

# Алгоритмы сегментации

# Сегментация

**Сегментация** — это процесс разделения цифрового изображения на несколько сегментов (множество пикселей, также называемых суперпикселями). Цель сегментации заключается в упрощении и/или изменении представления изображения, чтобы его было проще и легче анализировать. Сегментация изображений обычно используется для того, чтобы выделить объекты и границы (линии, кривые, и т. д.) на изображениях.



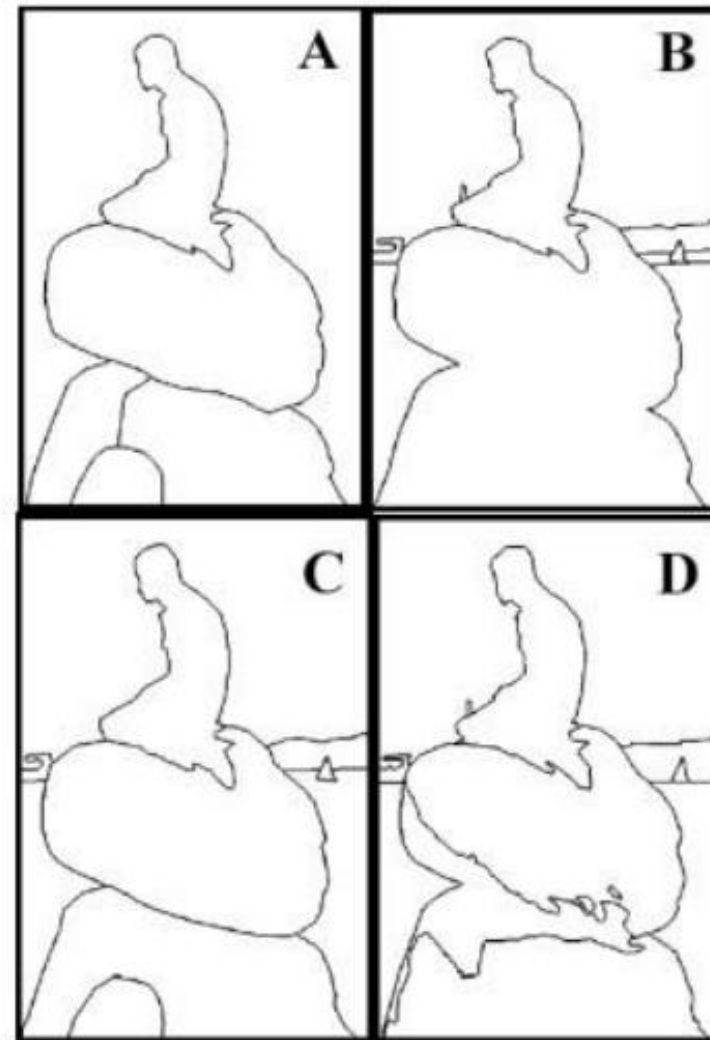
**Сегментация изображений** — это процесс присвоения таких меток каждому пикселю изображения, при которых пиксели с одинаковыми метками имеют общие визуальные характеристики.

- Медицинские изображения: обнаружение опухолей и других патологий, определение объёмов тканей, диагностика, изучение анатомической структуры
- Выделение объектов на спутниковых снимках
- Распознавание лиц
- Распознавание отпечатков пальцев
- Системы управления дорожным движением
- Обнаружение стоп-сигналов
- Машинное зрение
- Распараллеливание информационных потоков при передаче изображений высокого разрешения



## Оценка качества работы методов сегментации

- Целостность и однородность по некоторому признаку
- Отличие признака для смежных областей
- Отсутствие мелких отверстий внутри
- Гладкие границы

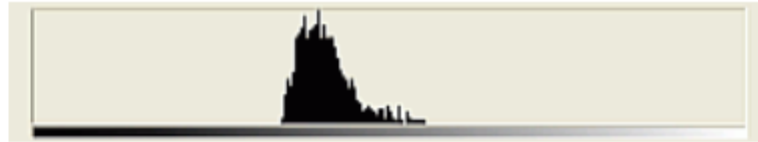


## Критерии «похожести» пикселей

- По расстоянию
- По яркости
- По цвету
- По текстуре

# Критерий однородности

- Гистограмма содержит не больше 1 значительного пика



- Отклонение любого пикселя от средней яркости  $< T_{avg}$

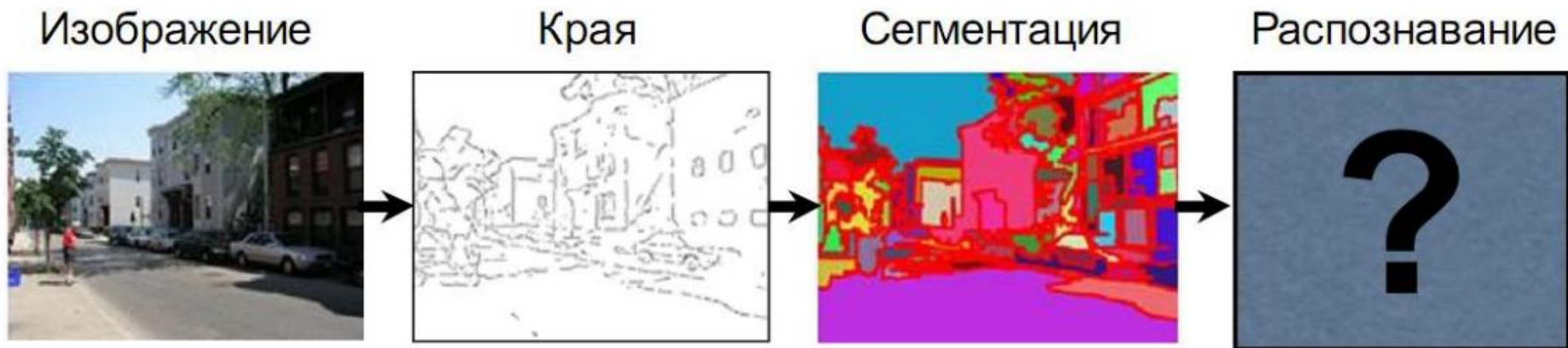
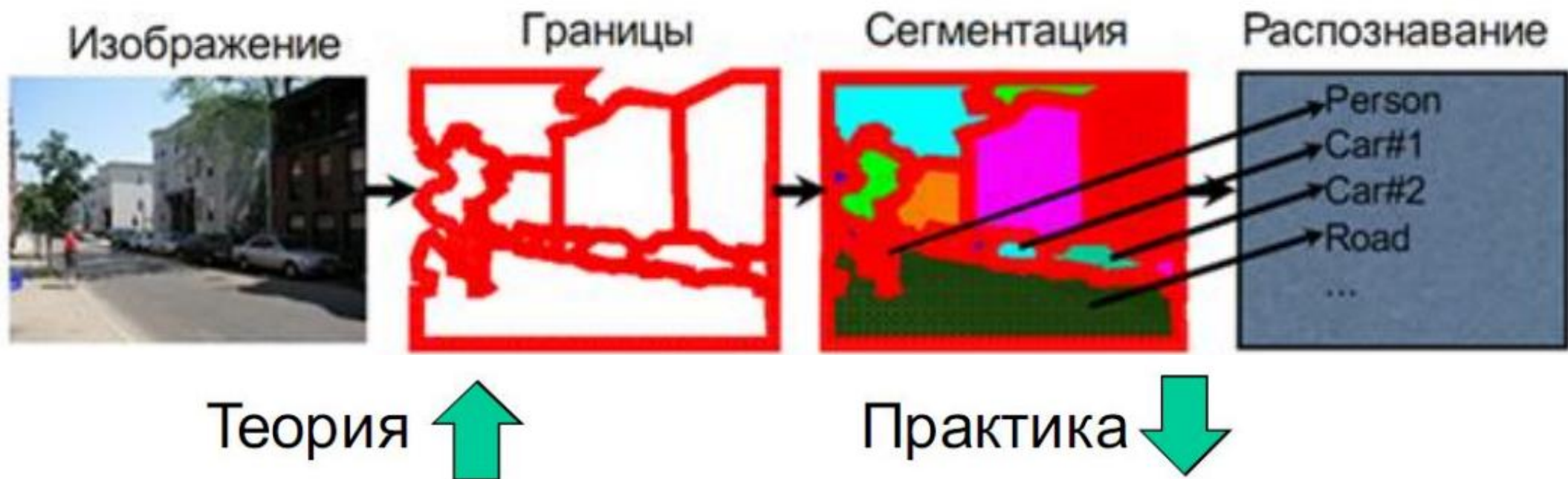
$$\forall p \in S \quad \left| I(p) - \frac{1}{N} \sum_{q \in S} I(q) \right| < T_{avg}$$

- Разница между соседними пикселями  $< T_{diff}$

$$\forall p \in S, \forall q \in N(p) \quad |I(p) - I(q)| < T_{diff}$$

- «Слабая» граница между регионами (только для слияния)

