

Компьютерное зрение. Математика в задачах обработки изображений

Горбачев Вадим Александрович, к.ф.-м.н.
МФТИ, ГосНИИАС



Компьютерное зрение

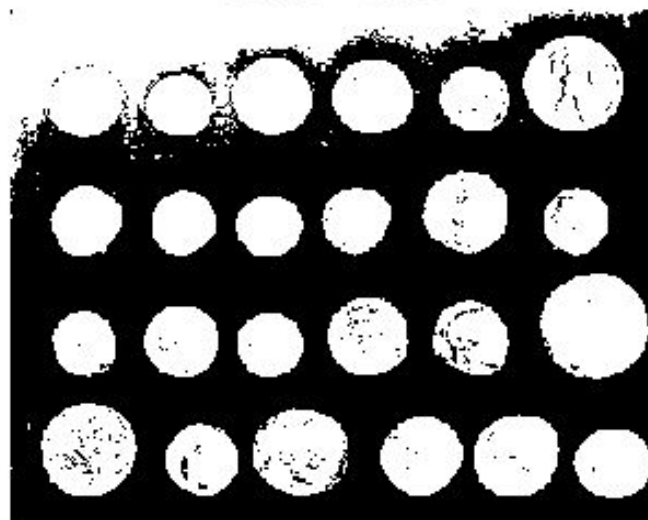


Примеры задач

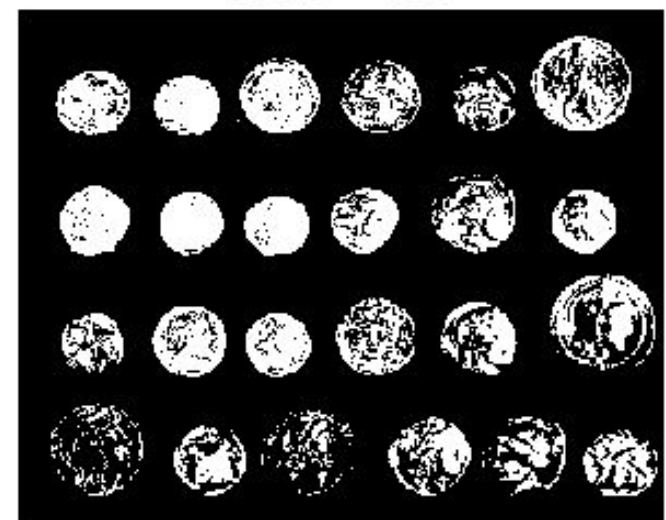
- Распознавание автомобильных номеров
- Классификация изображений
- Выявление движущихся объектов на видео, слежение
- Создание панорам из изображений
- Построение трёхмерных моделей объектов по снимкам
- Детектирование объектов определённого класса
- Ориентация робота в пространстве
- Синтез реалистичных текстур
- Поиск по изображениям
- Анализ медицинских изображений (подсчёт клеток и их типов)
- Биометрия

Пример:
Задача
выделения
объектов

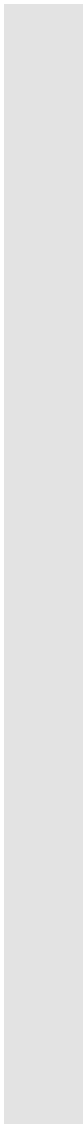

- Задача: выделить объекты, определив порог бинаризации



Низкий порог



Высокий порог

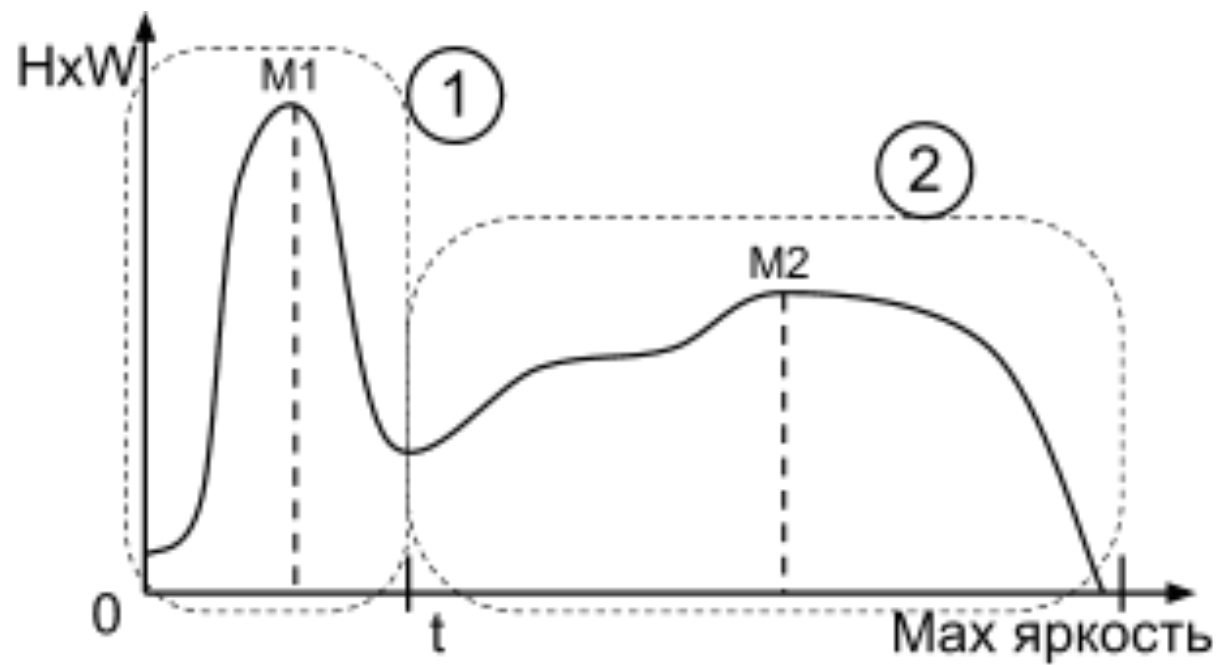


Яркости пикселей
объектов и фона имеют
различное
распределение

Гипотеза

Модель

- Бимодальное распределение



Метод Оцу

- Идея: найти порог оптимальным образом, так чтобы минимизировать дисперсию внутри одного класса (и максимизировать дисперсию между классами)

Гистограмма строится по значениям $p_i = \frac{ni}{N}$. В данной формуле N – общее количество пикселей изображения с уровнем яркости i . Порог t представляет собой целое значение от 0 до $L=\text{max}$. При помощи гистограммы все пиксели разделяются на «полезные» (объектные) и фоновые. Каждому виду соответствуют относительные частоты W_0 и W_1 :

$$W_0(t) = \sum_{i=1}^t p_i \quad (6)$$

$$W_1(t) = \sum_{i=t+1}^L p_i = 1 - W_0(t). \quad (7)$$

Далее вычисляются средние уровни для каждого вида изображения по формулам:

$$\mu_0(t) = \sum_{i=1}^t \frac{ip_i}{W_0} \quad (8)$$

$$\mu_1(t) = \sum_{i=t+1}^L \frac{ip_i}{W_1} \quad (9)$$

Далее ищется порог, который уменьшает дисперсию внутри вида пикселей, определяемую следующей формулой:

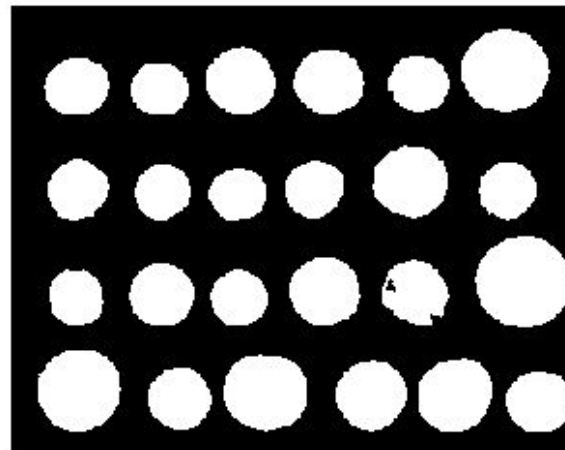
$$\delta_W^2(t) = W_1(t)\delta_1^2(t) + W_2(t)\delta_2^2(t) \quad (10)$$

Следующим шагом определяется межклассовая дисперсия, по формуле, представленной ниже:

$$\sigma_{\omega}^2(t) = W_0(t)W_1(t) * (\mu_1(t) - \mu_0(t))^2 \quad (11)$$

Затем вычисляется максимальное значение для оценки качества деления изображения на две части, которое соответствует искомому порогу:


$$\eta(t) = \max \left[\frac{\sigma_{\omega}^2(t)}{\delta_W^2(t)} \right] \quad (12)$$



Пример

- Задача: разделить изображение на области так чтобы каждая область соответствовала одному объекту





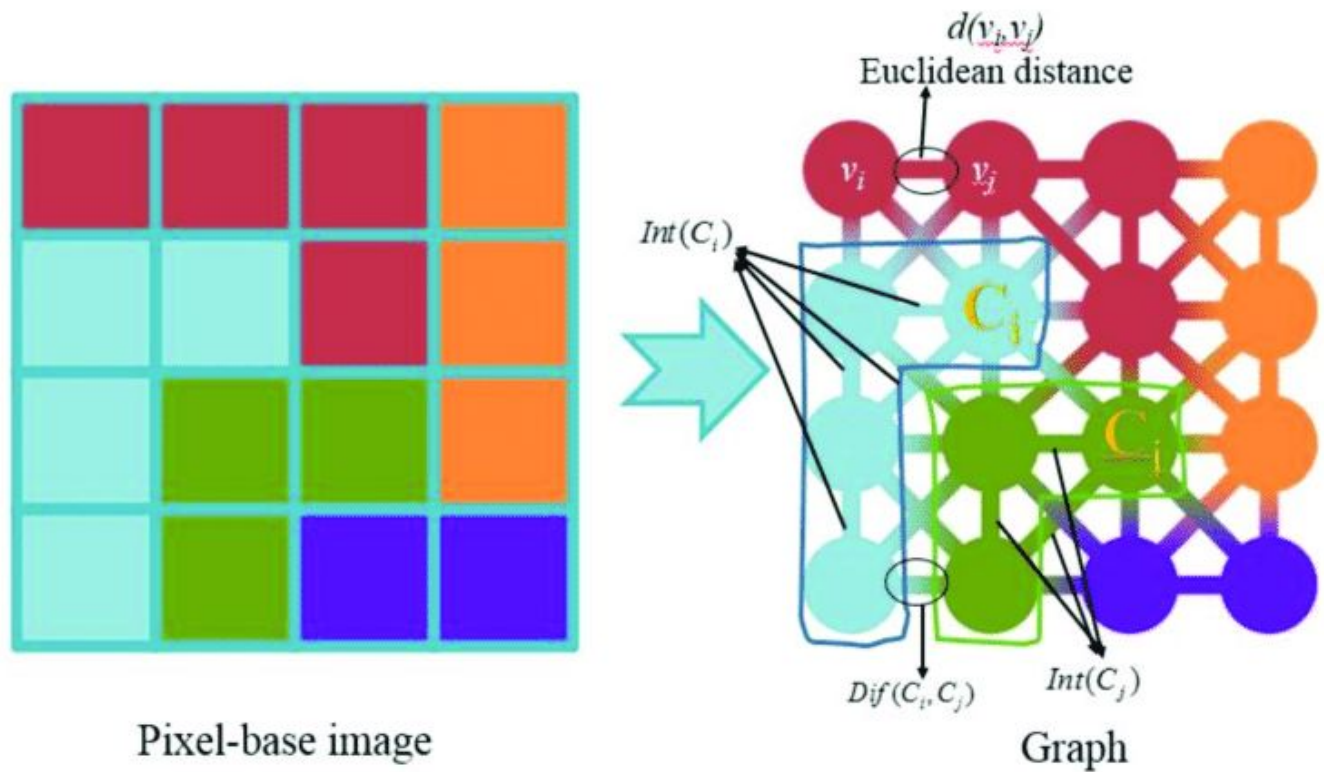
Соседние пикселы
внутри одного объекта
мало различаются по
яркости

Гипотеза



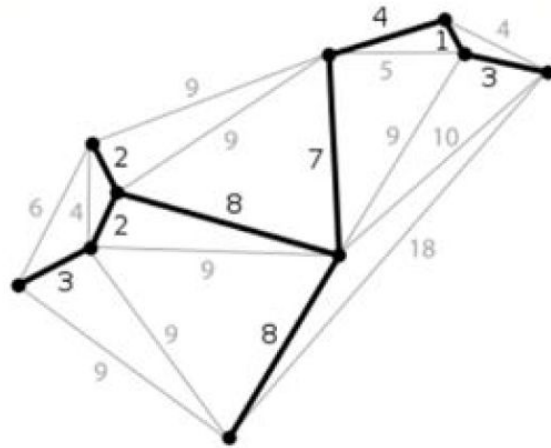
Модель

- Неориентированный граф. Вершины - это все пиксели. Рёбра ставятся между соседними пикселями. Чем больше разность яркости - тем больше вес ребра.



