

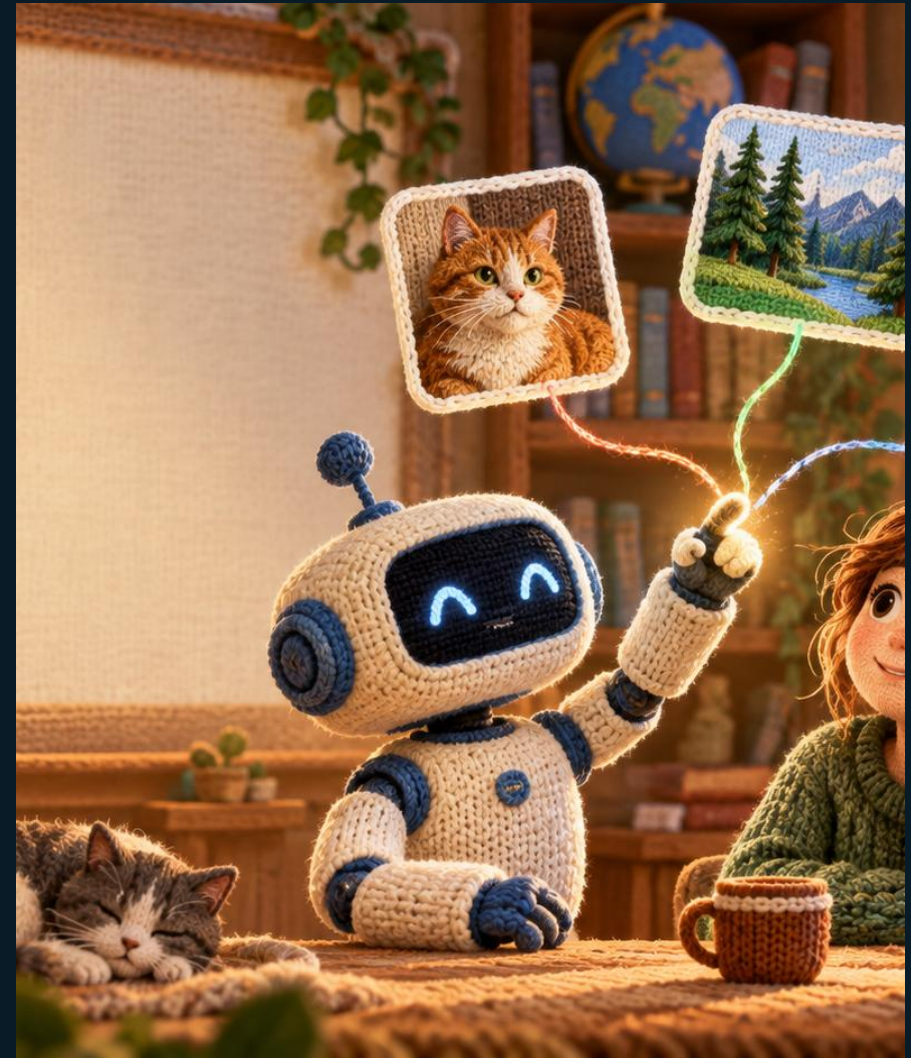
Компьютерное зрение: от пикселей к признакам

Как изображение превращается в данные, почему ручные признаки помогают, где они ломаются и как нейросеть строит embedding.

Урок 11

Сегодня снова: Стоп, проверяем!

Доктор Игорь Клейнер



Новая тема – анализ картинок

Магия нейросетей – этапы работы аналитика

картинка → числа

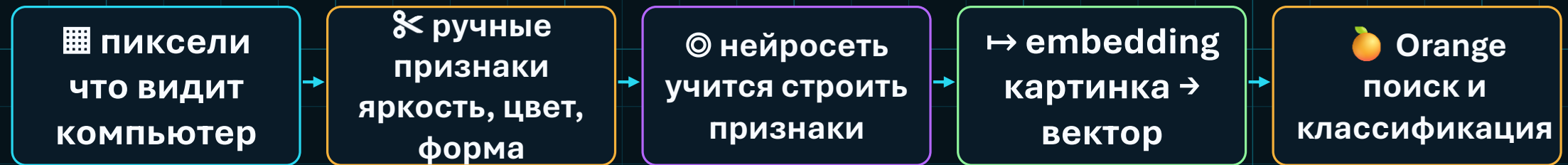
нейросеть → выученные признаки

ручные признаки → обычная таблица

embedding → снова знакомые модели
Orange

Маршрут урока

Учимся, задаем вопросы, понимаем, смотрим, шутим, используем



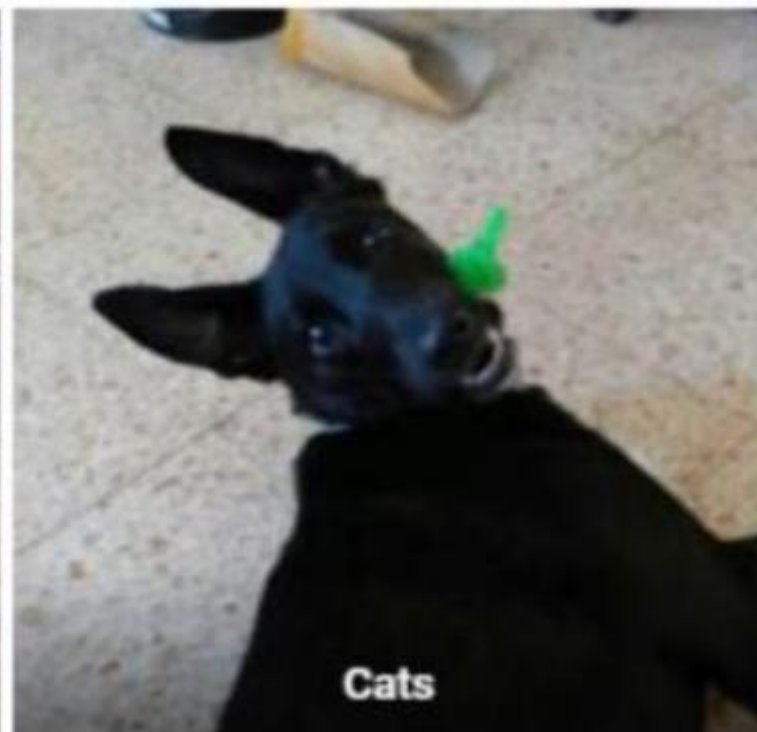
Мостик от кластеризации

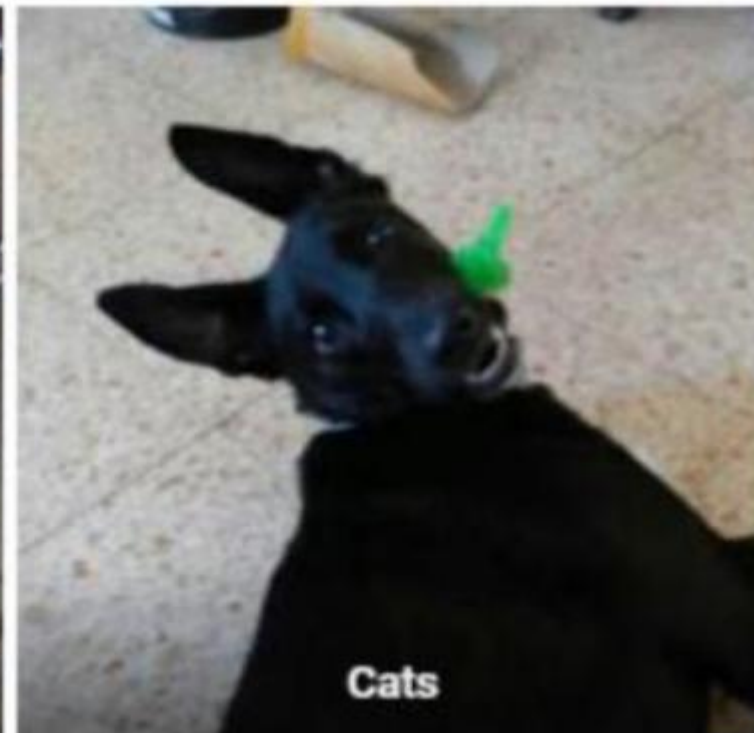
В прошлом уроке мы искали структуру без target. Теперь объектом становится фотография.

11.0

Сегодня снова: Стоп, проверяем!

Доктор Игорь Клейнер







Размер изображения:
869 × 869

Изображения других размеров не найдены.

[Похожие изображения](#) - [Пожаловаться на картинки](#)



2012

2020

Результатов: примерно 2 (1,48 сек.)



Размер изображения:
861 x 870

Изображения других размеров не найдены.



Что такое изображение



Прошлый урок: зависимой переменное не было

кластеризация искала похожие
объекты

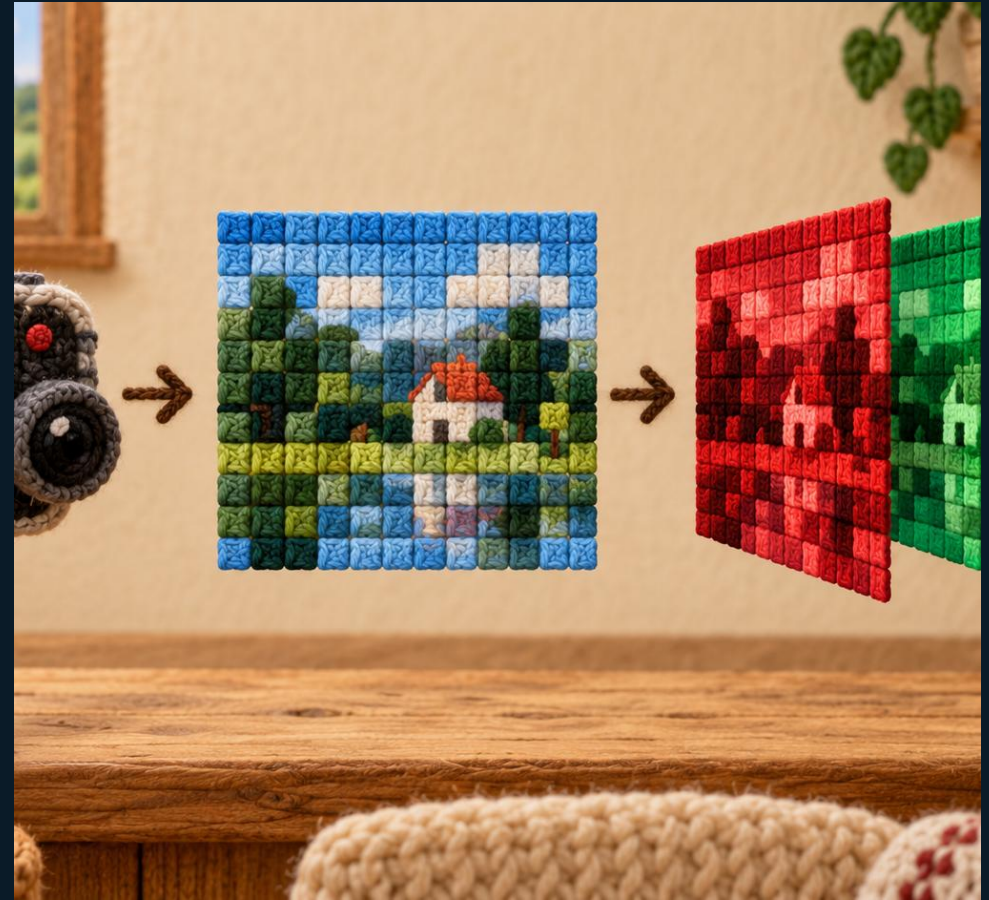
похожесть задавалась признаками и
расстоянием

результат был гипотезой, а не истиной

Сегодня объект сложнее

Не строка таблицы, а фотография

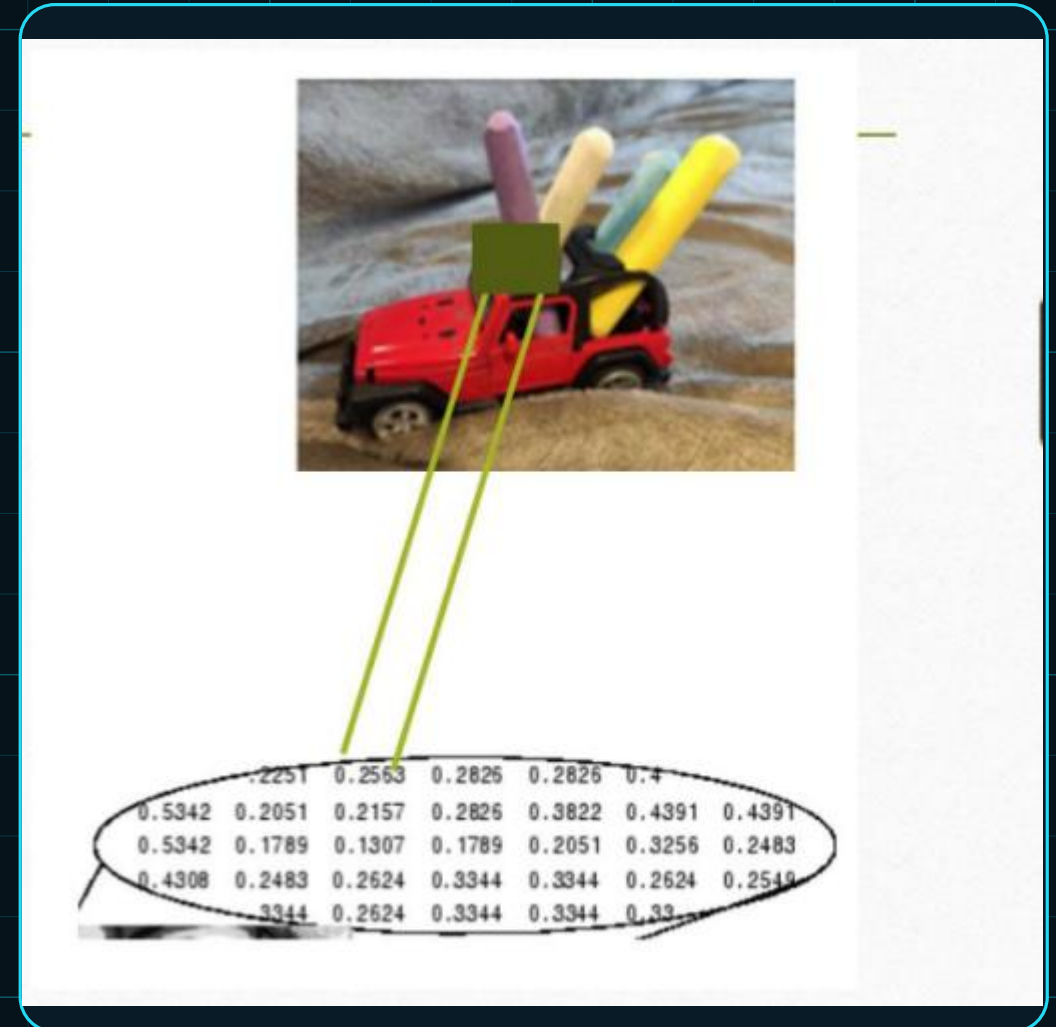
- у данных квартир были столбцы: площадь, качество, район
- у текстовых данных были слова и частоты
- а у изображения есть только пиксели



Сегодня объект сложнее

Не строка таблицы, а фотография

- у данных квартир были столбцы: площадь, качество, район
- у текстовых данных были слова и частоты
- а у изображения есть только пиксели



Вопрос

11.0 Мостик

?

Если у нас фотография кота, что будет строкой таблицы описывающей его?

Варианты: файл, пиксели, ручные признаки или embedding.

Ответ

11.0 Мостик



Строкой может стать всё изображение, но признаки нужно построить.

- самый сырой вариант — пиксели
- ручной вариант — яркость, цвет, форма
- современный вариант — embedding (вложение) от нейросети