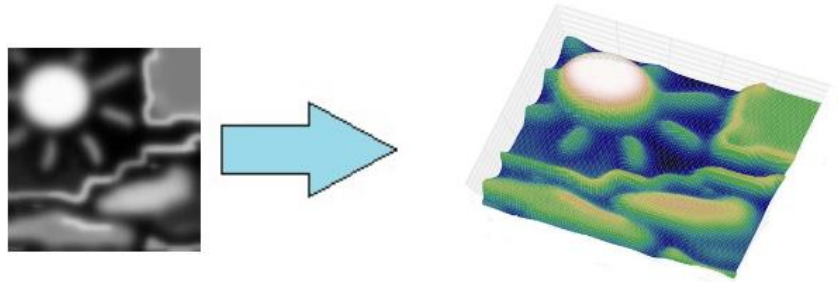
# План



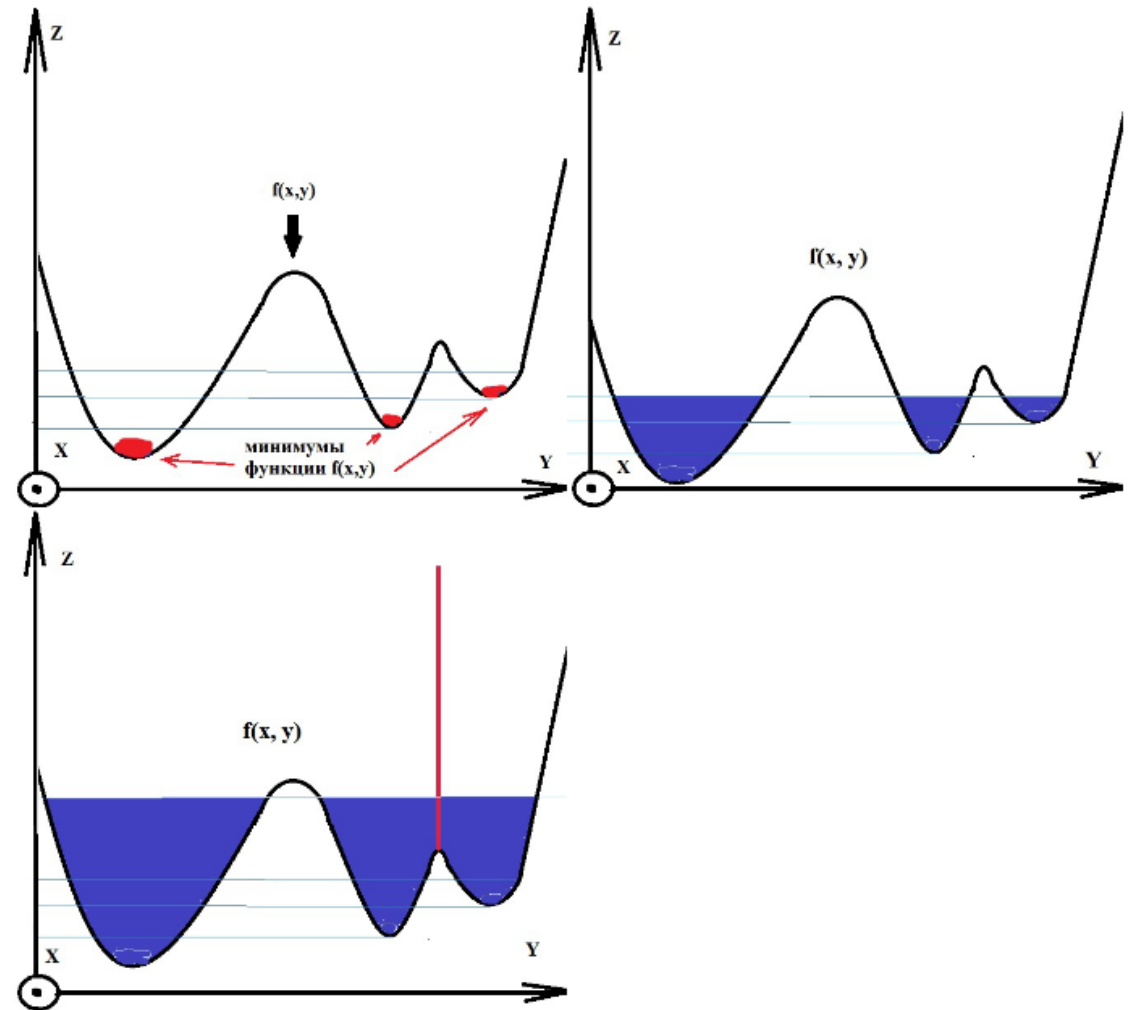


## Алгоритм сегментации по водоразделам (WaterShed)

Значением функции может быть интенсивность или модуль градиента. Для наибольшего контраста можно взять градиент от изображения. Если по оси OZ откладывать абсолютное значение градиента, то в местах перепада интенсивности образуются хребты, а в однородных регионах – равнины.



После нахождения минимумов функции  $f$ , идет процесс заполнения “водой”, который начинается с глобального минимума. Как только уровень воды достигает значения очередного локального минимума, начинается его заполнение водой. Когда два региона начинают сливаться, строится перегородка, чтобы предотвратить объединение областей. Вода продолжит подниматься до тех пор, пока регионы не будут отделяться только искусственно построенными перегородками (см. рис. справа).



Однако на практике сегментации по водоразделам приводит к значительной чрезмерной сегментации из-за шума или локальных неоднородностей в градиентном изображении.



Разработано множество методов, позволяющих избежать чрезмерной сегментации, используя различные маркеры или работая с иерархией контуров сегментации.

На следующем слайде можно увидеть некоторые примеры работы алгоритмов сегментации водоразделом, подробнее про них можно посмотреть по ссылке:

<https://people.cmm.minesparis.psl.eu/users/beucher/wtshed.html>



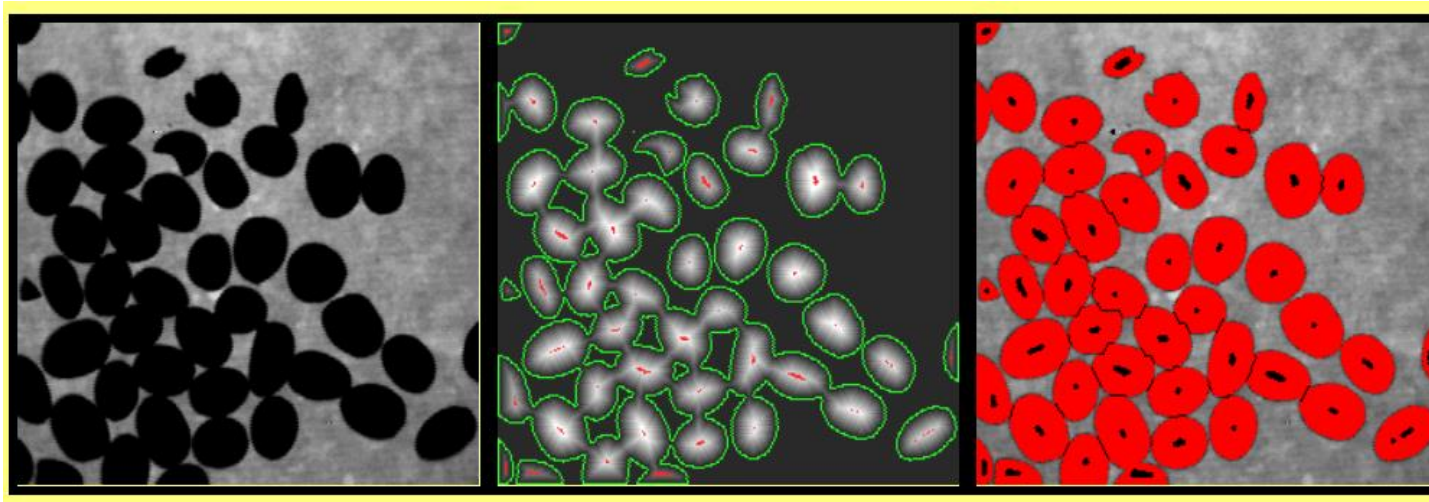
## Сегментация дороги

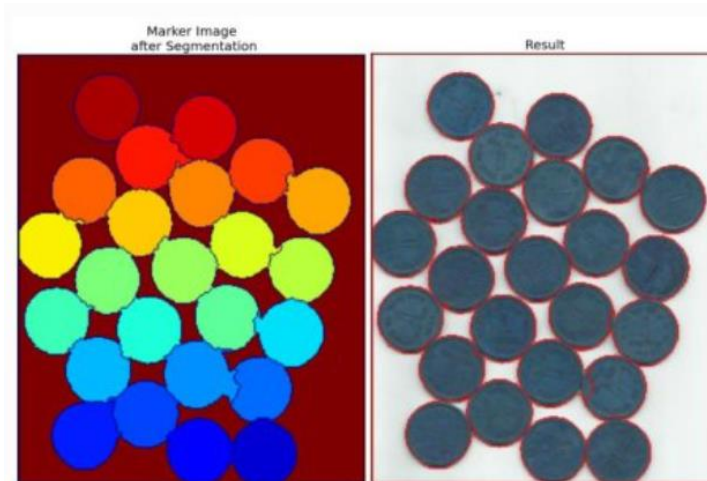
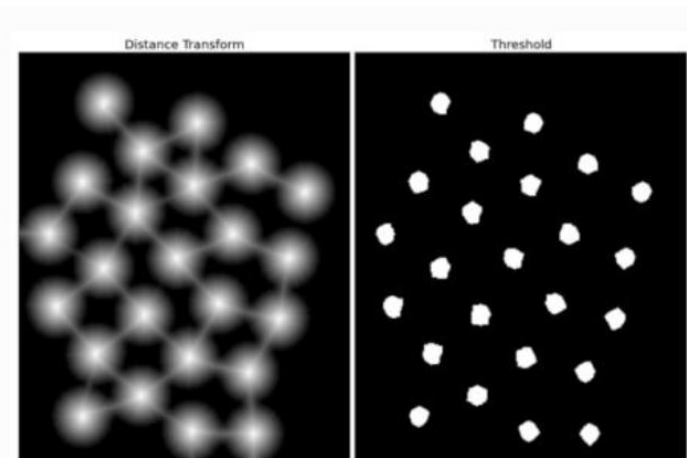
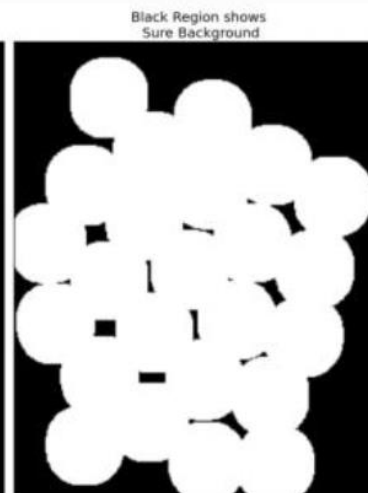
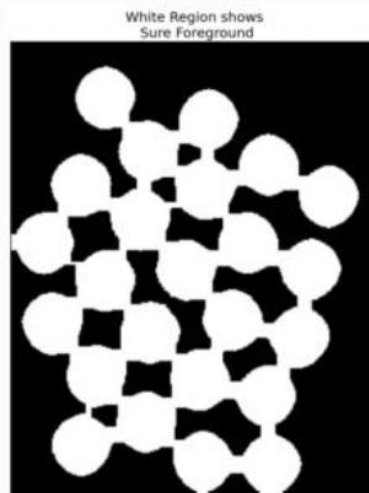
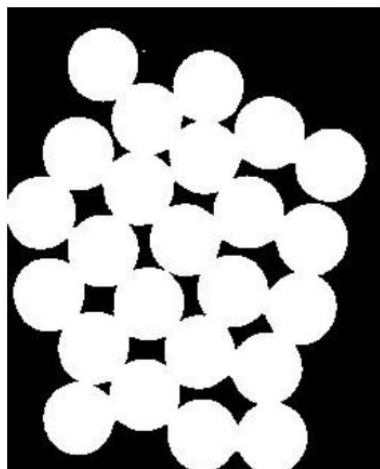
В этом примере маркеры были введены вручную.



