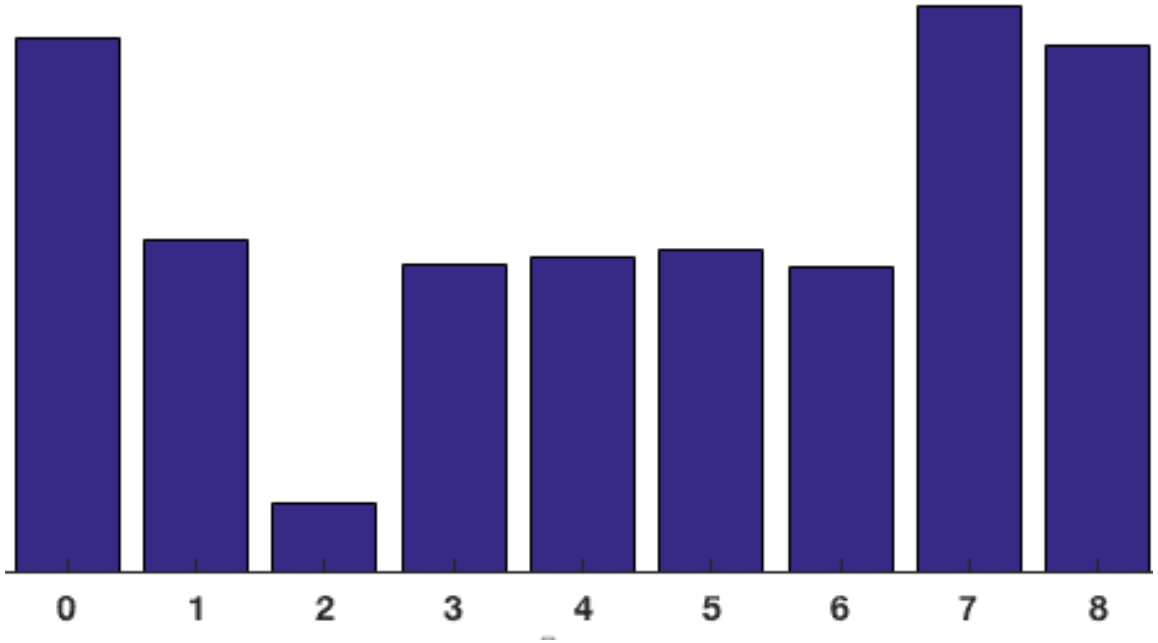


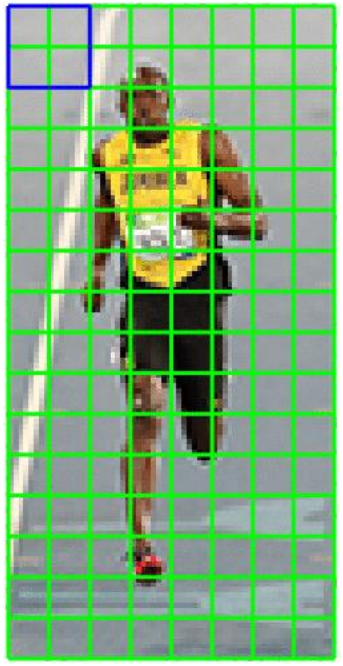
HOG (Histogram of Oriented Gradients)



HoG Descriptor

Получим гистограмму





4. Блочно нормализуем цвета

https://en.wikipedia.org/wiki/Histogram_equalization

HoG Descriptor

5. Конкатенируем вектора из каждого блока.

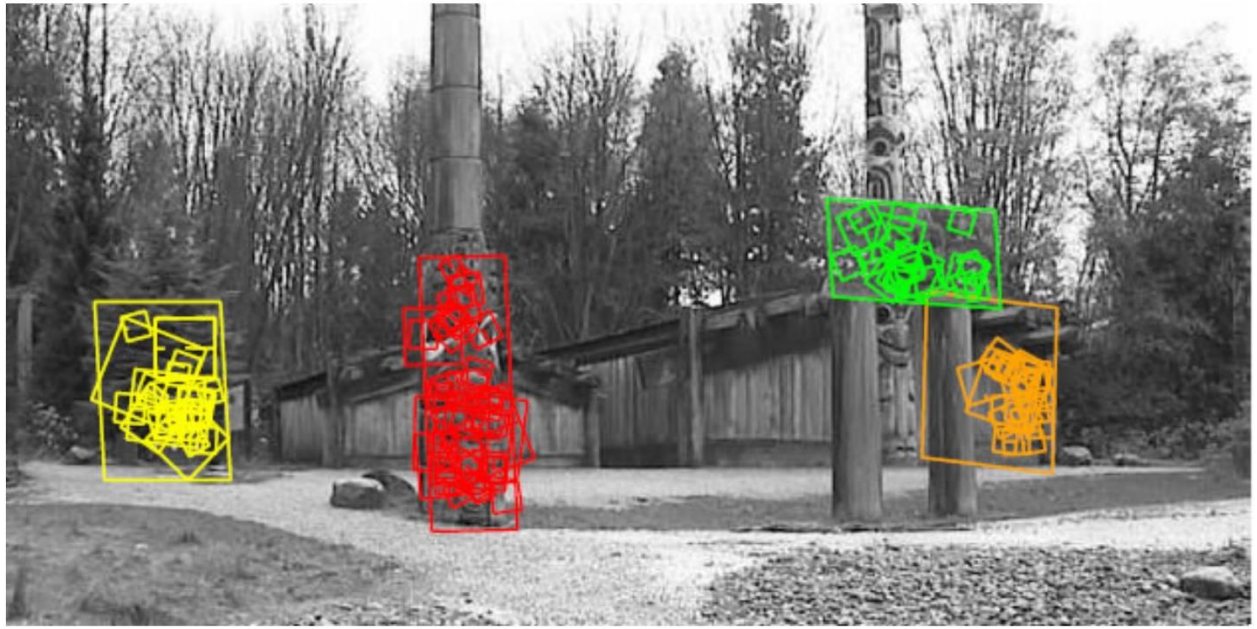
Получается:
7*15 блоков 16*16
36*1 вектор
Итого 3780-размерный вектор, который содержит информацию о признаках.

https://en.wikipedia.org/wiki/Histogram_equalization

SIFT (Scale Invariant Feature Transform)

Масштабно-инвариантный означает, что описание SIFT не меняется с:

- Масштабом
- Вращением
- Освещением
- Точкой обзора (аффинные искажения)



Source: http://aishack.in/tutorials/sift-scale-invariant-feature-transform-introduction/

Алгоритм SIFT

1. **Constructing a scale space** This is the initial preparation. You create internal representations of the original image to ensure scale invariance. This is done by generating a "scale space".

2. **LoG Approximation** The Laplacian of Gaussian is great for finding interesting points (or key points) in an image. But it's computationally expensive. So we cheat and approximate it using the representation created earlier.

