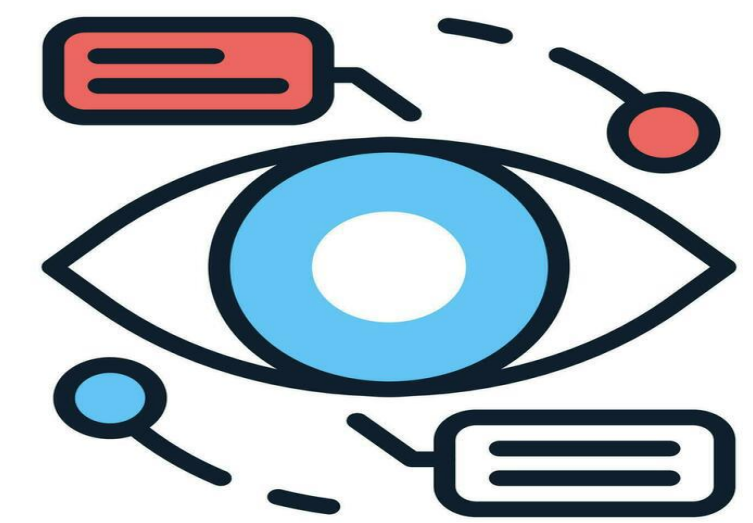


# **Основы искусственного интеллекта**

**Основы компьютерного зрения и его применения**

# Что такое компьютерное зрение?

- Междисциплинарная область, объединяющая информатику, искусственный интеллект, оптику и нейробиологию.
- **Цель Компьютерного Зрения:** Научить компьютеры "видеть" и интерпретировать визуальную информацию так же, как это делают люди.
- Компьютерное зрение стремится **к пониманию содержания изображения**, а не только к его модификации.
- **Краткий пример:** Распознавание кошки на фотографии.



# История компьютерного зрения

## Основные этапы

- Ранние эксперименты (1960-е - 1970-е): Простые задачи распознавания форм.
- Развитие алгоритмов обработки изображений (1980-е).
- Появление машинного обучения и первых нейронных сетей (1990-е- 2000-е).
- Революция глубокого обучения (2010-е - настоящее время): Прорыв в точности распознавания.



# Основные задачи компьютерного зрения

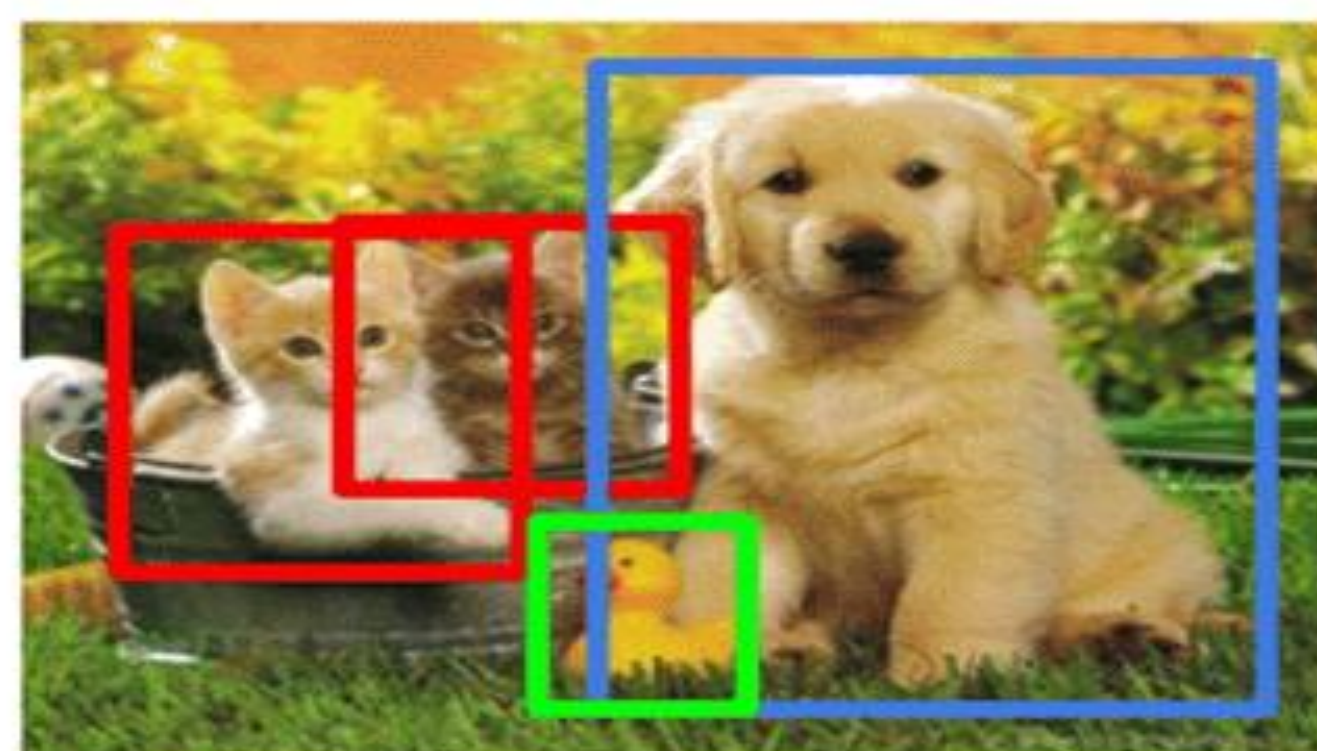
- **Классификация изображений:** Определение категории объекта на изображении (например, "кошка", "собака", "автомобиль").
- **Обнаружение объектов:** Нахождение и локализация нескольких объектов на изображении с помощью ограничивающих рамок (bounding boxes).