## Разделение кофейных зерен

В этом случае в качестве критерия используется функция расстояния исходного изображения.





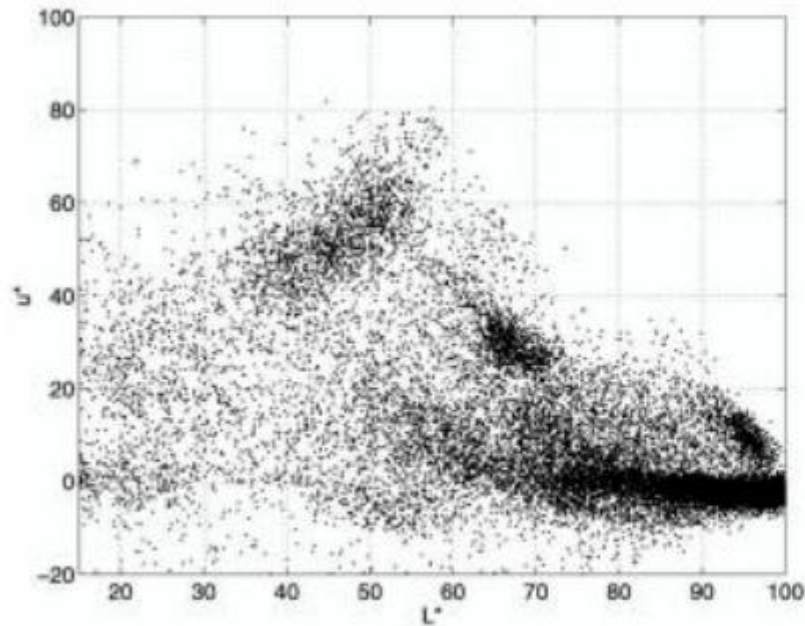
Поэтапное описание этого примера с сегментацией водоразделом по функции расстояния можно найти по ссылке:

[https://opencv24-python-tutorials.readthedocs.io/en/latest/py\\_tutorials/py\\_imgproc/py\\_watershed/py\\_watershed.html](https://opencv24-python-tutorials.readthedocs.io/en/latest/py_tutorials/py_imgproc/py_watershed/py_watershed.html)

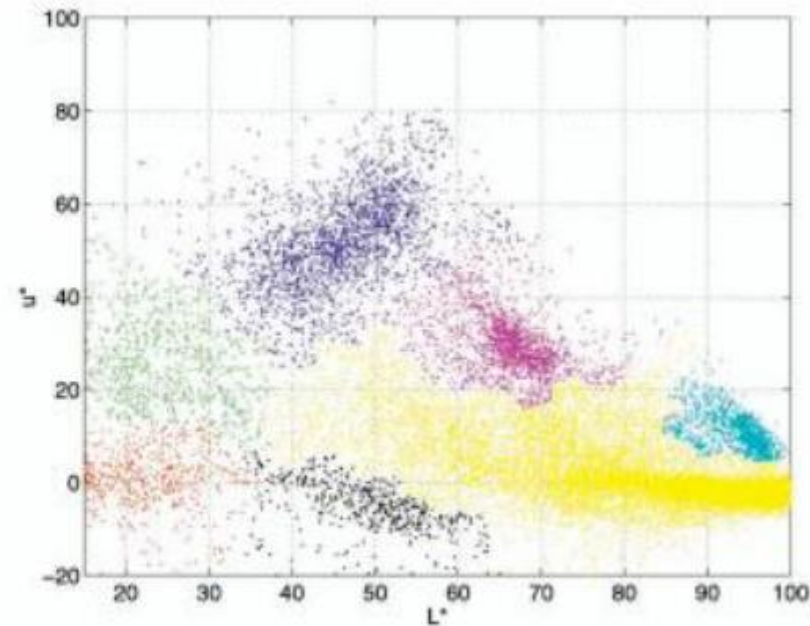
# Алгоритм сегментации MeanShift

MeanShift группирует объекты с близкими признаками. Пиксели со схожими признаками объединяются в один сегмент, на выходе получаем изображение с однородными областями.

Например, в качестве координат **в пространстве признаков** можно выбрать координаты пикселя  $(x, y)$  и компоненты **RGB** пикселя. Изобразив пиксели в пространстве признаков, можно заметить сгущения в определенных местах.

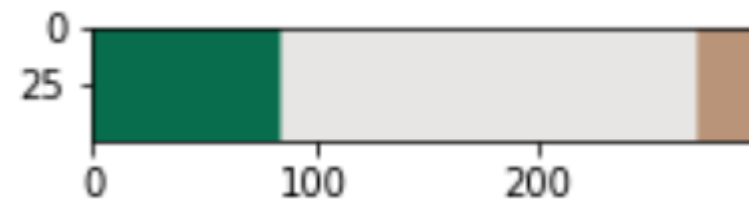
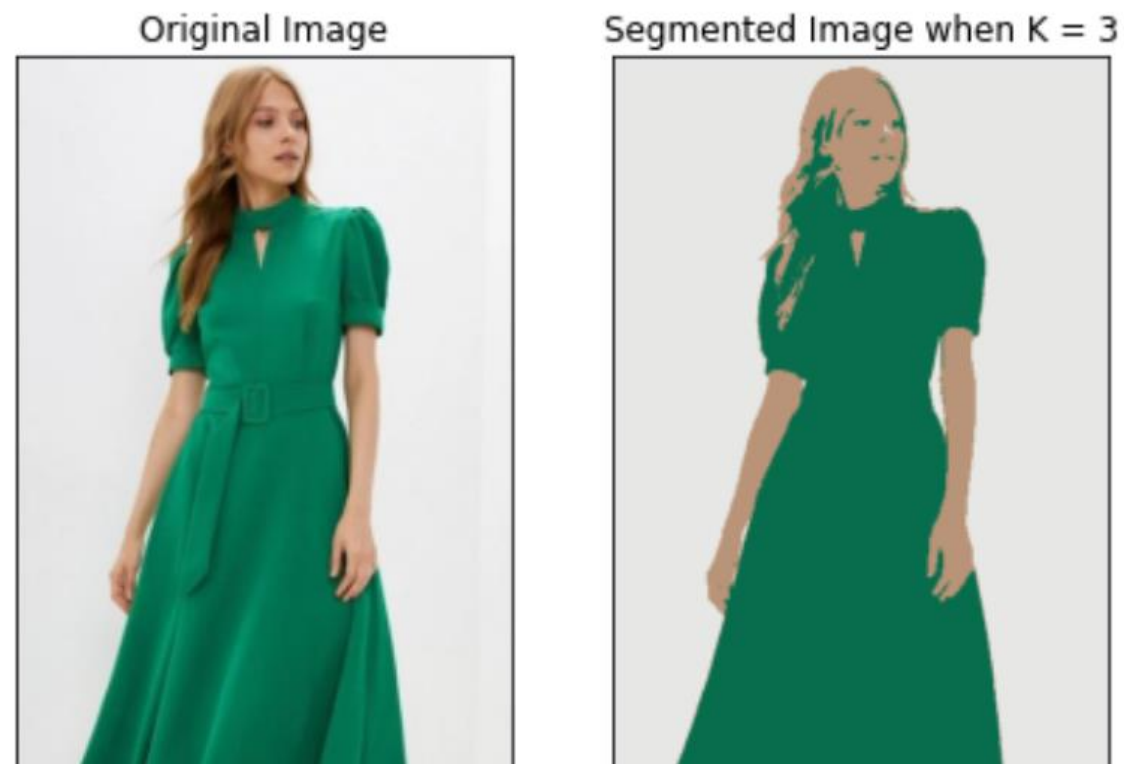
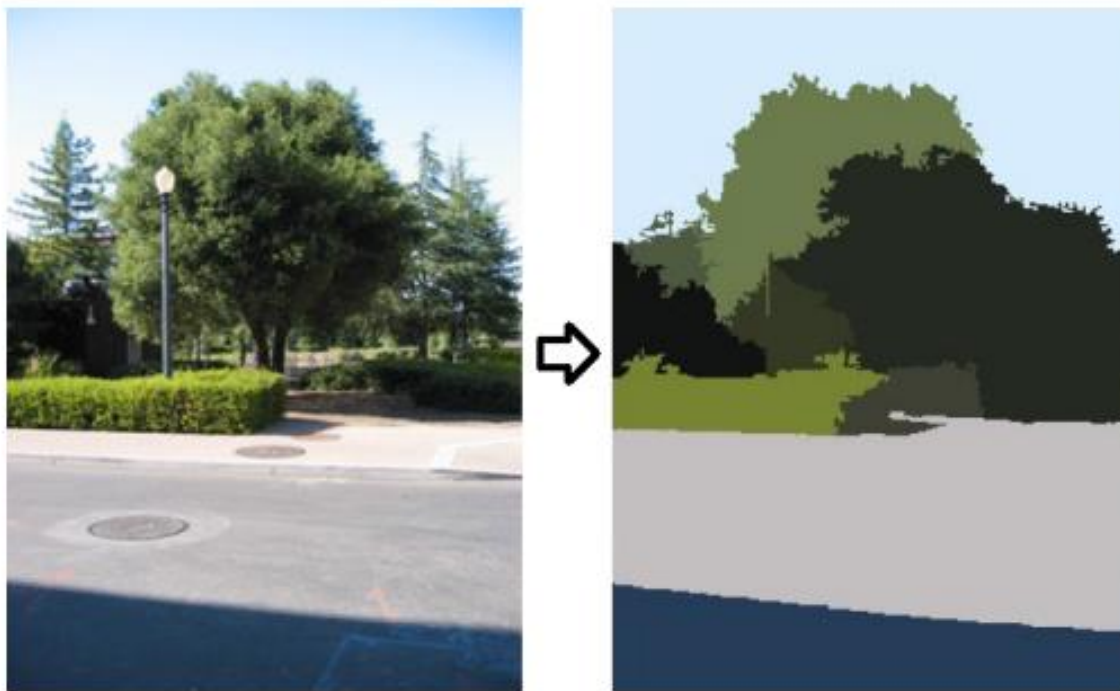