3. **Finding keypoints** With the super fast approximation, we now try to find key points. These are maxima and minima in the Difference of Gaussian image we calculate in step 2

4. **Get rid of bad key points** Edges and low contrast regions are bad keypoints. Eliminating these makes the algorithm efficient and robust. A technique similar to the Harris Corner Detector is used here.

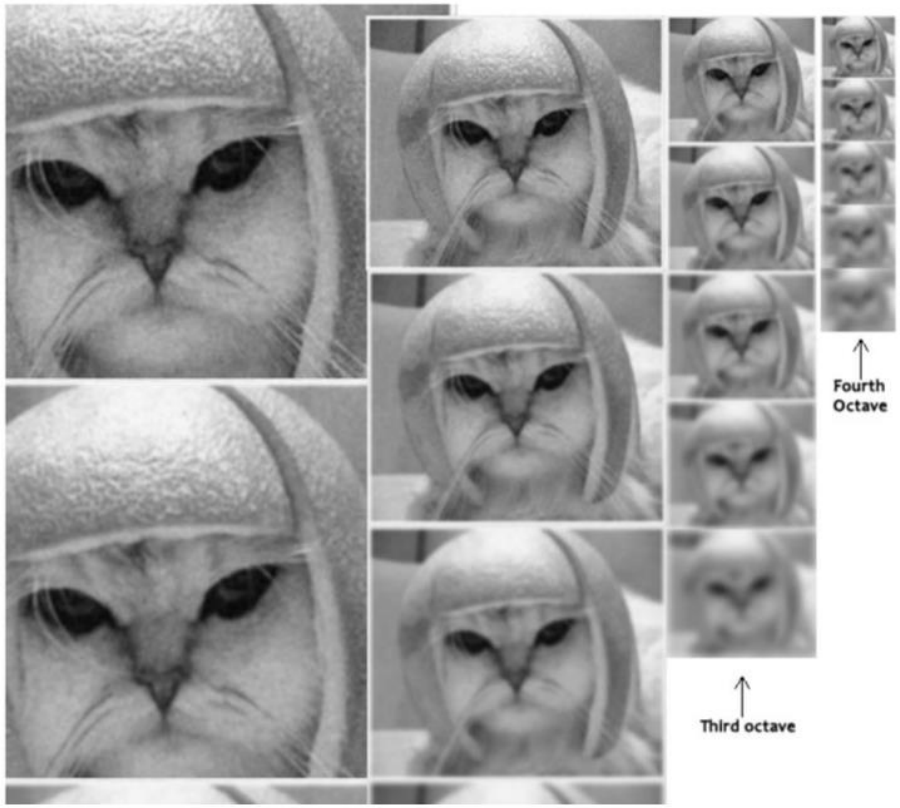
5. **Assigning an orientation to the keypoints** An orientation is calculated for each key point. Any further calculations are done relative to this orientation. This effectively cancels out the effect of orientation, making it rotation invariant.

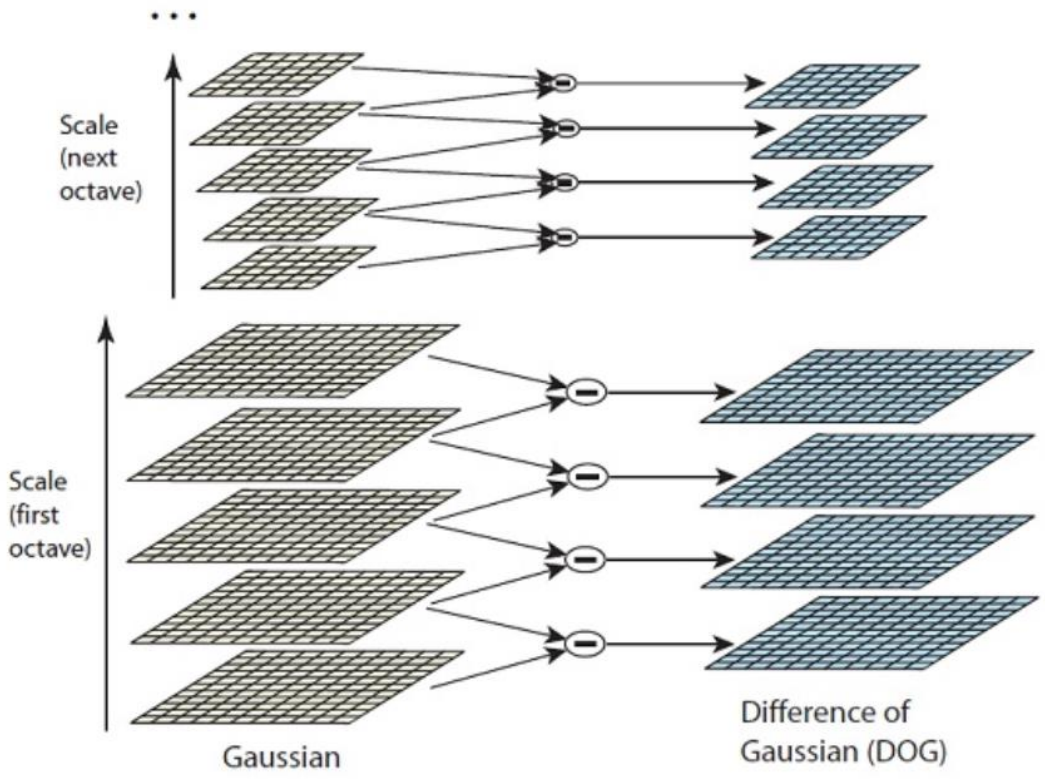
6. **Generate SIFT features** Finally, with scale and rotation invariance in place, one more representation is generated. This helps uniquely identify features. Lets say you have 50,000 features. With this representation, you can easily identify the feature you're looking for (say, a particular eye, or a sign board).

https://aishack.in/tutorials/sift-scale-invariant-feature-transform-introduction/

Алгоритм SIFT

1. **Constructing a scale space** This is the initial preparation. You create internal representations of the original image to ensure scale invariance. This is done by generating a "scale space".
2. **LoG Approximation** The Laplacian of Gaussian is great for finding interesting points (or key points) in an image. But it's computationally expensive. So we cheat and approximate it using the representation created

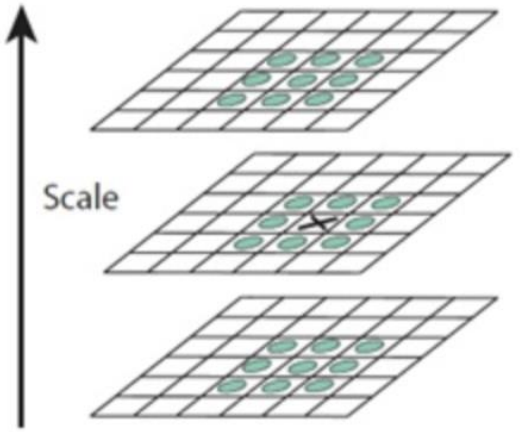




Алгоритм SIFT

Finding keypoints With the super fast approximation, we now try to find key points. These are maxima and minima in the Difference of Gaussian image we calculate in step 2