Summary

11.0 Мостик

в машинном обучении объект может
стать строкой таблицы

качество результата работы модели
зависит от признаков

для изображений **главное** испытание — построить хорошее
представление

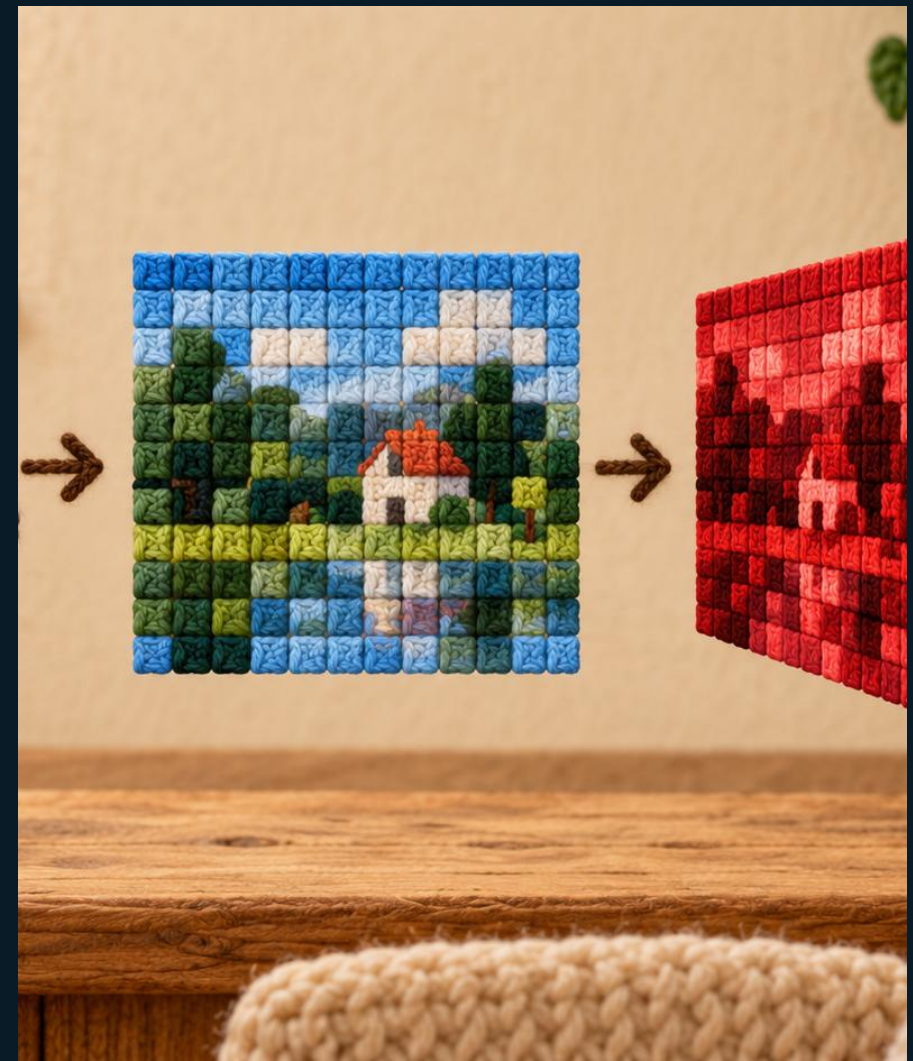
Что такое изображение для компьютера

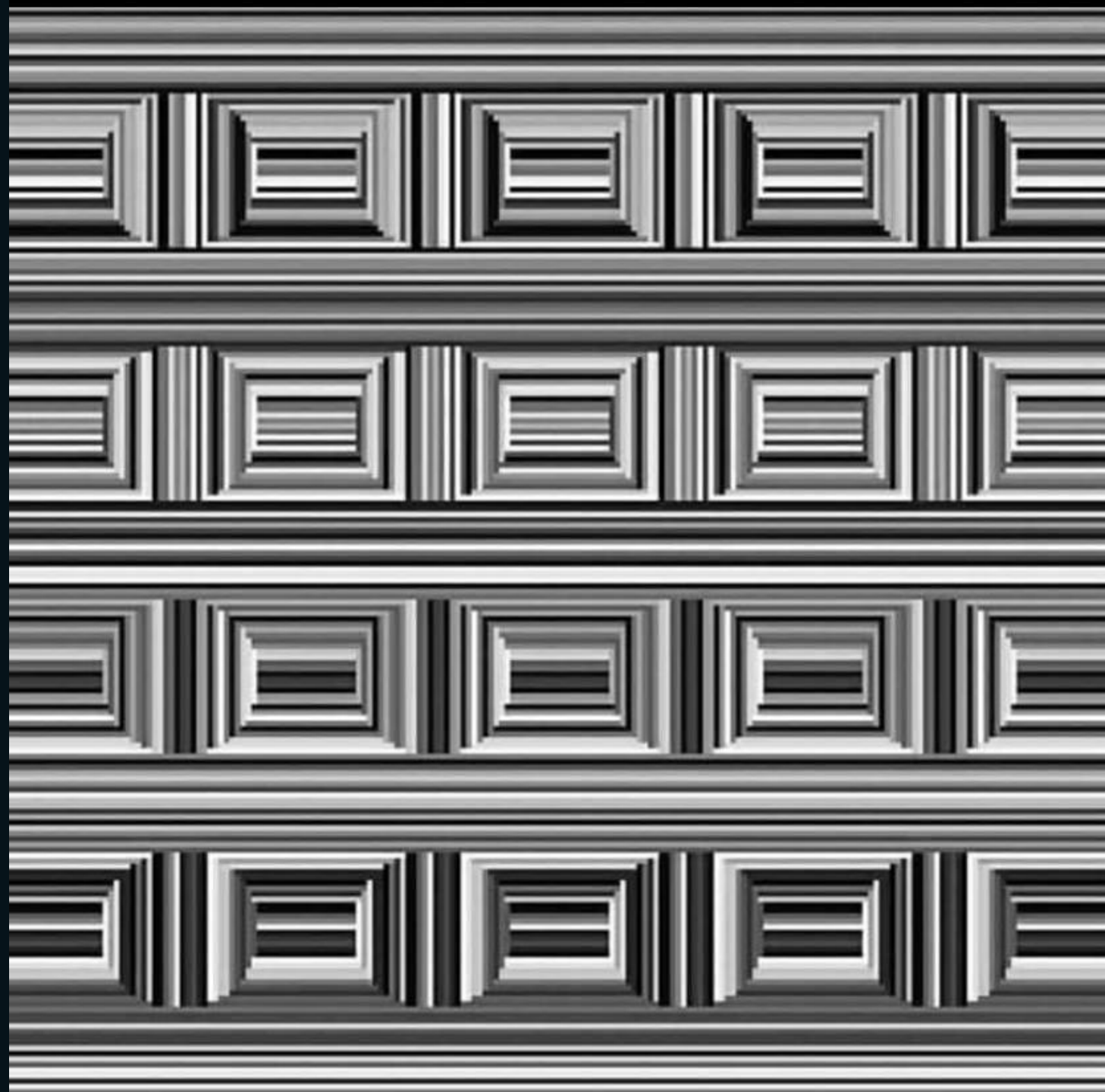
Человек видит смысл. Компьютер получает
числовую структуру.

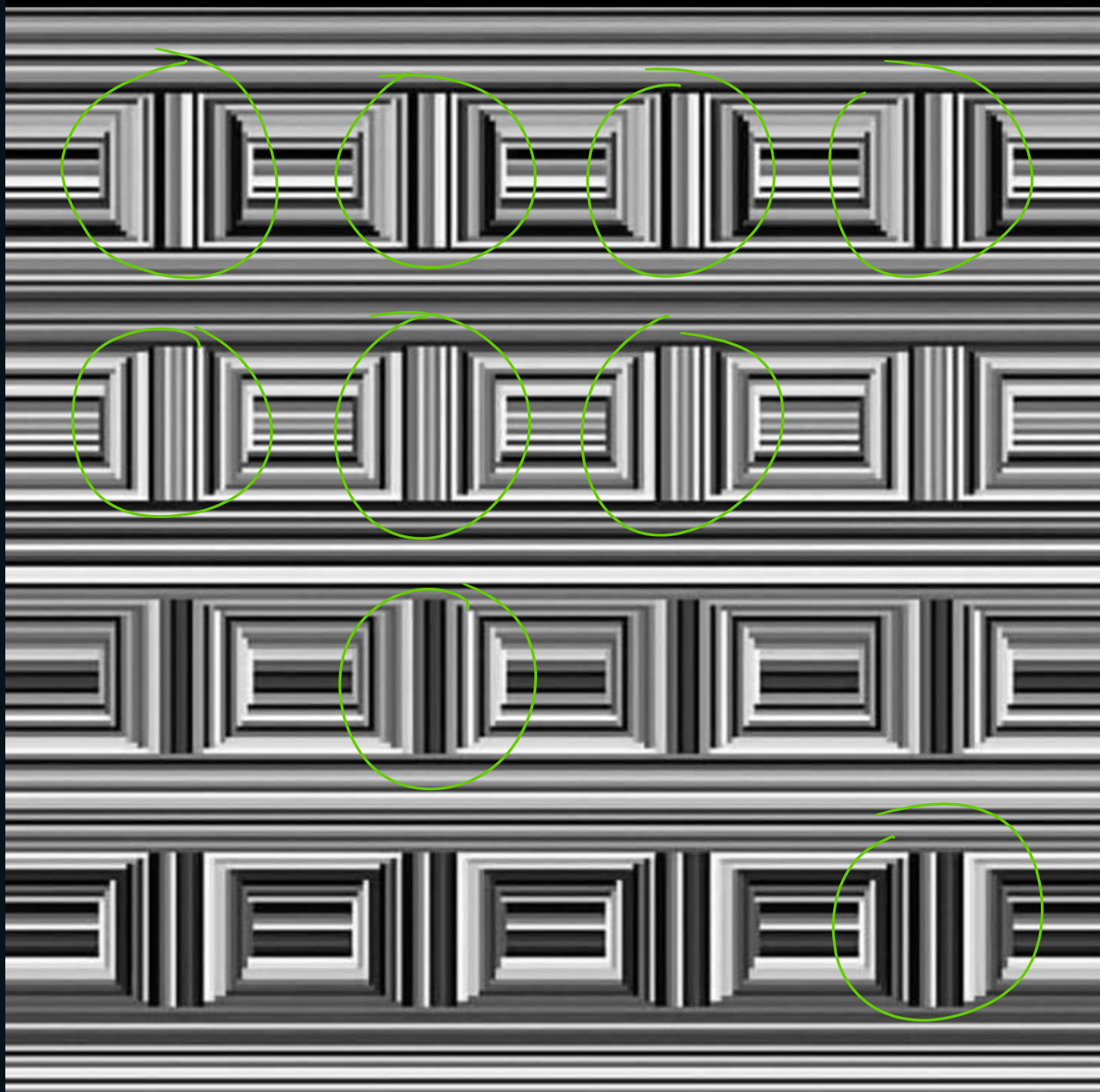
11.1

Сегодня снова: Стоп, проверяем!

Доктор Игорь Клейнер











Мы видим сцену целиком

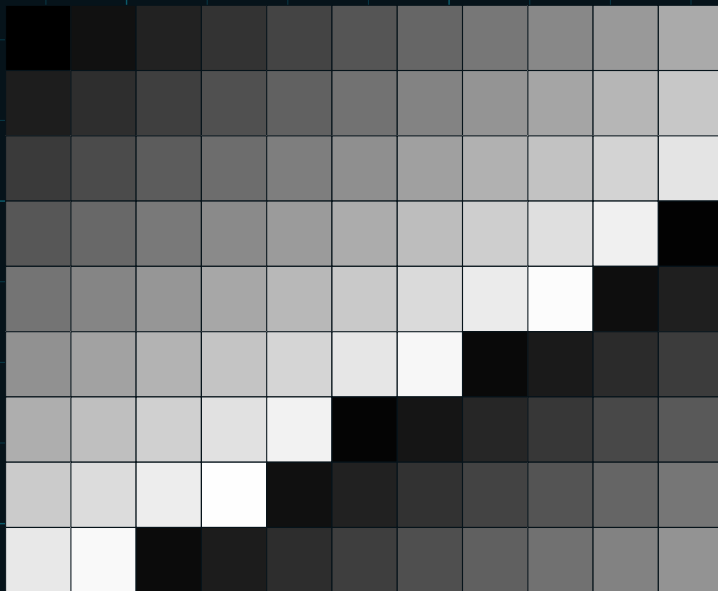
кот, машина, лес, еда — мозг
мгновенно достраивает смысл

мы используем опыт, контекст и
ожидания

мы почти не замечаем пиксели

Компьютер начинает с чисел

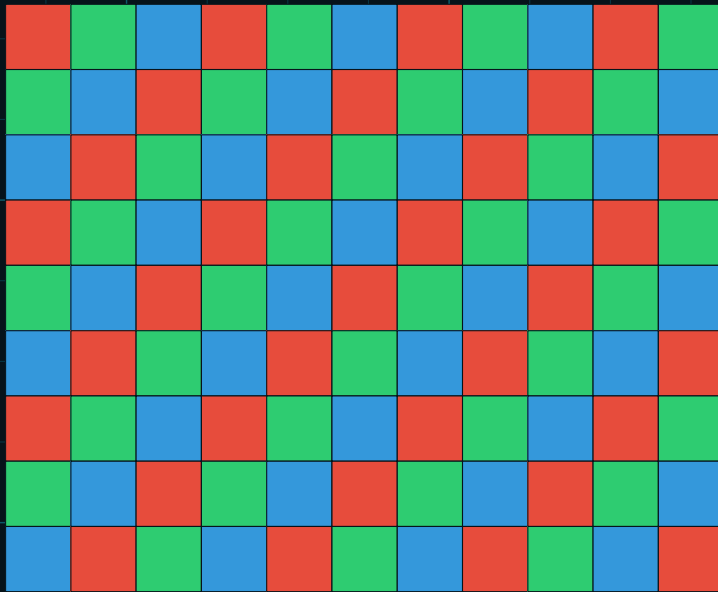
Серое изображение = одна матрица яркости



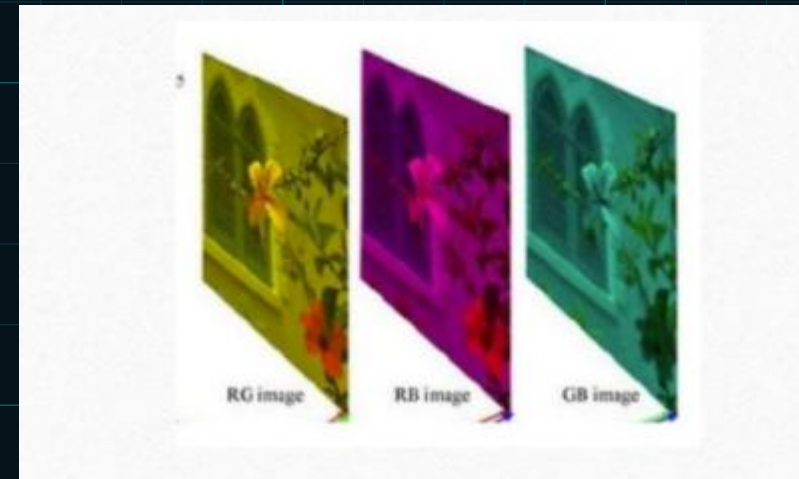
0, 37, 129, 255, ...
яркость = число в ячейке

Цветная картинка = три канала

RGB: красный, зелёный, синий



$R[x,y]$, $G[x,y]$, $B[x,y]$
цвет = три слоя чисел



ТЕХНОЛОГИИ
ДЕЛАЮТ
МИР ЯРЧЕ
♥

Что компьютер видит в картинке?

Не кошку, а числа

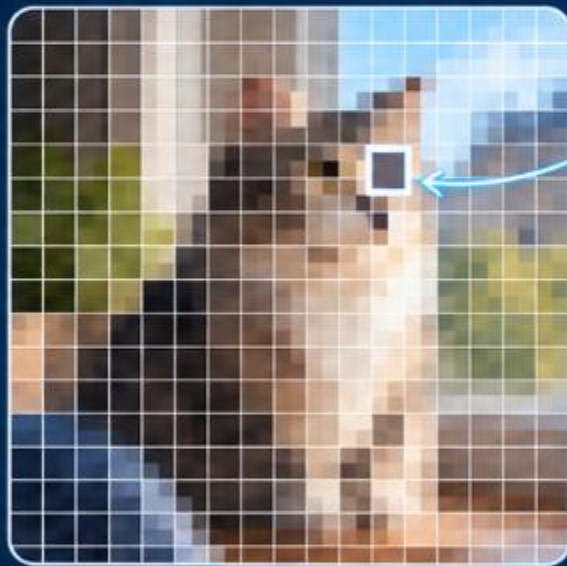
ДАЖЕ
КАРТИНКИ —
ЭТО ДАННЫЕ
♥

1 Картинка



Для нас это кошка ♥

2 Пиксели



Каждый
пиксель
имеет цвет

Картинка разбивается
на маленькие квадратики —
пиксели

3 Числа (RGB)

R					G					B				
120	118	130	125	112	200	195	180	170	160	230	220	210	200	190
122	135	140	128	110					188					185
130	145	142	135	120					175					170
128	140	138	130	118					160					160
115	125	123	120	110					150					155

У каждого пикселя есть три числа:
сколько **красного (R)**, **зелёного (G)**
и **синего (B)** света

RGB —
это цвет
в числах!
♥



Изображение превращается
в **таблицу чисел**

СНАЧАЛА
ЧИСЛА,
ПОТОМ
ВОЛШЕБСТВО
♥

АЛГОРИТМЫ

ГРАФИКА

БОЛЬШИЕ ВОЗМОЖНОСТИ

Видео — это не отдельная магия

кадр = изображение

фильм = последовательность кадров

добавляется время: движение, скорость,
порядок

Что можно делать с изображением?

Четыре разных вопроса — четыре разные задачи

🔍 **Classification**
что на картинке?

📍 **Detection**
где объект?

📐 **Segmentation**

≈ **Similarity**
что похоже?



Что можно делать с изображением?

Четыре разных вопроса — четыре разные задачи

🗂 **Classification**
что на картинке?

📏 **Detection**
где объект?

📐 **Segmentation**

≈ **Similarity**
что похоже?



КОМПЬЮТЕР
ВИДИТ
МИР

Что можно узнать из изображения?

Классификация, детекция, сегментация — одна картинка, разные задачи

Те же объекты —
больше
возможностей! ♥

1 Классификация

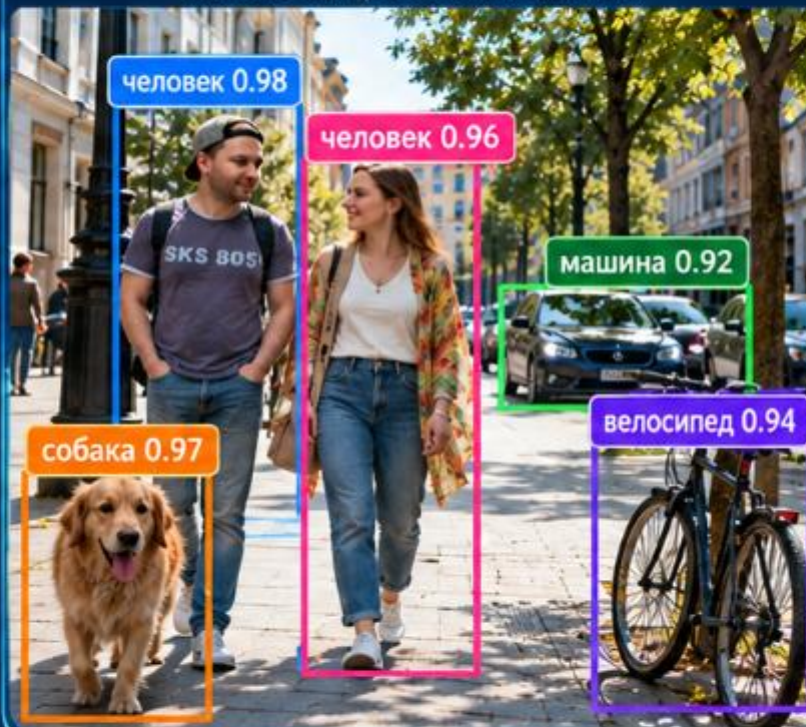
Определяет один общий класс
для всего изображения



 **городская сцена**
"улица", "люди", "собака"

2 Детекция

Находит объекты и определяет
их классы (рамки и подписи)



3 Сегментация

Выделяет каждый объект на уровне
пикселей (цветные маски)



Алгоритмы
замечают
важное ♥



Одна картинка —
много вопросов,
и на каждый есть ответ!

ДАННЫЕ
МОДЕЛИ
ЗАДАЧИ
РЕАЛЬНЫЙ МИР

БОЛЬШЕ
ЧЕМ
КАРТИНКИ
♥

Почему изображения сложны

один объект — много поз

один класс — много представителей

объект может быть частично скрыт

фон и освещение постоянно мешают

Один объект — много поз

Кот сидит, лежит, прыгает, прячется

человек узнаёт кота по смыслу

модели нужна устойчивость к позе и
масштабу

пиксели меняются почти полностью



МАШИННОЕ
ОБУЧЕНИЕ



Почему с изображениями сложно?

Один и тот же объект может выглядеть очень по-разному

Реальный
мир —
неидеален



другая поза



частично скрыт



темно



слишком ярко



другой масштаб



шум / размытие

Один
объект —
много
вариантов

Реальные
данные —
разные



Алгоритму нужно распознать объект
несмотря на все помехи

ДАННЫЕ

МОДЕЛИ

РЕАЛЬНЫЙ МИР

БОЛЬШИЕ
ЗАДАЧИ
С КАРТИНКАМИ

Один класс — много представителей

Все стулья — стулья, но выглядят очень по-разному

форма разная

материал разный

угол съёмки разный

часть объекта может быть закрыта



Один класс — много представителей

Все стулья — стулья, но выглядят очень по-разному

форма разная

материал разный

угол съёмки разный

часть объекта может быть закрыта



Освещение и шум

ночь и день меняют яркость

камера добавляет шум

контраст и цвет зависят от устройства

тень может выглядеть как объект



Освещение и шум

ночь и день меняют яркость

контраст и цвет зависят от устройства

камера добавляет шум

тень может выглядеть как объект



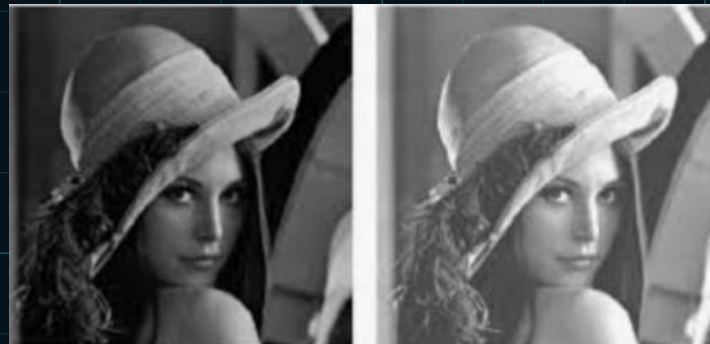
Освещение и шум

ночь и день меняют яркость

камера добавляет шум

контраст и цвет зависят от устройства

тень может выглядеть как объект



Лена (изображение)

Материал из Википедии — свободной энциклопедии

[\[править \]](#) [\[править ко](#)

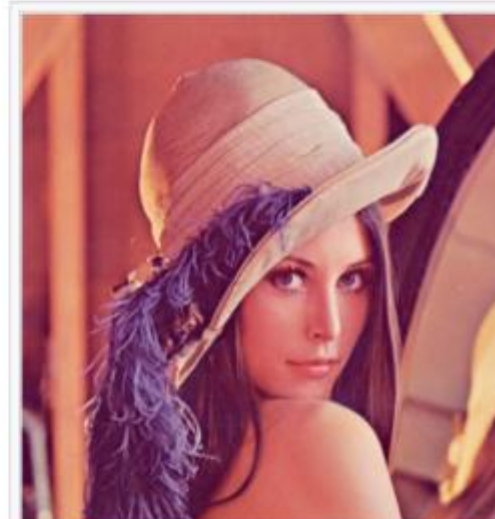
Текущая версия страницы пока [не проверялась](#) опытными участниками и может значительно отличаться от [версии](#), проверенной 20 апреля 2019; проверки требуют **3 правки**.