Метод «Efficient Graph-Based Image Segmentation »

- Идея: Будем добавлять рёбра между пикселями с минимальным весом, чтобы строить MST. Если ребро объединяет два дерева (региона), используем хитрое правило



- «Внутренняя разница» в компоненте C :

$$Int(C) = \max_{e \in MST(C, E)} w(e) \quad \text{где } w(e) - \text{вес ребра (мера различия двух пикселей)}$$

- Разница между областями:

$$Dif(C_1, C_2) = \min_{v_i \in C_1, v_j \in C_2, (v_i, v_j) \in E} w((v_i, v_j))$$

- Предикат присутствия границы между областями:

$$D(C_1, C_2) = \begin{cases} \text{true} & \text{if } Dif(C_1, C_2) > MInt(C_1, C_2) \\ \text{false} & \text{otherwise} \end{cases}$$

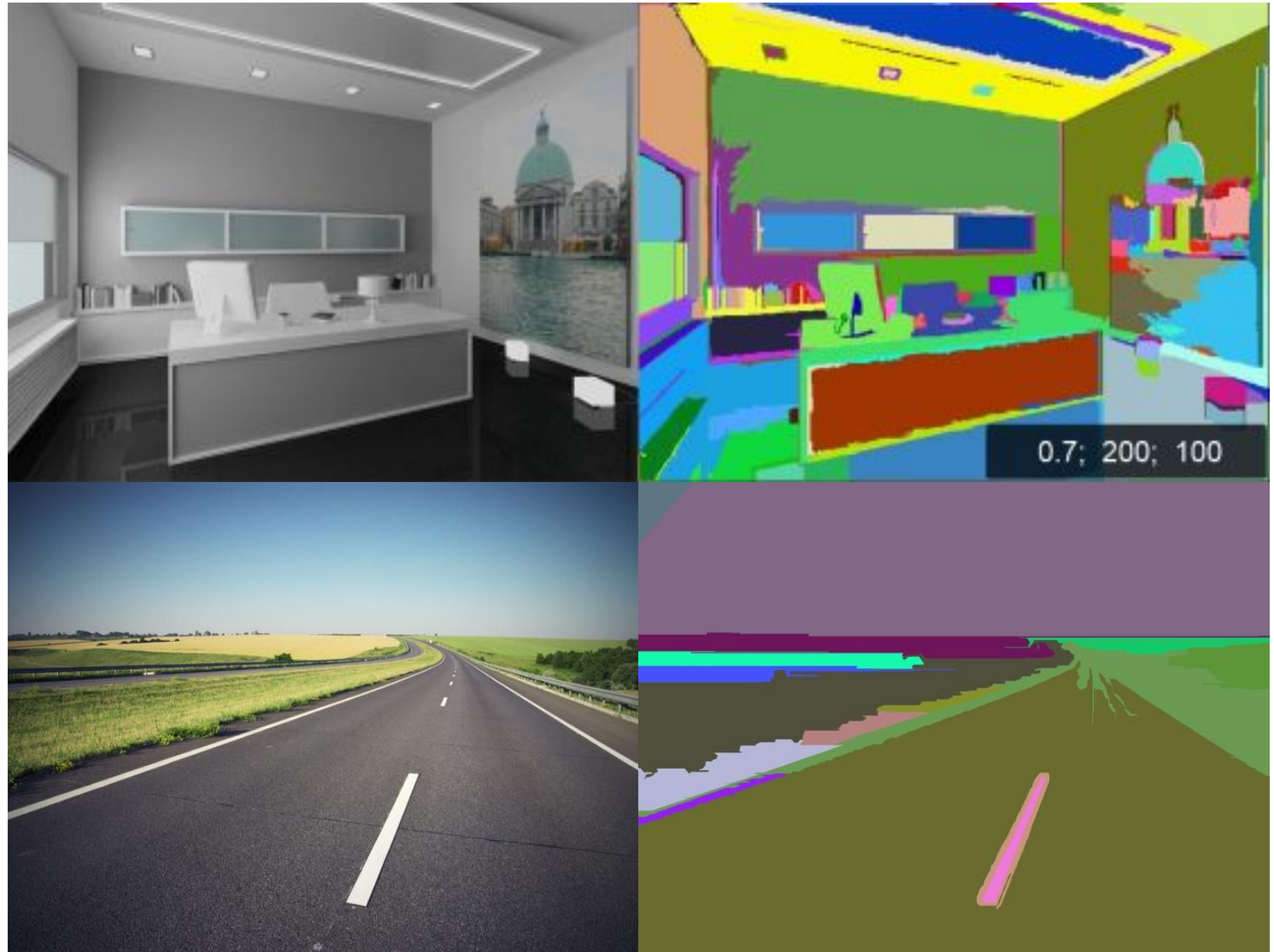
где $MInt$ – минимальные колебания интенсивности по областям
 $MInt(C_1, C_2) = \min(Int(C_1) + \tau(C_1), Int(C_2) + \tau(C_2))$

и регуляризационный параметр $\tau(C) = k/|C|$

где $|C|$ - размер области, k – коэффициент регуляризации

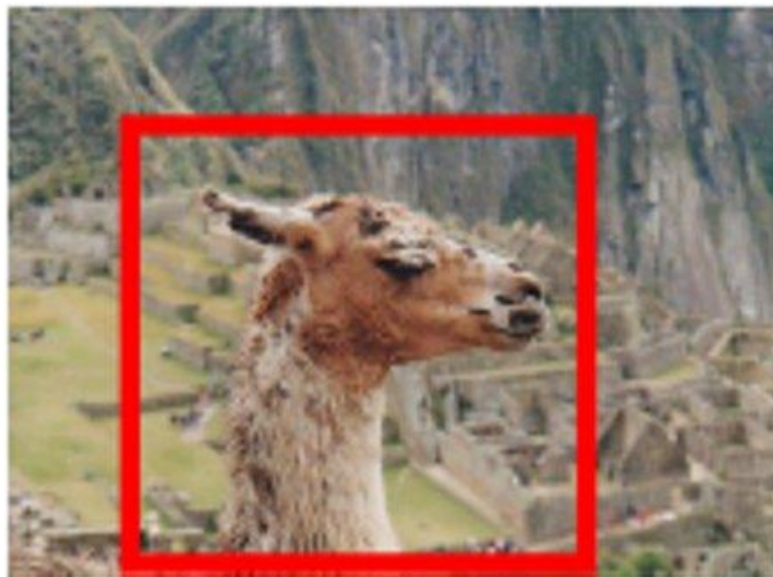


Graph-Based Image Segmentation



Пример

- Внутри областей имеются резкие границы, цвет неоднороден

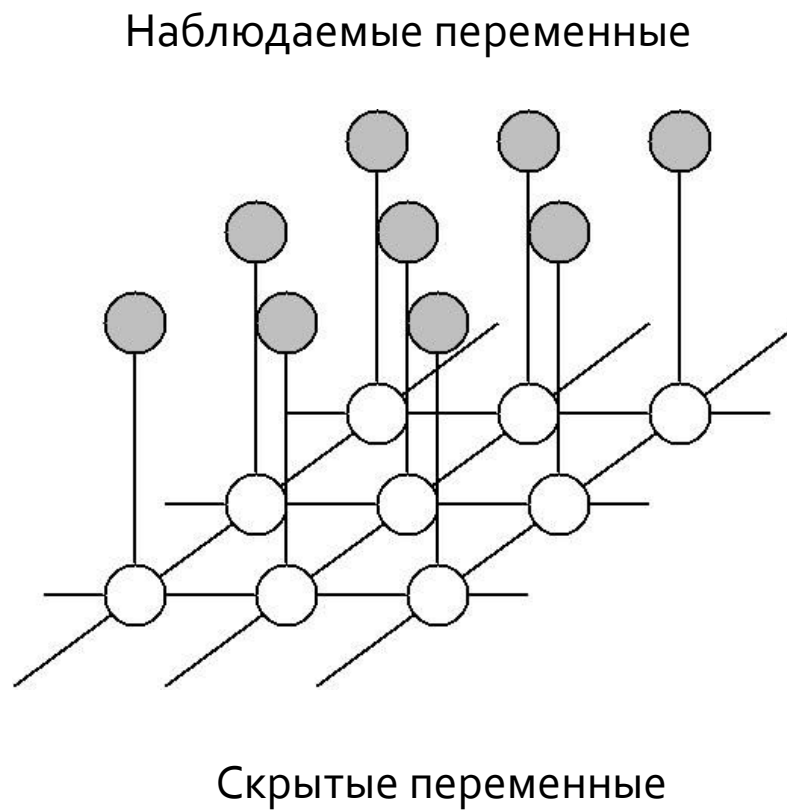


Будем штрафовать
точки границы, чтобы
она была только там
где точно
необходимо

Гипотеза

Модель

- Условные/марковские случайные поля

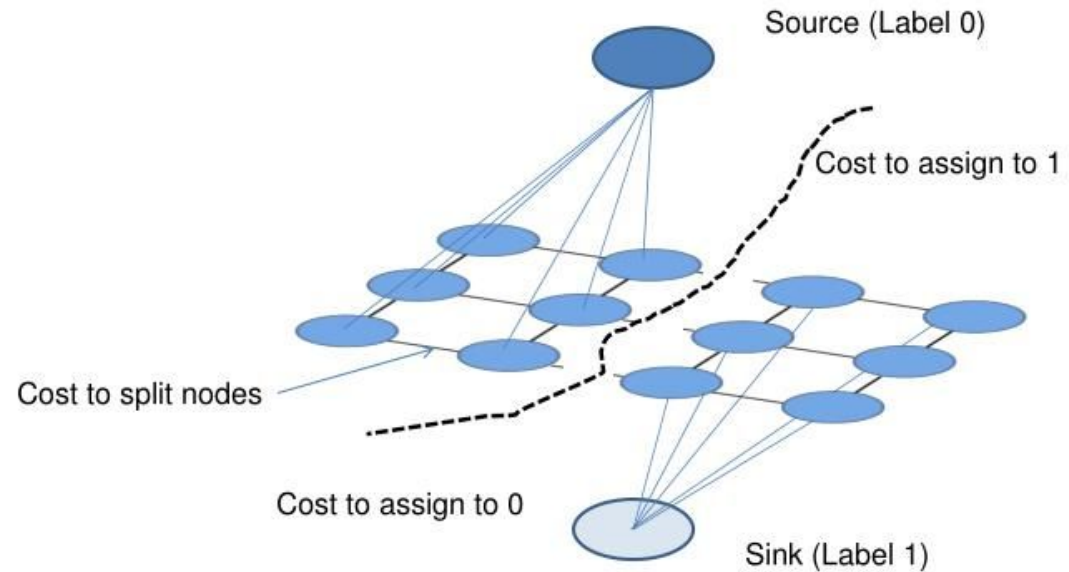


Вероятностная
модель связей

Метод GrubCut

- Идея: представим картинку как граф, веса рёбер зададим в соответствии с энергиями. Применим метод разреза графа

Solving MRFs with graph cuts




$$Energy(\mathbf{y}; \theta, data) = \sum_i \psi_1(y_i; \theta, data) + \sum_{i, j \in edges} \psi_2(y_i, y_j; \theta, data)$$

Пример

- На изображении много областей, они относительно однородны





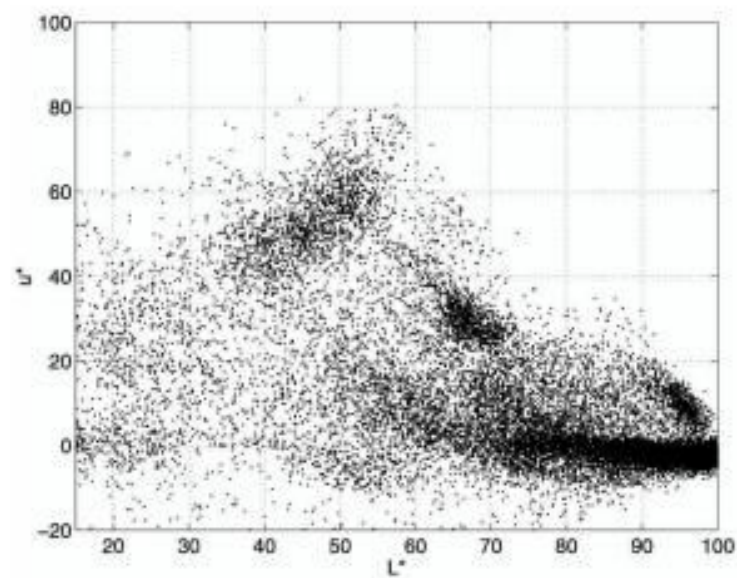
Пикселы одного объекта близки в пространстве яркость-координата

Гипотеза

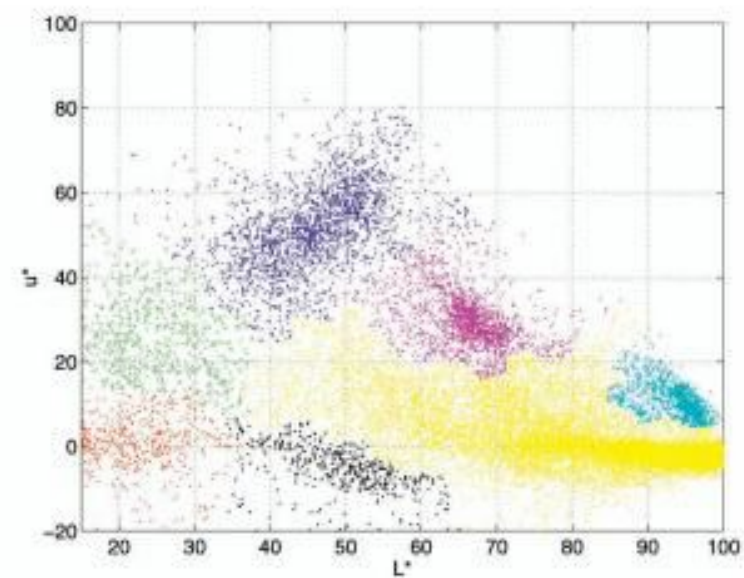


Модель

- Кластеры в многомерном пространстве (RGB-XY)