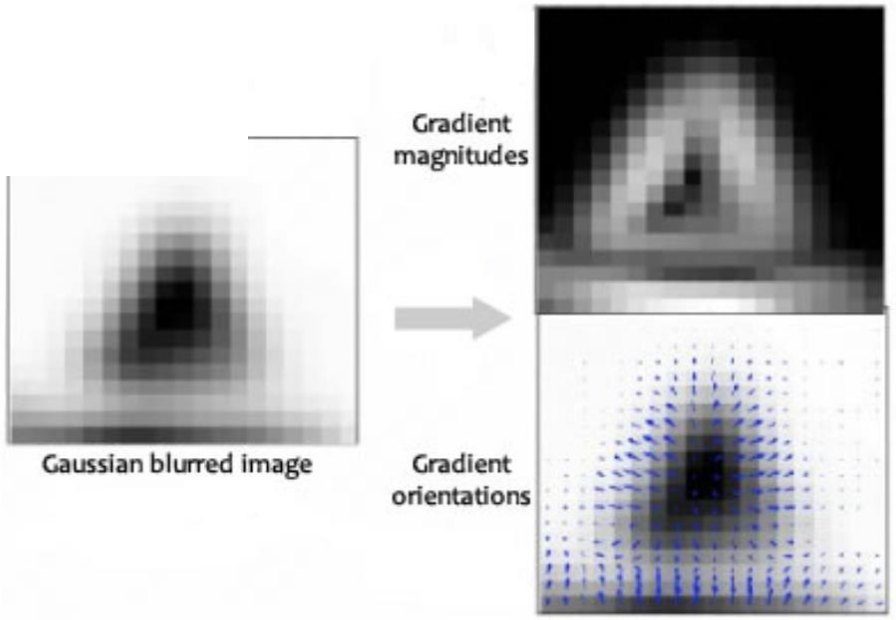
Get rid of bad key points Edges and low contrast regions are bad keypoints. Eliminating these makes the algorithm efficient and robust. A technique similar to the Harris Corner Detector is used here

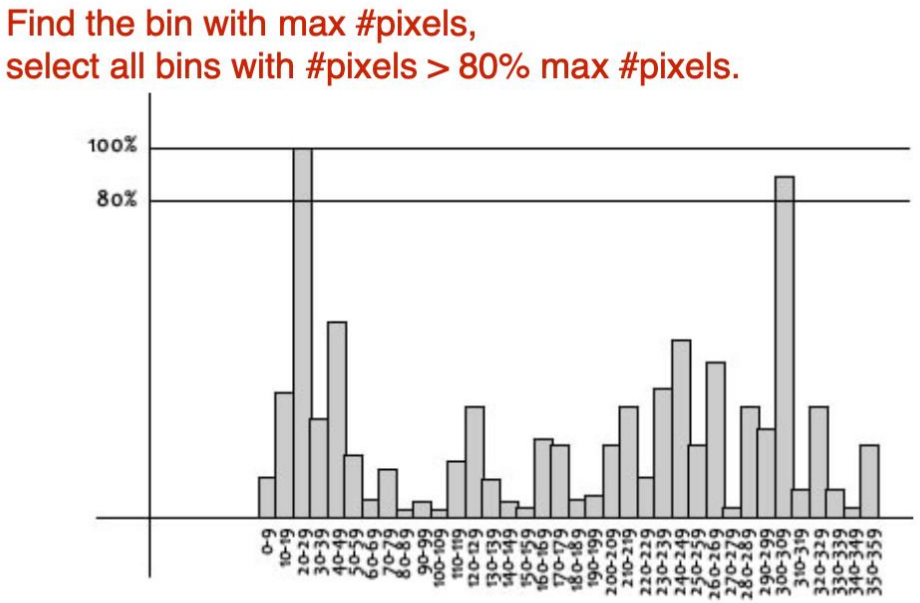


X marks the current pixel. The green circles mark the neighbours.

Алгоритм SIFT

Затем мы вычисляет HOG и определяем точки интереса для каждой ориентации пика. Ориентации пиков используются для сравнения повернутых изображений.





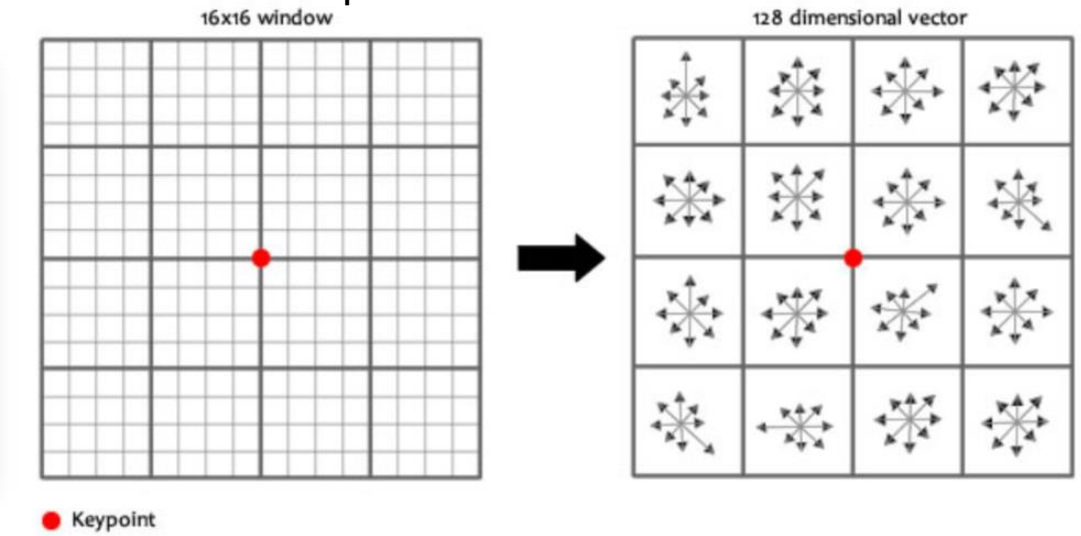
Assigning an orientation to the keypoints An orientation is calculated for each key point. Any further calculations are done relative to this orientation. This effectively cancels out the effect of orientation, making it rotation invariant.

Алгоритм SIFT

Наконец, HOG кодируются вокруг ключевых точек.

+ Вычесть ключевую ориентацию (для инвариантности ориентации)
SEP
+ Normalize gradient magnitude SEP (for illumination invariance)

16*16 окно разбивается на 16 4*4 окна



Generate SIFT features Finally, with scale and rotation invariance in place, one more representation is generated. This helps uniquely identify features. Lets say you have 50,000 features. With this representation, you can easily identify the feature you're looking for (say, a particular eye, or a sign board).