(a)



(b)

# Результаты сегментации



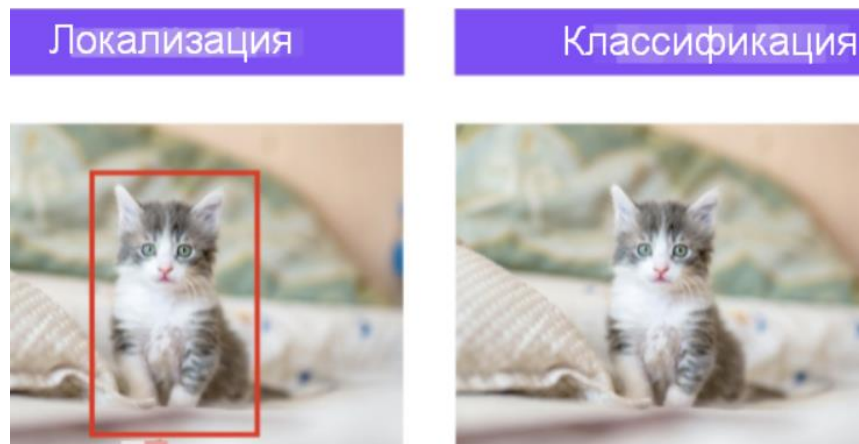


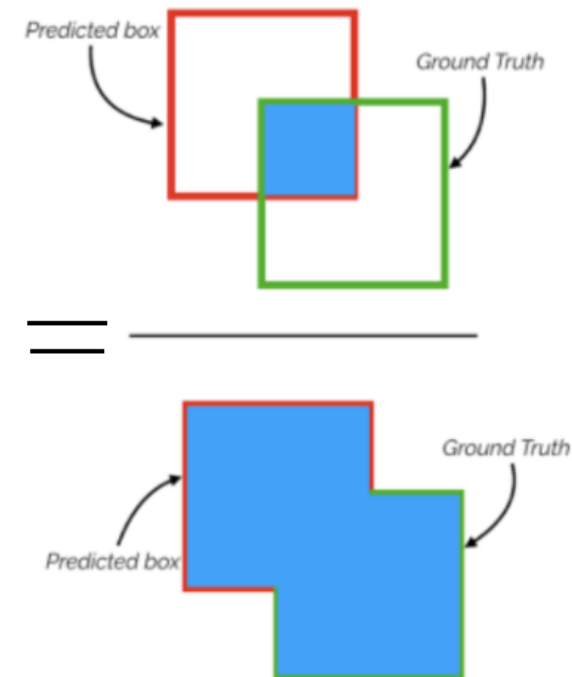
# Меры измерения точности при детектировании объектов на изображении

Детектирование состоит из локализации и классификации. Локализация определяет местоположение экземпляра (например, координаты ограничивающей рамки), а классификация сообщает вам, что это такое (например, собака или кошка).

Основные меры измерения точности детектирования **Precision**, **Recall**, и **F1-Score** связаны с понятиями **IoU**, **map**, **map50**.

Системы детектирования объектов делают прогнозы в ограничивающей рамке. Однако ограничивающая рамка не всегда точна, и для определения ее точности используется мера **IoU**



$$\text{IoU} = \frac{\text{Область пересечения}}{\text{Общая область}} = \frac{\text{Area of Intersection}}{\text{Area of Union}}$$




$$\text{Precision} = \frac{TP}{TP + FP}$$

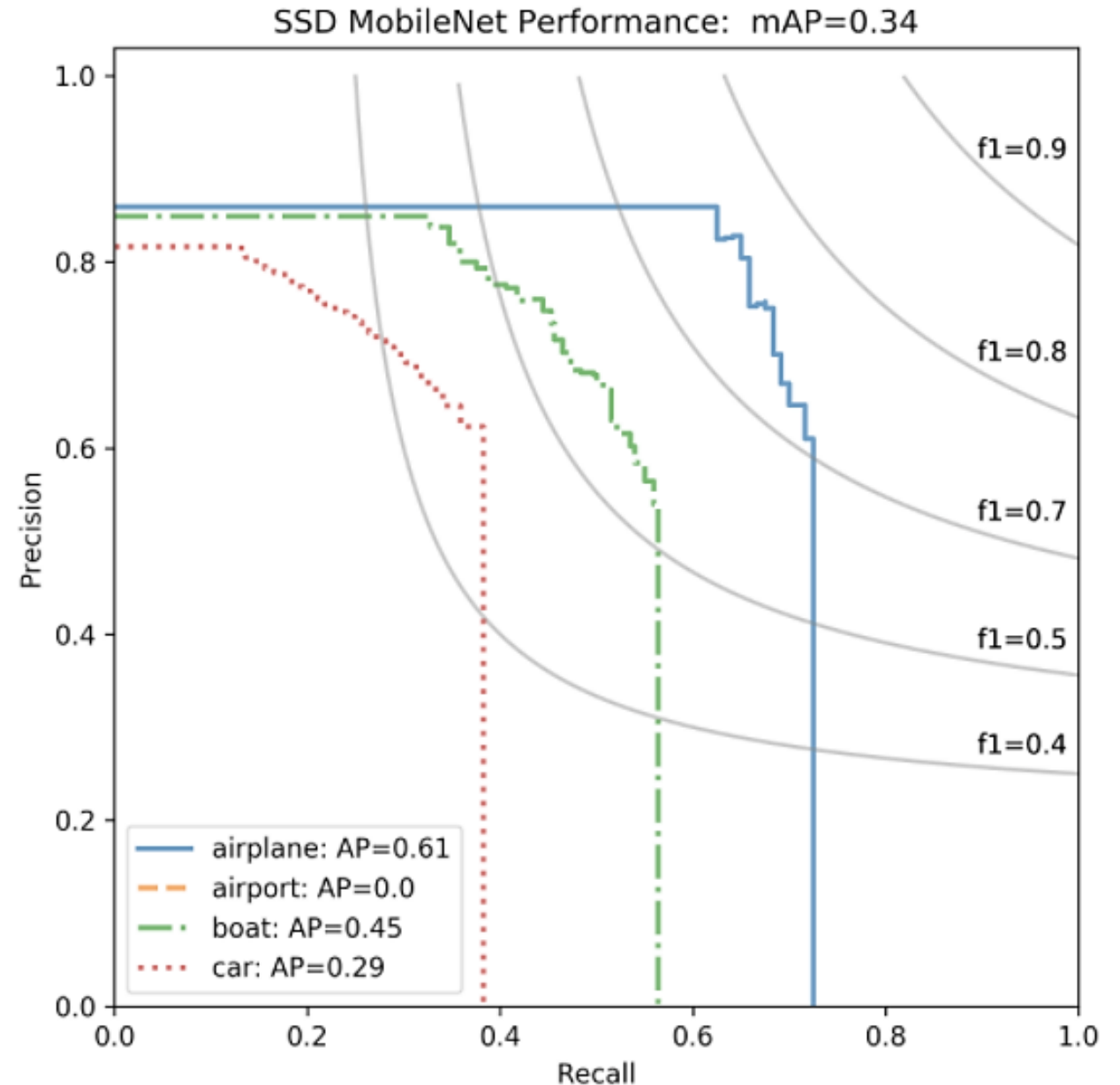
$$\text{Recall} = \frac{TP}{TP + FN}$$

$$F1 = 2 * \frac{\text{Precision} * \text{Recall}}{\text{Precision} + \text{Recall}}$$

$$F_{\beta} = (1 + \beta^2) * \frac{\text{precision} * \text{recall}}{(\beta^2 * \text{precision}) + \text{recall}}$$

Средняя оценка точности или **mAP** рассчитывается путем взятия среднего AP по всем классам и / или общим пороговым значениям IoU, в зависимости от различных существующих проблем обнаружения.

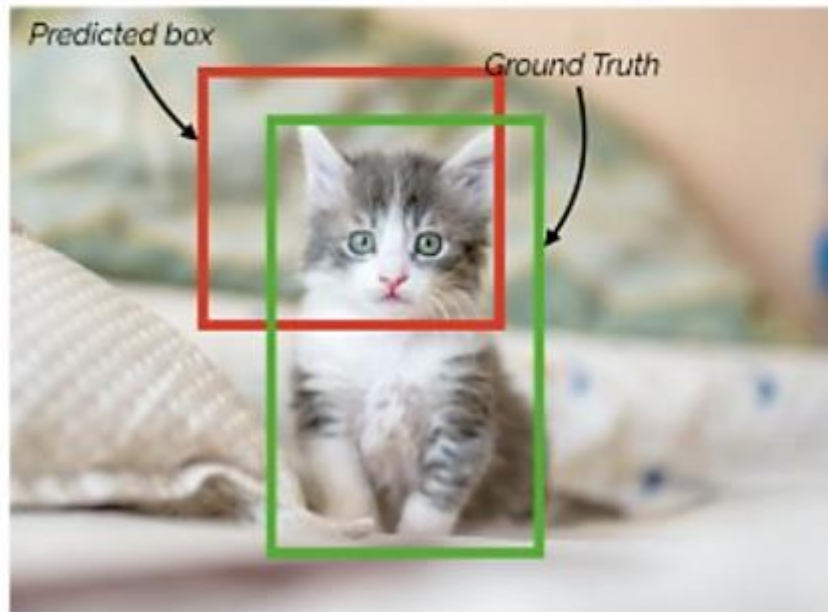
Пример: кривая точности-отзыва для модели распознавания для 4 классов объектов, где порог IoU равен 0,5:



Для задач обнаружения объектов рассчитываем Precision и Recall, используя значение IoU для заданного порогового значения IoU. Например, если порог IoU равен 0,5, а значение IoU для прогноза составляет 0,7, то классифицируем прогноз как истинно положительный (TF). С другой стороны, если IoU равно 0,3, классифицируем его как ложноположительный (FP).