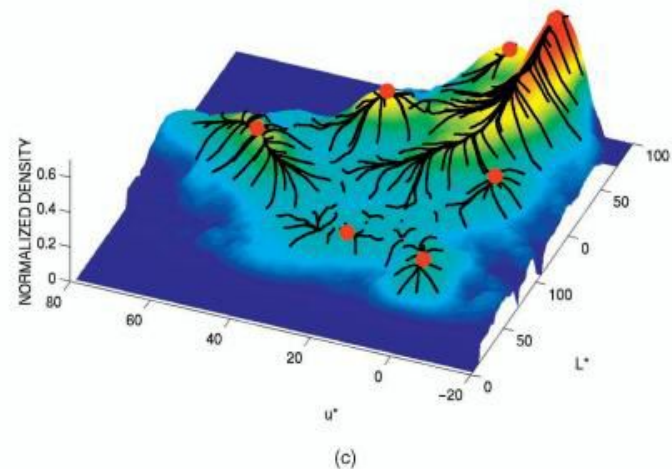
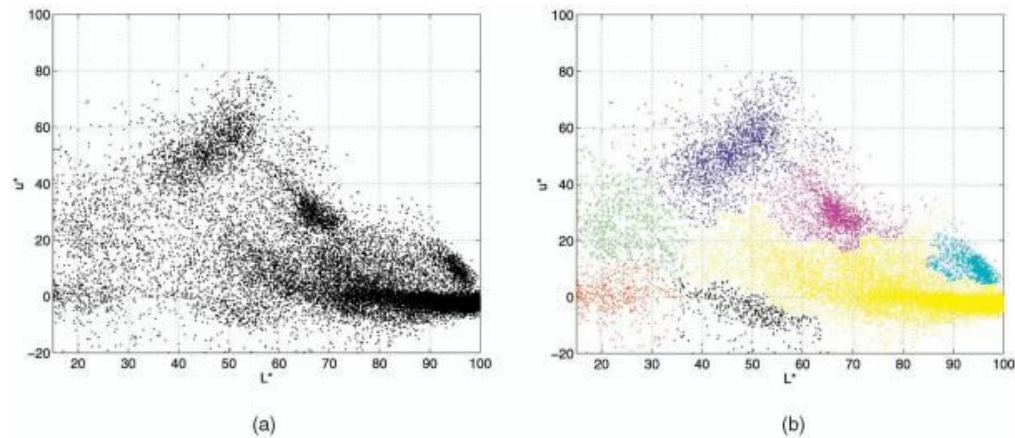
(a)



(b)

Метод Mean-Shift

- Идея: Сопоставим пикселям изображения точки в пространстве RGBXY. Выполним оценку среднего в окрестности точки. Запишем вектор смещения среднего относительно каждой точки. Все точки, от которых вектора смещаются в один и тот же центр отнесём к одному классу

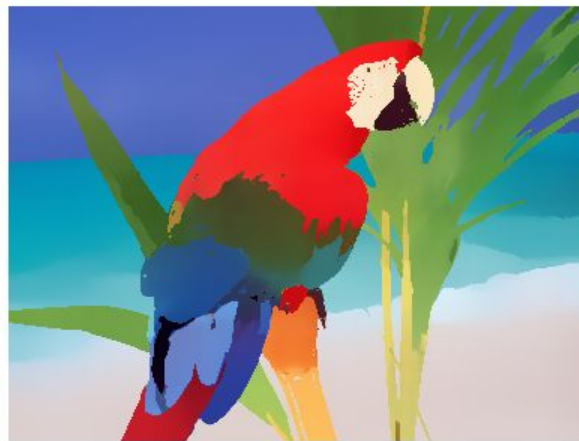


Mean-Shift

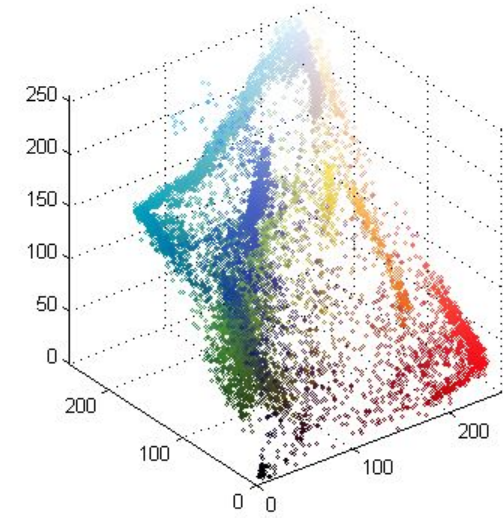
input image



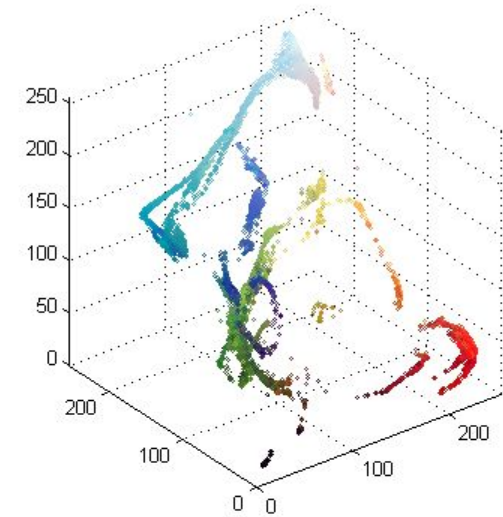
output image



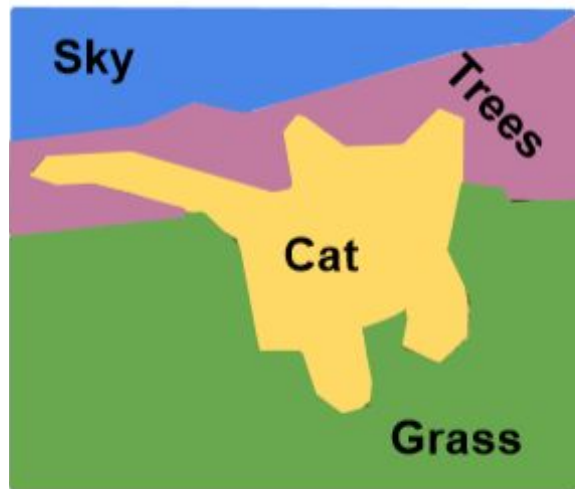
Pixel Distribution Before Meanshift



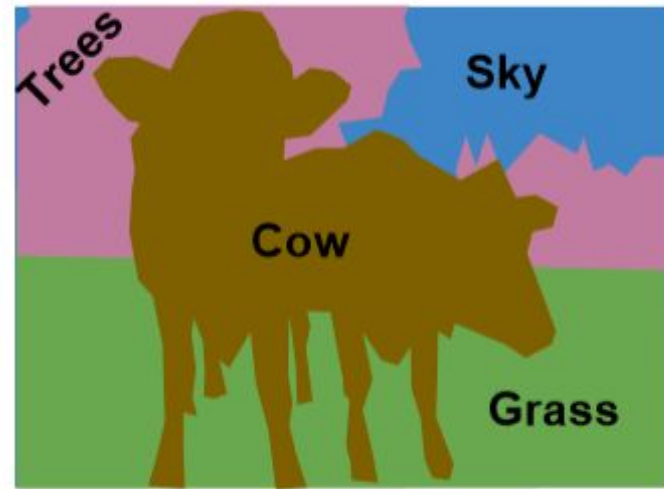
Pixel Distribution After Meanshift



Совсем
сложный
пример



This image is [CC0 public domain](#)



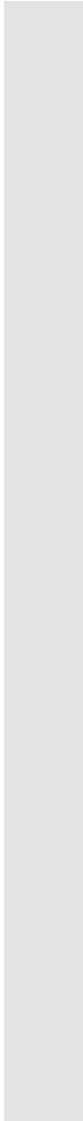



Нейросети могут всё!

Гипотеза

Модель

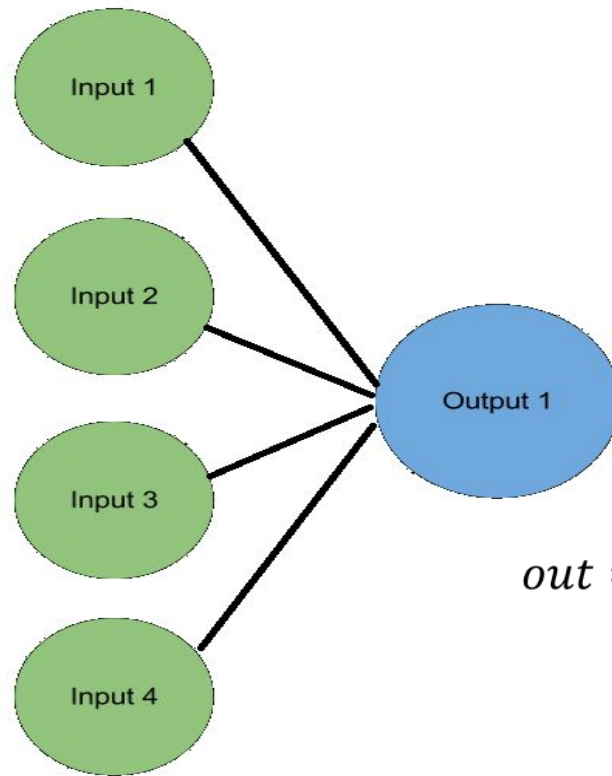
- Модель придумывать не будем
- Соберём побольше данных и закинем в нейросеть
- ...
- Profit!!!



Нейронные сети Эволюция архитектур

Линейный классификатор

- Сумма + Нелинейность (Сигмоид) = Logistic regression



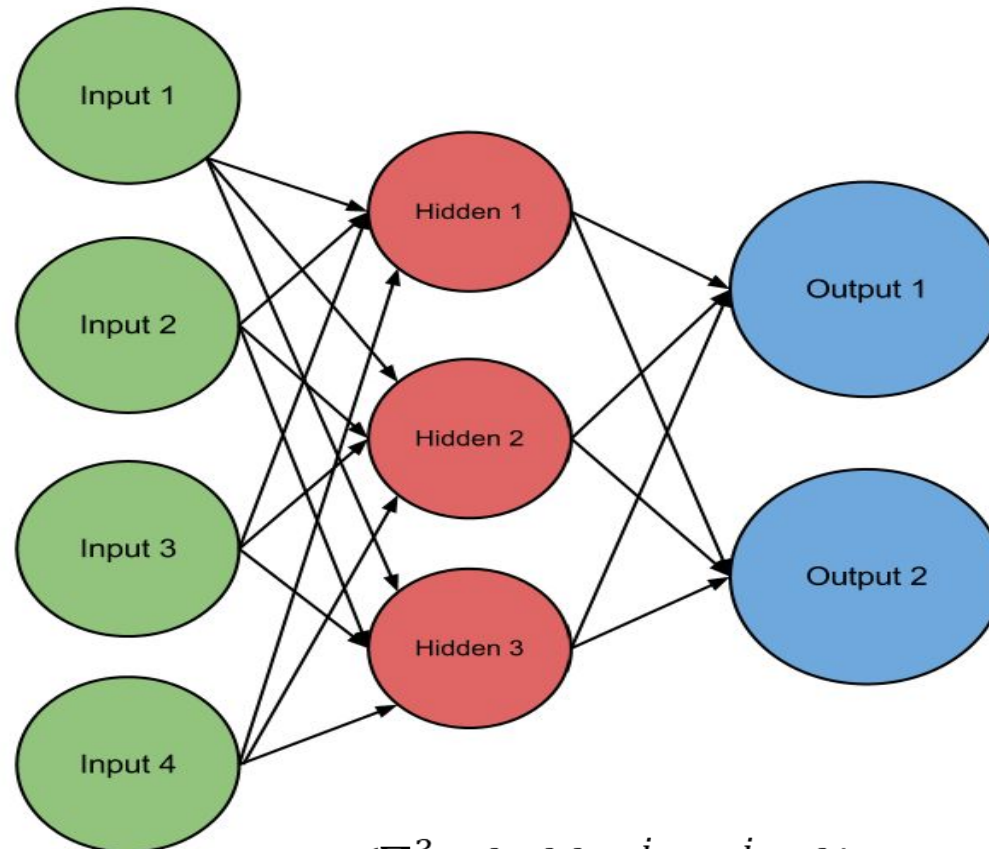
$$out = \text{sigm}(\sum_{i=1}^4 input_i * w_i + b)$$