SIFT результаты

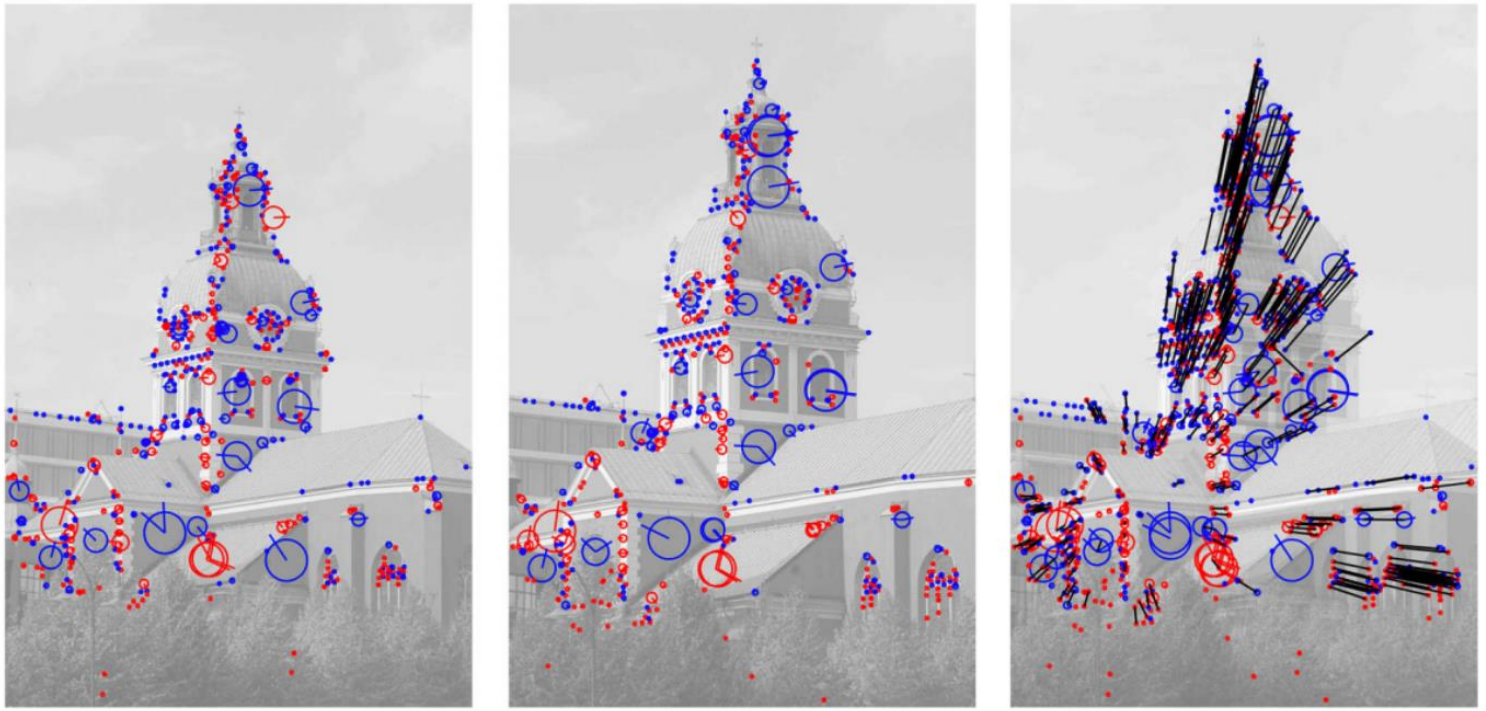


Fig. 1 Illustration of image matching using Laplacian interest points with locally adapted SIFT descriptors computed around each interest point. (*left*) and (*middle*) Two images of a building in downtown Stockholm taken from different 3-D positions with the interest points shown as *circles* overlaid on a bright copy of the original image with the size of each *circle* proportional to the locally adapted scale estimate and with the orientation estimate used for aligning the orientation of the SIFT descriptor drawn as an *angular line* from the center of the interest point. The colour of the *circle* indicates the polarity of the interest point, with *red* corresponding to *bright blobs* and *blue* corresponding to *dark blobs*. (*right*) Matching relations between the interest points drawn as *black lines* on *top* of a superposition of the original grey-level images

Сегментация

Сегментация - постановка

Сегментация — это поиск согласованных областей на изображении.

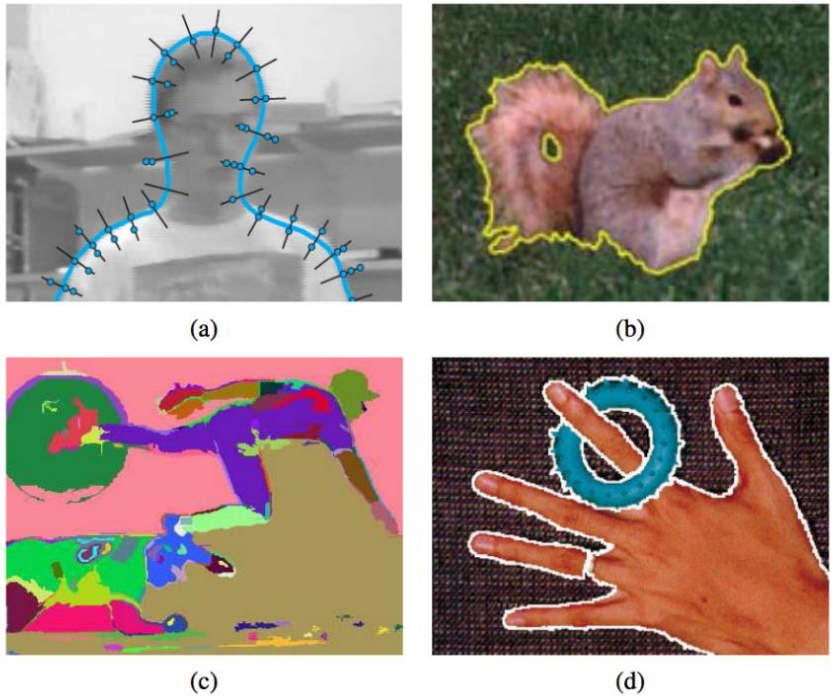
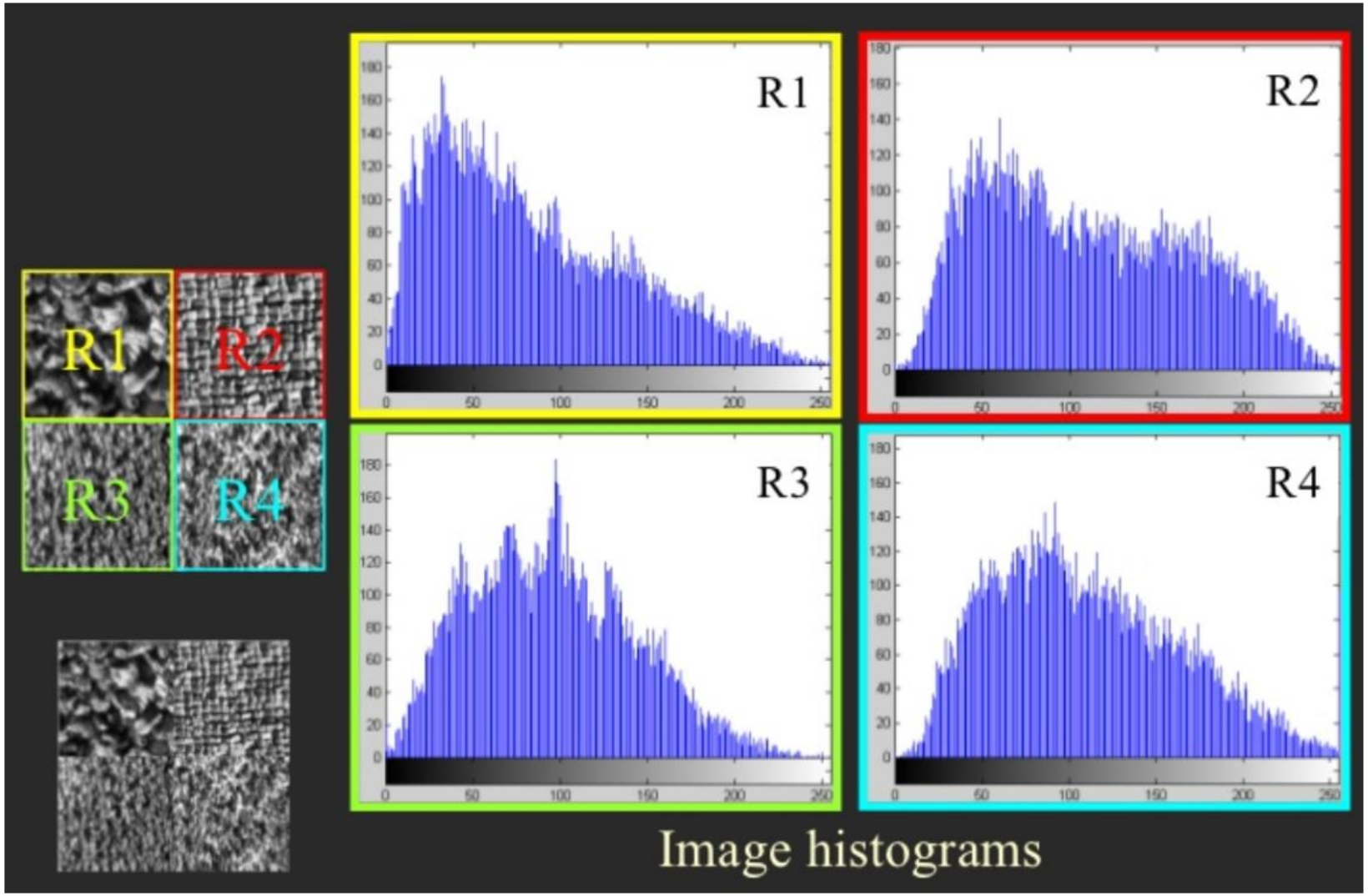