Классификация



КОТ

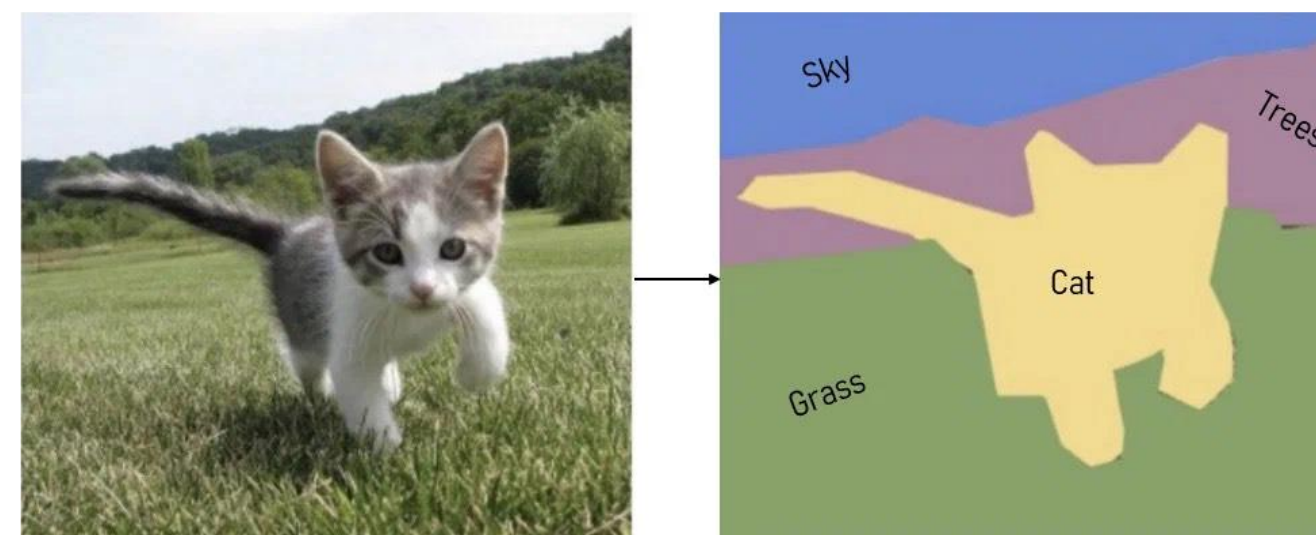
Обнаружение объектов



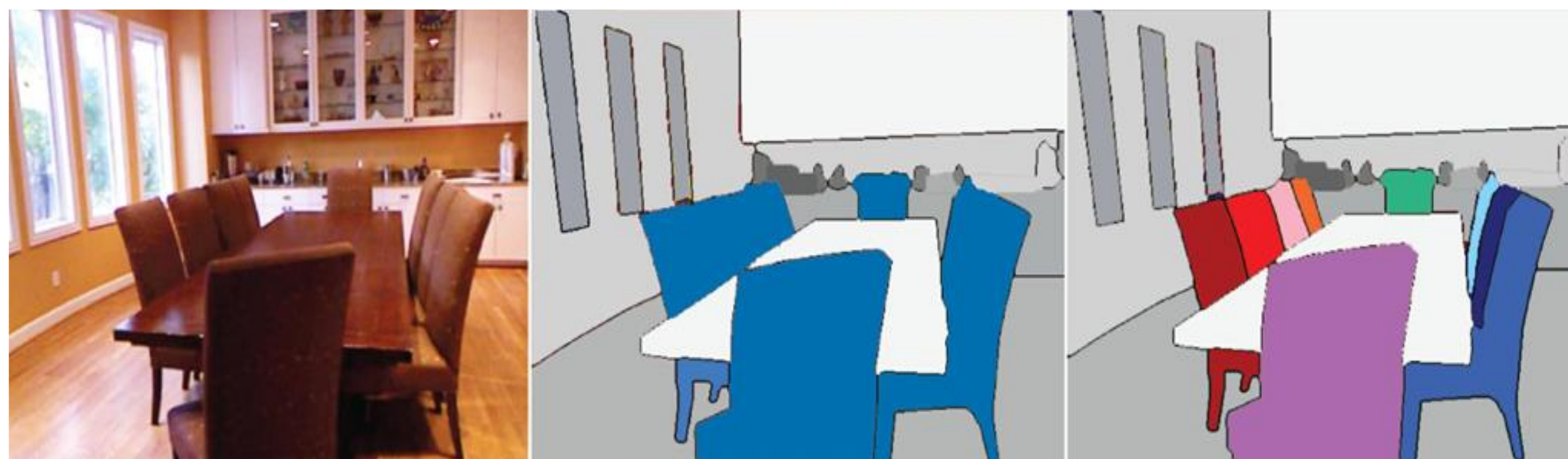
КОТ, СОБАКА, УТКА

# Основные задачи компьютерного зрения

- **Сегментация изображений:**



- **Семантическая сегментация:** Классификация каждого пикселя изображения по принадлежности к определенному классу.
- **Сегментация экземпляра:** Выделение отдельных экземпляров объектов одного класса.



Входное изображение

Семантическая сегментация

Сегментация экземпляра

# Основные задачи компьютерного зрения

- **Отслеживание объектов:** Определение положения объекта в последовательности видеокадров.
- **Оценка позы:** Определение положения ключевых точек тела человека или объекта.
- **Генерация изображений:** Создание новых реалистичных или стилизованных изображений.
- **Восстановление изображений:** Улучшение качества изображений (удаление шума, размытия, дорисовка недостающих частей).

# Основы обработки изображений

## Как компьютер "видит" изображение?

- **Представление изображений:**
  - Пиксели и их значения (цвет, яркость).
  - Цветовые модели (RGB, HSV, Grayscale).
  - Разрешение изображения.

# Основы обработки изображений

## Как компьютер "видит" изображение?

- **Базовые операции обработки изображений:**
  - Фильтрация (сглаживание, резкость).
  - Геометрические преобразования (масштабирование, поворот, сдвиг).
  - Гистограмма и ее применение (контрастирование).

# Основы обработки изображений

## Как компьютер "видит" изображение?

- **Извлечение признаков**

- **Цель:** Выделение информативных характеристик изображения, которые могут быть использованы для решения задач компьютерного зрения.
- **Классические методы:**
  - Детекторы углов (Harris Corner Detector).
  - Детекторы границ (Sobel, Canny).
  - Дескрипторы (SIFT, SURF, HOG).

# Машинное обучение как основа компьютерного зрения

## Основные концепции машинного обучения

- 1. Обучающие данные и разметка.
- 2. Задача обучения с учителем.
- 3. Модель машинного обучения.
- 4. Процесс обучения и оценки модели.