- это и есть нейрон

Нейронные сети

- LogReg + LogReg = Neural Network

$$hidden^j = \text{sigm}(\sum_{i=1}^4 input_i * w_i^j + b^j)$$

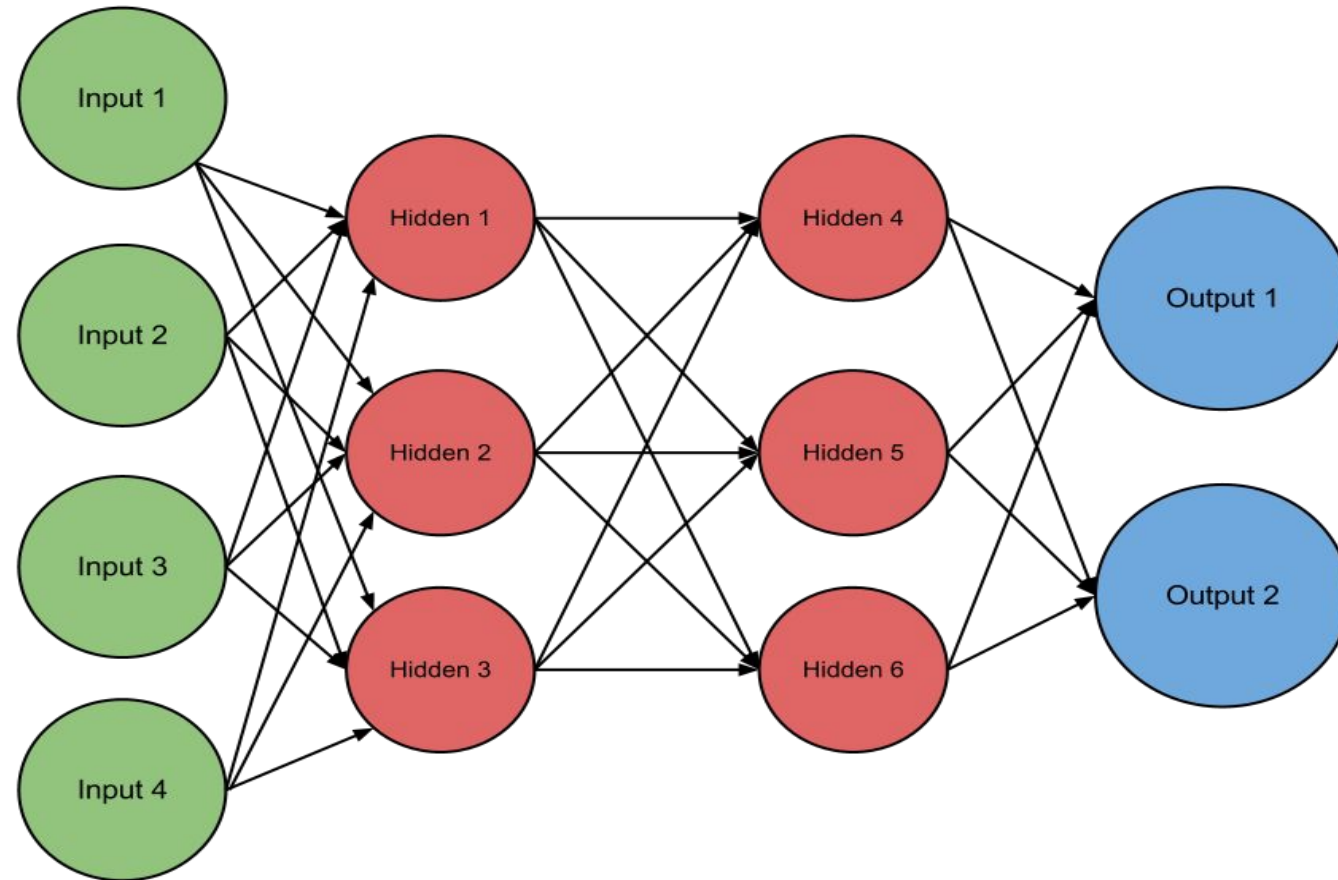


$$out = \text{sigm}(\sum_{j=1}^3 hidden^j * w^j + b)$$

- За счёт нескольких нелинейностей нейросеть не сводится к линейной функции

Многослойные нейронные сети

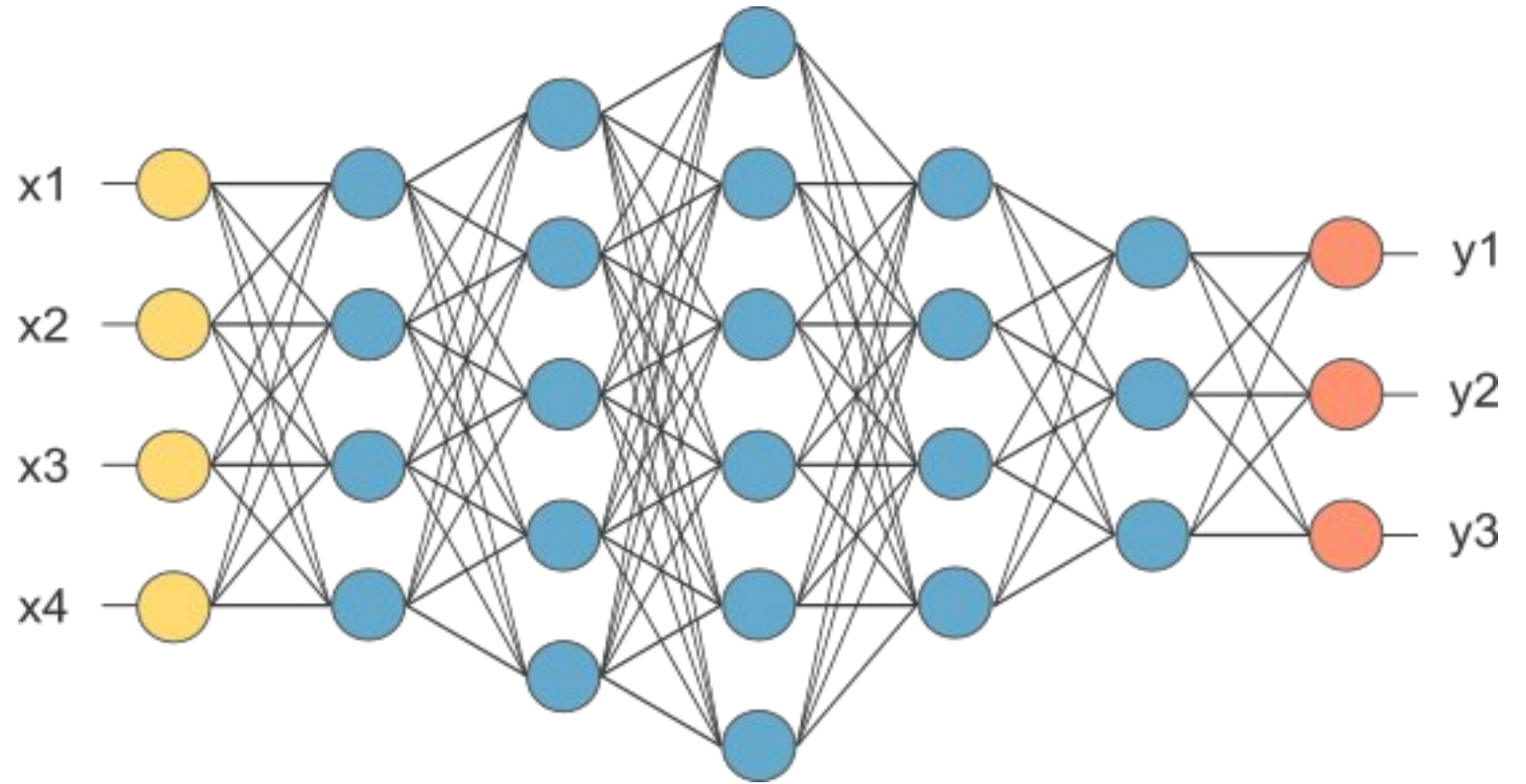
- Больше скрытых слоёв!



- Это всё ещё комбинация суммирования и нелинейности

Многослойные нейронные сети

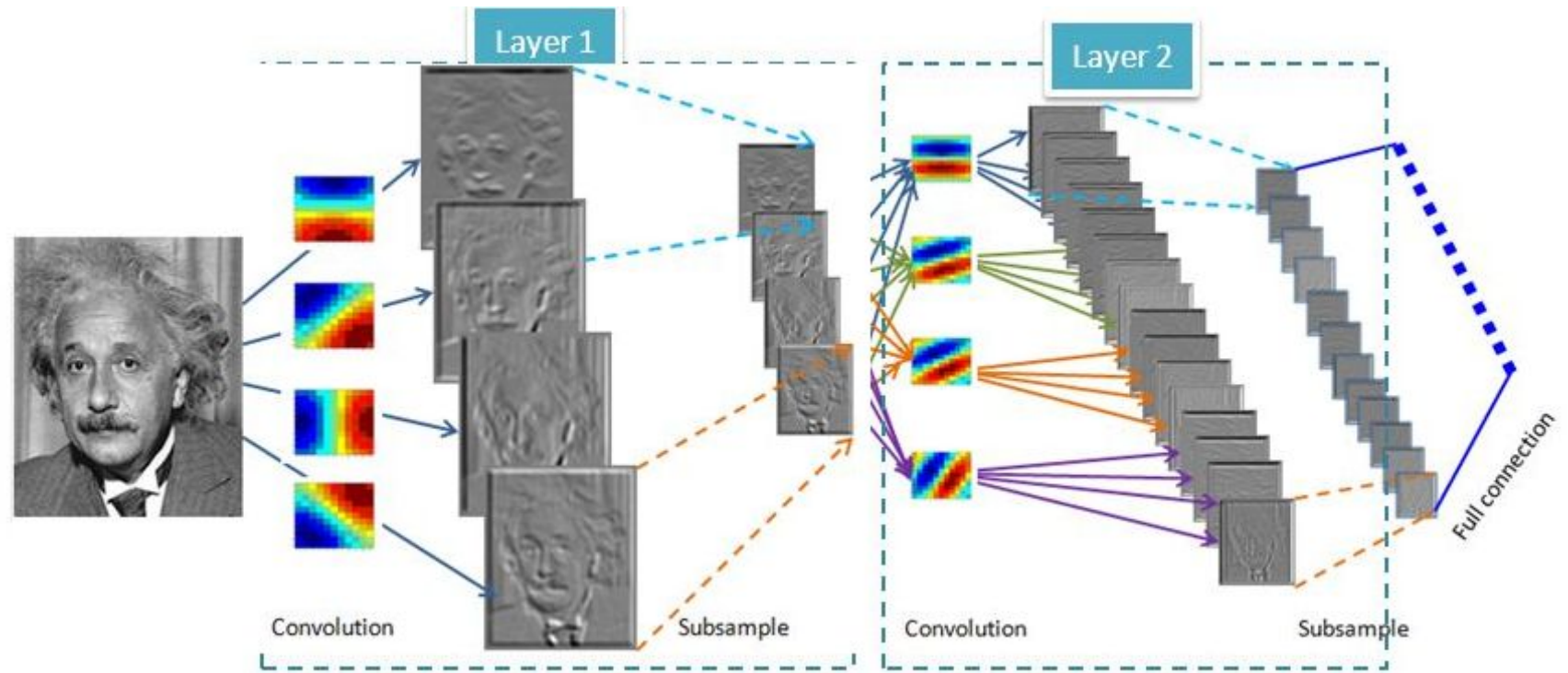
- ЕЩЁ БОЛЬШЕ!