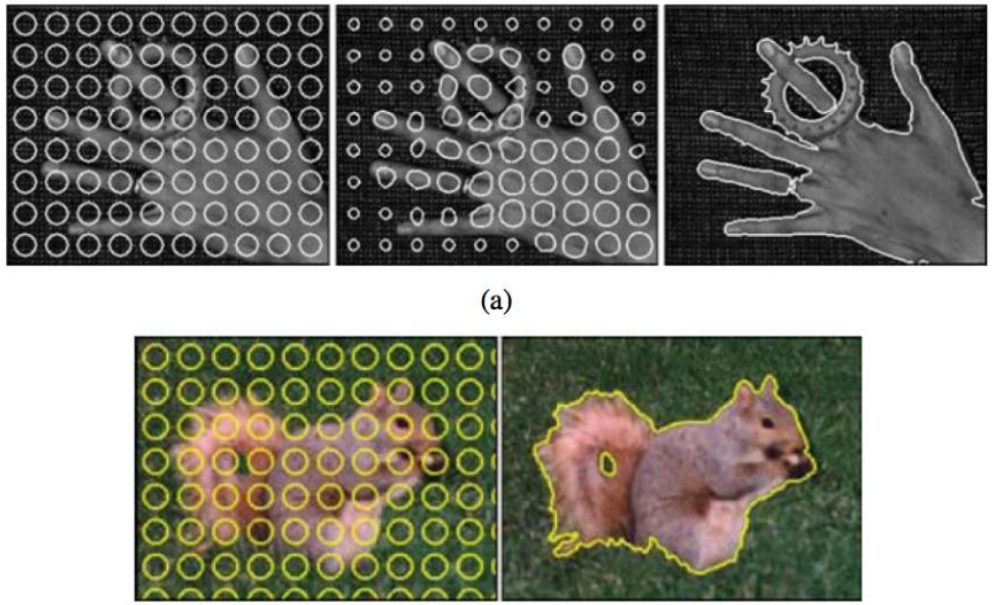
Figure 5.1 Some popular image segmentation techniques: (a) active contours (Isard and Blake 1998) © 1998 Springer; (b) level sets (Cremers, Rousson, and Deriche 2007) © 2007 Springer; (c) graph-based merging (Felzenszwalb and Huttenlocher 2004b) © 2004 Springer; (d) mean shift (Comaniciu and Meer 2002) © 2002 IEEE; (e) texture and intervening contour-based normalized cuts (Malik, Belongie, Leung *et al.* 2001) © 2001 Springer; (f) binary MRF solved using graph cuts (Boykov and Funka-Lea 2006) © 2006 Springer.

Сегментация – статистика



Сегментация без классификации

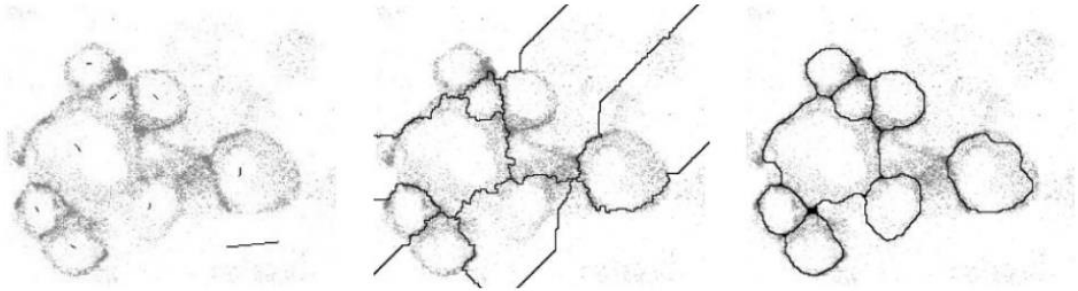
Часто подходы базируются на сравнении соседних регионов



(a)

(b)

Figure 5.11 Level set segmentation (Cremers, Rousson, and Deriche 2007) © 2007 Springer: (a) grayscale image segmentation and (b) color image segmentation. Uni-variate and multi-variate Gaussians are used to model the foreground and background pixel distributions. The initial circles evolve towards an accurate segmentation of foreground and background, adapting their topology as they evolve.