У этого термина существуют и другие значения, см. [Лена \(значения\)](#).

«**Ле́на**» (англ. Lenna, нередко также *Lena*; швед. Lena) — название [стандартного тестового изображения](#), широко используемого в научных работах для проверки и иллюстрации [алгоритмов обработки изображений](#) (сжатия, шумоподавления и т. д.). Тестовое изображение представляет собой оцифрованный портрет шведской модели [Лены Сёдерберг](#), который в свою очередь является фрагментом разворота из журнала *Playboy* за ноябрь 1972 года^[1]. Написание *Lenna* предложила сама Лена — по-английски её имя читалось бы «Лина».

В 1973 году Александру Савчуку из [Университета Южной Калифорнии](#) для иллюстрации статьи на тему обработки изображений понадобился фотопортрет с хорошим динамическим диапазоном. Савчук отсканировал фрагмент постера из Playboy. Он использовал сканер с разрешением 100 линий на дюйм, и результирующее изображение получилось 512 на 512 точек. Вскоре эта картинка превратилась для индустрии в [стандарт де-факто](#): на нём проверялись и отрабатывались всевозможные приёмы коррекции изображений, оттачивались новые алгоритмы обработки^[1].

В 1997 году Лена Сёдерберг посетила юбилейную конференцию [Society for Imaging Science and Technology](#)^[en] в [Кембридже](#)^[2].



Лена Сёдерберг

<https://en.wikipedia.org/wiki/Lenna>

Совет от учителя

Не путайте пиксели и смысл

Для человека картинка уже наполнена смыслом. Для алгоритма смысл надо построить из чисел.

Вопрос

11.1 Изображение

?

**Почему нельзя просто дать модели все пиксели и
успокоиться?**

Есть технический и смысловой ответ.

Ответ

11.1 Изображение



Можно дать пиксели, но это редко самый удобный подход

- пикселей очень много
- малый сдвиг меняет тысячи чисел
- нужны признаки, устойчивые к позе, свету и фону

Summary

11.1 Изображение

изображение хранится как массив
чисел

задачи vision бывают разными

для цвета нужны три канала

главная трудность — построить
признаки, близкие к смыслу

Summary



Ручные признаки: иногда это очень сильно

Сначала мы сами придумываем признаки. Это не примитивно — это аналитика.

11.2

Сегодня снова: Стоп, проверяем!

Доктор Игорь Клейнер



Feature engineering по-человечески

Хороший ручной признак может выиграть у большой модели на простой задаче.

Проблема не в ручных признаках, а в том, что сложная сцена требует слишком много признаков.

Вопрос

11.2 Ручные признаки

?

Упражнение 1: как отличить день от ночи одним числом?

Подсказка: ищем одно число, которое обычно выше днём и ниже ночью.

Ответ

11.2 Ручные признаки

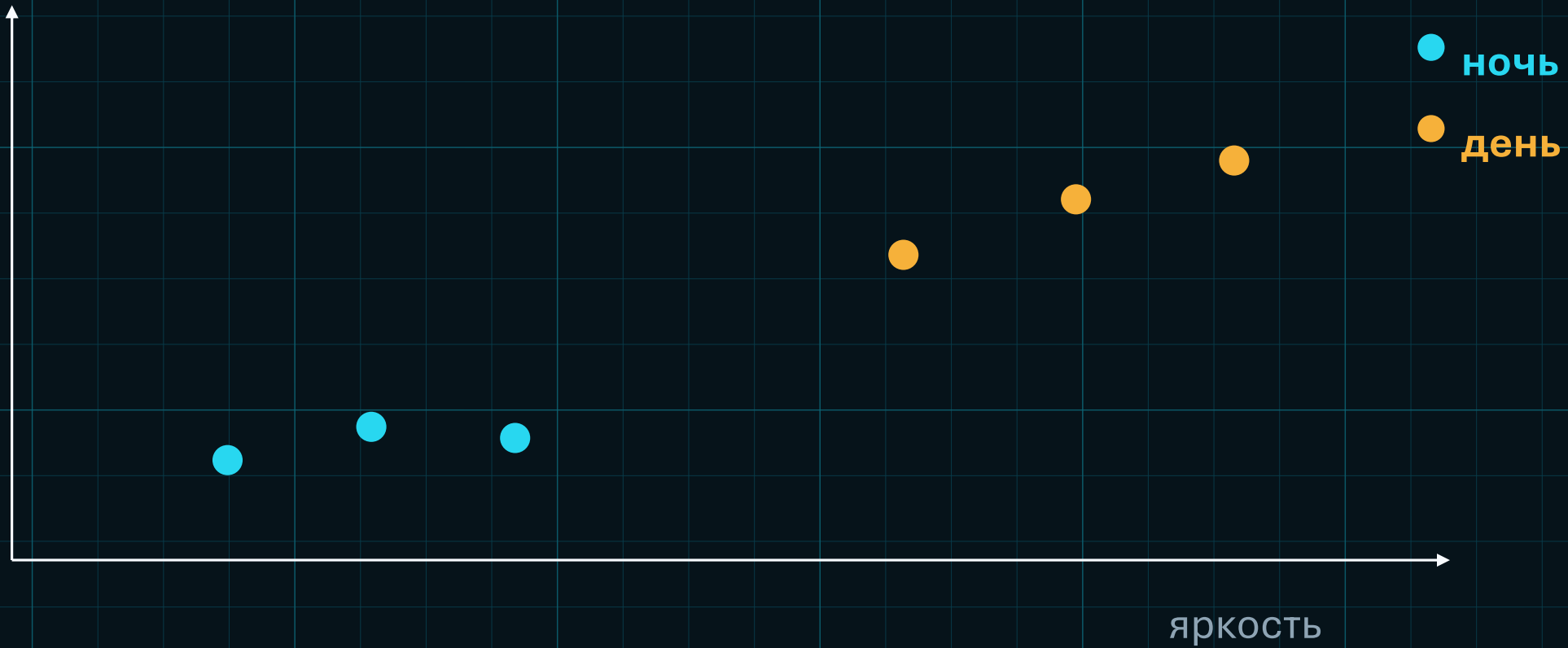


Средняя яркость часто даёт сильный первый признак.

- берём среднюю яркость изображения
- строим порог: ниже — ночь, выше — день
- это простой, понятный и проверяемый baseline

День / ночь на одной оси

Один признак — одна граница



АНАЛИЗ
ДАННЫХ



Задача: день или ночь?

Простой ручной признак: средняя яркость

Маленькие
признаки —
большие
возможности ♥



НОЧЬ



ПОРОГ



ДЕНЬ



0

яркость

255

Более тёмные изображения
чаще относятся к ночи

Более яркие изображения
чаще относятся к дню



Один удачный признак
иногда решает задачу

Данные
делают
мир яснее ♥

ДАННЫЕ

МОДЕЛИ

БОЛЬШИЕ ИДЕИ

ЛУЧШЕ
ДАННЫЕ
ЯРЧЕ
ЗАВТРА



Совет

Сначала ищем простой признак

Перед сложной моделью проверьте, не решает ли задачу одна понятная величина: яркость, доля цвета, размер, число объектов.

Ловушка

Яркость может обмануть

Ночная улица с фонарями и дневная фотография в тени могут оказаться рядом. Признак полезный, но не святой.

Вопрос

11.2 Ручные признаки

?

Упражнение 2: как найти солнечный лес?

Нужно не одно число, а минимум две идеи.

Ответ

11.2 Ручные признаки



Берём два признака: яркость и долю зелёного.

- яркость помогает найти солнечные снимки
- доля зелёного помогает найти лес
- вместе они дают двумерное пространство

Солнечный лес: уже две оси





Задача: солнечный лес

Два простых признака: яркость и доля зелёного

Данные превращают
идеи в ответы



Простые
признаки —
большие
результаты

количество
зелёного
(доля зелёного
в изображении)

БОЛЬШЕ
ПРИЗНАКОВ
— БОЛЬШЕ
ВОЗМОЖНОСТЕЙ



Ручной поиск признаков может помочь,
но нужно догадаться, какие признаки важны

МАЛЕНЬКИЕ
ПРИЗНАКИ
ВЕДУТ
К БОЛЬШИМ
ОТКРЫТИЯМ

Упражнение 3: много неба

доля синего цвета — первый намёк

небо чаще сверху, море может быть
снизу

но важно, где расположен синий цвет

Ответ

11.2 Ручные признаки



Для неба нужен признак положения.

- доля синего в верхней части кадра
- граница между небом и землёй
- цвет + геометрия лучше, чем один цвет

Упражнение 4: красный круглый объект

красный — цветовой признак

круглый — признак формы

а если объект частично закрыт?

а если красный фон?

Ответ

11.2 Ручные признаки



Цвета мало: нужна форма и контекст.

- ручной признак может быть сильным
- но нужно решать, что считать кругом
- и как отделять объект от фона

Вопрос

11.2 Ручные признаки

?

Упражнение 5: человек в красной куртке рядом с собакой — какие признаки нужны?

Это уже не одна красивая ось.

Ответ

11.2 Ручные признаки



Нужен набор признаков и отношений.

- есть человек
- есть одежда и её цвет
- есть собака
- объекты рядом, а не просто на одной картинке
- появляется геометрия сцены

Упражнение 6: удивлённый человек держит подарок

Какие признаки пришлось бы придумать вручную?

- эмоция лица: человек удивлён?
- предмет в руках: где он и что это?
- похож ли предмет на подарок?
- рядом ли нужные объекты друг с другом?
- есть ли контекст праздника?

Вывод: ручные признаки помогают, но сложная сцена быстро превращает их в головоломку.

Совет от учителя

Ручной признак — не старьё

Если признак понятный, дешёвый и работает, не стыдно его использовать. Стыдно не проверить его честно.

Ловушка

Не влюбляйтесь в придуманный признак

**Чем красивее ваша идея, тем сильнее желание не замечать
примеры, где она ломается.**

Вопрос

11.2 Ручные признаки

?

Так в чём настоящая проблема ручного feature engineering?

Подсказка: ручные признаки полезны, но плохо масштабируются.

Ответ

11.2 Ручные признаки



Он полезен, но плохо масштабируется.

- для простой задачи достаточно одного признака
- для сложной сцены нужны десятки правил
- каждая новая задача требует новой ручной работы

Summary

11.2 Ручные признаки

ручные признаки могут быть очень сильными

они превращают картинки в обычную таблицу

каждый признак — гипотеза аналитика

сложные сцены требуют слишком много ручных признаков

отсюда естественный переход к нейросетям

Проверяем не запоминание слов, а понимание маршрута.

АНАЛИЗ
ДАННЫХ



Старая и новая парадигма в анализе изображений

Технологии
меняются,
любопытство
остаётся

Раньше

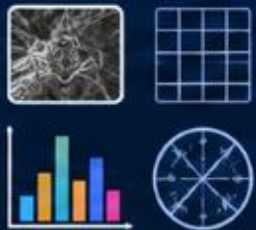
Признаки придумывает человек

Данные



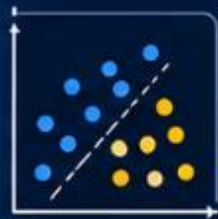
Изображения

Ручной поиск
признаков



Края, текстуры,
цвета и др.

Алгоритм



Классические
модели
(SVM, деревья и др.)

Результат



Кошка

94%

Сейчас

Признаки учится сама сеть

Данные



Изображения

Глубокая сеть



Сеть **сама учится**
полезным признакам
из данных

Результат



Кошка

94%



Главный сдвиг:
признаки всё чаще учатся
автоматически

Больше,
чем просто
картинки

ДАННЫЕ

МОДЕЛИ

БОЛЬШИЕ ИДЕИ

ЛУЧШИЕ
ДАННЫЕ
ДЛЯ
ЛУЧШЕГО
ЗАВТРА

Нейронная сеть на пальцах

Не магия. Много простых вычислений, которые учатся по ошибке.

11.3

Сегодня снова: Стоп, проверяем!

Доктор Игорь Клейнер

Главная идея

Нейросеть сама учится строить полезное представление данных.

Не вручную “яркость + зелёный + форма”, а параметры, найденные по примерам.

Один нейрон: взвешенное голосование

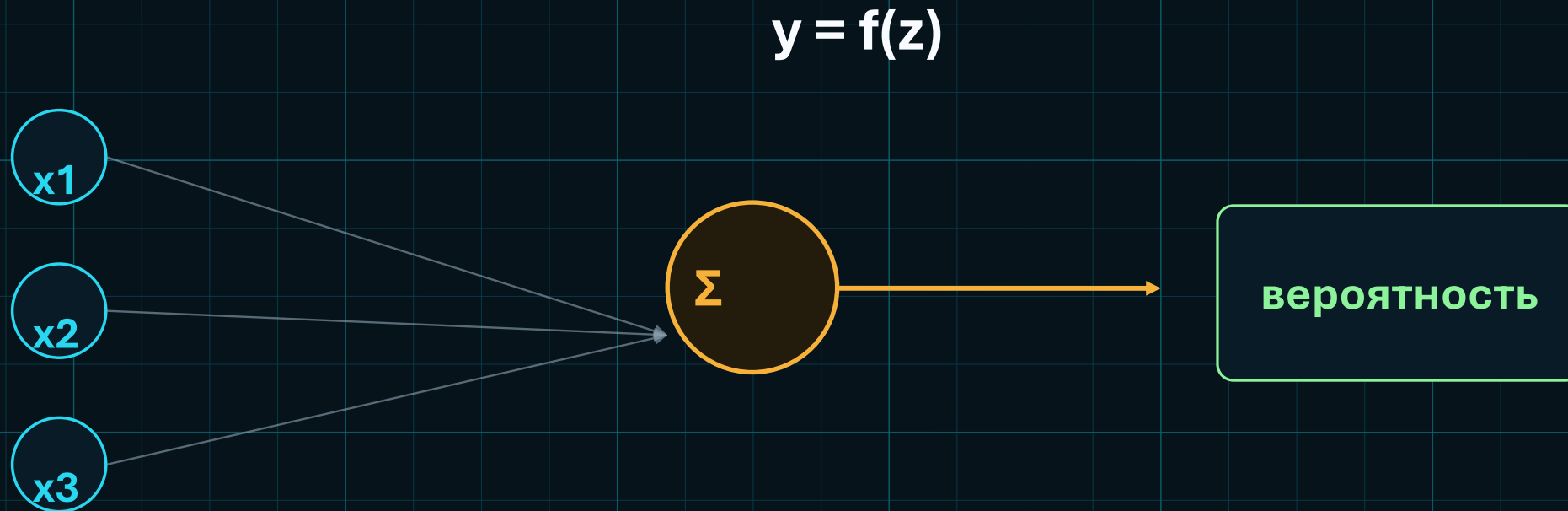
Нейросеть начинается с простого блока

$$z = w_1x_1 + w_2x_2 + w_3x_3 + b$$



Нелинейность: маленький поворот, большая польза

Нейросеть начинается с простого блока



нелинейность нужна, чтобы сеть не была одной большой линейной формулой

Пример: день или ночь

входы: яркость, доля тёмных
пикселей, яркие точки

на выходе — число: насколько похоже на
ночь

нейрон складывает сигналы с весами

Вопрос

11.3 Нейросети

?

Откуда берутся веса нейрона?

Подсказка: веса меняются после ошибок, а не появляются из воздуха.