- ЭТО МАГИЯ! Это всё ещё комбинация суммирования и нелинейности

Свёрточные сети

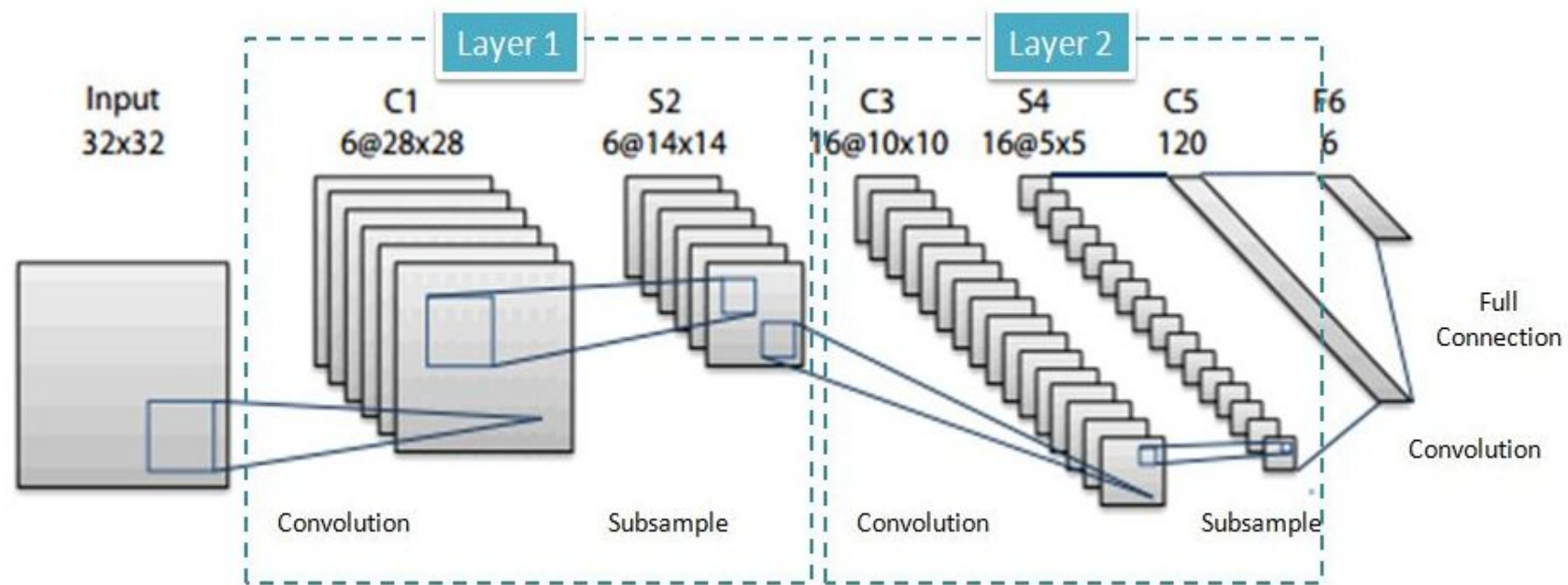
- Свёртка + Нейросеть = Свёрточные сети



- При свёртке входов сети гораздо меньше, чем количество пикселей

Свёрточные сети

- Свёртка + Нейросеть = Свёрточные сети

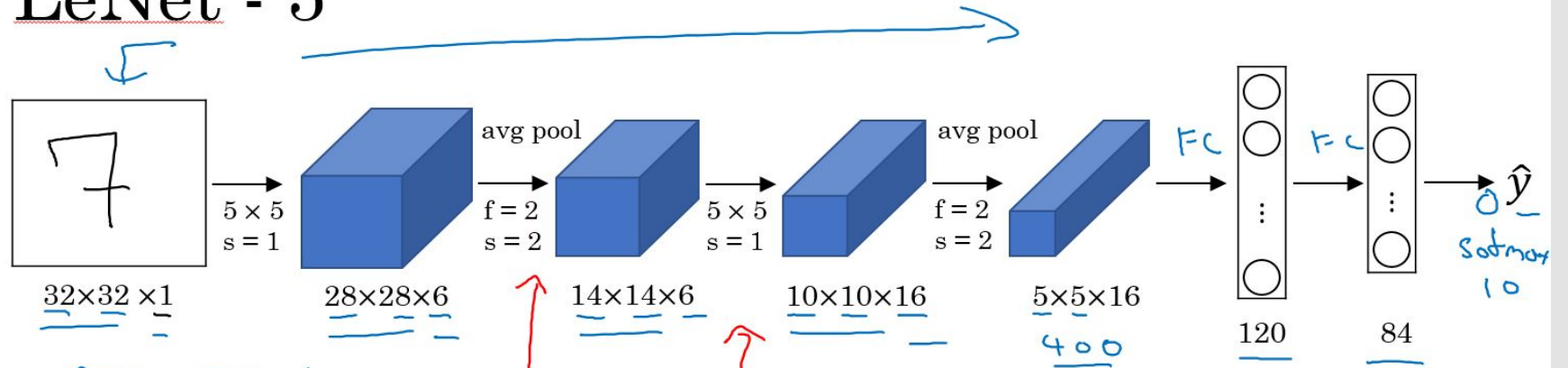


- Да, ещё используется уменьшение размера картинки (Pooling/Subsampling)

Свёрточные сети

- Одна из первых сетей для классификации

LeNet - 5



60K parameters.
 $n_H, n_W \downarrow$ $n_C \uparrow$
 conv pool conv pool fc fc output
 Advanced: sigmoid/tanh ReLU

non-linearity after pooling $n_H \times n_W \times n_C$ $f \times f \times n_C$



Неплохо, но так просто уже никто не делает.
 Гляньте на GoogLeNet

[LeCun et al., 1998. Gradient-based learning applied to document recognition]

Применение

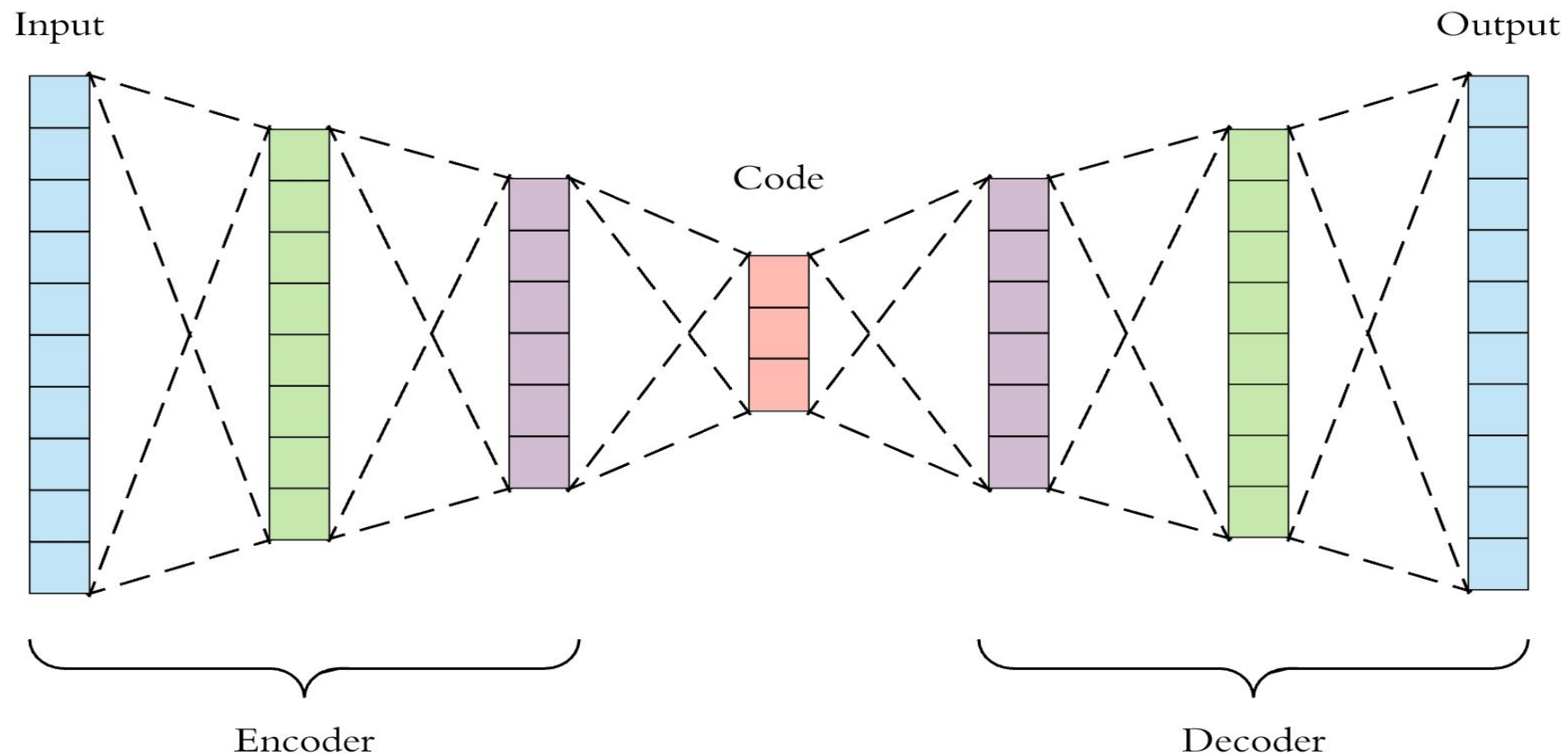
- Классификация
- Регрессия
- Детектирование
- Обучение представлений (Embeddings)
- ...

За кадром

- ReLU
- Dropout
- 1x1 convolution
- Residual Blocks
- Inception Blocks
- Batch Normalization
- Fully Convolutional Networks

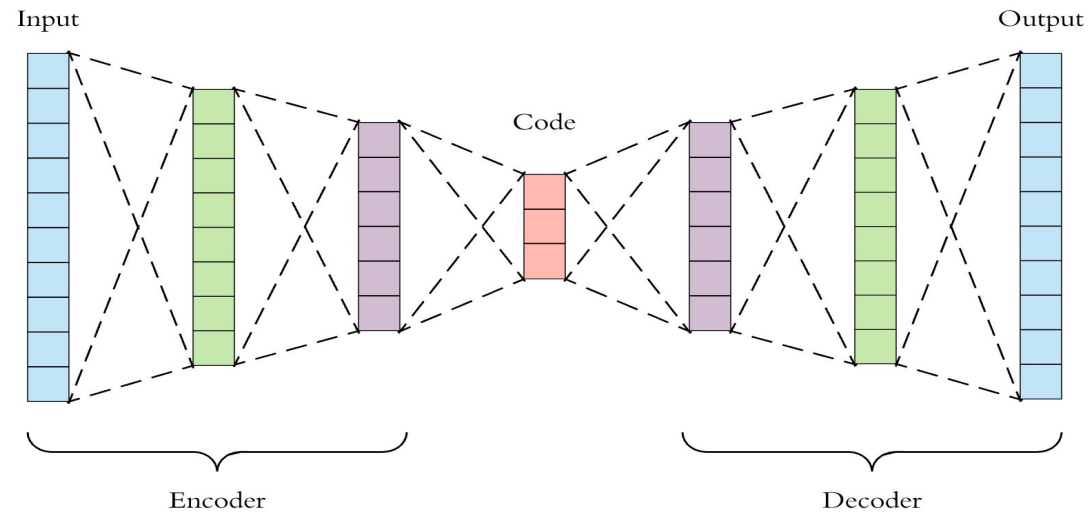
Автоэнкодер

- Нейросеть + Нейросеть наоборот = Автоэнкодер



Автоэнкодер

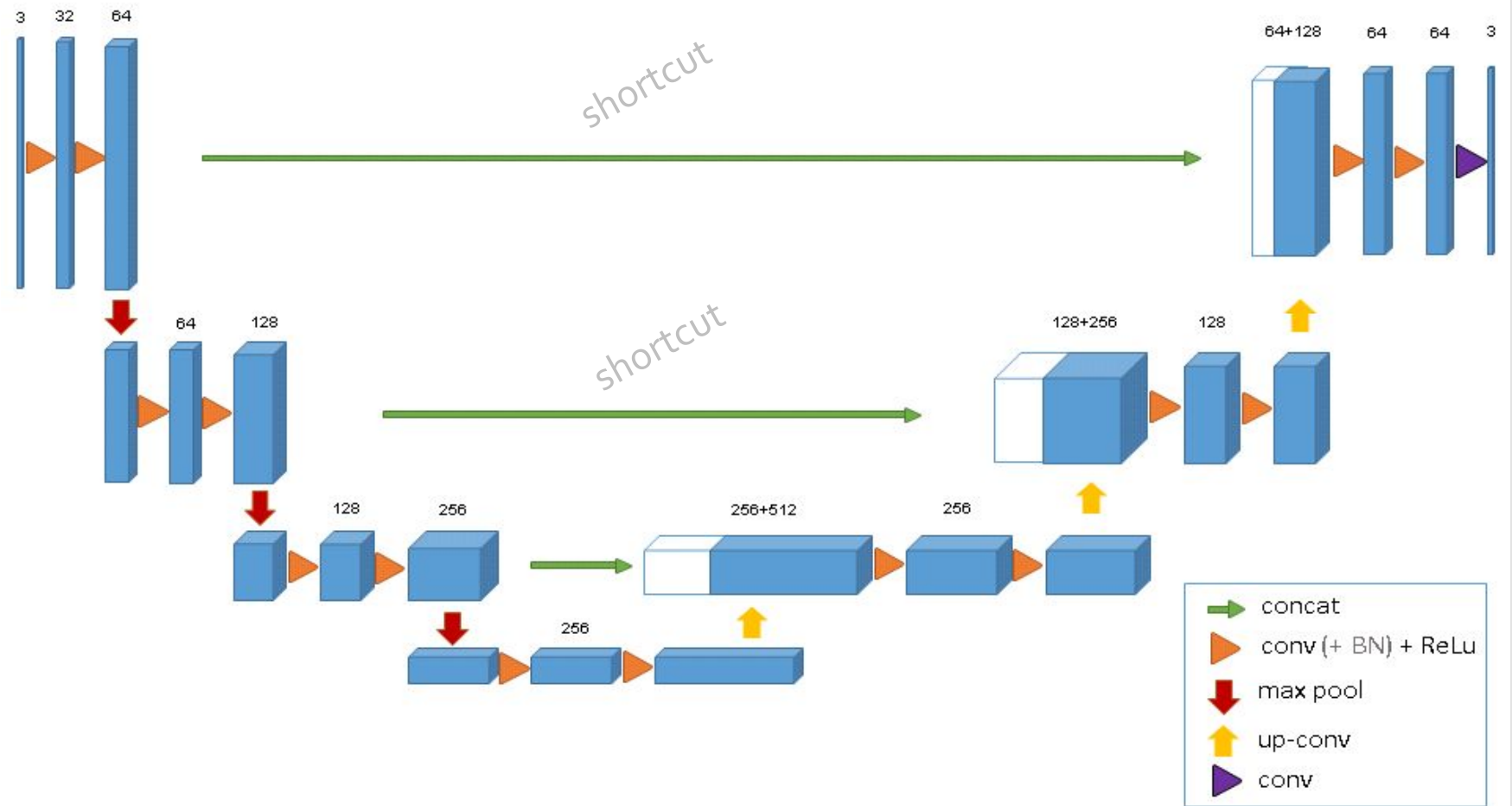
- Нейросеть + Нейросеть наоборот = Автоэнкодер