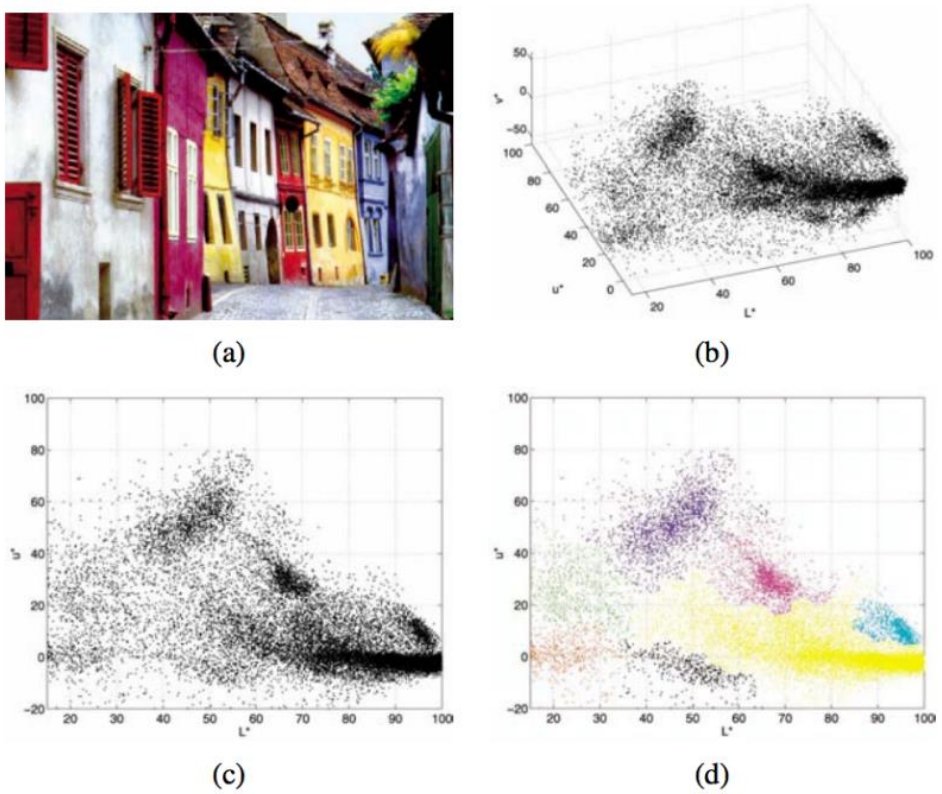


(a) (b) (c)

Figure 5.13 Locally constrained watershed segmentation (Beare 2006) © 2006 IEEE: (a) original confocal microscopy image with marked seeds (line segments); (b) standard watershed segmentation; (c) locally constrained watershed segmentation.

Сегментация без учителя

Кластеризация и нейронные сети* могут использоваться для группировки похожих участков.
*например, самоорганизующиеся карты (SOM)



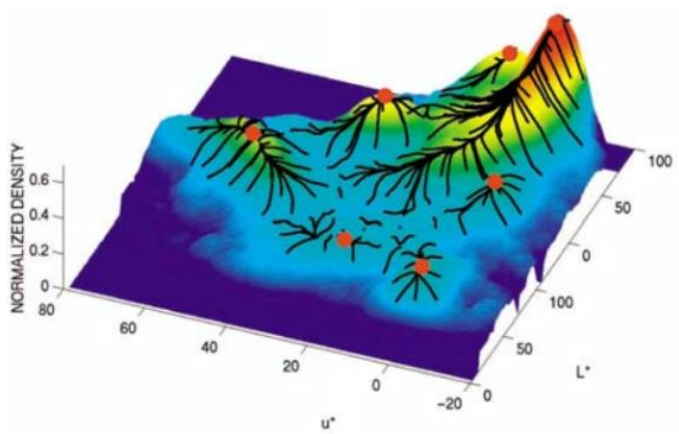
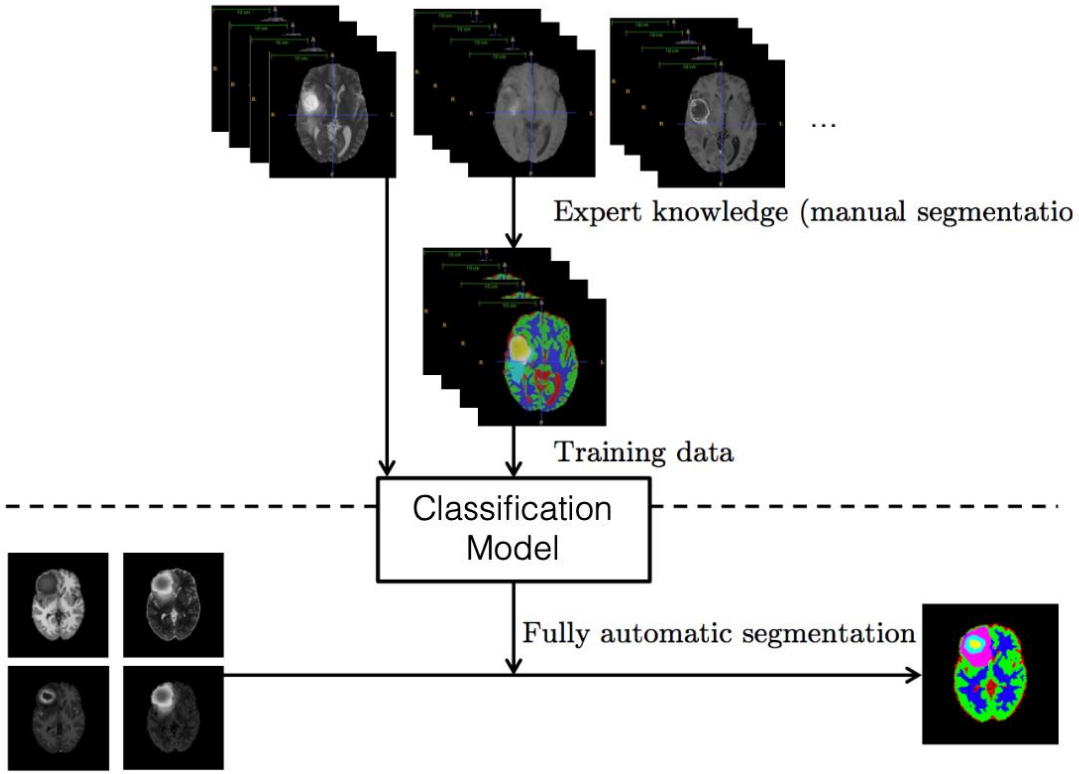


Figure 5.16 Mean-shift image segmentation (Comaniciu and Meer 2002) © 2002 IEEE: (a) input color image; (b) pixels plotted in L*u*v* space; (c) L*u* space distribution; (d) clustered results after 159 mean-shift procedures; (e) corresponding trajectories with peaks marked as red dots.