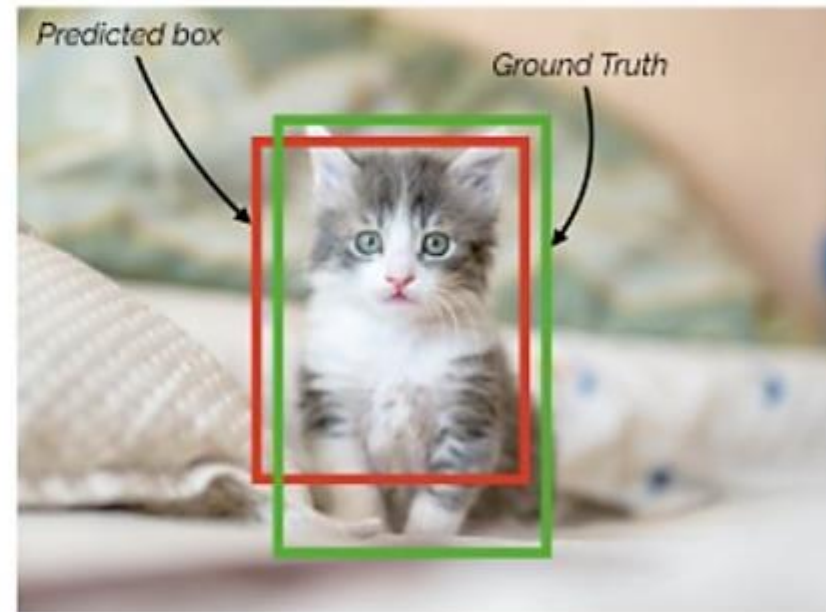
*If IoU threshold = 0.5*

*False Positive (FP)*



*$IoU = \sim 0.3$*

*True Positive (TP)*



*$IoU = \sim 0.7$*

# Ансамбли в детектировании объектов

<http://recog.ru/ensemble-object-detection/>

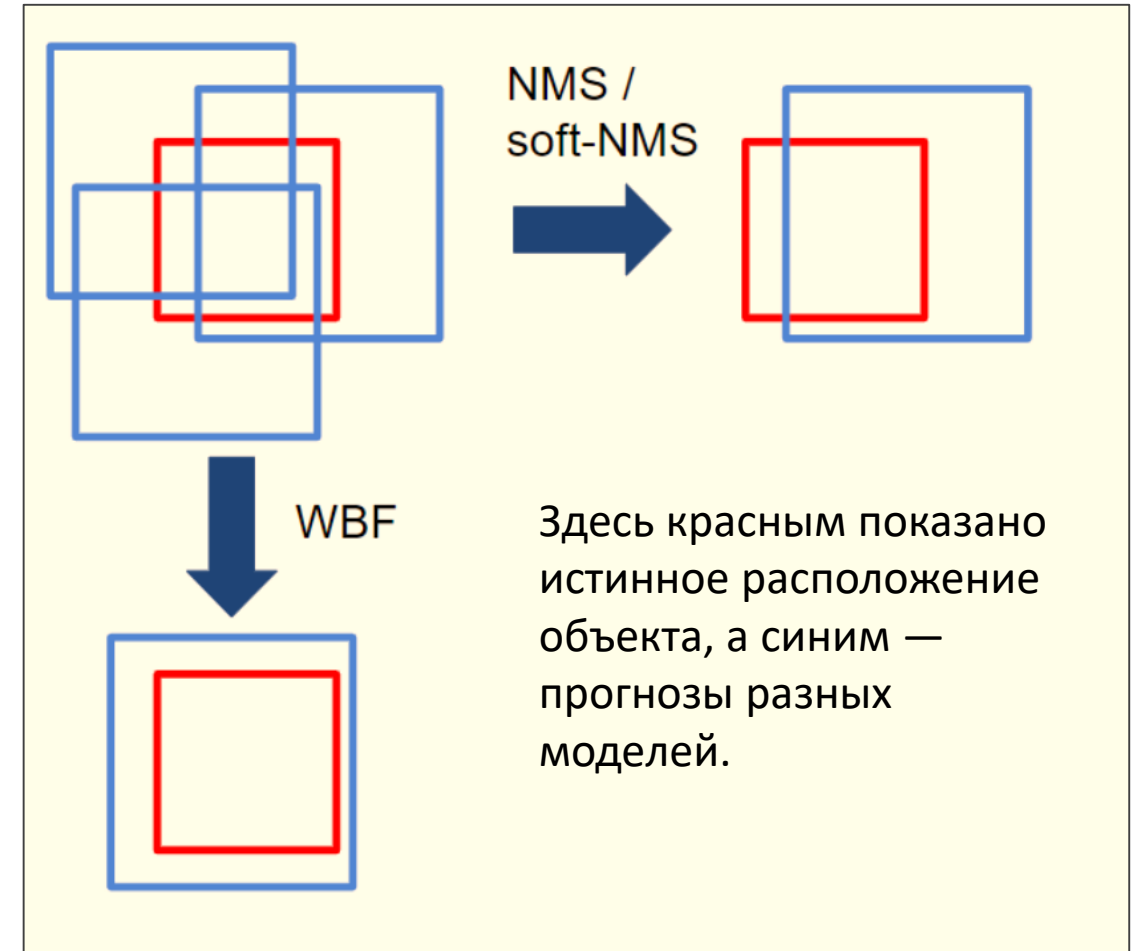
Для повышения качества часто детектирования используются несколько моделей, объединяя их в ансамбли. **Основной проблемой ансамблей является объединение результатов для детектированного одного объекта.**

Существуют несколько известных методов объединения результатов:

- NMS (non-maximum suppression)
- soft-NMS
- NMW (non-maximum weighted)
- WBF (Weighted boxes fusion)

Например, **NMS** сортирует все области по их доверительной вероятности. Затем выбирается область с максимальной оценкой достоверности. В то же время все другие области, которые значительно перекрывают эту область, отфильтровываются, а методы **NMW** и **WBF** работают с IoU более сложным образом

Отличия работы методов объединения результатов:



# Проблемы разметки изображений для дальнейшей классификации

При детектировании объектов базовые модели часто возвращают выровненные по вертикали и горизонтали прямоугольники. Это приводит к проблемам, например, при детектировании квадрата повернутого на  $45^\circ$ , площадь ограничивающей рамки в два раза больше квадрата.



Еще одной проблемой при обучении моделей на выровненных прямоугольниках является то, что часто объекты находятся рядом, и при их повороте эти области пересекаются или даже вкладываются друг в друга. В результате получается детектор, который может пересчитывать или недосчитывать объекты там, где есть кластеры одного и того же или сходного класса. Повернутые ограничивающие рамки могут обеспечить более высокую точность и полноту.

Например, выровненный по оси прямоугольник вокруг человека на рисунке содержит много неба и немного мотоцикла. В повернутой коробке меньше неба и почти нет мотоцикла.





Предполагаемые ограничивающие рамки (красные) и ограничивающие рамки истинности (зеленые) для модели с выравниванием по осям (слева) и модели с повернутым прямоугольником (справа).

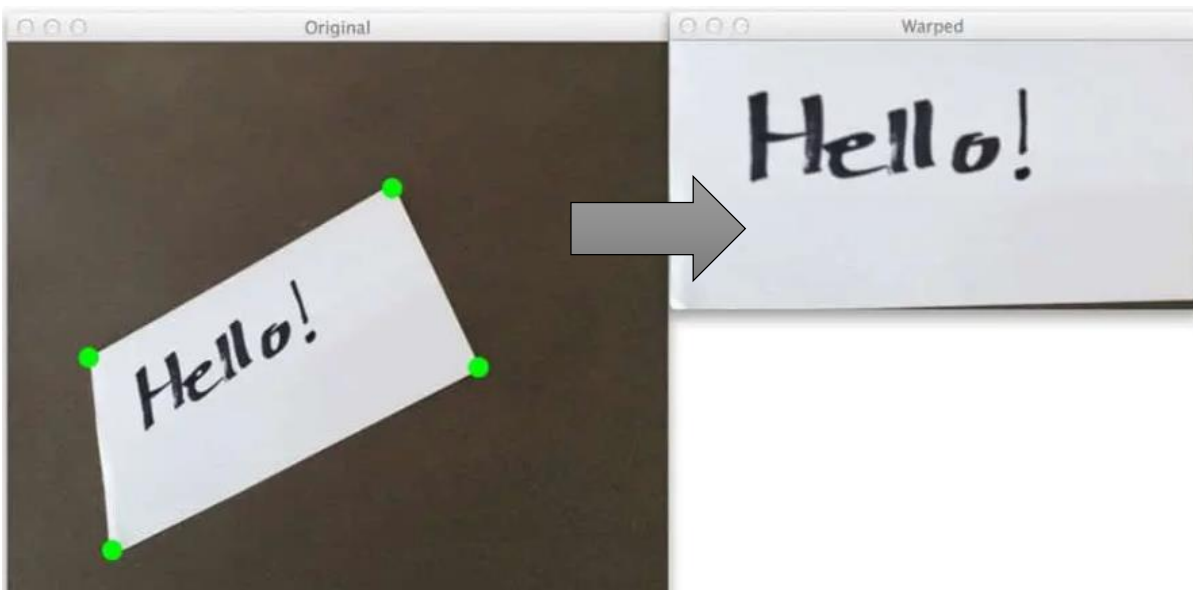


Оценка качества:

- обычная (выровненная по осям) модель. Precision = 0.37, Recall = 0.55, F1-score = 0.44;
- **модель с повернутыми прямоугольниками. Precision = 0.77, Recall = 0.76, F1-score = 0.76;**



# Как можно выровнять искаженный перспективой документ?



с помощью OpenCV это можно сделать всего за три простых шага:

- Шаг 1: Обнаружение краев.
- Шаг 2: Используйте края изображения, чтобы найти контур (контур), представляющий сканируемый лист бумаги.
- Шаг 3: Примените перспективное преобразование, чтобы получить вид документа сверху вниз.

```
# convert the image to grayscale, blur it, and find edges in the image
```

```
gray = cv2.cvtColor(image, cv2.COLOR_BGR2GRAY)
gray = cv2.GaussianBlur(gray, (5, 5), 0)
edged = cv2.Canny(gray, 75, 200)
```

```
# find the contours in the edged image, keeping only the
# largest ones, and initialize the screen contour
```

```
cnts = cv2.findContours(edged.copy(), cv2.RETR_LIST,
                        cv2.CHAIN_APPROX_SIMPLE)
```

```
cnts = imutils.grab_contours(cnts)
```

```
cnts = sorted(cnts, key = cv2.contourArea, reverse = True)[:5]
```

```
# loop over the contours
```

```
for c in cnts:
```

```
    # approximate the contour
```

```
    peri = cv2.arcLength(c, True) #True указывает, должна ли кривая быть замкнутой
```

```
    approx = cv2.approxPolyDP(c, 0.02 * peri, True)
```

```
    # if our approximated contour has four points, then we
```

```
    # can assume that we have found our screen
```

```
    if len(approx) == 4:
```

```
        screenCnt = approx
```

```
        break
```





```
# находим контуры в изображении и подсчитываем число книг
cnts = cv2.findContours(closed.copy(), cv2.RETR_EXTERNAL, cv2.CHAIN_APPROX_SIMPLE)[1]
total = 0
# выполняем цикл по контурам
for c in cnts:
    # сглаживаем контур
    peri = cv2.arcLength(c, True)
    approx = cv2.approxPolyDP(c, 0.02 * peri, True)

    # если у контура есть четыре вершины, это, скорее всего, книга
    if len(approx) == 4:
        cv2.drawContours(image, [approx], -1, (0, 255, 0), 4)
        total += 1
```

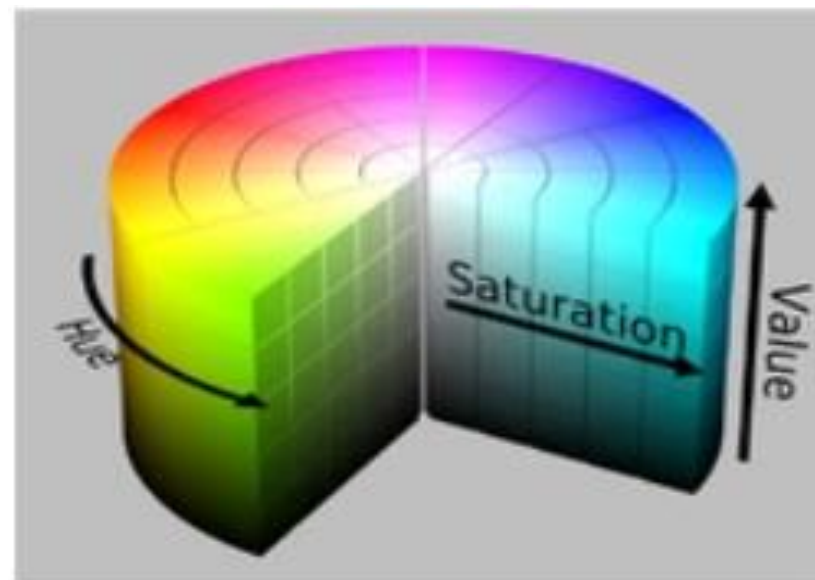
# Как можно определить площадь леса на спутниковом снимке?



Самым простым критерием при выделении объектов классификации на изображении является цвет. Поэтому задачу разметки изображений можно свести к задаче выделения объектов по цветам.



По умолчанию в OpenCV цветное изображение хранится в палитре BGR. В RGB цвет пикселя определяется насыщенностью красным, зеленым и голубым. Таким образом, выбор диапазона оттенков одного и того же цвета становится не самой простой задачей. При использовании цветовой модели HSV сделать это гораздо проще. Эта цветовая схема определяется тремя компонентами (рисунок 1): Hue - цветовой тон, Saturation – насыщенность и Value - яркость. **Поскольку канал тона моделирует тип цвета, он очень полезен в задачах обработки изображений, которым необходимо сегментировать объекты на основе их цвета.**



```
cv.inRange(src, lowerb, upperb[, dst]) ->dst
```

```
frame_HSV = cv.cvtColor(frame, cv.COLOR_BGR2HSV)
frame_threshold = cv.inRange(frame_HSV, (low_H, low_S, low_V), (high_H, high_S, high_V))
```

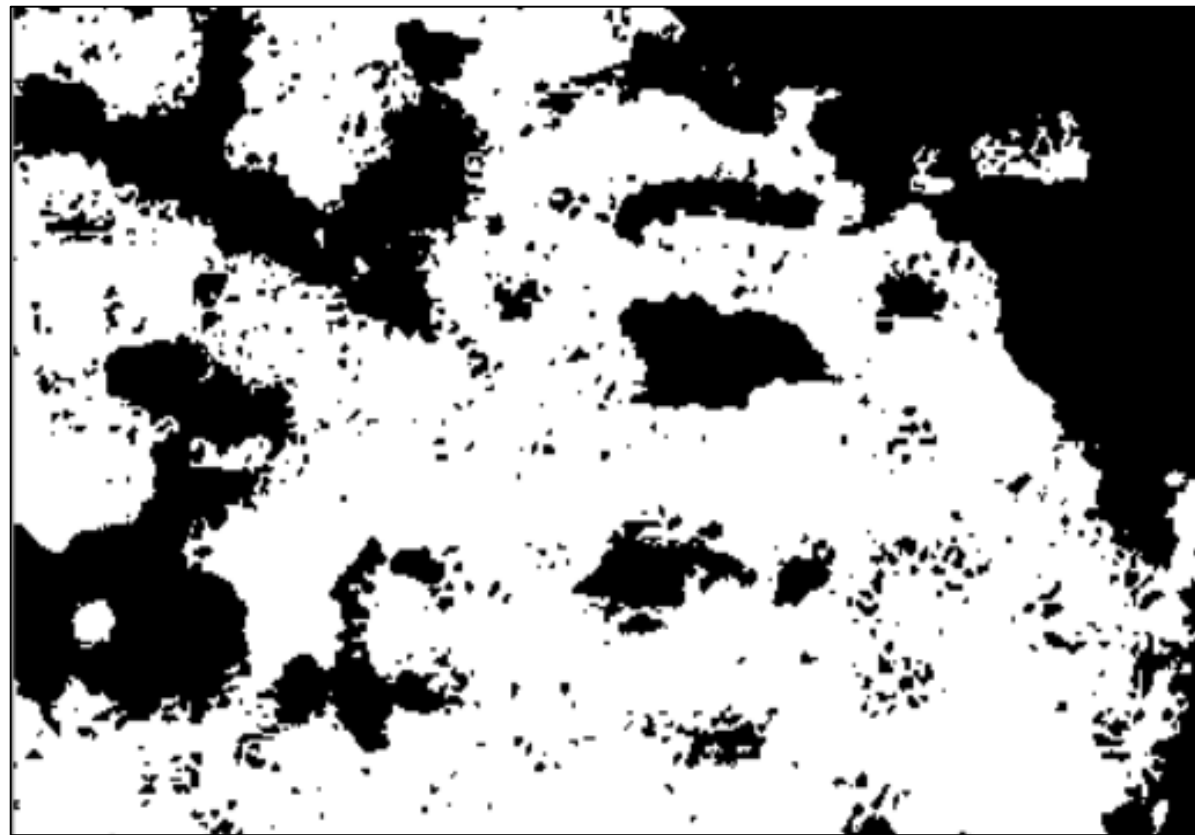


В рассматриваемом для зеленого цвета лесного массива приняты следующие ограничения

```
# диапазон изменения тона
low_H = 50;
high_H = 100;

# диапазон изменения насыщенности
low_S = 20;
high_S = 256;

# диапазон изменения яркости
low_V = 10;
high_V = 100;
```



Как можно улучшить?

После применения морфологических операций находим контуры, сортируем по длине и удаляем самые короткие. Затем можно найти площадь контуров и получить площадь лесных массивов на данной территории.





# Обработка спутниковых снимков

Источниками снимков земной поверхности являются спутники дистанционного зондирования Земли (ДЗЗ), оснащённые различными видами съёмочной аппаратуры. В настоящее время ряд организаций свободно предоставляет снимки различного пространственного (геометрического), спектрального, временного и радиометрического разрешения научным организациям и частным лицам

DigitalGlobe



**WorldView-4**

AIRBUS



**Pleiades**

planet.