- Если сигнал на выходе будет близок ко входу, то Code будет представлять собой хорошее компактное описание сигнала
- Система может снижать уровень шума сигнала, восстанавливать изображения
- Система может переводить данные в пространство меньшей размерности
- Decoder может преобразовывать сигнал в другую область, например переводить предложения или генерировать ответы на фразы

U-net

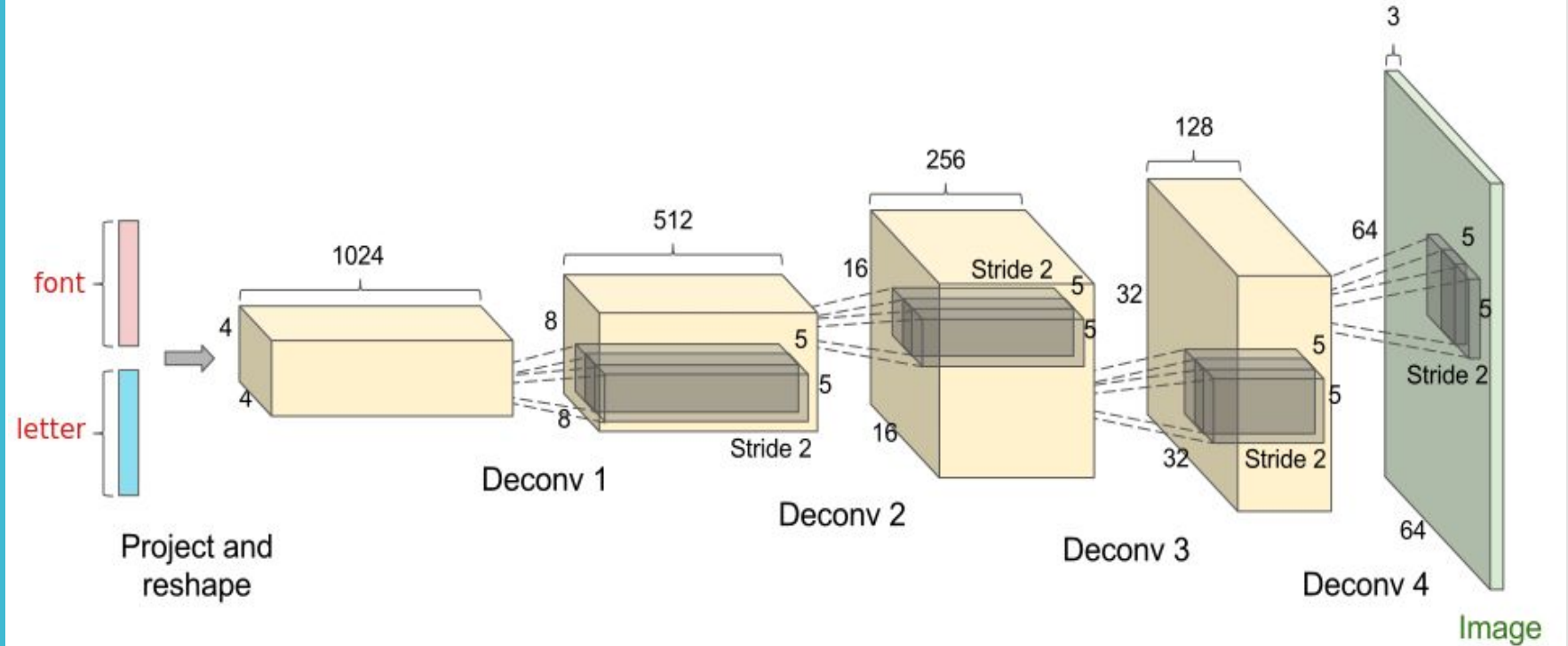
- Автоэнкодер + Skip-connection = U-net



- Можно делать сегментацию изображений!

Генерирующая сеть

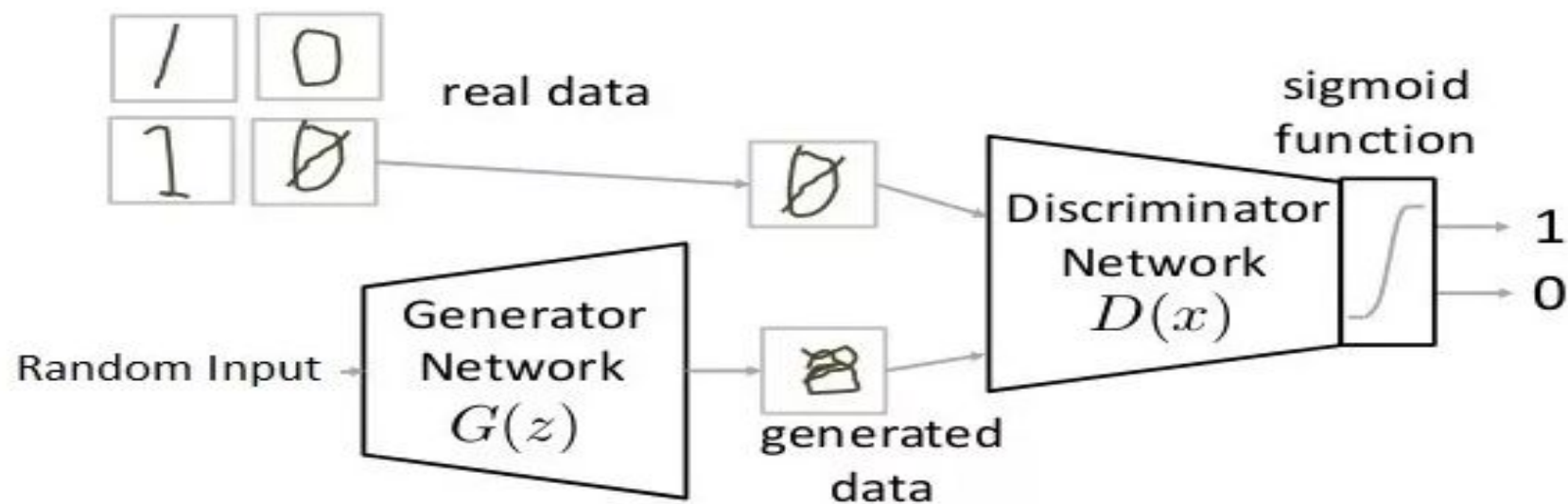
- Случайный вход + Нейросеть наоборот = Генерирующая сеть



- Хотим генерировать реалистичные изображения

Generative adversarial network

- Генерирующая сеть + Классифицирующая Нейросеть = GAN



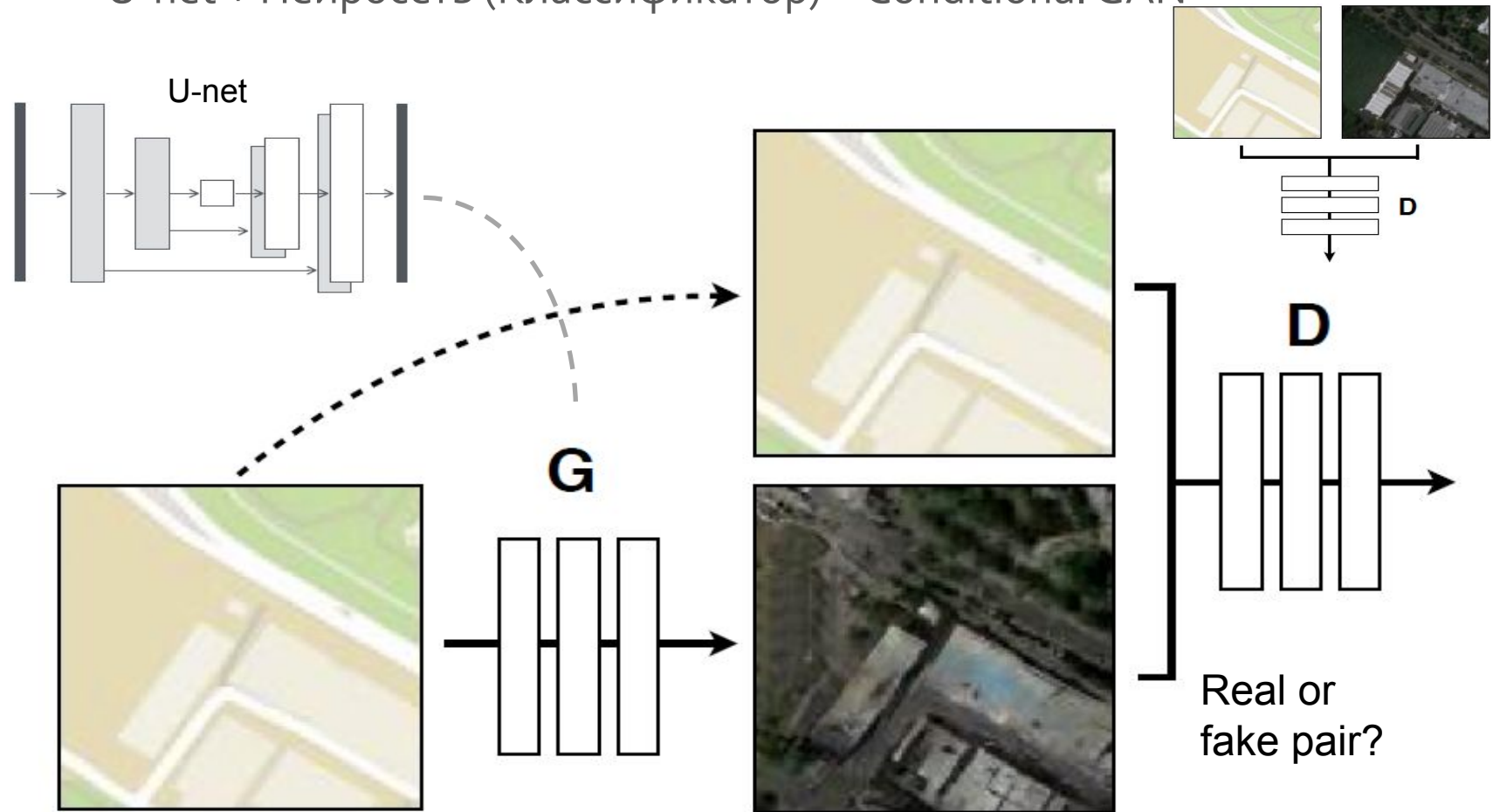
Этих людей
не
существует



- Их нет! Изображения сгенерированы сетью по случайным входам

Conditional GAN

- U-net + Нейросеть (Классификатор) = Conditional GAN



- Ещё более хитрый способ решения задачи сегментации

Pix2pix

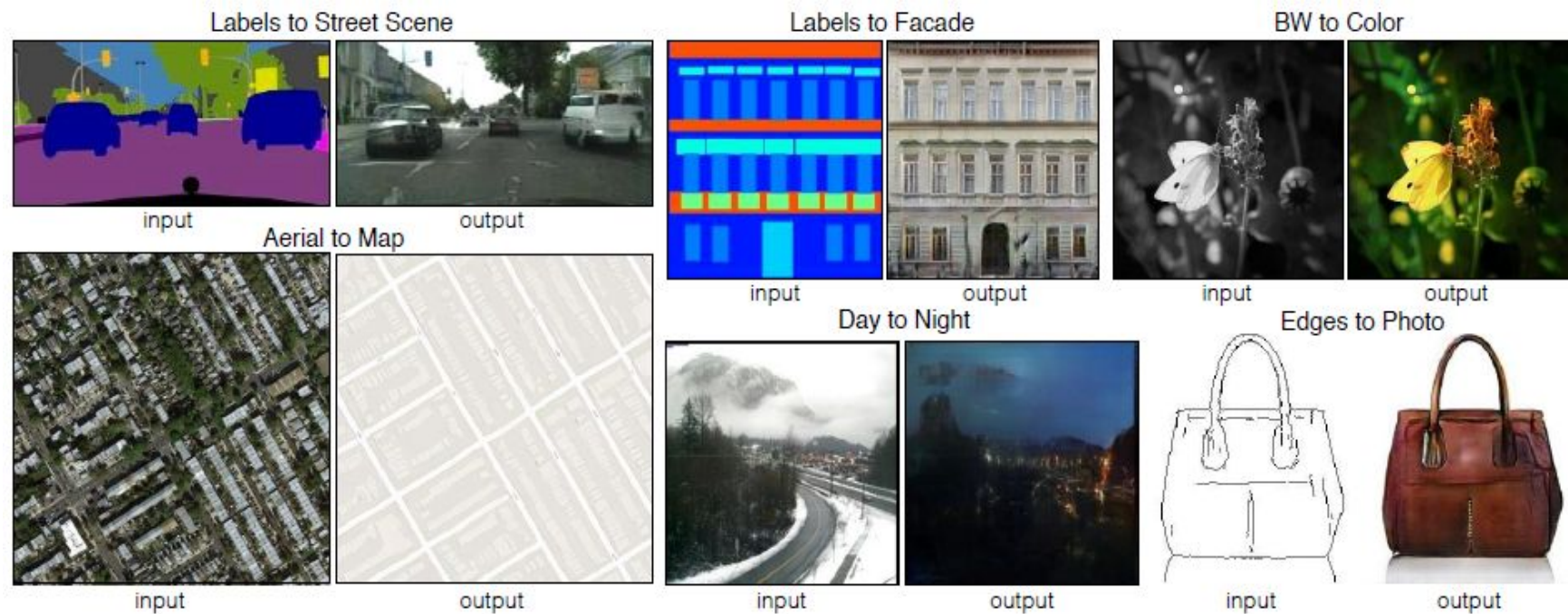
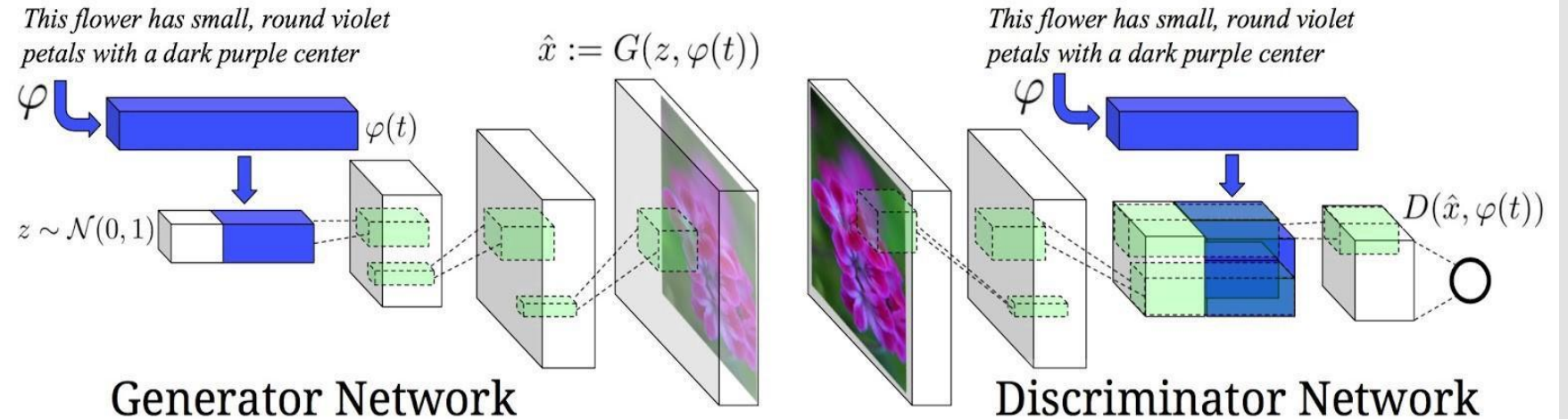