Сегментация с учителем

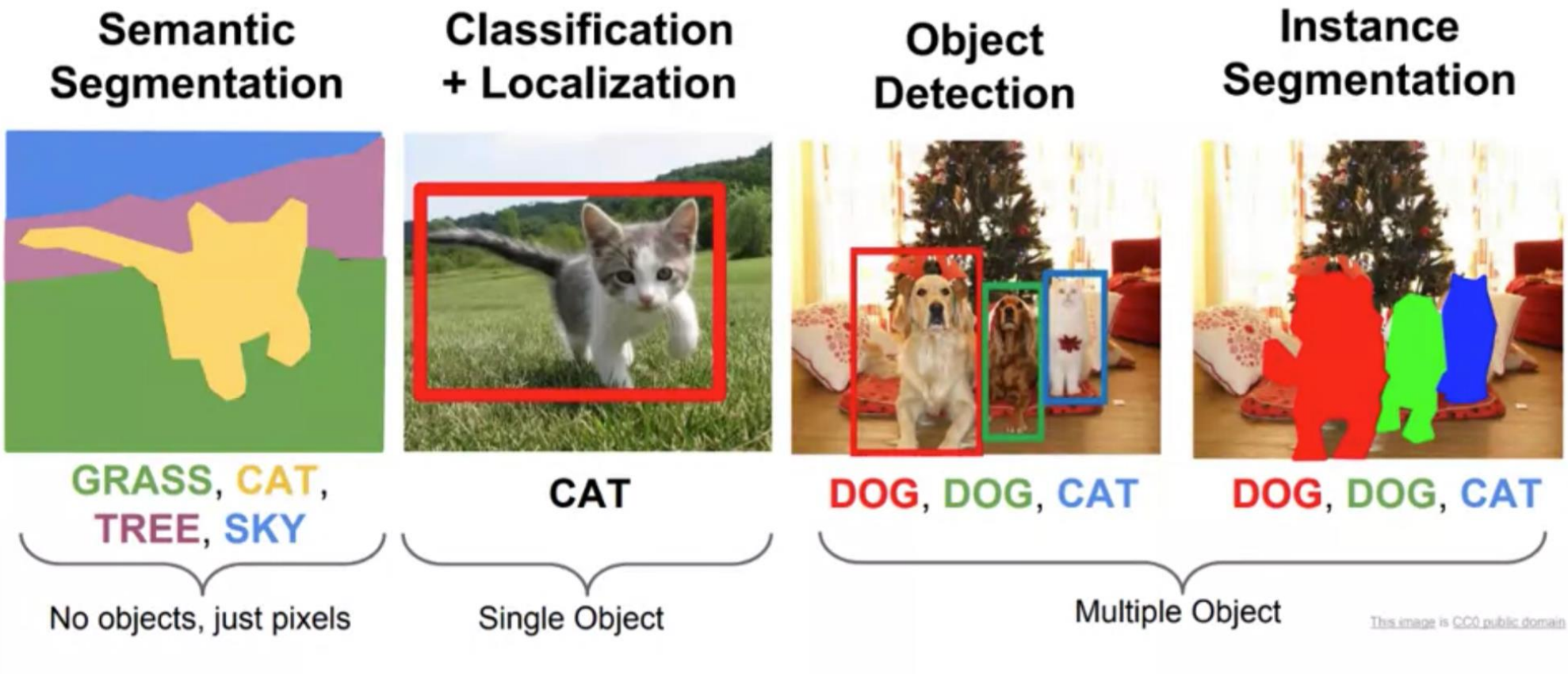
К размеченным данным можно применять многие методы классификации.

Применяются стандартные методы обучения с учителем.