**Planetscope (Dove)**

esa



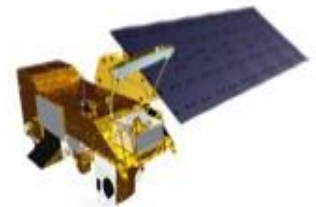
**Sentinel-2**

NASA USGS

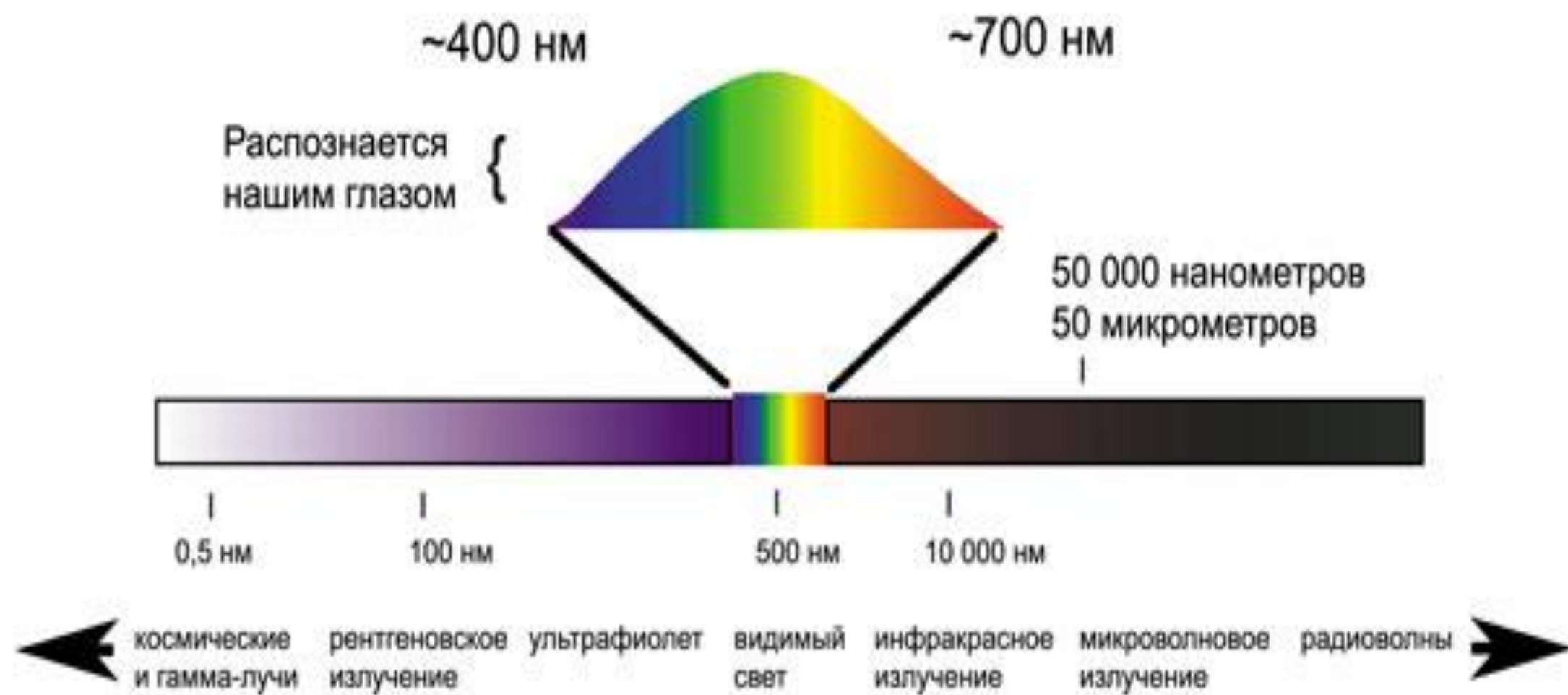


**Landsat-8**

NASA



**Aqua (MODIS)**



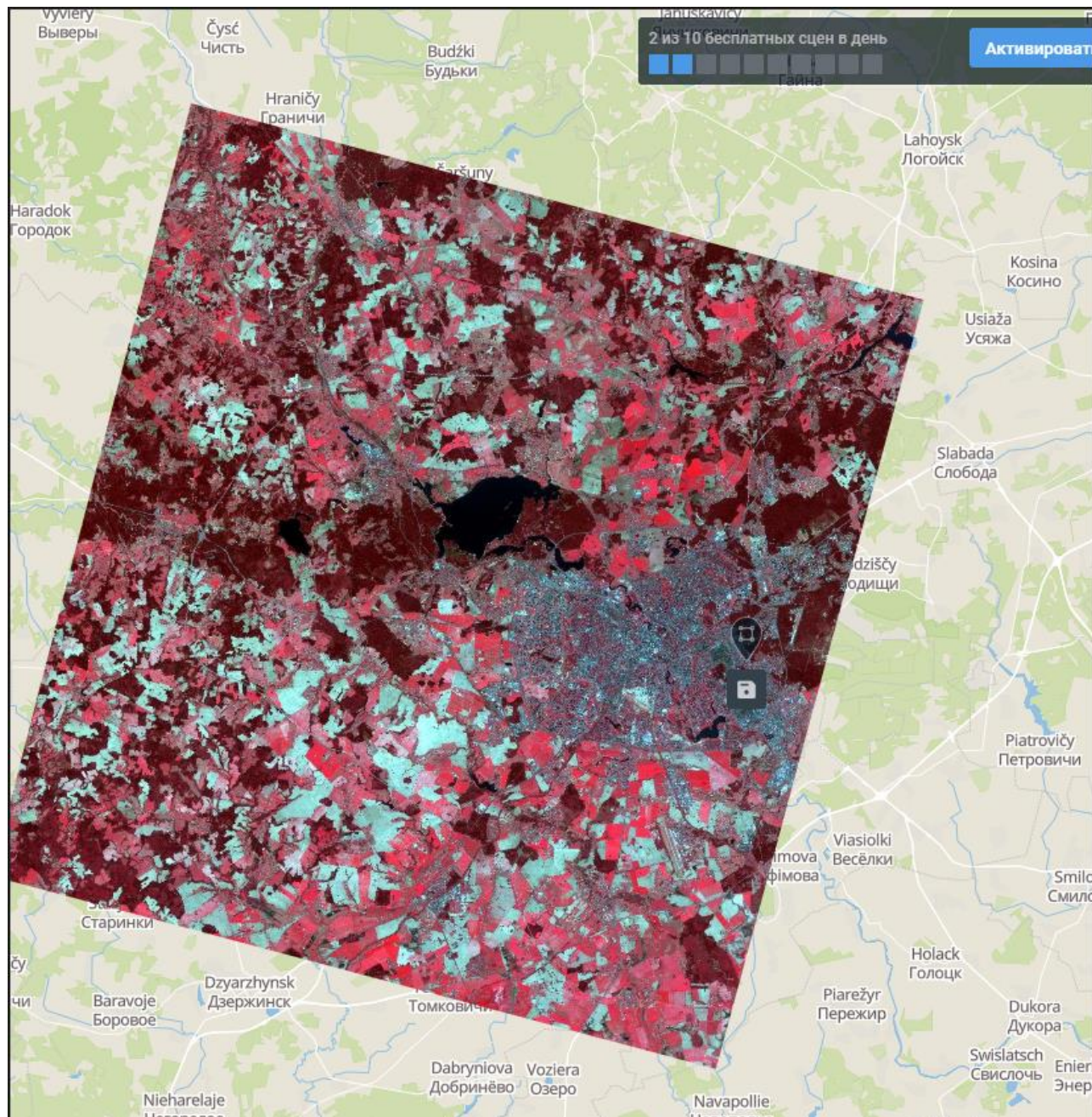
Для уменьшения зависимости от прозрачности атмосферы на космических аппаратах используется многоканальное оборудование, регистрирующее электромагнитное излучение в различных диапазонах длин волн. Поэтому спутниковые снимки состоят из множества снимков, каждый из которых соответствует определенному каналу.

Практика использования каналов зависит в первую очередь от тех задач, которые ставит перед собой исследователь. Комбинация различных каналов позволяет выделять на снимках интересные объекты, например, одинаково зеленые растения при просмотре их в спектральных каналах отличных от видимых изменяют свой цвет в зависимости от интенсивности процессов фотосинтеза, что позволяет делать выводы о насыщении почвы водой и микроэлементами.




Номер канала	Пространственное разрешение, м/пиксел	Спектральное/радиометрическое разрешение	
		Начало спектра, нм	Конец спектра, нм
1	30	450	515
2	30	525	605
3	30	630	690
4	30	760	900
5	30	1550	1750
6	60	10 400	12 500
7	30	2080	2350
8	15	520	900














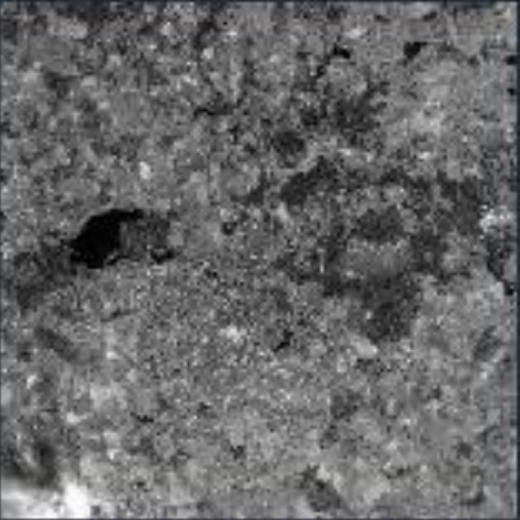
# Настройки поиска

Снимки высокого разрешения (для аналитики)




 2019-04-01 - 2019-09-01 


 ≤ 19% 





1 050 USD


01 июля 2019


2 m/pxl 

0.17km² 


100% 


Ziyuan-3 

0° 







0% 


Добавить в корзину

SCENE SEARCH




All Filters  
High resolution images


  
  
  
  
  






312 USD


18 Dec 2018


0.5 m/px 

0.7295km² 

100% 

Pleiades 1B 

20.00° 

0.00% 

Add to cart

Источником снимков земной поверхности в видимой части спектра с высоким и сверхвысоким разрешением могут послужить сервисы Google и Яндекс. Например, Яндекс приобретает изображения с самым высоким разрешением — со спутников Ikonos, QuickBird и WorldView2, на таких снимках можно разглядеть даже дорожную разметку

**Одним из основных препятствий получения качественного и точного изображения на космическом снимке являются многочисленные атмосферные помехи.** И речь здесь идет не только о плотных облаках, которые совсем не пропускают свет и могут полностью скрывать обширные территории от сканирующей аппаратуры спутника. Это относится и к чистой земной атмосфере, которая поглощает и рассеивает солнечные лучи, причем в различных частях спектра по-разному.