# Классические алгоритмы машинного обучения в компьютерном зрении

- • Метод ближайших соседей (K-Nearest Neighbors, KNN).
- • Линейная регрессия и логистическая регрессия.
- • Деревья решений и случайный лес (Random Forest).
- • Метод опорных векторов (Support Vector Machines, SVM).

# Глубокое обучение в компьютерном зрении

## Сверточные нейронные сети (Convolutional Neural Networks, CNN)

- Основные слои CNN
  - сверточные слои (convolutional layers)
  - слои подвыборки (pooling layers)
  - слои активации (activation layers)
  - полносвязные слои (fully connected layers).
    - Идея свертки: обнаружение локальных признаков.
    - Идея пулинга: уменьшение размерности и повышение инвариантности.
- Архитектуры CNN: LeNet, AlexNet, VGG, ResNet, EfficientNet

# Глубокое обучение в компьютерном зрении

## Рекуррентные нейронные сети (Recurrent Neural Networks, RNN)

- Обработка последовательностей (например, видео).
- LSTM и GRU

# Глубокое обучение в компьютерном зрении

## Трансформеры (Transformers)

- Внимание (Attention mechanism).
- Применение в задачах, требующих глобального контекста.

# Применение Компьютерного зрения

- Автономный транспорт

- Распознавание дорожных знаков, пешеходов, других транспортных средств.
- Системы помощи водителю (ADAS).
- Беспилотные автомобили и дроны.

- Медицина:

- Анализ медицинских изображений (рентген, КТ, МРТ).
- Диагностика заболеваний.
- Хирургическая навигация.

# Применение Компьютерного зрения

- **Производство и контроль качества**
  - Автоматический контроль дефектов продукции.
  - Роботизированная сборка.
  - Мониторинг производственных процессов..
- **Безопасность и наблюдение:**
  - Распознавание лиц.
  - Анализ поведения.
  - Контроль доступа..

# Применение Компьютерного зрения

- Другие применения

- Ритейл (анализ поведения покупателей, автоматические кассы).
- Сельское хозяйство (мониторинг урожая, обнаружение болезней растений).
- Развлечения (компьютерные игры, дополненная и виртуальная реальность).
- Поиск и классификация изображений в интернете.



# Зачем нужна обработка изображений?

- А. Подготовка изображений для последующего анализа или перед передачей в ML/нейросети
- В. Улучшение качества изображений для выделения полезной информации.
- С. Извлечение признаков и характеристик из изображений.
- Д. Уменьшение шума и артефактов.
- Е. Коррекция искажений.



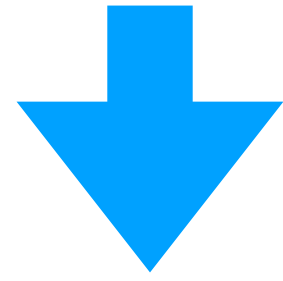
# Зачем нужна обработка изображений?

## Примеры

- Повышение резкости и контраста перед OCR (распознанием текста)
- Удаление шума в медицинских снимках
- Геометрические преобразования при анализе спутниковых снимков

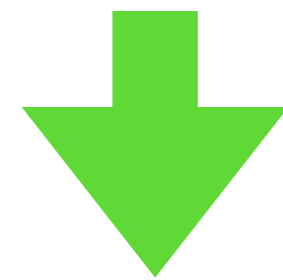
# Этапы типичной системы Компьютерного Зрения

1.Сбор данных



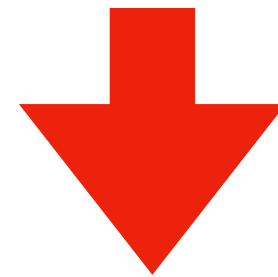
Получение изображений с камер, баз данных и т.д.

2.Предварительная обработка изображений



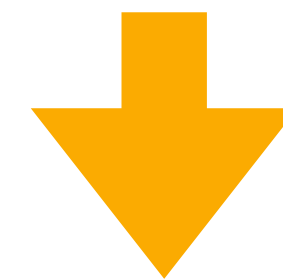
Улучшение качества, коррекция, подготовка к анализу.

3.Извлечение признаков



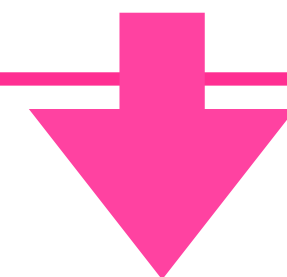
Определение ключевых характеристик изображения (границы, углы, текстуры, цветовые гистограммы и т.д.).