Figure 1: Many problems in image processing, graphics, and vision involve translating an input image into a corresponding output image. These problems are often treated with application-specific algorithms, even though the setting is always the same: map pixels to pixels. Conditional adversarial nets are a general-purpose solution that appears to work well on a wide variety of these problems. Here we show results of the method on several. In each case we use the same architecture and objective, and simply train on different data.

- Код в open-source

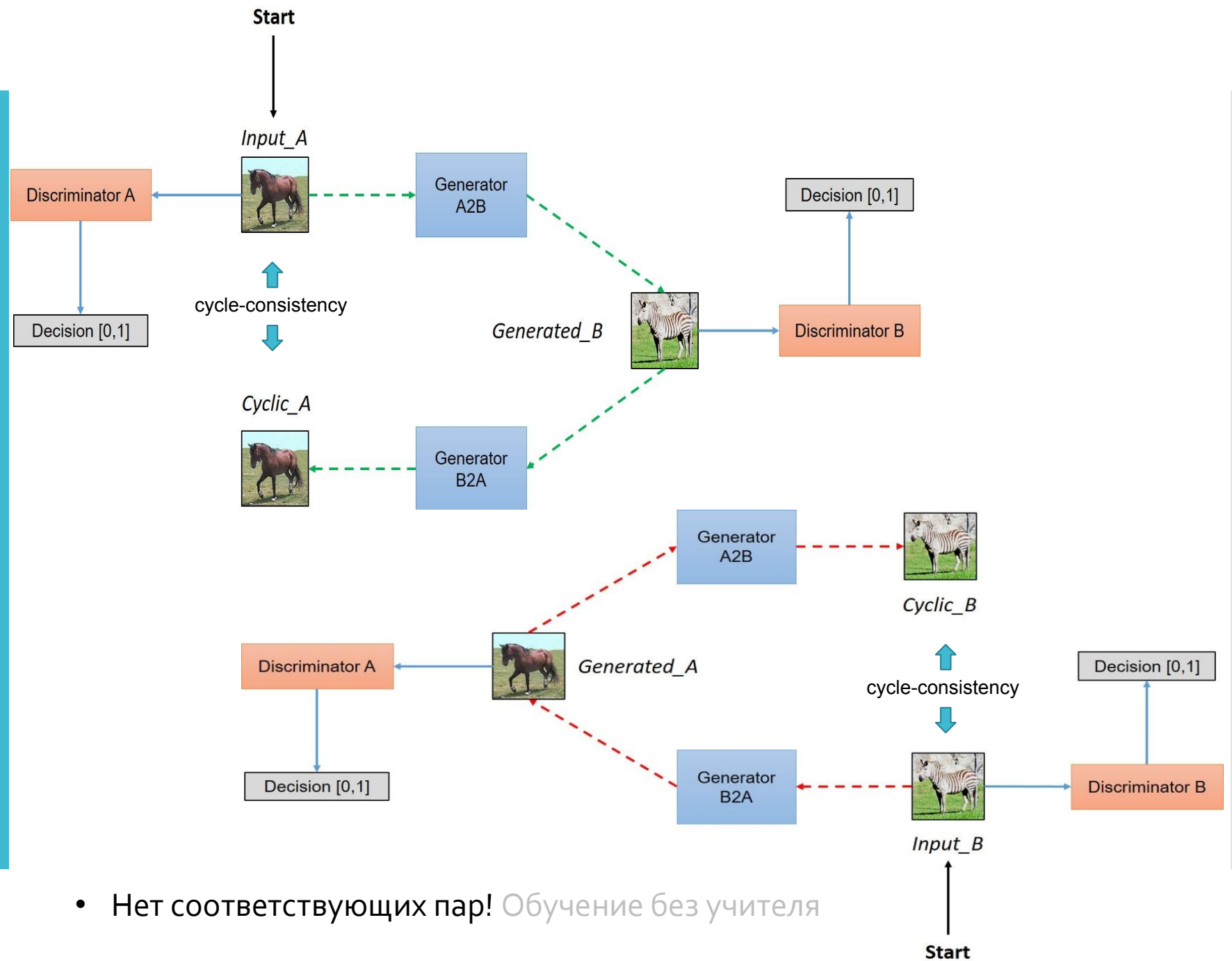
Conditional GAN

- Информация + Генератор + Дискриминатор = cGAN



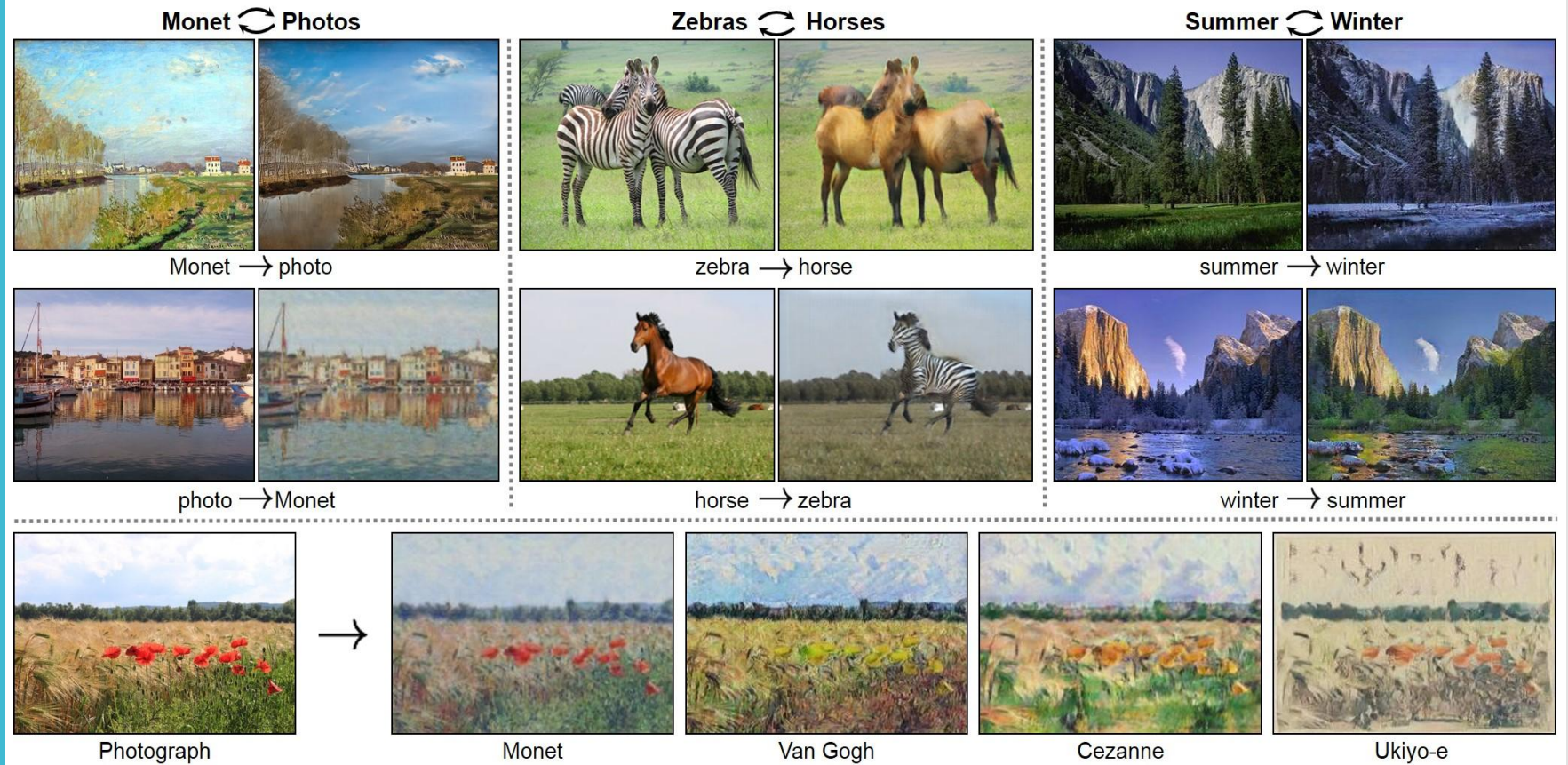
- Можно генерировать изображения по текстовому описанию

CycleGAN




- Нет соответствующих пар! Обучение без учителя

CycleGAN



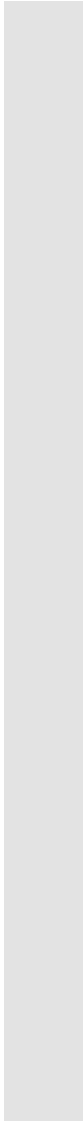
Два умных мужика и два китайца делают чудеса

- Unpaired Image-to-Image Translation using Cycle-Consistent Adversarial Networks. Zhu, Park, Isola, Efros



Паровоз ещё не
уехал!

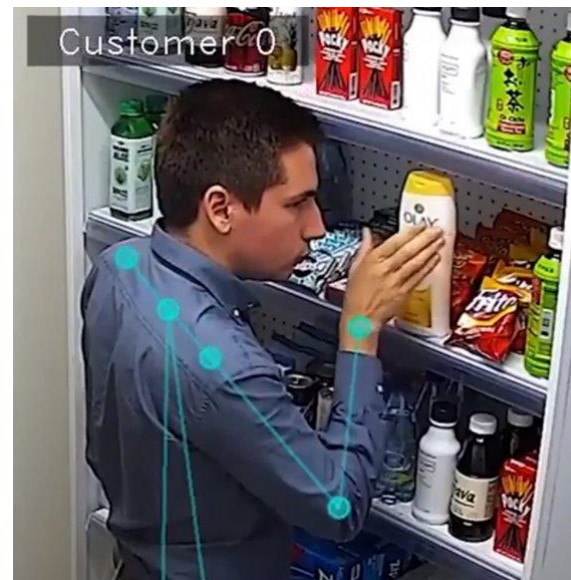
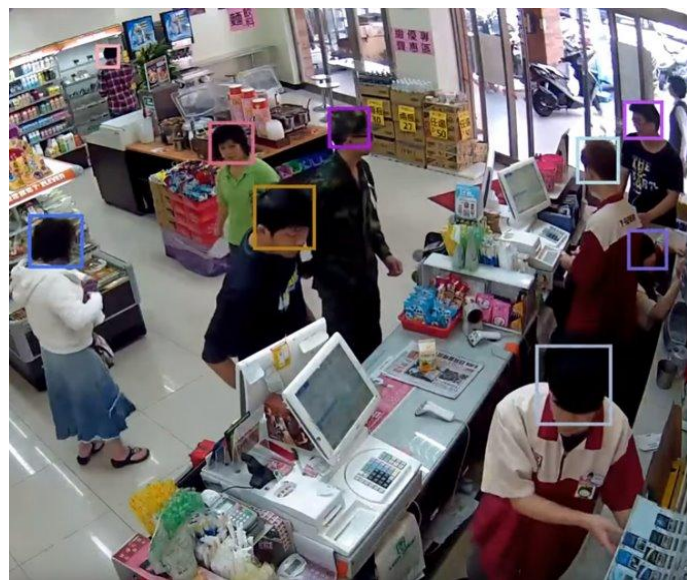
Новые архитектуры и
модели впереди!