Ответ

11.3 Нейросети



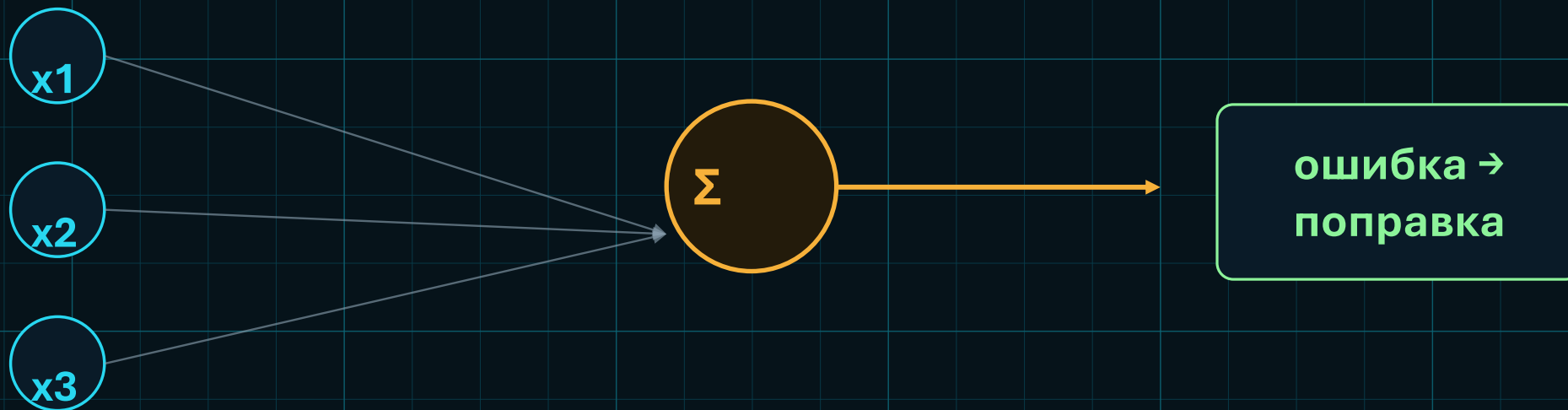
**Весы в начале могут быть случайны.
Их улучшают в процессе обучения сети**

- модель делает прогноз
- сравнивает с правильным ответом
- получает ошибку
- немного меняет веса
- повторяет это много раз

Обучение: прогноз → ошибка → поправка

Нейросеть начинается с простого блока

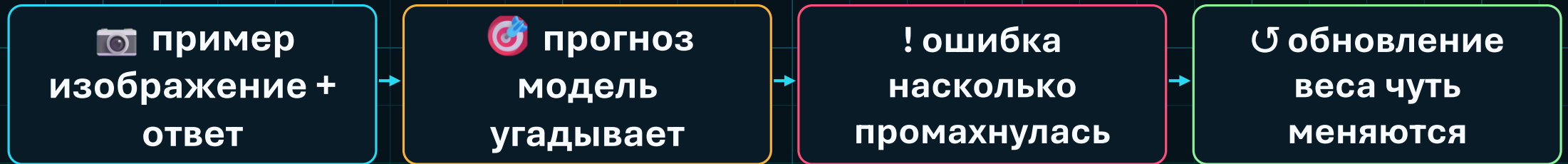
веса немного меняются



нелинейность нужна, чтобы сеть не была одной большой линейной формулой

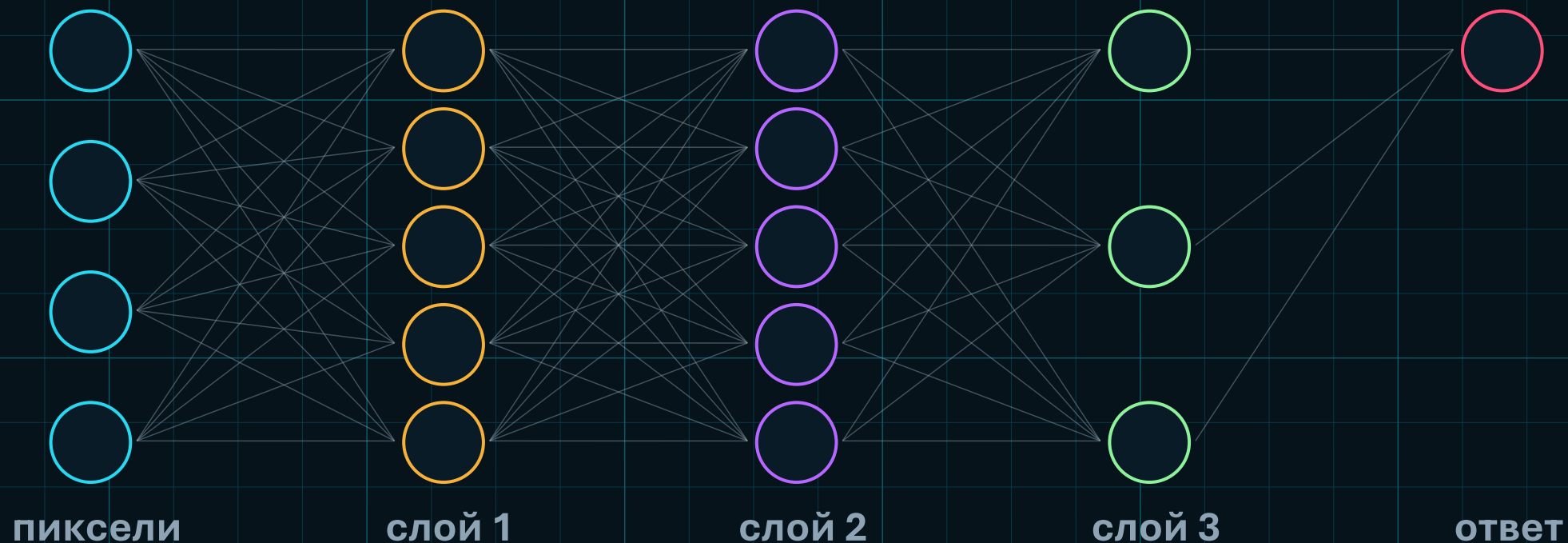
Цикл обучения

Один и тот же шаг повторяется много раз



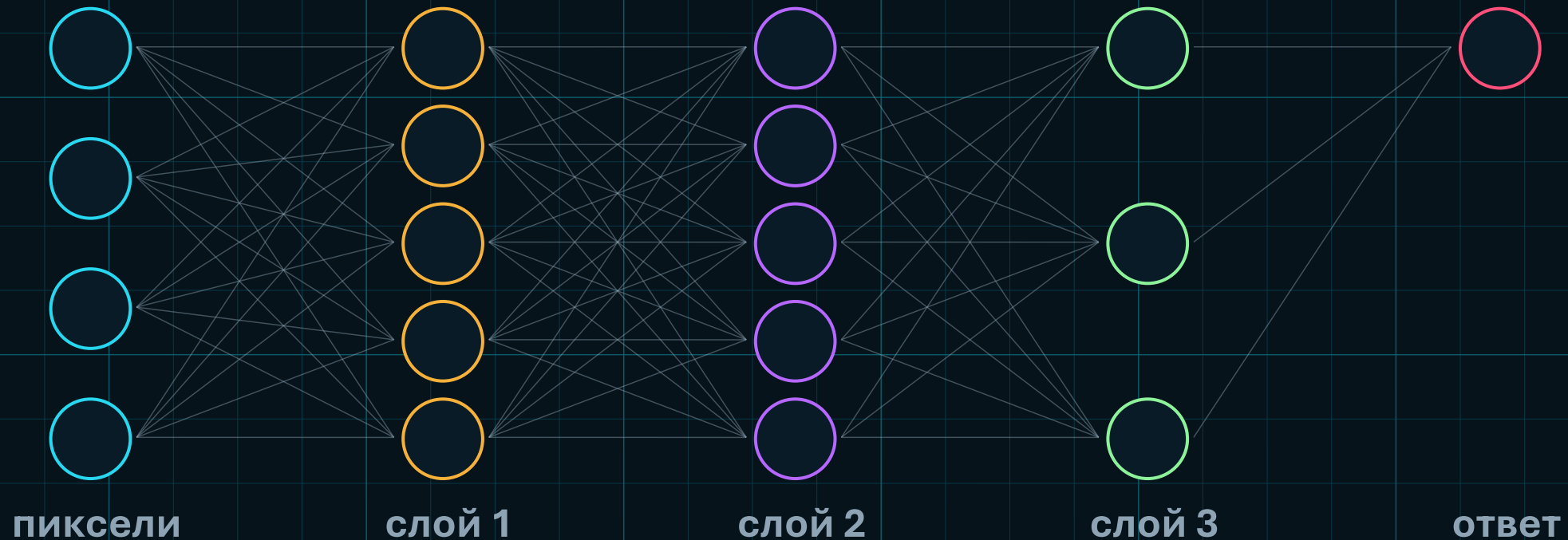
Одного нейрона мало. Нужны слои! Много слоев!

Много нейронов параллельно ищут разные сигналы



Глубина: представление строится постепенно

Слой за слоем данные становятся удобнее



Ранние уровни

простые локальные паттерны

маленькие изменения цвета

края и контрасты

эти признаки ещё далеки от слова “кот”

Средние уровни

текстуры

части форм

углы

повторяющиеся детали

Глубокие уровни

комбинации частей

пространство становится полезным для
решения задачи

похожие объекты начинают сближаться

БОЛЬШИЕ
ИДЕИ
НАЧИНАЮТСЯ
С ДАННЫХ



Как работает нейронная сеть?

От пикселей к смыслу

Маленькие
признаки —
большие
возможности

ПИКСЕЛИ



Изображение
представлено числами
(яркость пикселей)

ПРОСТЫЕ КРАЯ



Первые слои находят
простые признаки:
линии, края, контрасты

ТЕКСТУРЫ



Средние слои
распознают текстуры
и повторяющиеся
структуры

ЧАСТИ ОБЪЕКТА



Дальше сеть находит
части объекта:
глаза, уши, нос, усы

ЦЕЛЫЙ ОБЪЕКТ



Глубокие слои
объединяют все признаки
и распознают объект
(например, «кошка»)

РАННИЕ СЛОИ

Выделяют простые признаки
из данных

СРЕДНИЕ СЛОИ

Комбинируют простые признаки
в более сложные

ГЛУБОКИЕ СЛОИ

Собирают части в целостный образ
и принимают решение



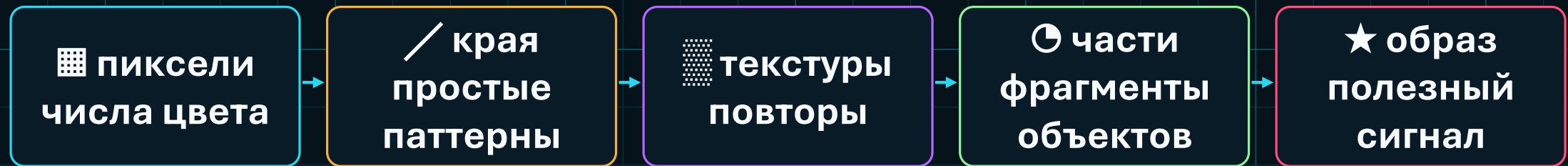
Глубокие слои собирают
сложные признаки из простых

Данные
делают
мир яснее

ДАННЫЕ
МОДЕЛИ
БОЛЬШИЕ ИДЕИ

ЛУЧШИЕ
ДАННЫЕ
ЯРЧЕ
ЗАВТРА

Пиксели → признаки → более сложные признаки



Representation learning

Сеть учит не только ответ, но и язык описания объекта

**ручной подход: человек выбирает
признаки**

**глубокий подход: сеть учит признаки по
данным**

**полученное представление можно
использовать дальше**

Совет

Не надо мистифицировать нейросети

Сеть не “понимает как человек”. Она настраивает много параметров так, чтобы уменьшить ошибку на примерах.

Ловушка

Больше слоёв не всегда лучше

Если мало данных, плохая разметка или неправильная проверка, большая сеть может уверенно выучить ерунду.

Вопрос

11.3 Нейросети

?

Если нейросеть сама строит признаки, человек уже не нужен?

Это хороший провокационный вопрос.

Ответ

11.3 Нейросети



Нужен ещё больше: ставить задачу и проверять результат.

- выбрать данные
- проверить разметку
- оценить ошибки
- заметить перекосы
- решить, можно ли применять модель

Summary

11.3 Нейросети

нейрон делает взвешенную сумму и
нелинейное преобразование

обучение — это многократное
исправление весов по ошибке

слои строят всё более удобные признаки

глубокая сеть полезна как фабрика
представлений

проверка человеком никуда не исчезает

Проверяем не запоминание слов, а понимание маршрута.

Почему CNN подходят изображениям

Свёрточная сеть ищет локальные паттерны и собирает их в более сложные.

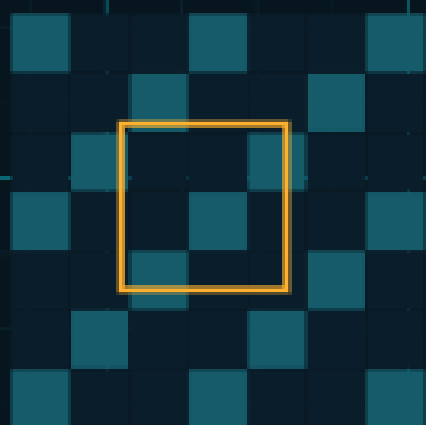
11.4

Сегодня снова: Стоп, проверяем!

Доктор Игорь Клейнер

Filter ищет один паттерн

Идея свёртки без тяжёлой математики



пиксели



маленькое окно

filter / kernel

ищет локальный
паттерн

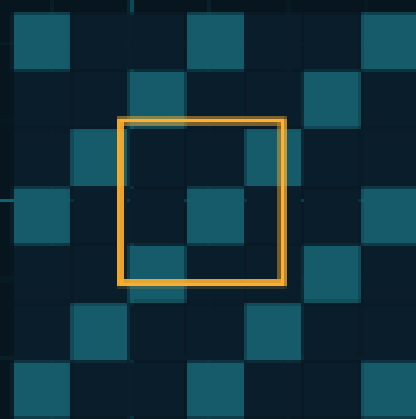


признак

Один filter не понимает всю картинку. Он проверяет: «есть ли здесь мой маленький паттерн?»

Feature map: карта находок

Где filter увидел свой паттерн

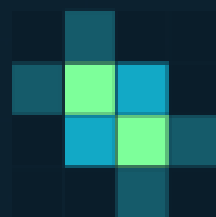


пиксели



feature map

карта: где паттерн найден



признак

не одно число,
а маленькая
карта

Feature map показывает не просто «нашёл / не нашёл», а где именно filter увидел свой паттерн.

Почему это удобно для картинок

объект может появиться в разных
местах

локальная структура важнее, чем номер
пикселя в таблице

один и тот же filter можно применять по
всей картинке

Много filters — много карт признаков

Один ищет края, другой текстуры, третий углы



Вопрос

11.4 CNN

?

Почему CNN не нужно заново учиться искать край в каждом углу изображения?

Проверка идеи shared filter.

Ответ

11.4 CNN



Потому что один filter проходит по всей картинке.

- паттерн может быть слева, справа, сверху, снизу
- поиск локальный, но применяется везде
- это экономит параметры и помогает обобщать

Summary

11.4 CNN

свёртка ищет локальный паттерн

несколько filters дают разные карты

feature map показывает, где паттерн найден

слои собирают простое в более сложное

Проверяем не запоминание слов, а понимание маршрута.

ImageNet и перелом 2012 года

Когда стало ясно: сеть может учить признаки для изображений очень хорошо.

11.5

Сегодня снова: Стоп, проверяем!

Доктор Игорь Клейнер

ImageNet: огромная задача классификации

миллионы изображений

тысячи категорий

нужно отличать не “день/ночь”, а
множество объектов

ImageNet соревнования по классификации изображений

- Около 14 миллионов изображений
- Около 20 тысяч категорий



2012: AlexNet меняет ожидания

глубокая сверточная сеть резко
улучшила результат

началась новая эпоха компьютерного
зрения

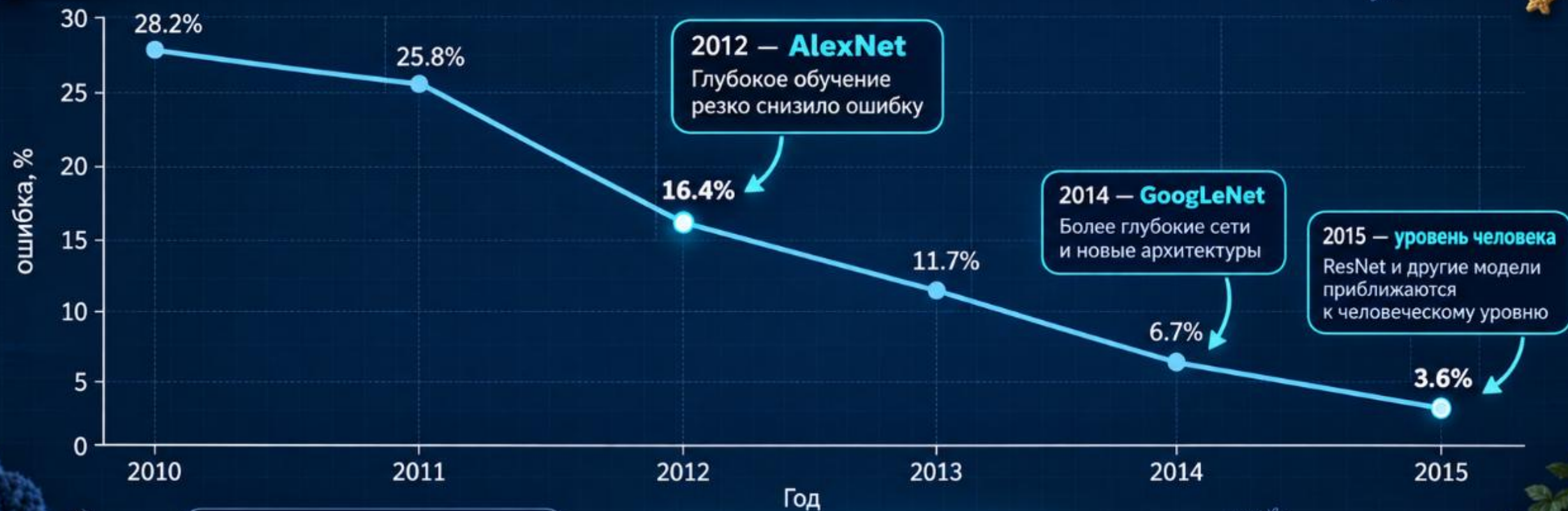
ручной feature engineering перестал быть
единственным путём



ImageNet: перелом после 2012 года

Топ-5 ошибка классификации, меньше = лучше

Большие
данные —
большие
возможности



До 2012 доминировали ручные признаки и более классические методы.



AlexNet показал силу глубокого обучения и открыл **новую эру** в компьютерном зрении.