4.Анализ и интерпретация



Классификация, обнаружение объектов, сегментация, распознавание образов.

5.Принятие решений/Действие



На основе результатов анализа

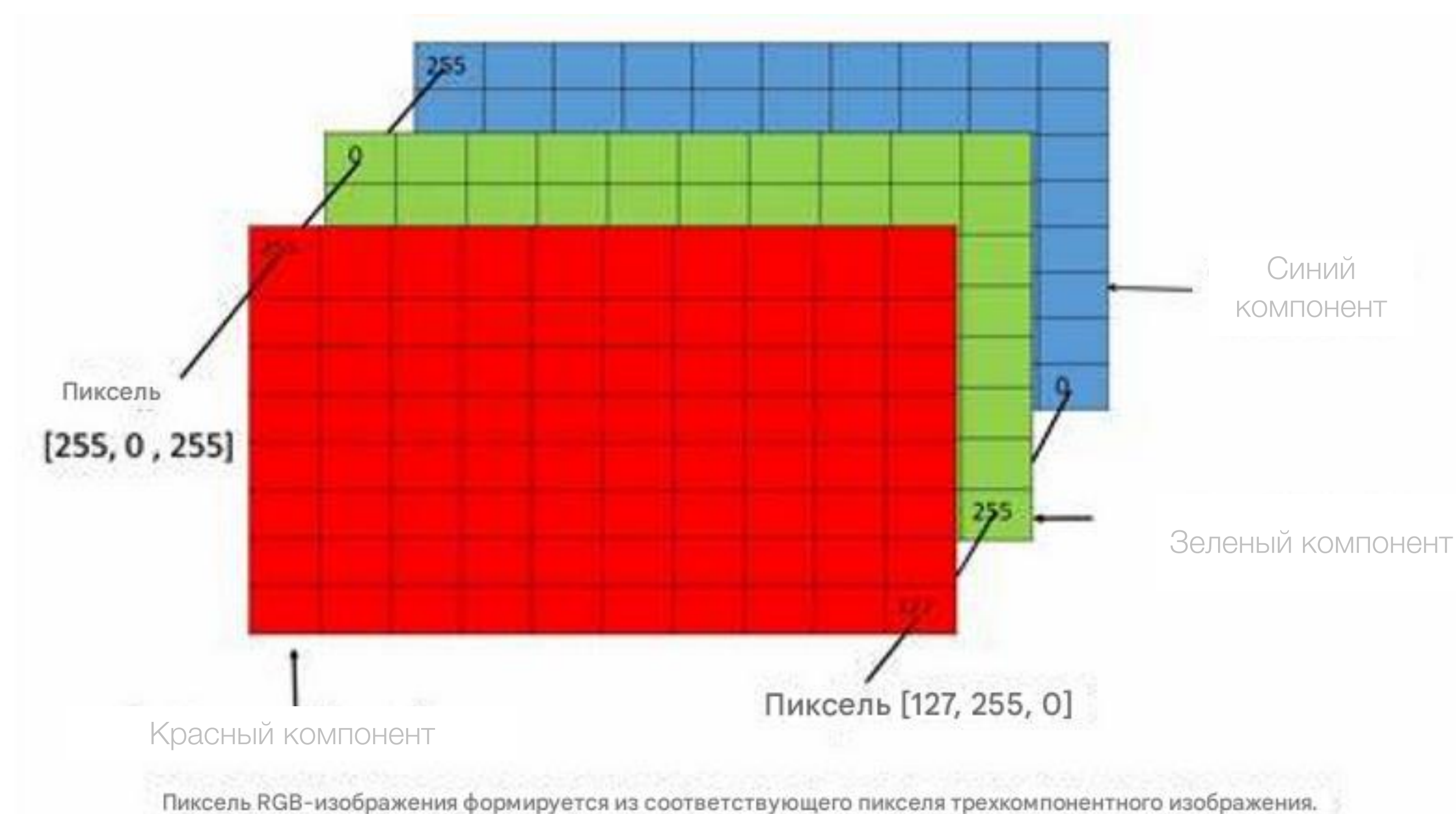
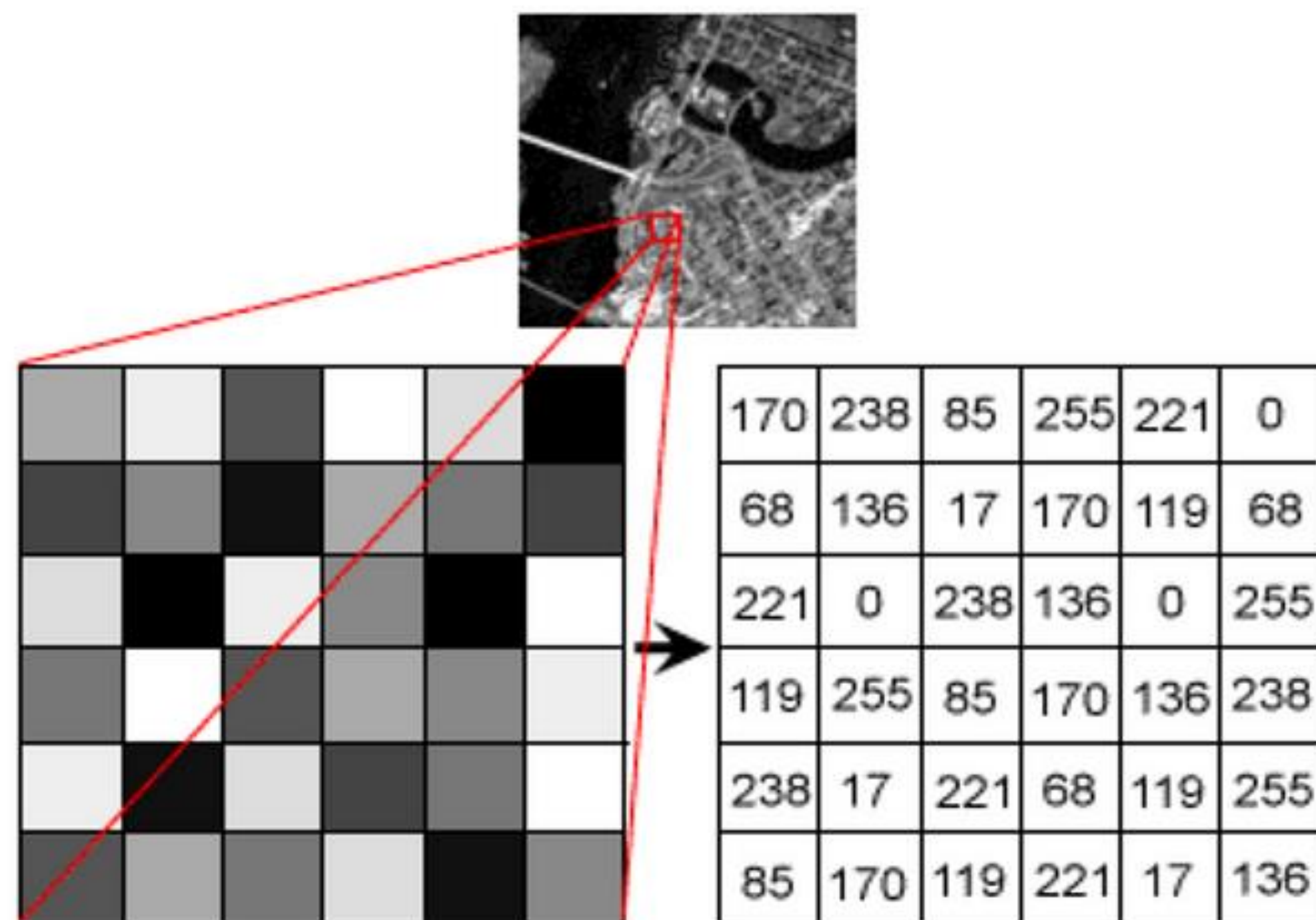


# Что такое цифровая картинка?

- **Цифровое изображение** — это матрица (таблица) чисел, представляющих яркость или цвет каждого пикселя.
- **Пиксель (pixel)** — минимальный элемент изображения, содержащий информацию о цвете.