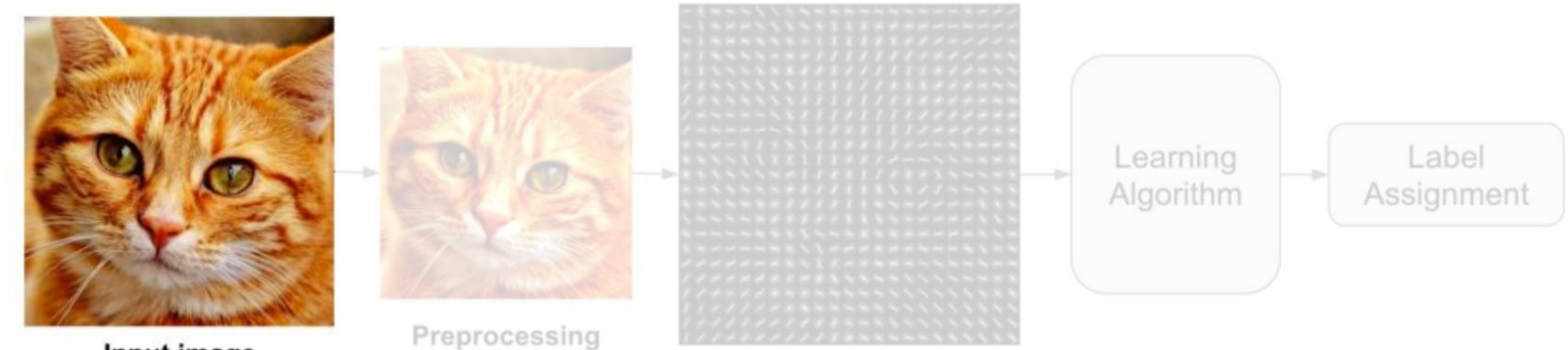


Обнаружение объектов

Возможные задачи

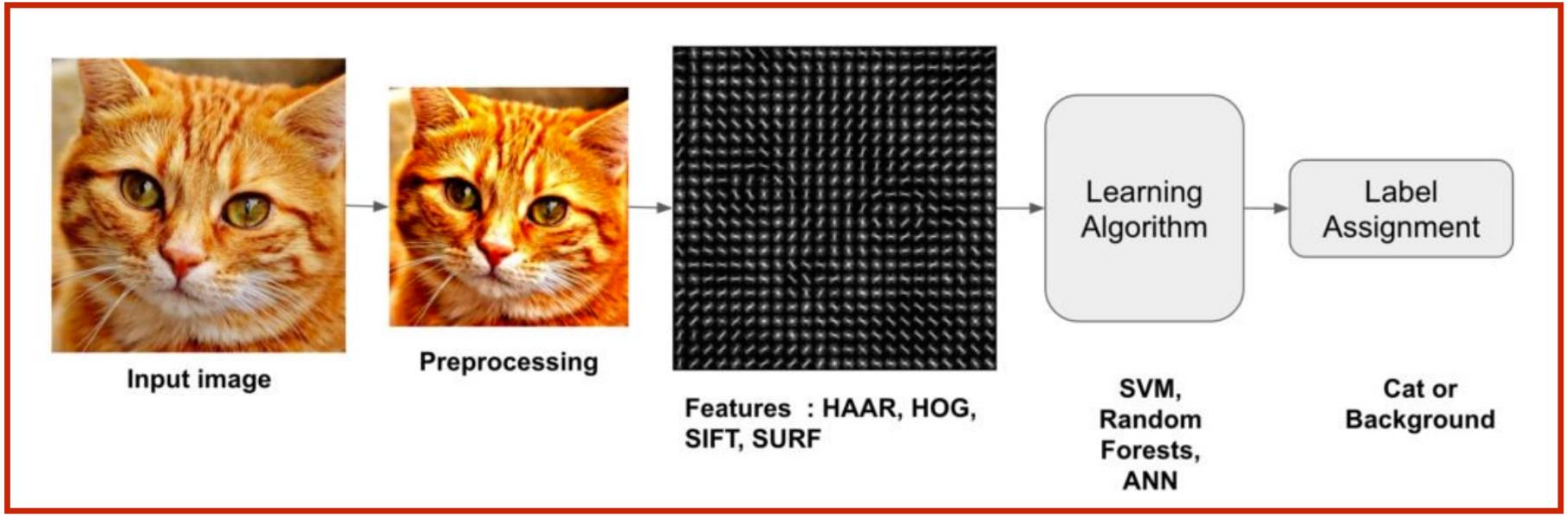


Стандартный пайплайн



From segmentation or object detection
(classifying segments or pixels as background or not)

Стандартный пайплайн



This pipeline is also used for segmentation or object detection

…but these can be unsupervised classification

Проблемы

- Окклюзии могут скрывать ключевые части объектов.
- Точки обзора могут не показывать все ключевые части объектов.
- Освещение может значительно изменить внешний вид объектов (например, тени).
- Изменение формы можно ожидать от многих объектов. (например, тело может изгибаться)
- Внутриклассовые различия могут быть значительными. (например, стулья могут сильно отличаться друг от друга)

Подходы

- Нейронные сети и глубокое обучение являются наиболее распространенными методами.
- Опорные векторные машины (SVM) изучают границы между классами в выбранном пространстве признаков.
- Бустинг объединяет слабые (крайне простые) классификаторы.

Аугментация

Базовый подход

- Для правильного обучения моделям глубокого обучения обычно требуется много данных.
- Часто бывает полезно получить больше данных из существующих моделей с помощью методов дополнения данных.

Original	Flip	Rotation	Random crop
			
• Image without any modification	• Flipped with respect to an axis for which the meaning of the image is preserved	• Rotation with a slight angle • Simulates incorrect horizon calibration	• Random focus on one part of the image • Several random crops can be done in a row

Color shift	Noise addition	Information loss	Contrast change
			
• Nuances of RGB is slightly changed • Captures noise that can occur with light exposure	• Addition of noise • More tolerance to quality variation of inputs	• Parts of image ignored • Mimics potential loss of parts of image	• Luminosity changes • Controls difference in exposition due to time of day

This X Does Not Exist!

This Person Does Not Exist

The site that started it all, with the name that says it all. Created using a style-based generative adversarial network (StyleGAN), this website had the tech community buzzing with excitement and intrigue and inspired many more sites.

Created by Phillip Wang.

This Cat Does Not Exist

These purr-fect GAN-made cats will freshen your feeline-gs and make you wish you could reach through your screen and cuddle them. Once in a while the cats have visual deformities due to imperfections in the model – beware, they can cause nightmares.

Created by Ryan Hoover.

This Rental Does Not Exist

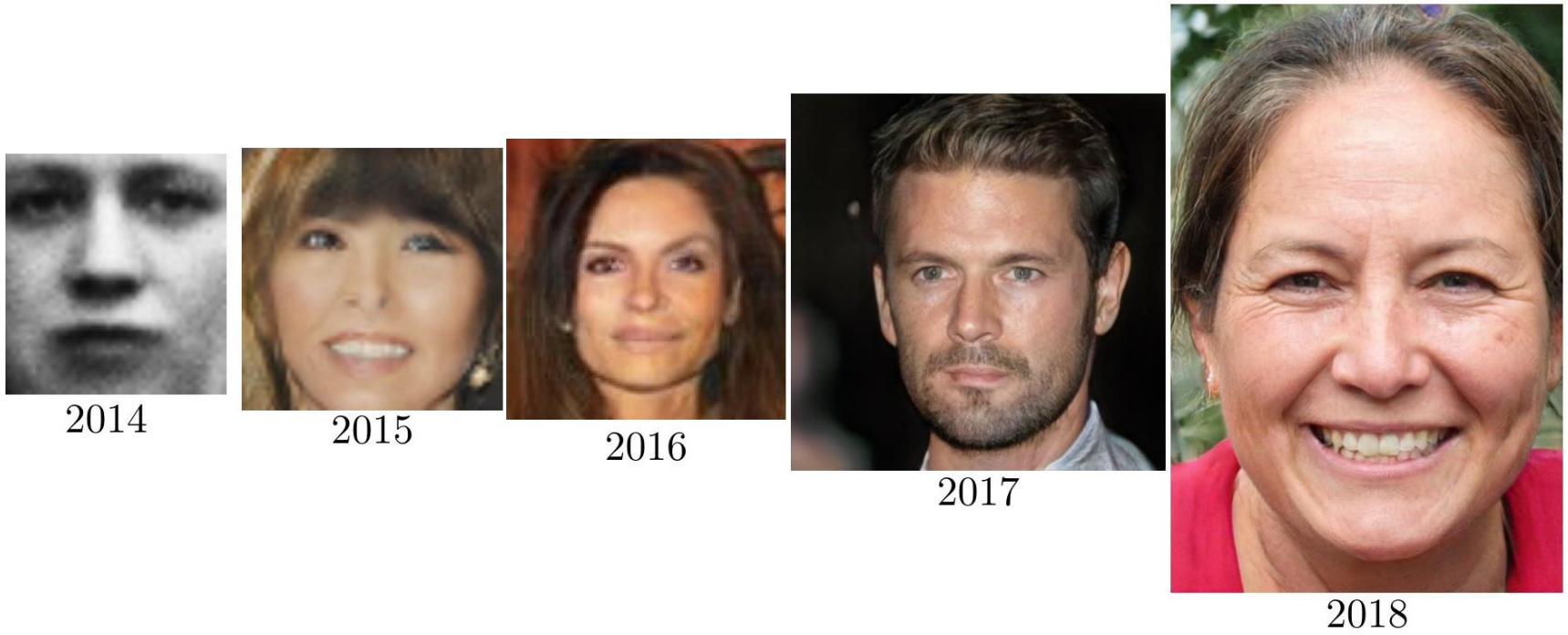
Why bother trying to look for the perfect home when you can create one instead? Just find a listing you like, buy some land, build it, and then enjoy the rest of your life.

Created by Christopher Schmidt.

https://thisxdoesnotexist.com/

Generative Models Progress

The news are well motivated.



- Enormous progress in recent years.
- Technology is ready for new tasks.

https://twitter.com/goodfellow_ian/status/1084973596236144640

Dirty Road Signs Generation

Class 0

Class 1

Class 2

Class 6

Class 7

Class 8

- Road signs from the book are too clean.
- Need to put mud and shadows on the signs.

https://arxiv.org/abs/1907.12902
https://www.hse.ru/sci/diss/426009543