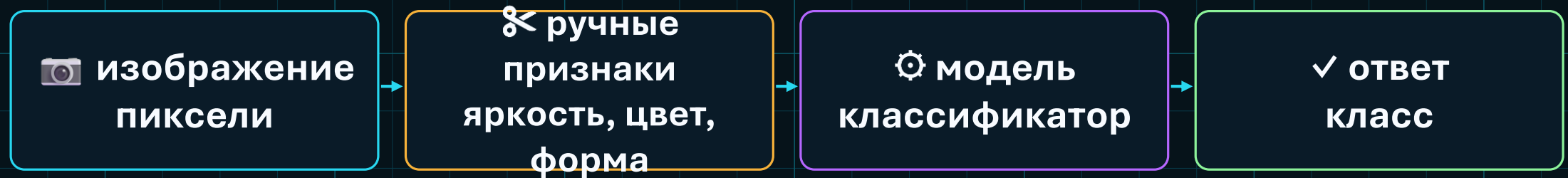
Те же данные.
Больше смыслов

ДАННЫЕ
МОДЕЛИ
БОЛЬШИЕ ИДЕИ

ЛУЧШИЕ
ДАННЫЕ
ЯРЧЕ
ЗАВТРА

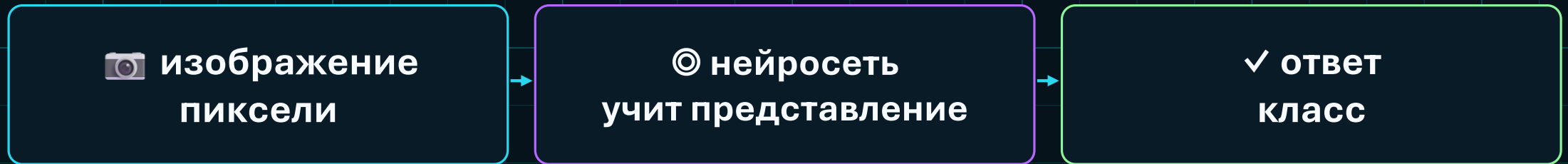
Старая парадигма

Человек сам строит признаки



Новая парадигма

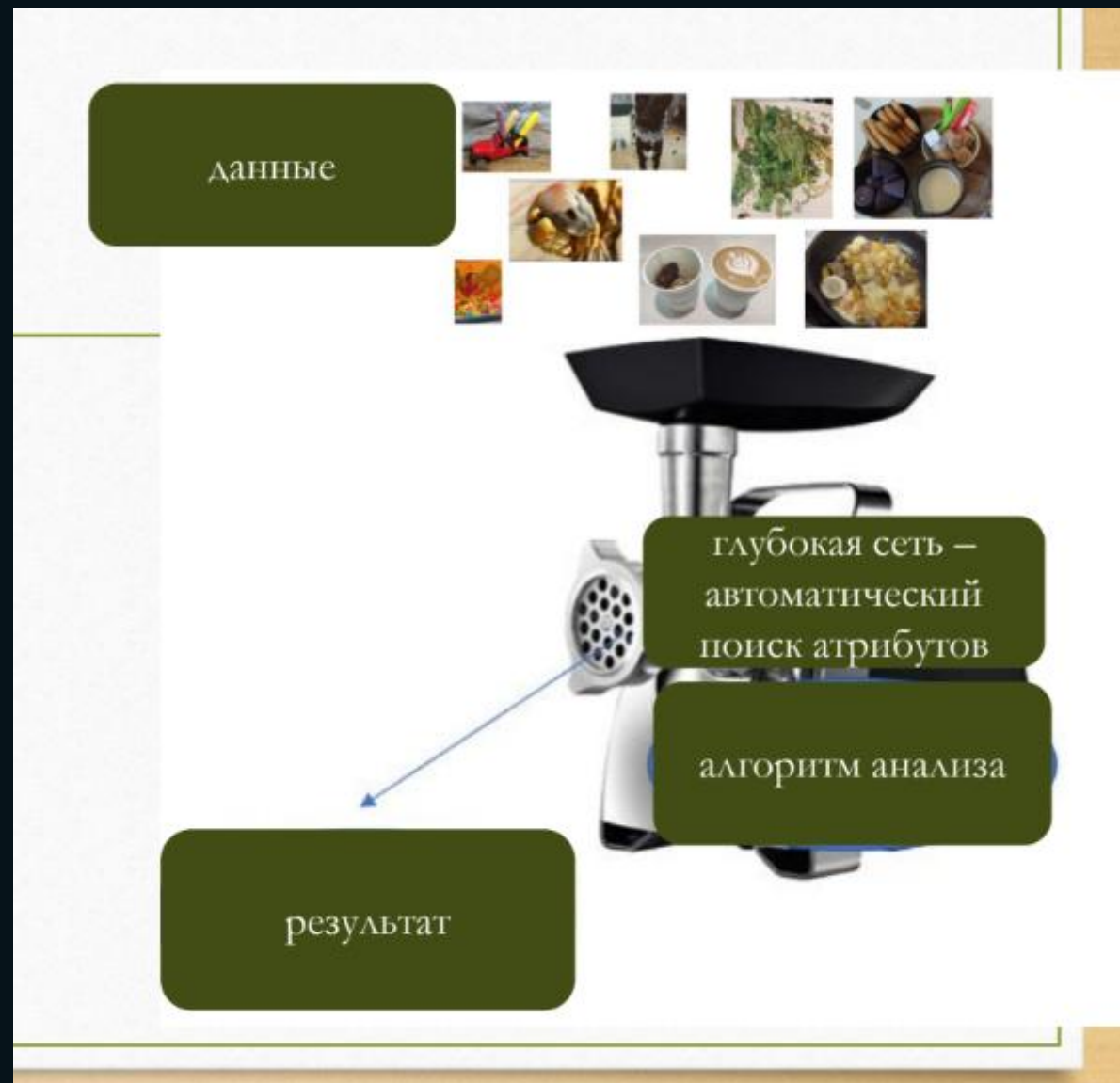
Сеть учит признаки по данным



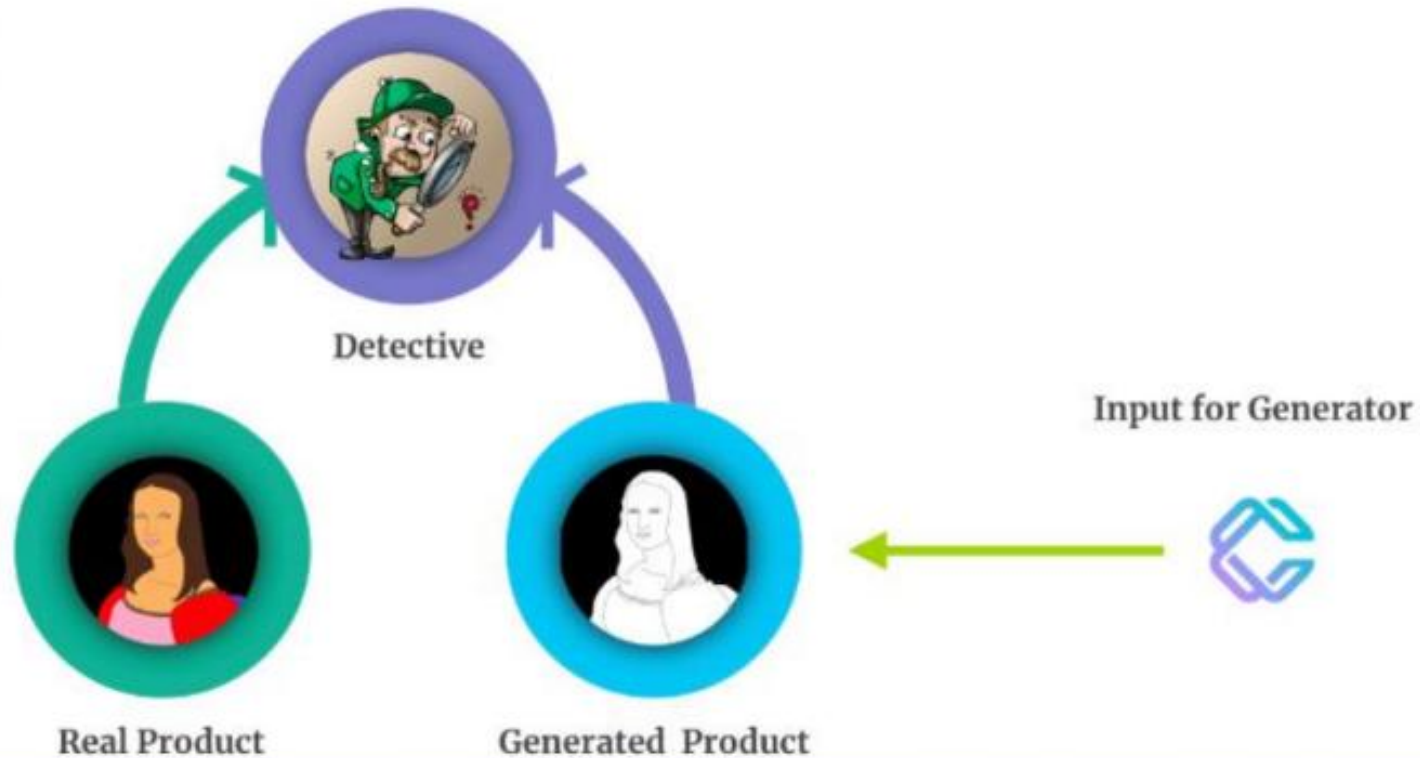
Старая парадигма



Новая парадигма



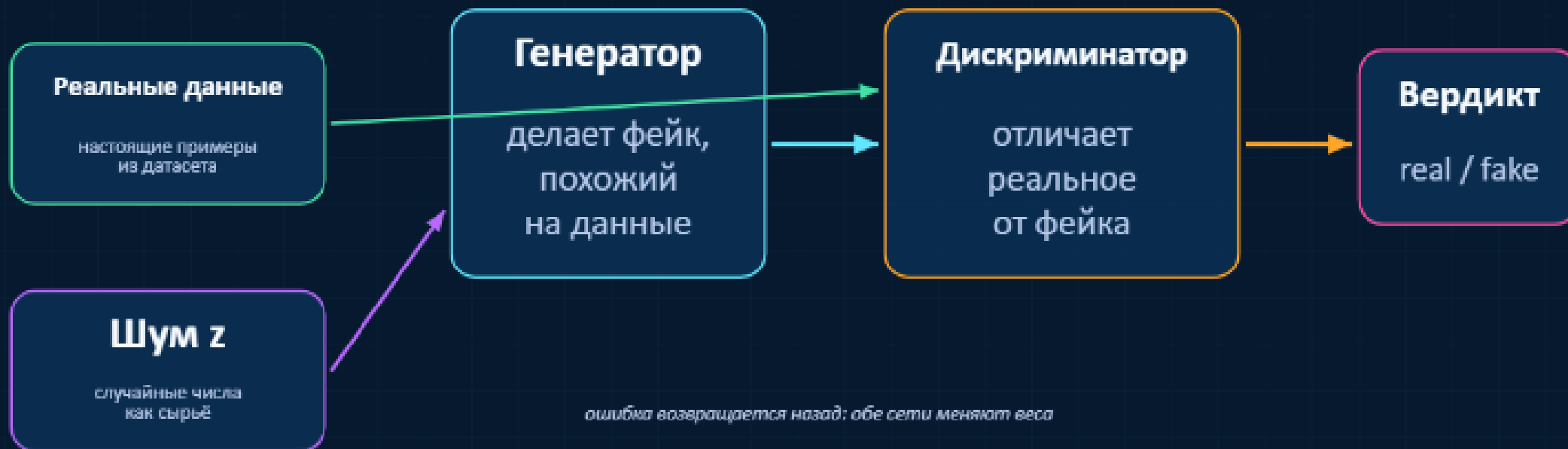
Generative Adversarial Networks



Beware
fake news
fake videos
fake fakes

GAN: две сети играют в игру

Generative Adversarial Networks — генератор против детектора



Идея: одна сеть учится обманывать, другая — разоблачать

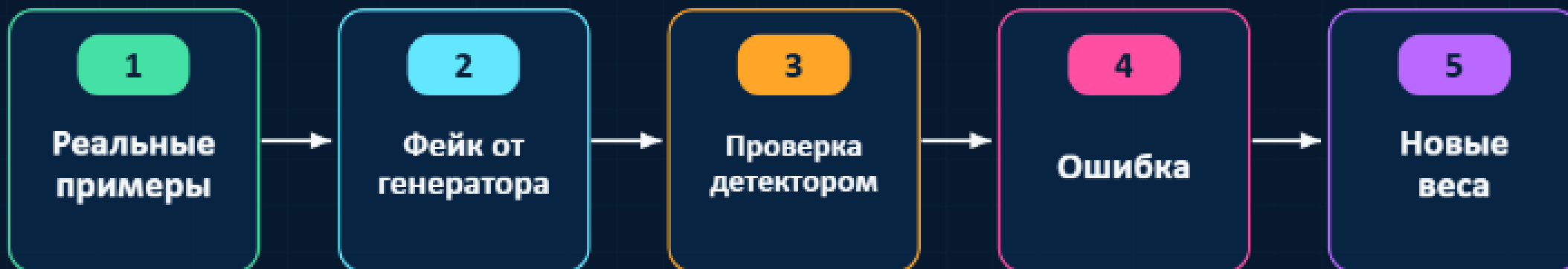
ИДЕИ
РАБОТАЮТ

GAN



Как GAN учится?

понятная петля обучения без тяжёлых формул



Генератор: сделать фейк, который трудно поймать

Дискриминатор: отличить настоящее от фейка

Чем сильнее один игрок, тем труднее становится второму

ИДЕИ
РАБОТАЮТ

Вопрос

11.5 ImageNet

?

Означает ли это, что ручные признаки умерли?

Важно не создать ложную дихотомию.

Ответ

11.5 ImageNet



Нет. Они остались полезны, но теперь есть ещё один мощный путь.

- простые признаки хороши для простых и понятных задач
- нейросети полезны для сложных представлений
- часто лучший подход — сочетать здравый смысл и learned features

Summary

11.5 ImageNet

**2012 важен как перелом в практике
computer vision**

ручные признаки не исчезли

нейросеть может учить признаки из
данных

главная ценность для нас —
representation learning

Проверяем не запоминание слов, а понимание маршрута.

Embedding: главный мост к нашему курсу

Нейросеть превращает картинку в вектор.
Дальше снова знакомый анализ данных.

11.6

Сегодня снова: Стоп, проверяем!

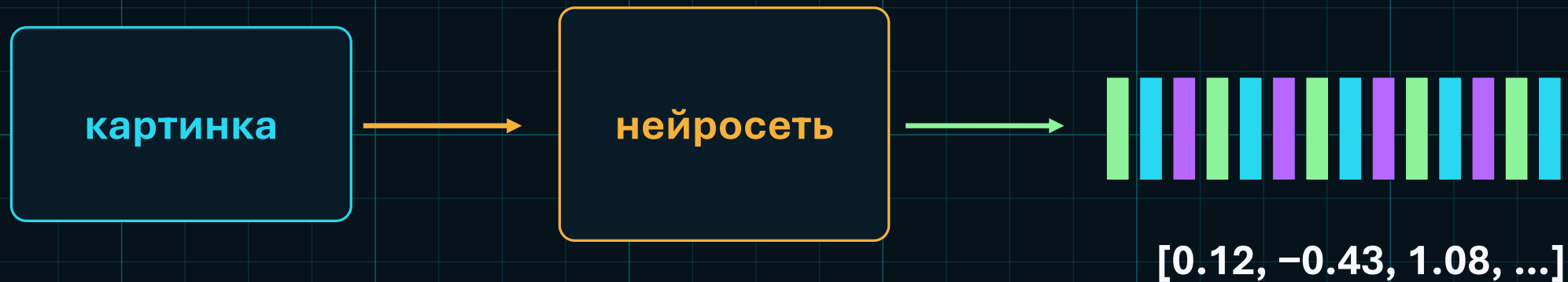
Доктор Игорь Клейнер





Сеть может дать не только класс

Можно получить от сети внутреннее представление изображения – удобные свойства



embedding = числовой отпечаток изображения

Embedding — числовой отпечаток

длинный вектор чисел

Полученный из внутреннего слоя сети

обычно похожие картинки получают
близкие векторы

числа не обязаны быть понятны как
“яркость” или “зелёный”

И снова обычная таблица



Почему похожие изображения оказываются рядом

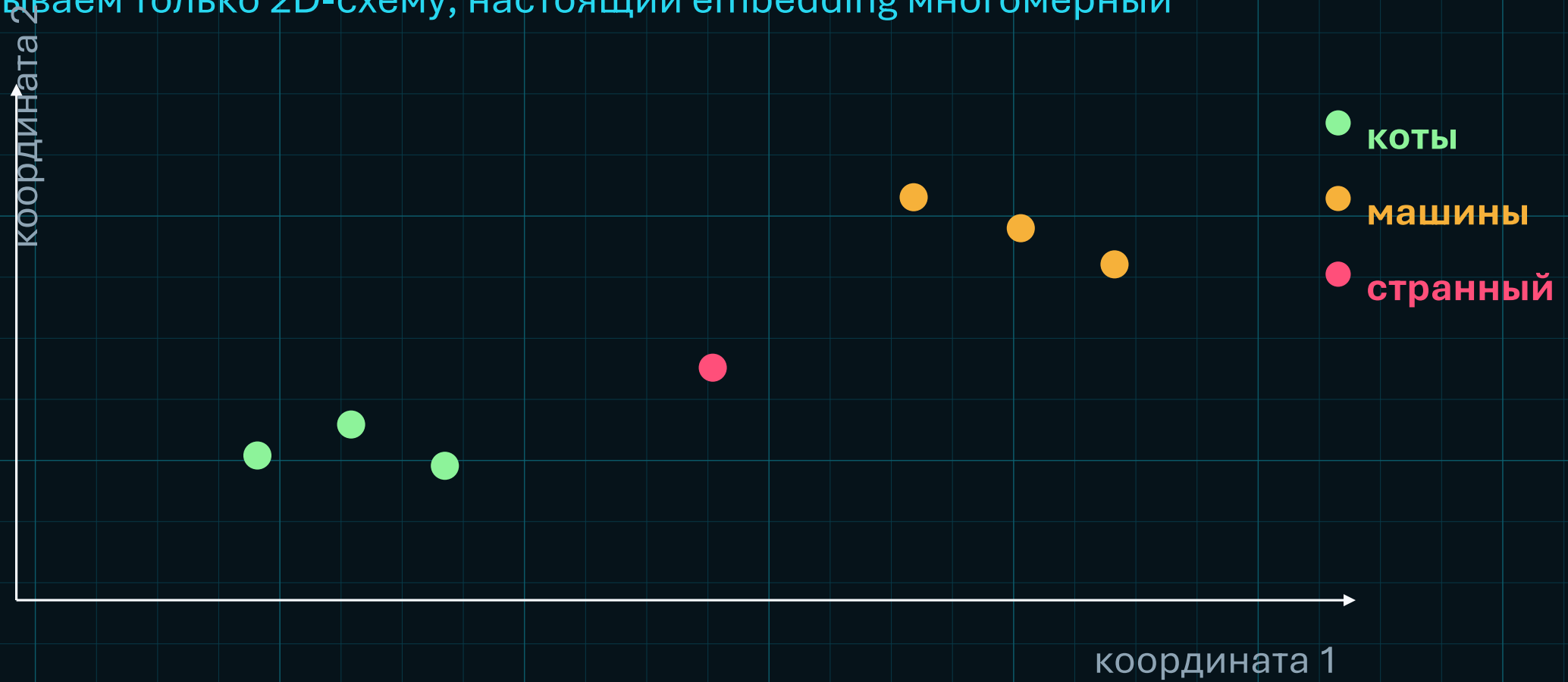
сеть обучалась различать визуальные
паттерны

расстояние между embedding-векторами
становится полезным

внутренние векторы сохраняют много
визуальной информации

Идея близости в embedding-пространстве

Показываем только 2D-схему; настоящий embedding многомерный



Embedding можно использовать много раз

Один вектор признаков — много следующих задач

1. Поиск похожих
Neighbors / similarity
search

2. Кластеризация
группы без target

IMAGE EMBEDDING
числовой отпечаток
картинки

3. Классификация
модель получает
признаки

4. Поиск выбросов
кто далеко от
остальных

5. Визуализация
PCA / t-SNE / UMAP

💡 Embedding — это не ответ, а удобное представление данных.

Transfer learning на пальцах

сеть уже обучалась на большой
коллекции изображений

для маленького учебного датасета это
часто практичнее, чем учить сеть с нуля

мы используем её как готовую фабрику
признаков

Ловушка

Embedding — это не объяснение

Вектор может отлично помогать модели, но отдельные координаты часто трудно интерпретировать человеческим языком.

Вопрос

11.6 Embeddings

?

Что общего между Bag of Words и Image Embedding?

Очень важная связь с прошлым уроком и NLP.

Ответ

11.6 Embeddings



Оба превращают сложный объект в числовую строку.

- текст → частоты слов или вектор
- изображение → embedding-вектор
- после этого можно считать расстояния и строить модели

Совет от Игоря

Думайте не “нейросеть вместо анализа”, а “нейросеть как источник признаков”

Это лучший мост для нашего курса: сложный объект превращается в таблицу, а дальше снова работают знакомые инструменты.

Summary

11.6 Embeddings

embedding — числовой отпечаток изображения

он берётся из внутреннего слоя сети

похожие картинки часто близки в embedding-пространстве

после embedding изображения становятся обычной таблицей

дальше можно применять знакомые методы анализа данных

Проверяем не запоминание слов, а понимание маршрута.

Orange: изображения превращаются в таблицу

Показываем не магию, а конкретный workflow.

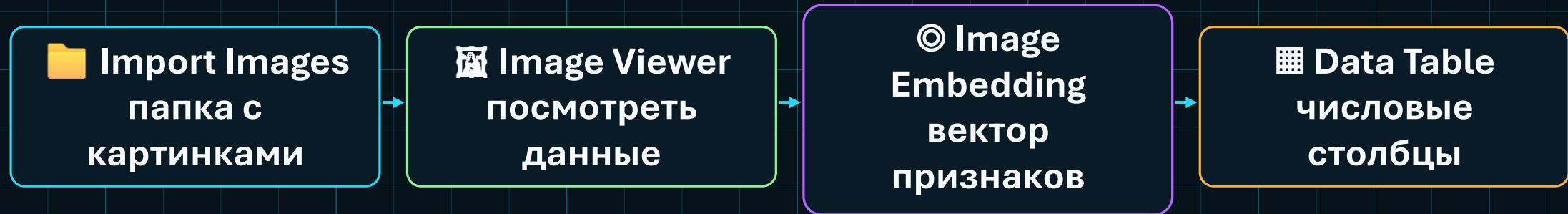
11.7

Сегодня снова: Стоп, проверяем!