# Что такое цифровая картинка?



# Цветовая модель RGB

- RGB (Red, Green, Blue) — основная модель представления цвета на экранах.
- Каждый пиксель кодируется тремя числами от 0 до 255:
  - **R** — интенсивность красного
  - **G** — интенсивность зелёного
  - **B** — интенсивность синего

# Цветовая модель RGB

## Пример

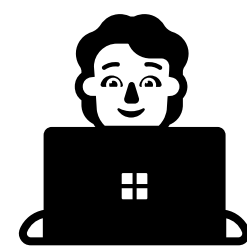
$(255, 0, 0)$  — чисто красный

$(0, 255, 0)$  — чисто зелёный

$(0, 0, 255)$  — чисто синий

$(255, 255, 255)$  — белый

$(0, 0, 0)$  — чёрный



# На Python:

```
import cv2

img = cv2.imread('Lenna.jpg')

pixel = img[100, 100]

print("пиксель:", pixel)
```

# Категории техник обработки изображений



Точечные  
операции

Преобразование  
значения каждого  
пикселя



Локальные  
операции  
(Фильтры)

Преобразование  
значения пикселя на  
основе значений его  
соседей в определенной  
окрестности



Глобальные  
операции  
(Преобразов  
ания)

Преобразование  
изображения  
целиком,  
учитывая все  
пиксели

# Точечные операции: Изменение яркости и контраста

- **Яркость:** Добавление или вычитание константы ко всем пикселям.
  - Увеличение яркости делает изображение светлее.
  - Уменьшение яркости делает изображение темнее.
- **Контраст:** Растяжение или сжатие диапазона значений пикселей.
  - Увеличение контраста делает светлые области светлее, а темные - темнее.
  - Уменьшение контраста сближает значения пикселей.

# Пример на Python (используя библиотеку OpenCV)

Контекст: Коррекция недостаточной или избыточной освещенности, улучшение видимости деталей.

```
import cv2
import numpy as np

# Загрузка изображения
img = cv2.imread('Lenna.jpg')

# Изменение яркости (добавление 50 к каждому пикселю)
brightness = 50
bright_img = np.clip(img.astype(np.int16) + brightness, 0, 255).astype(np.uint8)

# Изменение контраста (умножение на коэффициент)
contrast = 1.5
contrast_img = np.clip(((img - np.mean(img)) * contrast + np.mean(img)), 0, 255).astype(np.uint8)
```

# Локальные операции: Фильтры

- **Фильтр (ядро):** Матрица весов, которая применяется к окрестности каждого пикселя.
- **Свёртка (convolution):** Процесс применения фильтра к изображению. Значение центрального пикселя заменяется на взвешенную сумму значений пикселей в его окрестности.

# Локальные операции: Фильтры

## Основные типы фильтров

- **Размытие (Blurring):** Уменьшение шума и сглаживание деталей.
  1. **Средний фильтр (Average Filter):** Заменяет пиксель средним значением его соседей.
  2. **Гауссов фильтр (Gaussian Filter):** Использует веса, распределенные по нормальному закону (большой вес у центральных пикселей).
- **Выделение границ (Edge Detection):** Обнаружение резких изменений интенсивности.
  1. **Фильтр Собеля (Sobel Filter):** Вычисляет градиент изображения по горизонтали и вертикали.
  2. **Фильтр Лапласа (Laplacian Filter):** Обнаруживает области быстрого изменения интенсивности.

```
import cv2

# Загрузка изображения
img = cv2.imread('Lenna.jpg')

# Среднее размытие (ядро 5x5)
blur_avg = cv2.blur(img, (5, 5))

# Гауссово размытие (ядро 5x5, стандартное отклонение 0)
blur_gauss = cv2.GaussianBlur(img, (5, 5), 0)

# Фильтр Собеля (вычисление градиента по оси X)
sobelx = cv2.Sobel(img, cv2.CV_64F, 1, 0, ksize=5)
sobely = cv2.Sobel(img, cv2.CV_64F, 0, 1, ksize=5)
sobel_combined = cv2.magnitude(sobelx, sobely)
sobel_combined = cv2.normalize(sobel_combined, None, 0, 255, cv2.NORM_MINMAX,
cv2.CV_8U)
```

Спасибо за внимание!