Мои проекты

- ФГУП "ГосНИИАС" (ГНЦ РФ)
(Кафедра управляющих и информационных систем МФТИ)
- Лаборатория гибридных интеллектуальных систем МФТИ
- Лаборатория МФТИ-Сбербанк

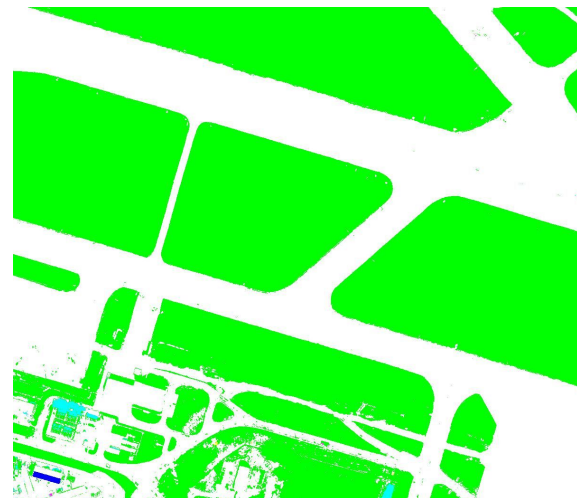
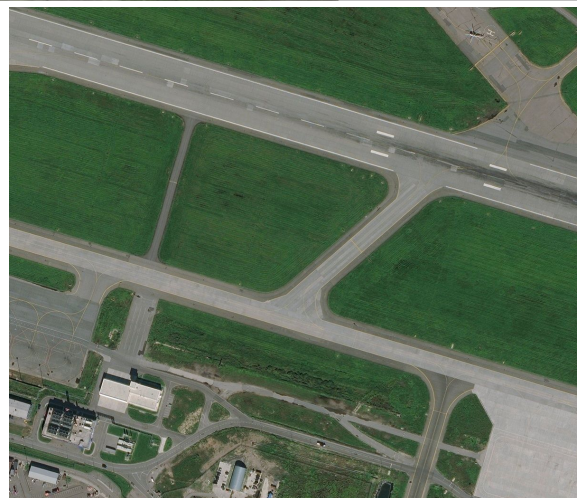
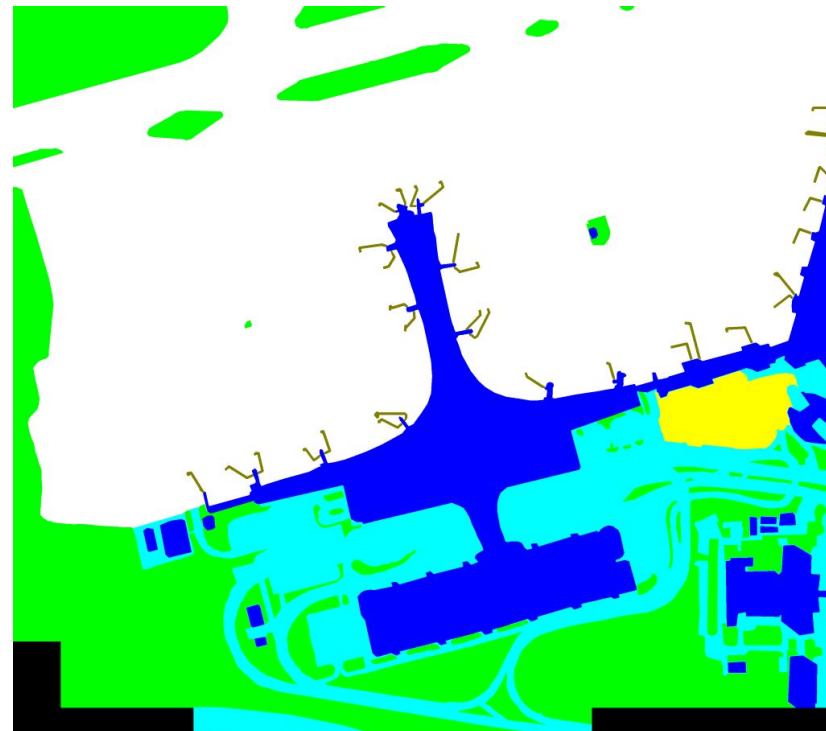
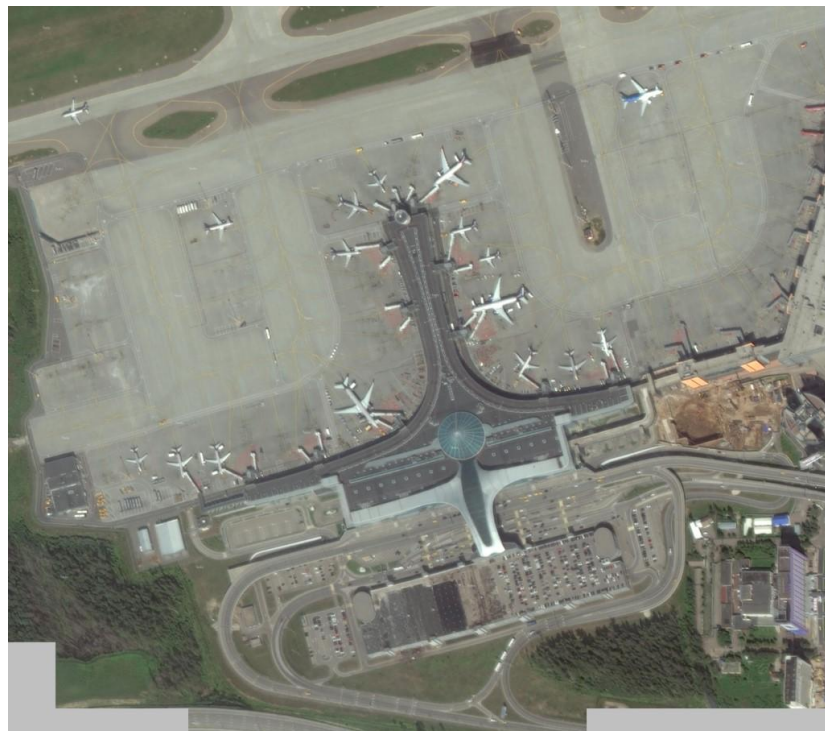
Магазин будущего



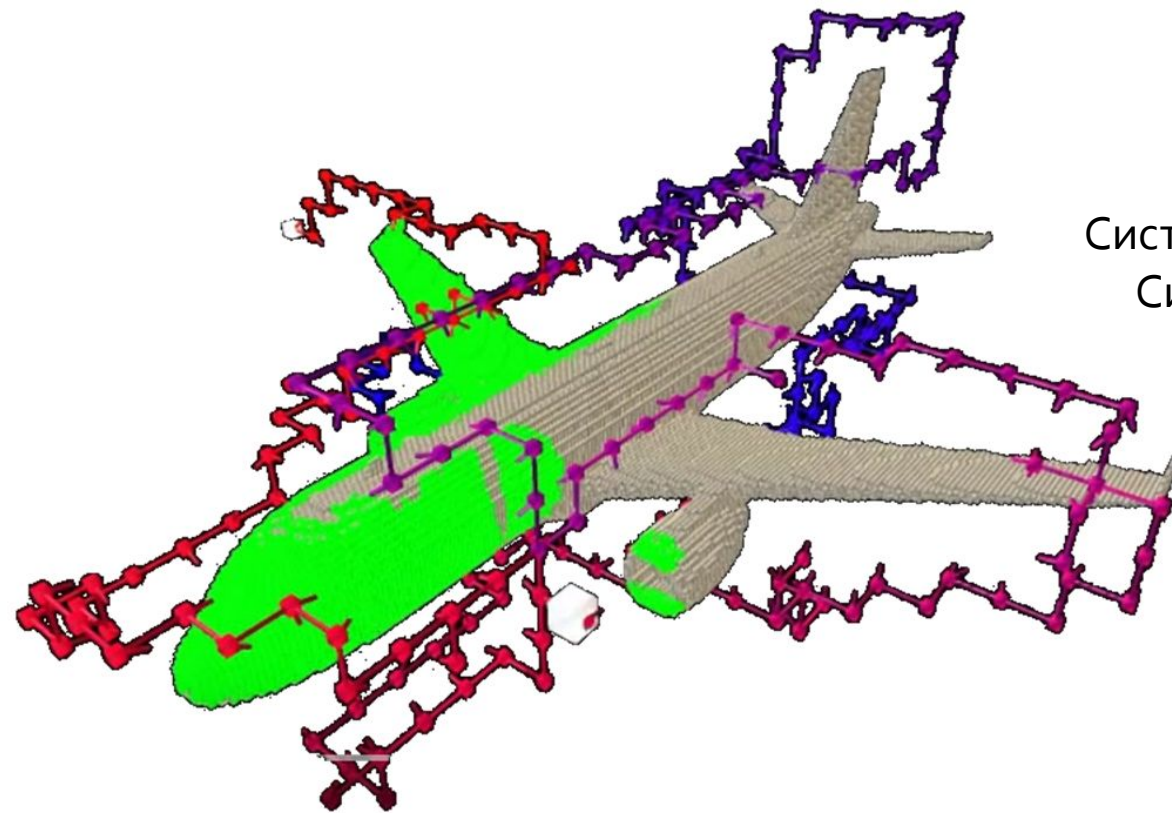
Магазин будущего

- Детектирование товаров
- Трекинг товаров
- Распознавание действий и позы покупателей
- Идентификация покупателей
- Трекинг покупателей по залу
- Составление виртуальной корзины покупок и привязка к покупателю для автоматической оплаты
- Отслеживание пустот на полках

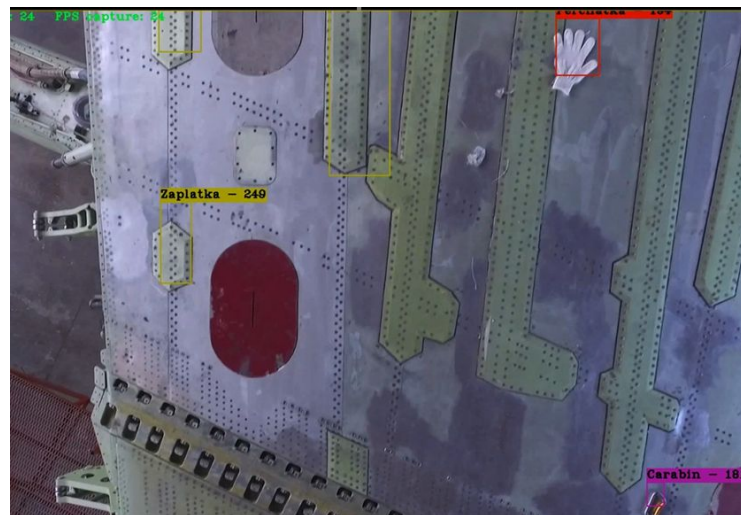
Семантическая сегментация аэро- и космоснимков



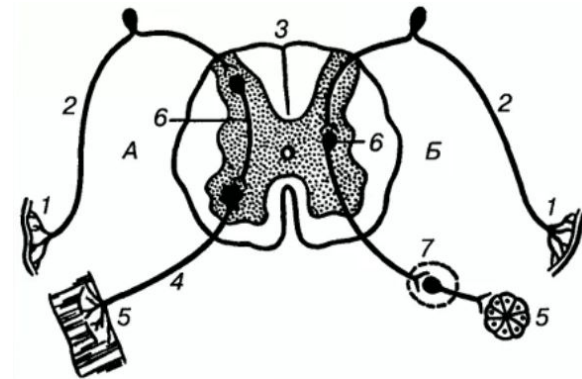
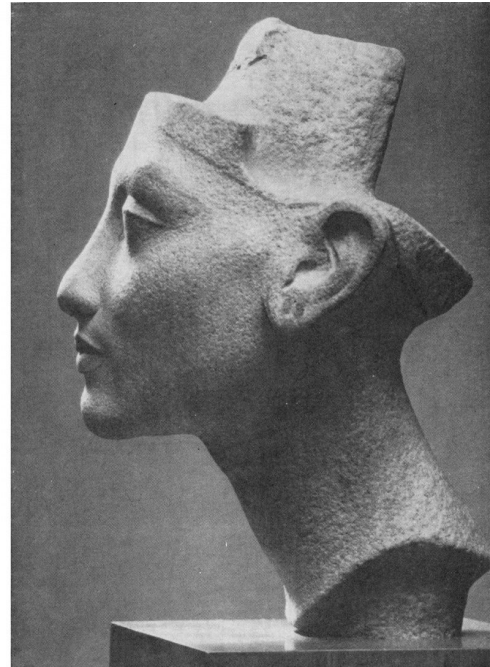
Визуальный
осмотр
воздушных
судов
с помощью
дронов и ИИ



Дрон + Детектор +
Система ориентирования +
Система управления =
Автоматический
визуальный осмотр



Новая биометрия



Случайный
Стимул



Реакция



Нейросеть



Идентификатор

- Динамические паттерны вместо статических + Случайные стимулы =
Защита от копирования биометрических шаблонов



СБЕРБАНК



Спасибо за внимание!

Горбачев Вадим Александрович, к.ф.-м.н.

79164363030@yandex.ru