Доктор Игорь Клейнер

Базовый workflow Image Analytics

Современная логика работы в Orange

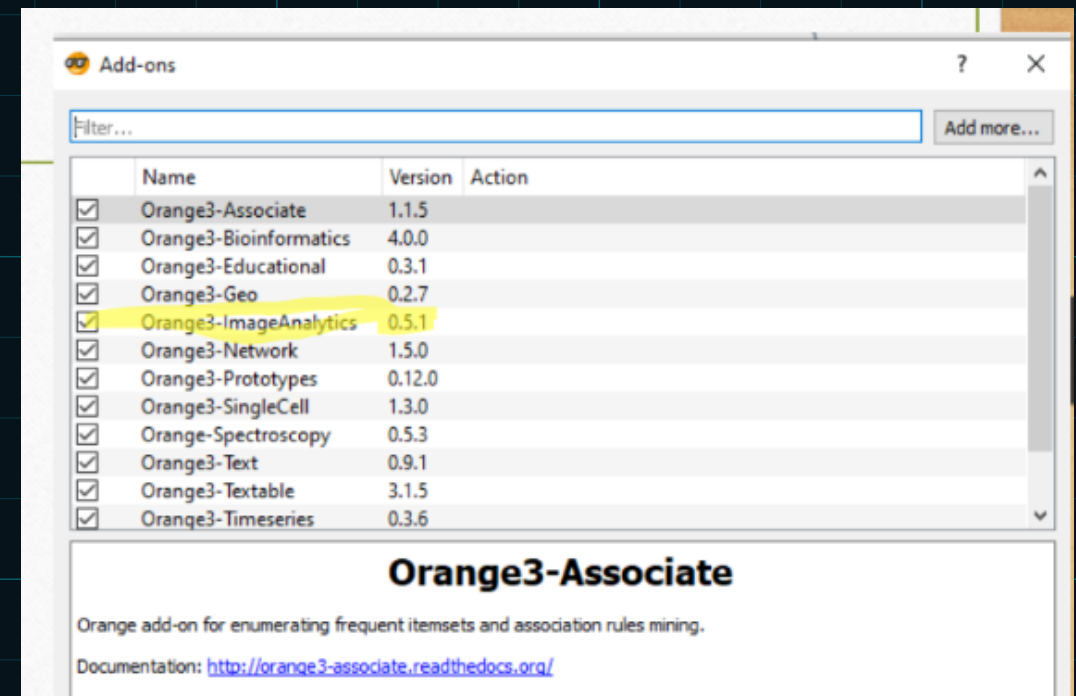


Шаг 0: Image Analytics

проверяем, установлен ли add-on

если виджетов нет — сначала
устанавливаем расширение

виджеты: Import Images, Image Viewer,
Image Embedding, Image Grid



Import Images

загружает картинки из папки

добавляет путь, имя, ширину, высоту,
размер

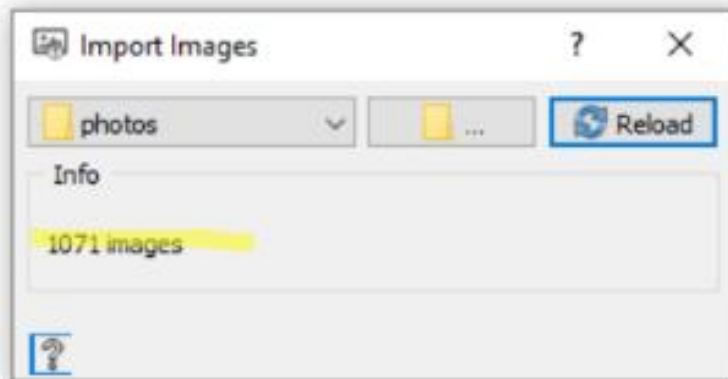
одна строка = одно изображение

подпапки могут стать классами

Загрузка изображений



Import Images

 [Select Top Level Directory](#)

← → ↶ ↷ 📁 > This PC > Downloads > celeb

Organize ▾ New folder

Name	Date modified	Type
img_align_celeba	1/11/2020 3:01 AM	File folder
people	6/8/2020 12:37 AM	File folder
small_1500	1/11/2020 12:06 PM	File folder

Image Viewer

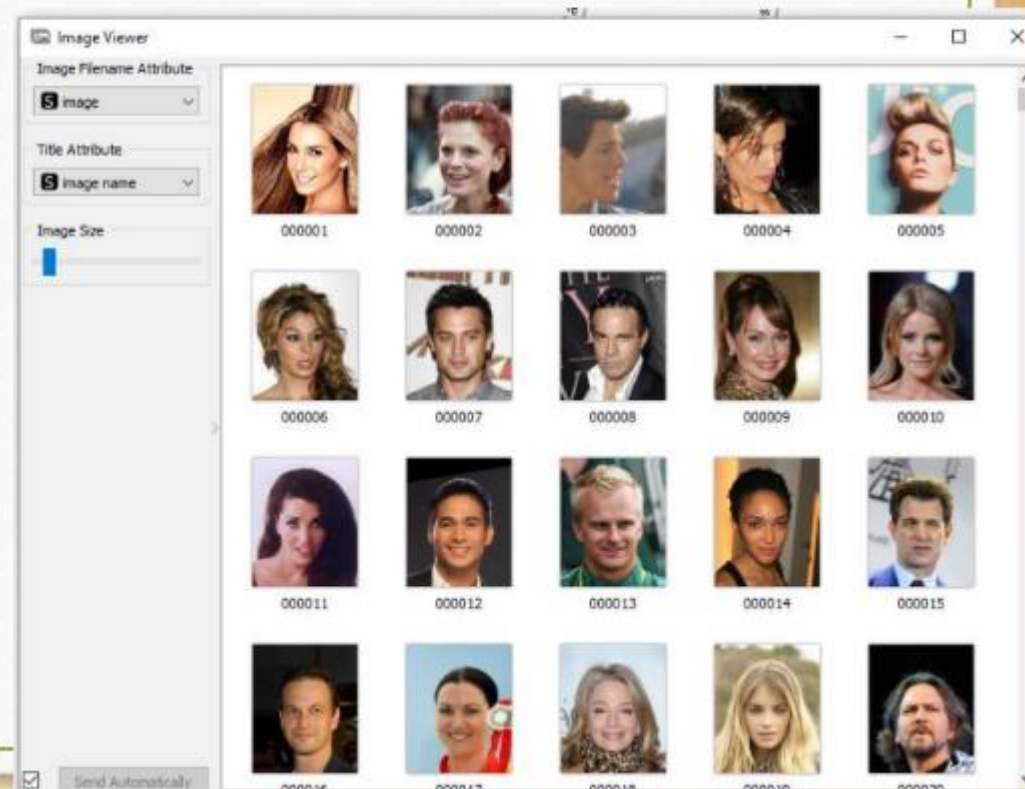
сначала смотрим глазами, что
загрузилось

проверяем, не перепутаны ли папки

ищем битые, лишние и странные
картинки

Загрузка изображений

- Модуль import images загружает изображения из папки
- Наши данные — изображения с лицами людей



Модуль image viewer

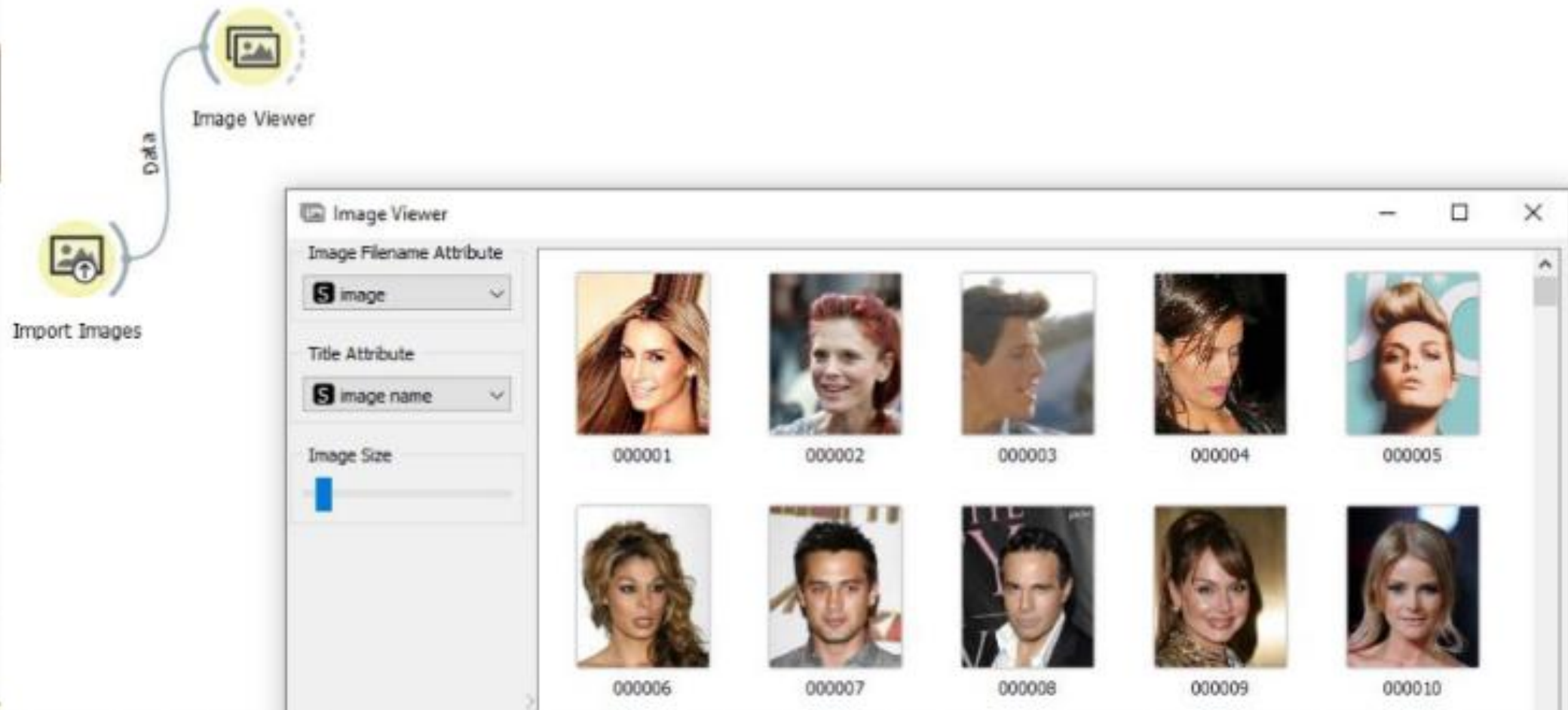


Image Embedding

выбираем embedder

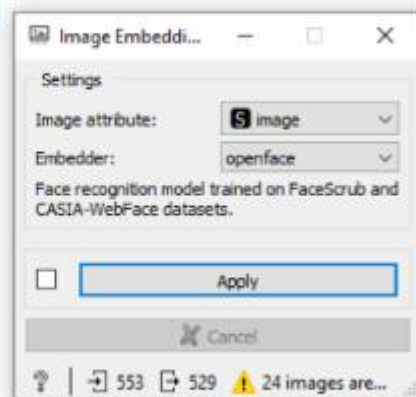
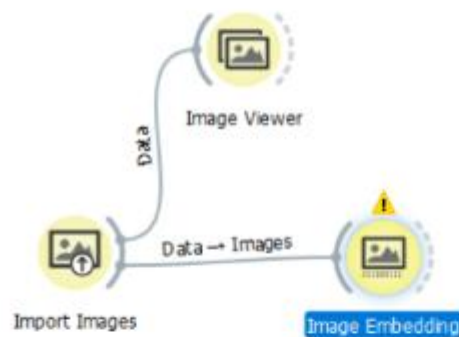
модуль может быть медленным

часть моделей работает локально, часть
может требовать интернет

на выходе — таблица с новыми
числовыми признаками

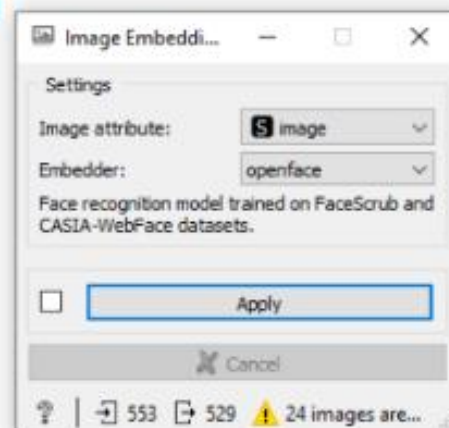
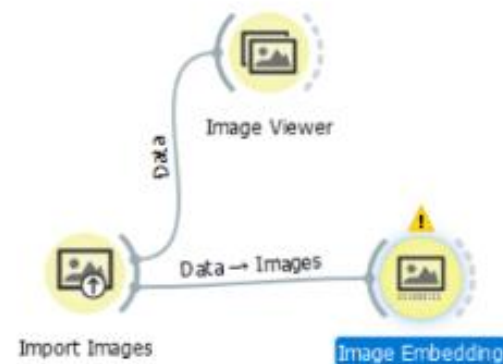
Модуль image embedding

- Передает данные выбранной нейронной сети
- Некоторые сети требуют пересылки ваших данных по интернету
- Некоторые сети работают локально
- Выбор сети зависит от задачи
- Медленный модуль ☹



Модуль image embedding

- Модуль переводит каждое изображение в формат атрибутов АВТ
- Используются атрибуты полученные одним из уровней нейронной сети
- Очень удобно – наши данные в формате АВТ 😊
- Понять атрибуты – сложно 😞





hidden origin type	image name	image	size	width	height	n0 True	n1 True	n2 True	n3 True	n4 True	n5 True
		adya klein/Downl image									
1	000001	000001.jpg	11440	178	218	-0.053012	0.0340883	-0.0529928	-0.0635259	0.122357	0.0137896
2	000002	000002.jpg	7448	178	218	-0.0524473	-0.108529	0.0290732	-0.0653206	0.0302915	0.0548248
3	000005	000005.jpg	6351	178	218	0.0249734	-0.0470992	-0.0349977	-0.0225646	0.0654447	-0.0124718
4	000006	000006.jpg	8073	178	218	-0.0528982	-0.0375369	-0.0230314	-0.0587647	0.0885369	0.0832653
5	000007	000007.jpg	8203	178	218	0.0235286	0.0172986	0.142785	-0.0174786	0.013685	0.198834
6	000008	000008.jpg	7725	178	218	-0.0915683	0.010774	-0.0172467	-0.0710286	0.0033494	0.0860399
7	000009	000009.jpg	8641	178	218	0.0795521	-0.161666	-0.110721	-0.0794649	0.10777	-0.0669041
8	000010	000010.jpg	7003	178	218	0.0620204	-0.233974	0.0628555	0.104396	-0.000814112	-0.0798988
9	000011	000011.jpg	4948	178	218	0.0310639	-0.188207	-0.119044	-0.0103557	0.190869	0.00447682
10	000012	000012.jpg	5086	178	218	-0.00100698	0.0800772	0.0526064	0.0946974	-0.0271877	0.205562

МАШИННОЕ
ЗРЕНИЕ
ДЕЛАЕТ МИР
ЯРЧЕ

Эмбеддеры в Orange

Какой выбрать и для чего он нужен?



Один
виджет —
много
возможностей

Эмбеддер	Онлайн?	Для чего подходит	Плюсы / минусы
 SqueezeNet (local)	 нет, локально	 Быстрое универсальное эмбеддинг изображений, когда нет интернета	 Быстрый, локальный, удобно для быстрого старта  Обычно проще/слабее, чем большие модели
 Inception v3	 да	 Универсальные природные фото и общие задачи с изображениями	 Сильный универсальный baseline  Требуется интернет
 VGG-16	 да	 Универсальные признаки изображений, классический baseline	 Понятная и классическая модель  Более тяжёлая/медленная, требует интернет
 VGG-19	 да	 Похожа на VGG-16, но немного глубже	 Более богатые признаки  Ещё более тяжёлая, требует интернет
 Painters	 да	 Картины, стиль живописи, сходство художников	 Хорошо работает для изображений искусства  Не подходит для обычных фото-задач
 DeepLoc	 да	 Микроскопические изображения, дрожжевые клетки	 Специализированное для биологических изображений клеток  Практически бесполезен для обычных фотографий
 openface	 да	 Лица и портреты, сходство лиц	 Полезен для задач, связанных с лицами  Имеет смысл только когда лица — основной объект

Картинки
в цифры



Эмбеддер не ставит ярлык сам по себе —
он превращает картинку в числовой
вектор признаков.

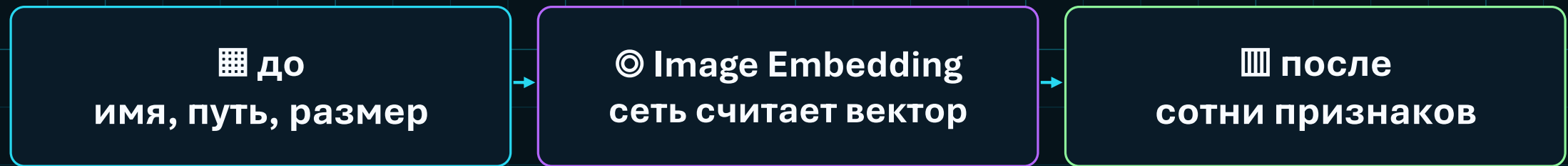


В Orange только
SqueezeNet работает
без интернета;
остальные требуют интернет.