Например, **синие лучи рассеиваются примерно в 16 раз сильнее, чем красные.** Поэтому цвет неба (рассеянный солнечный свет) — синий, а цвет Солнца (прямой солнечный свет) — желтый. Кроме того, в запыленных слоях атмосферы рассеиваются не только синие и голубые лучи, но даже зеленые и часть желтых. И **только красные и оранжевые лучи беспрепятственно проникают сквозь атмосферу.** Таким образом, при составлении цветных снимков из спектральных составляющих необходимо учитывать все эти особенности.

От момента, как снимок получен со спутника, до возможности его анализировать должен пройти целый цикл процедур по приведению его в вид, удобный для получения и последующего анализа визуальной информации.

В цикл предварительных процедур по обработке снимка входят следующие:

- **радиометрическая коррекция** – устраняет варьирование значений яркости пикселей, которое происходит в результате неправильной работы детекторов, влияния рельефа и атмосферы;
- **атмосферная коррекция** – коррекция влияния атмосферы, которая определяет расположение диапазонов съемки за счет окон прозрачности;
- **геометрическая коррекция** включает исправление таких искажений снимка, как полосчатость, выпадение строк, также геокодирование – привязывание снимка таким образом, что каждой точке изображения задается координата соответствующей точки на местности;
- **ортотрансформирование** – при нем устраняются ошибки изображения за счет перепадов высот рельефа местности.

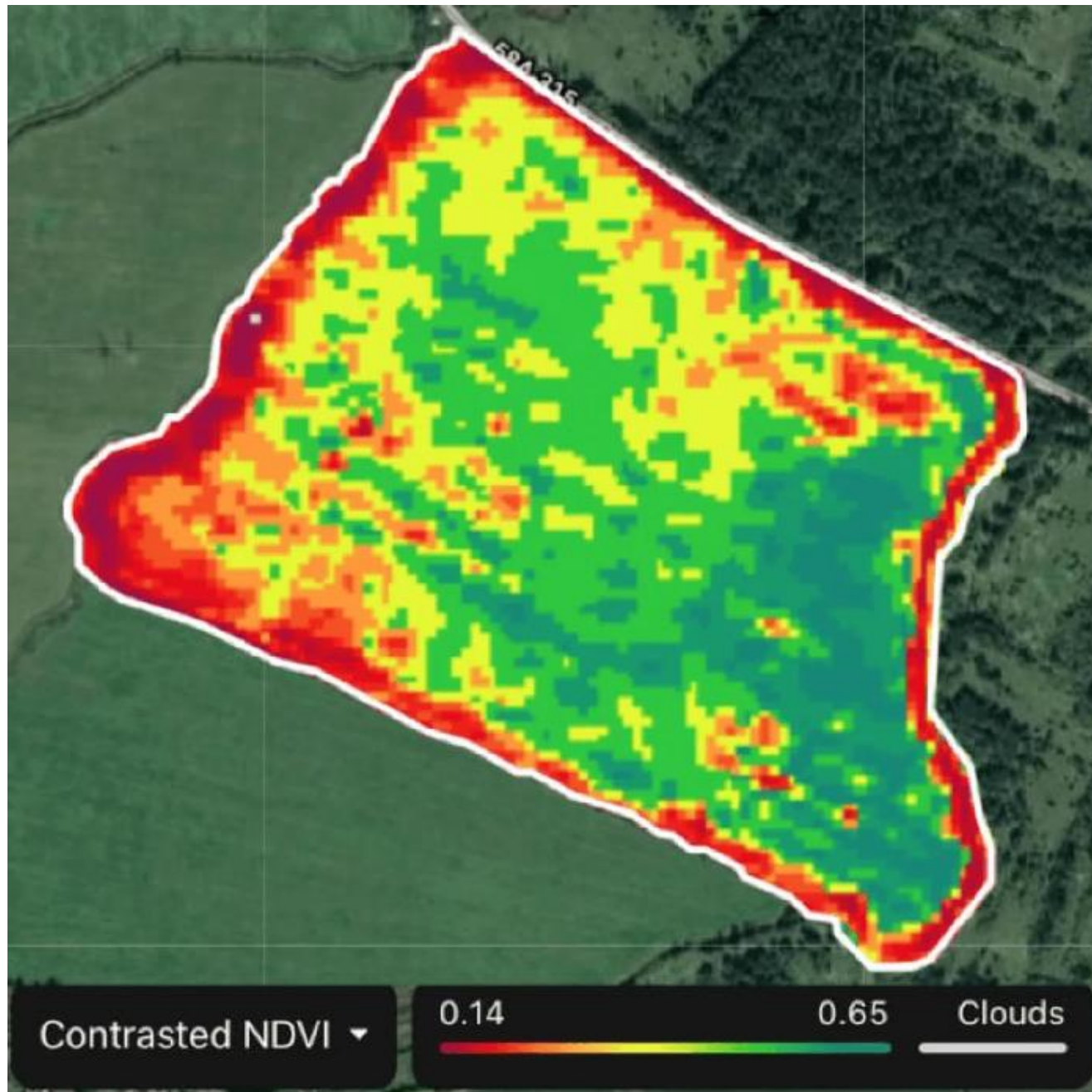


**NDVI (Normalized difference vegetation index, Нормализованный вегетационный индекс)** — это числовой показатель качества и количества растительности на участке поля. Он рассчитывается по спутниковым снимкам и зависит от того, как растения отражают и поглощают световые волны разной длины.

Например, растения для нашего глаза зелёные потому, что пигмент хлорофилл хорошо отражает зелёные волны. Также хлорофилл поглощает красные волны: за счёт этого происходит фотосинтез, то есть культура растёт и развивается. А клеточная структура растения отражает ближние инфракрасные волны. Получается, что **здоровое растение, в котором много хлорофилла и хорошая клеточная структура, активно поглощает красный свет и отражает ближний инфракрасный. Больное растение — с точностью до наоборот.**

Чтобы понять состояние здоровья растения, надо вычислить отношение между разностью интенсивностей отраженного света в красном (Red) и инфракрасном диапазоне (Nir) и их суммой. Это и есть NDVI.

$$\text{NDVI} = \frac{\text{NIR} - \text{RED}}{\text{NIR} + \text{RED}}$$



Чем выше показатель **NDVI** тем здоровее растения на этом участке

Можно  
корректировать  
удобрение, полив и тд

# Применение компьютерного зрения

Приложения дополненной  
реальности

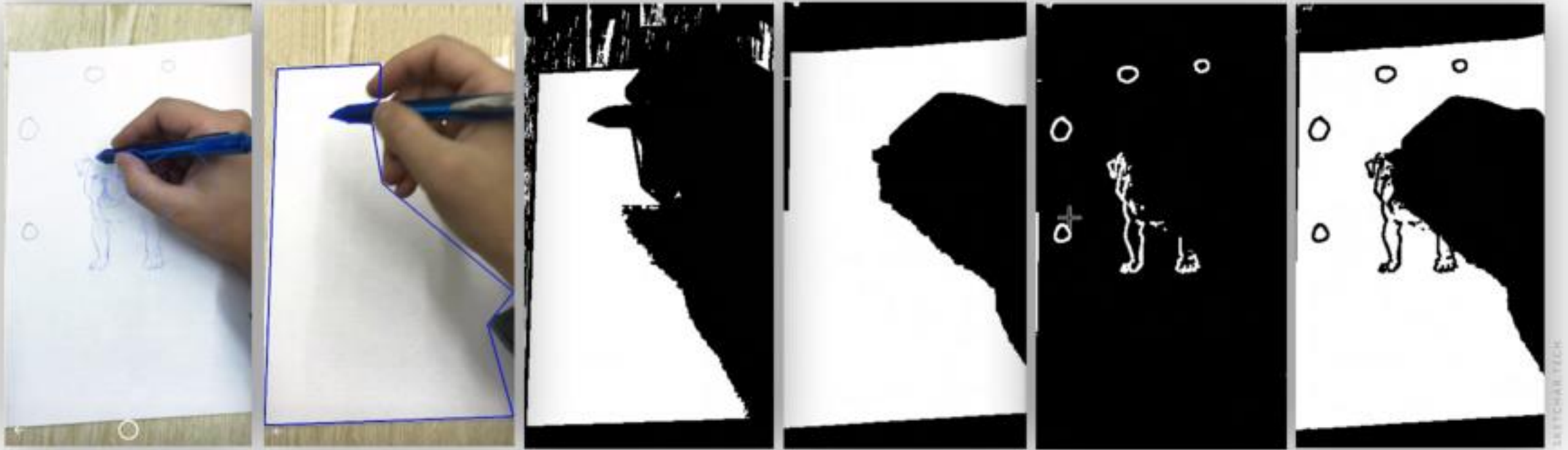
# SketchAR (опыт разработки)

Плоскости, на которых рисуют, бывают разных фактур, цветов и детализаций. В случае, если грубо определить окружающую среду, алгоритм будет «отвлекаться» на посторонние “шумы”. Эта проблема решается фильтрами.

Унифицировать поверхности сложно, если, конечно, обучать нейронные сети на реальных данных от пользователей. **Но обучение нейронных сетей — процесс долгий, поэтому рассматривать их как самостоятельный метод не верно.** Только дополнять основной метод.







Решение: разделить на слои

1. виртуальное изображение расположенное на поверхности видимое пользователю на экране смартфона.
2. Определение листа на поверхности
3. Грубое отделение листа от фона. Виден шум на поверхности стола и тень руки определяется как единое с рукой.
4. Четкое отделение окружения от листа и руки.
5. Определение только нарисованного контента от остального
6. Определение нарисованного контента относительно окружения.

# Темы для поощрительного доклада на лекции:

## ✓ Реализация своего проекта при помощи библиотеки MediaPipe

Выбрать какую-либо модель MediaPipe на официальном сайте <https://mediapipe.dev/> и реализовать с помощью неё конкретную прикладную задачу, к примеру:

- Распознавание определенных жестов с какими-либо управляющим воздействиями (например, производится снимок с сохранением или открывается какое-либо приложение)
- Подсчет приседаний или отжиманий для фитнес-приложений
- Наложение маски с сохранением фото
- Любая другая задача на ваше усмотрение . . .

## ✓ Проект по обработке спутниковых снимков

## ✓ Проект распознавания изображений с использованием любой обученной модели нейронной сети.