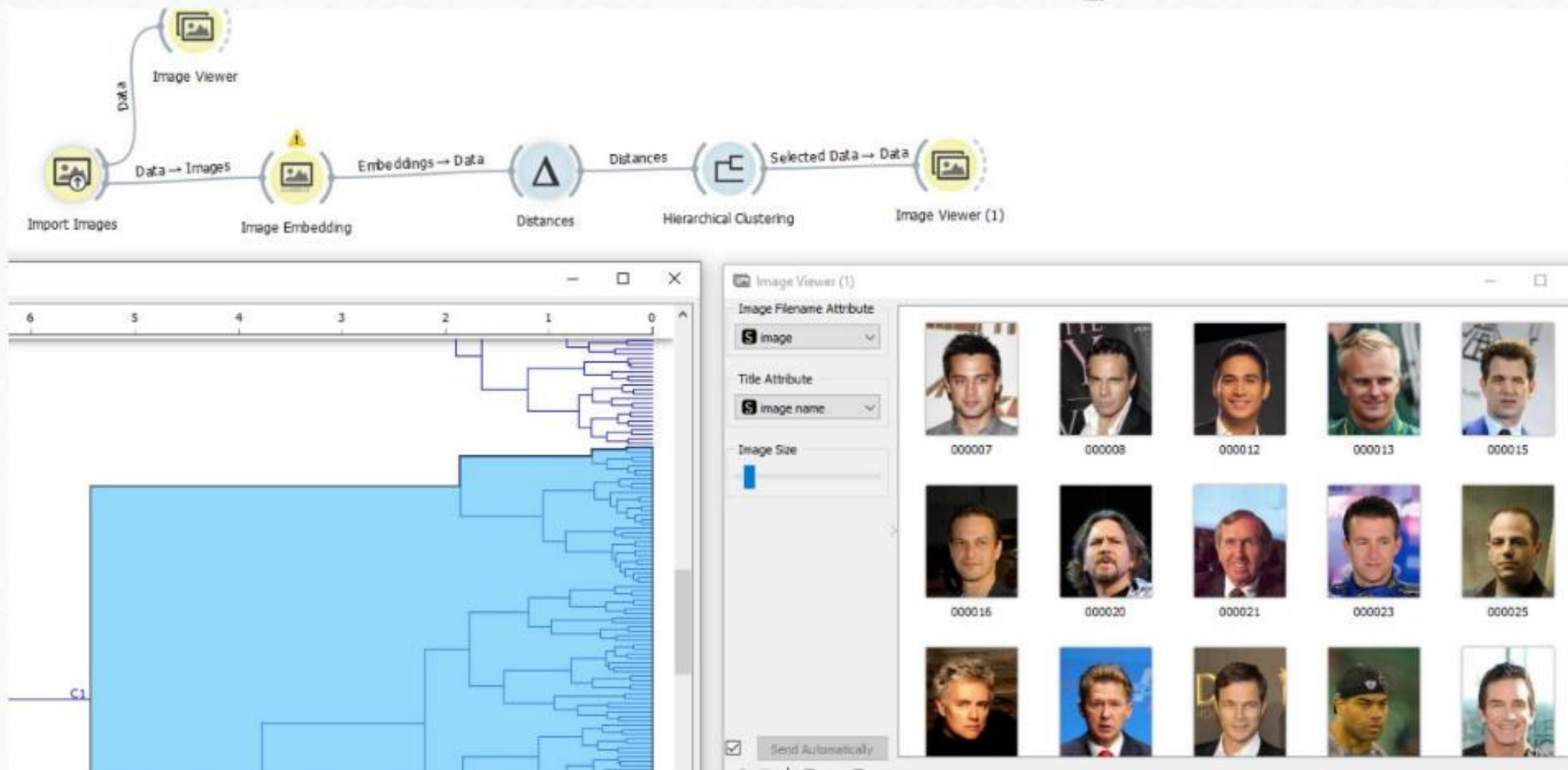
БОЛЬШЕ
ЗНАНИЙ
ЯРЧЕ
МИР

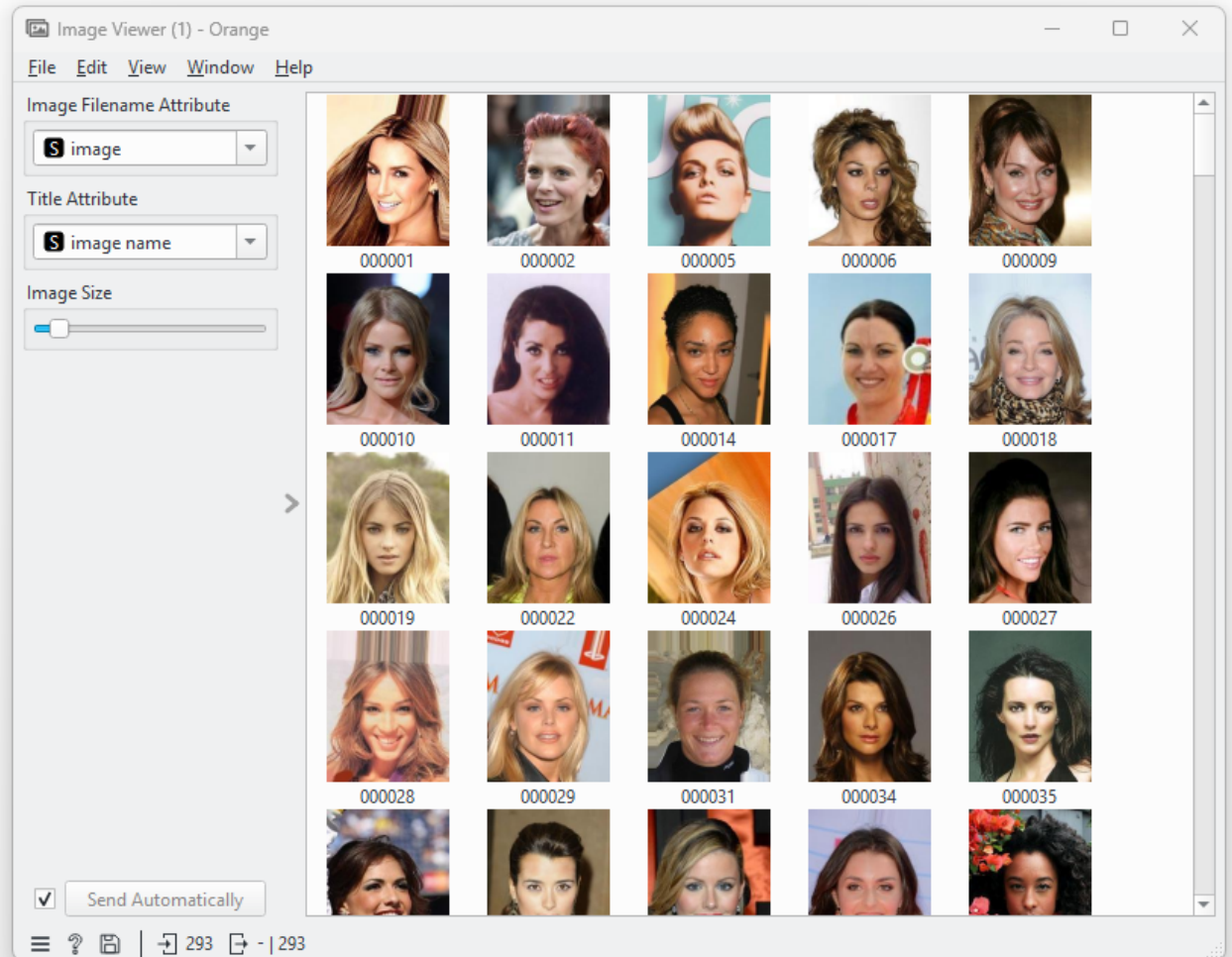
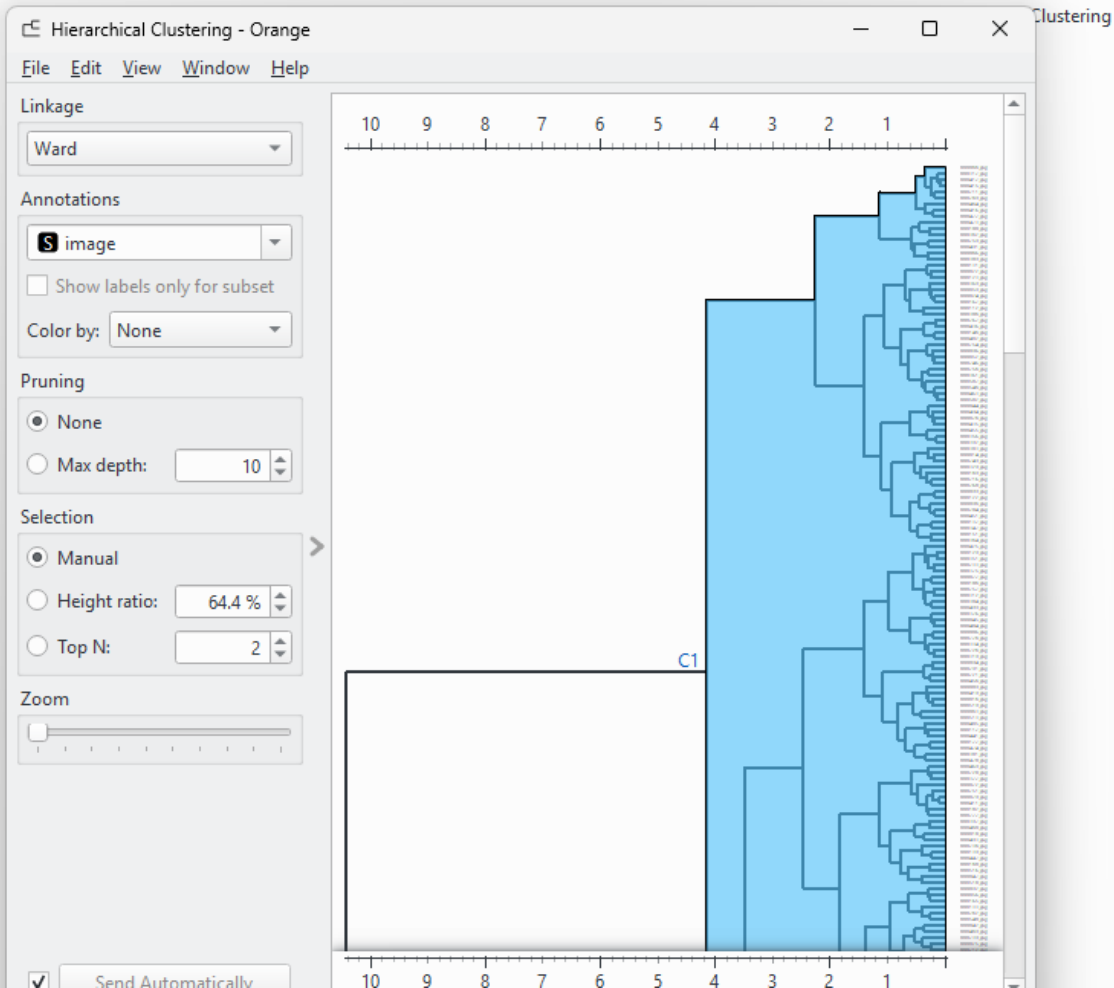
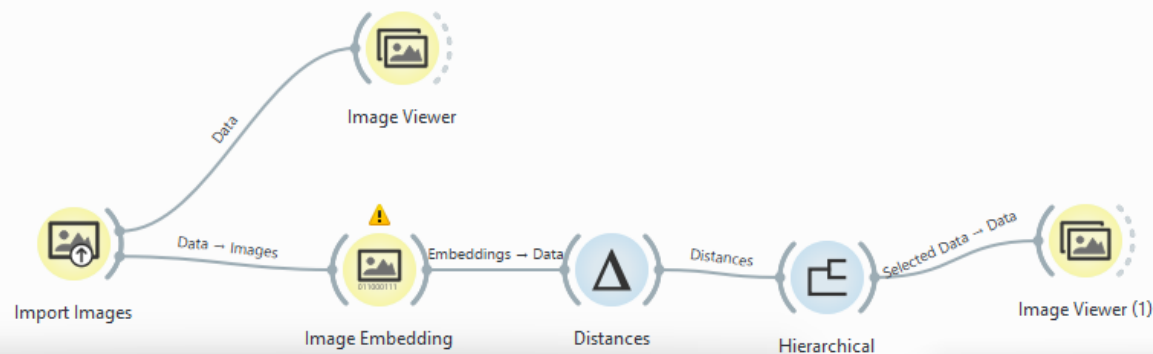
Что меняется после Image Embedding?

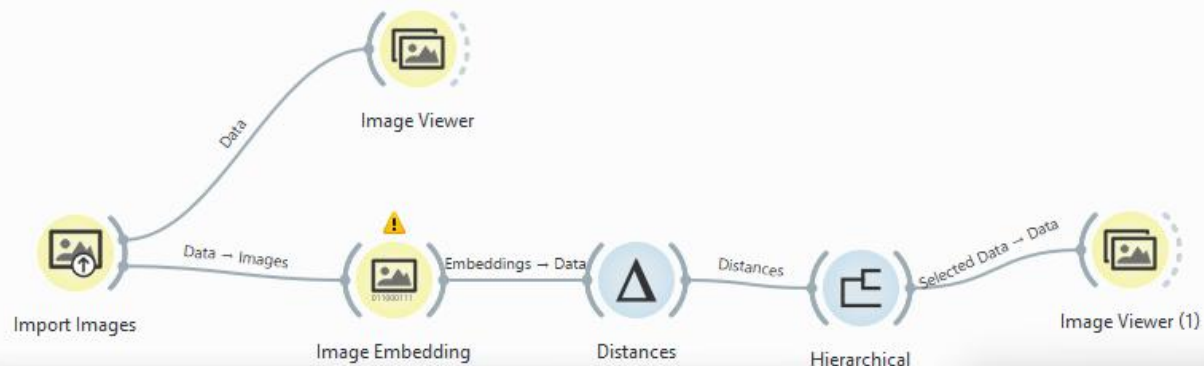
То же изображение, но таблица стала богаче



Магия нейронных сетей







Clustering - Orange

File Edit View Window Help

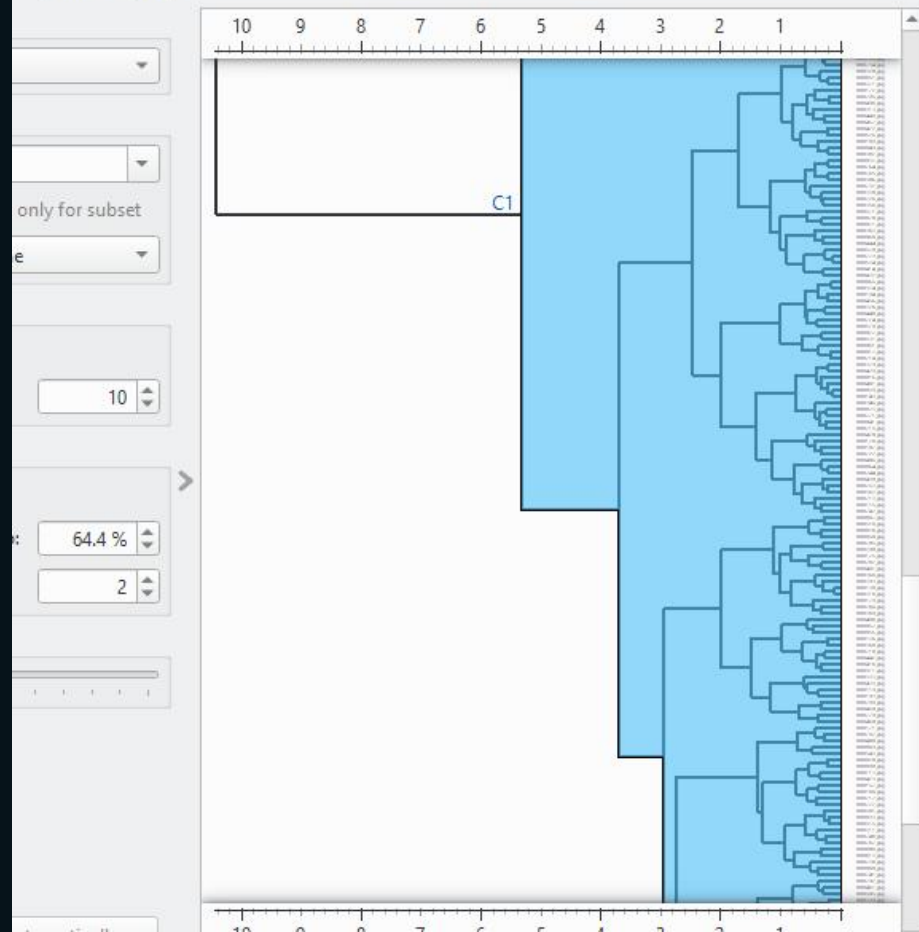


Image Viewer (1) - Orange

File Edit View Window Help

Image Filename Attribute

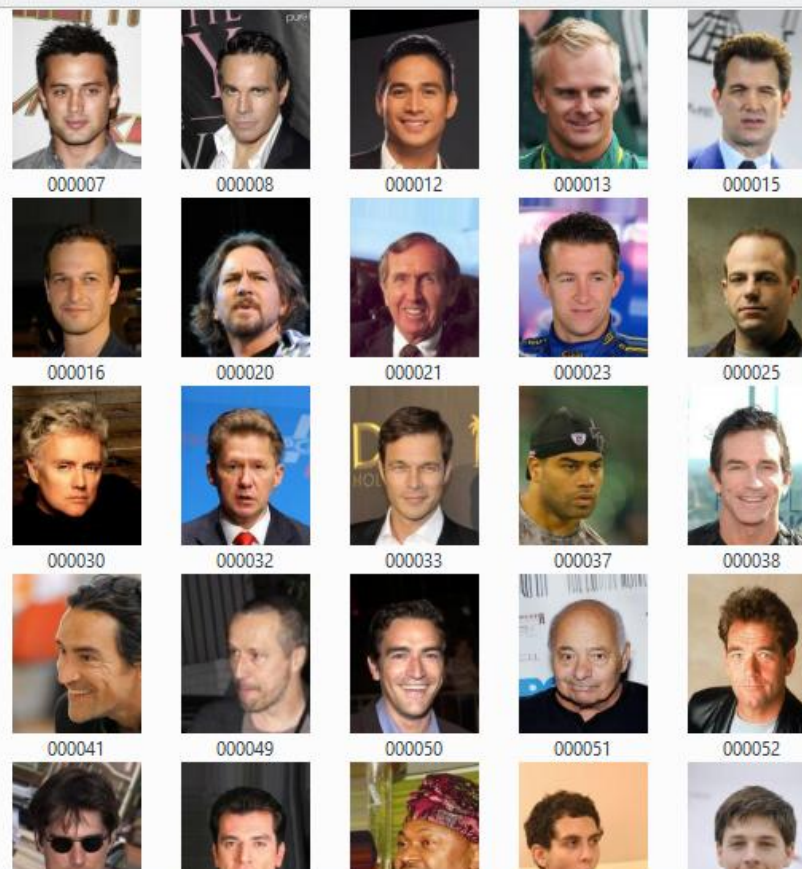
image

Title Attribute

image name

Image Size

Slider control for image size.



☒ Send Automatically

222 222

Вопрос

11.7 Orange

?

Почему перед Image Embedding всё равно надо открыть Image Viewer?

Проверяем привычку смотреть данные.

Ответ

11.7 Orange



Потому что embedding честно посчитает признаки даже для мусора.

- если в папке лишняя картинка — она станет строкой
- если классы перепутаны — модель будет учиться на ошибке
- нейросеть не заменяет проверку данных

Совет от Игоря

Делайте маленький пробный запуск

Сначала 20–30 изображений, потом весь набор. Так вы проверите путь, скорость, интернет и выбранный embedder.

Summary

11.7 Orange

Import Images создаёт строки изображений

Image Viewer нужен для проверки глазами

Image Embedding добавляет признаки из нейросети

после этого данные снова выглядят как обычная таблица

Проверяем не запоминание слов, а понимание маршрута.

Практика 1: поиск похожих изображений

Когда embedding уже есть, можно искать соседей.

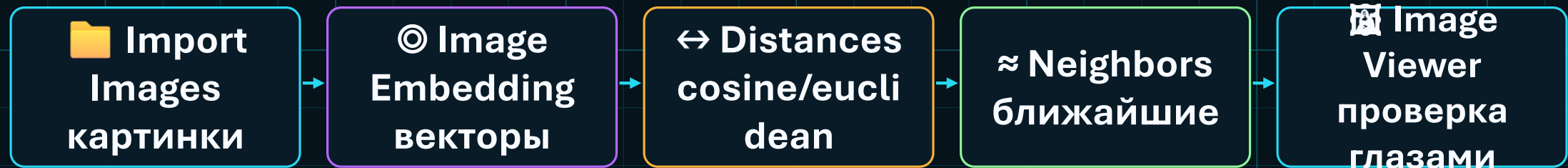
11.8

Сегодня снова: Стоп, проверяем!

Доктор Игорь Клейнер

Workflow похожих изображений

Берём одну картинку и ищем ближайшие



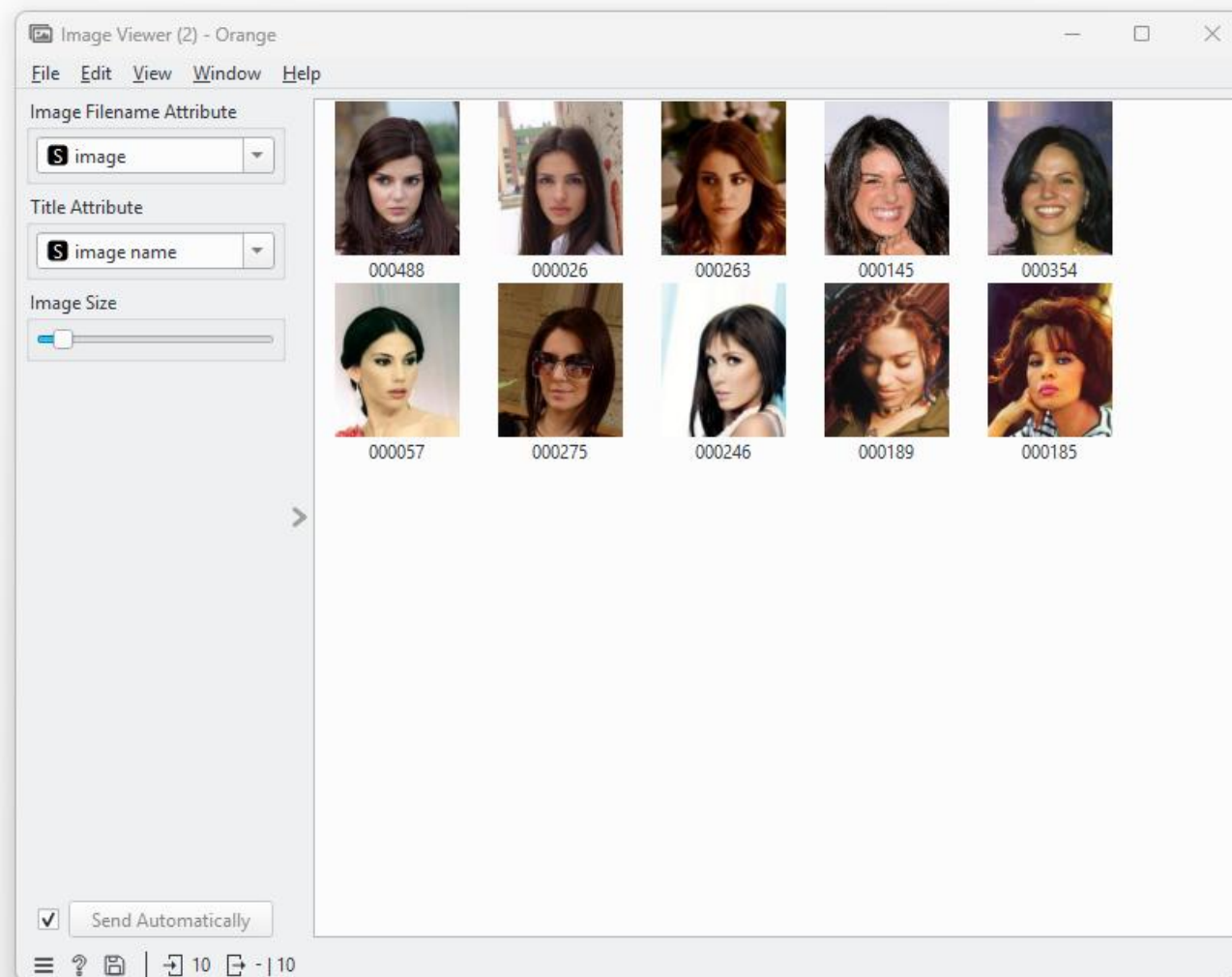
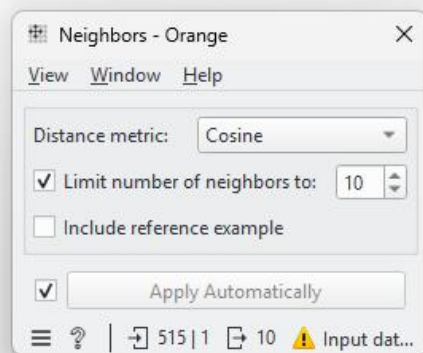
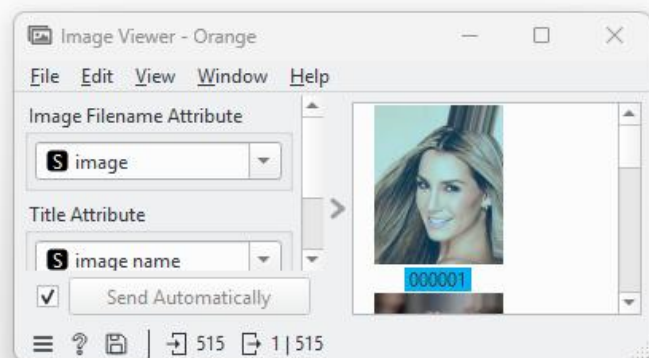
Что значит “похожее”?

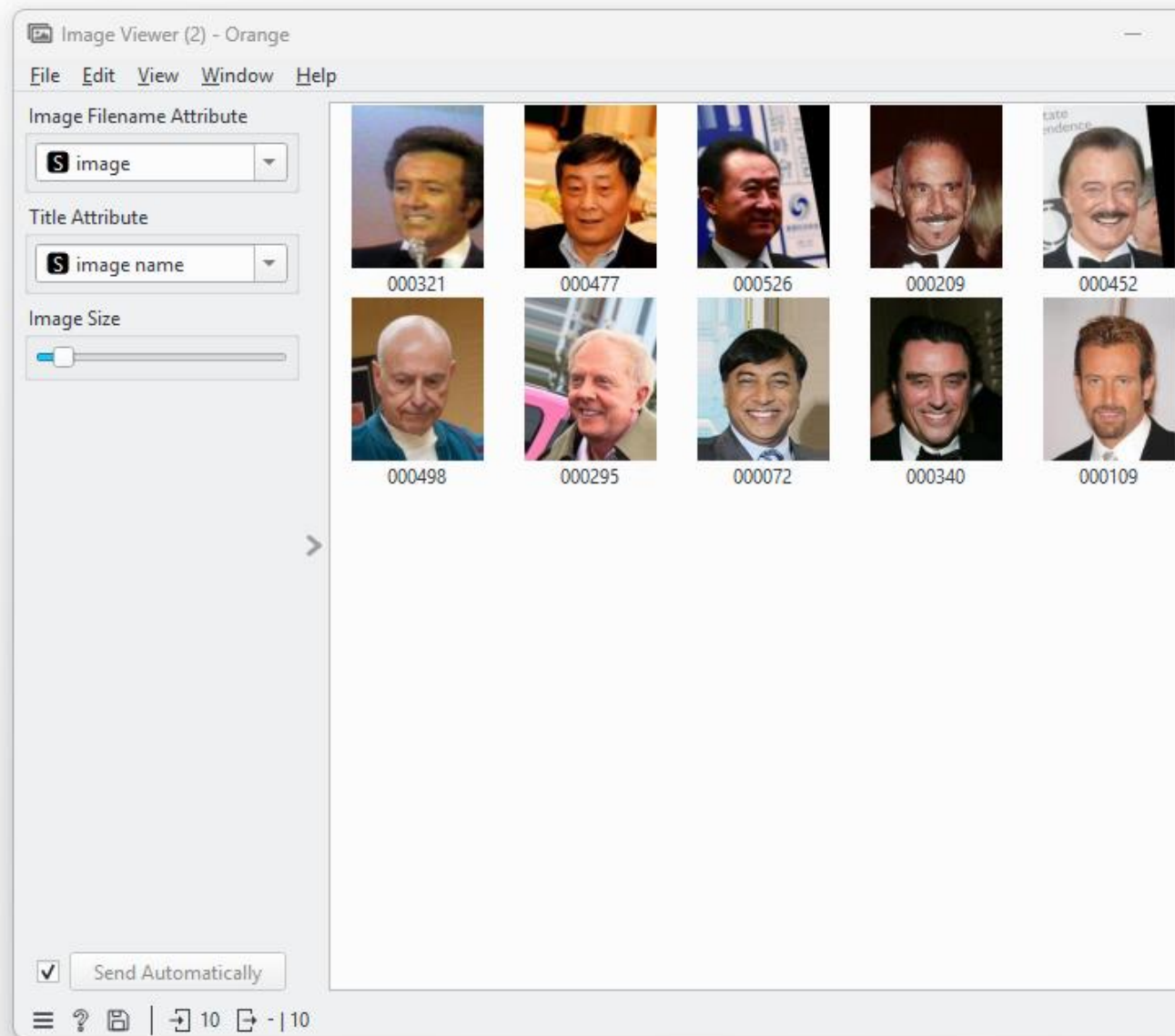
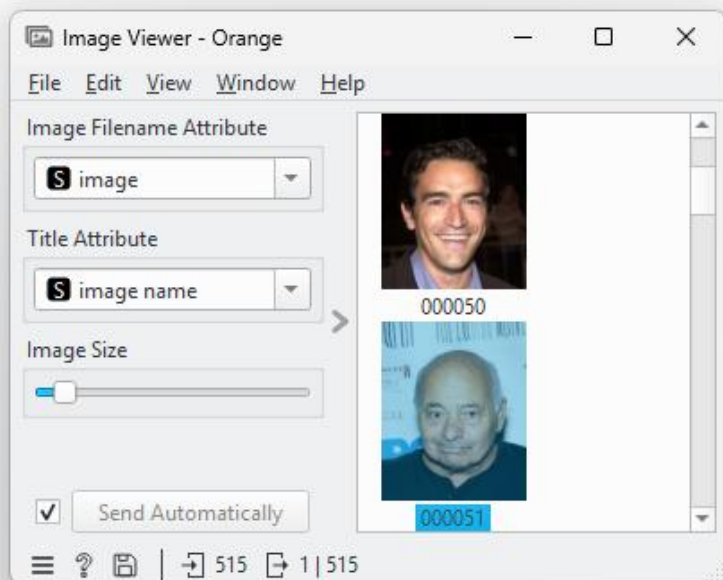
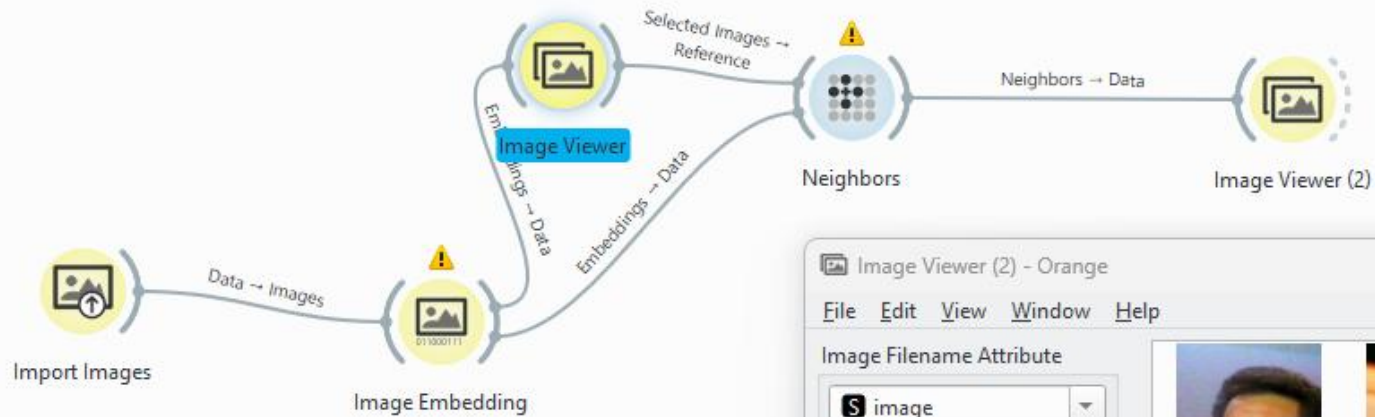
можно быть похожими по цвету

можно быть похожими по смыслу

можно быть похожими по форме

embedding выбирает свой язык
похожести







nature
communications

Article | [Open Access](#) | Published: 07 October 2019

Democratized image analytics by visual programming through integration of deep models and small-scale machine learning

Primož Godec, Matjaž Pančur, Nejc Ilenič, Andrej Čopar, Martin Stražar, Aleš Erjavec, Ajda Pretnar, Janez Demšar, Anže Starič, Marko Toplak, Lan Žagar, Jan Hartman, Hamilton Wang, Riccardo Bellazzi, Uroš Petrovič, Silvia Garagna, Maurizio Zuccotti, Dongsu Park, Gad Shaulsky & Blaž Zupan 

Read
Me

Read
Me

Look-alike Images

By: Blaž Zupan, Jan 8, 2020

There is a cool and perhaps not often used widget in Orange called Neighbors. The widget accepts the data and a reference data item and outputs the nearest neighbors of that item.

Related: [Image Analytics Workshop at AIUCD 2018](#)

Here I will show how to use it to display a set of images most similar to a selected reference image. I will use the following workflow:

Вопрос

11.8 Similarity

?

Если сеть нашла “похожее”, надо ли ей верить?

Провокация после красивого результата.

Ответ

11.8 Similarity



Верить — нет. Проверять — да.

- открываем найденные изображения
- смотрим, что именно модель считает похожим
- сравниваем с человеческой задачей
- корректируем данные или метод

Ловушка

Похожесть модели — это не обязательно похожесть человека

Если задача человеческая, человека надо возвращать в цикл проверки.

Совет

Смотрите не только удачные пары

Самые полезные примеры часто там, где модель нашла странного соседа. Это окно в её представление.

Summary

11.8 Similarity

embedding позволяет считать
расстояния между изображениями

результат обязательно проверяем
глазами

Neighbors ищет ближайшие объекты

ошибочные соседи помогают понять
язык модели

Проверяем не запоминание слов, а понимание маршрута.

Практика 2: классификация изображений

Папки становятся target, embeddings становятся признаками, дальше работает обычная ML-логика.

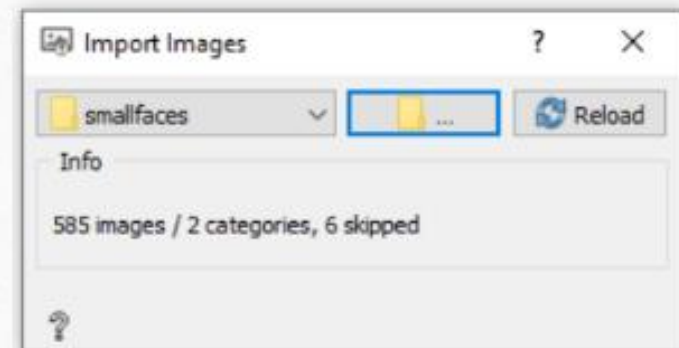
11.9

Сегодня снова: Стоп, проверяем!

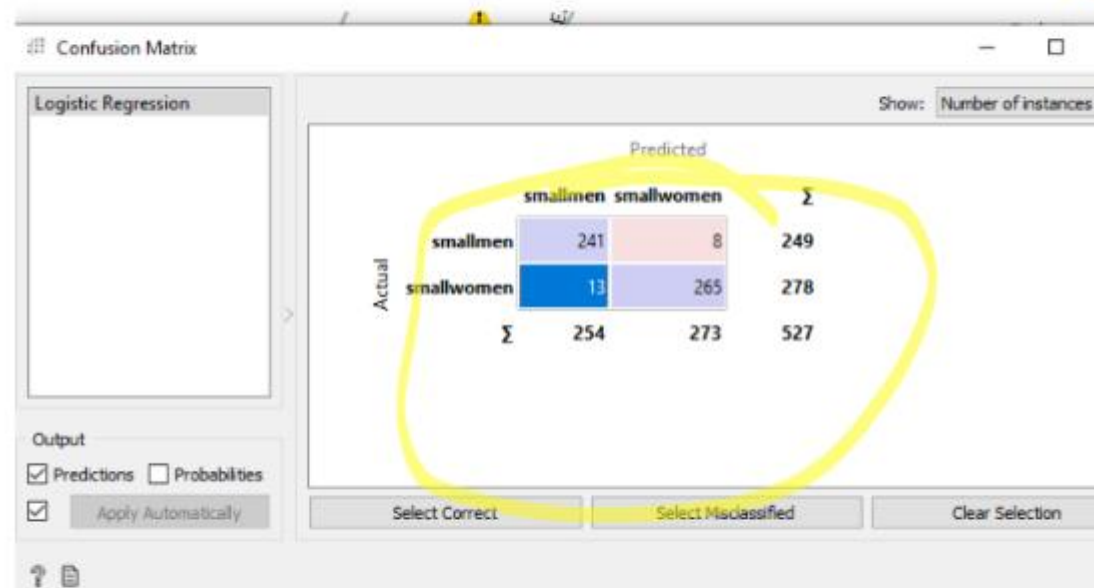
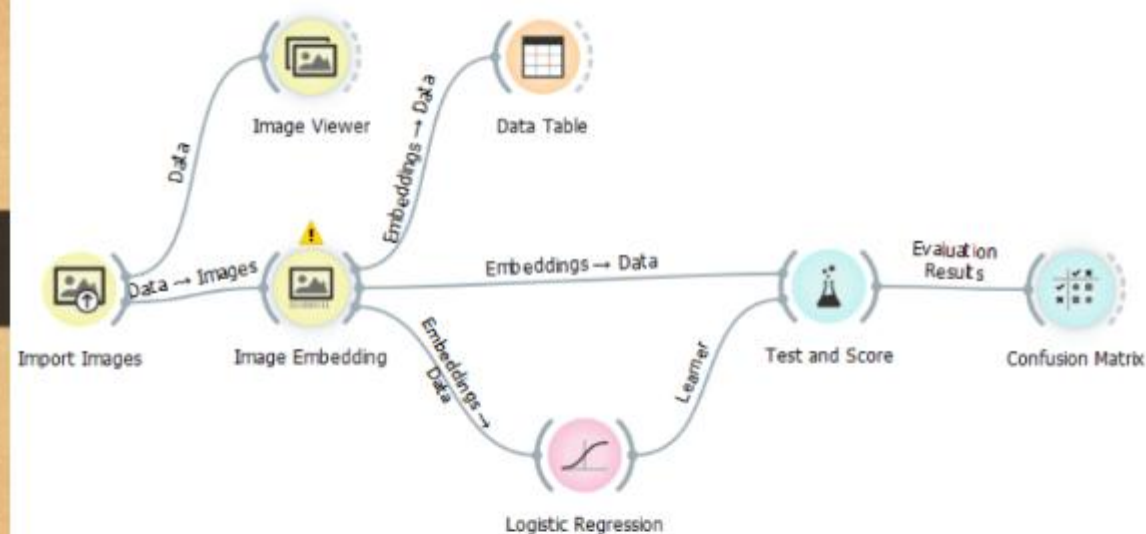
Доктор Игорь Клейнер

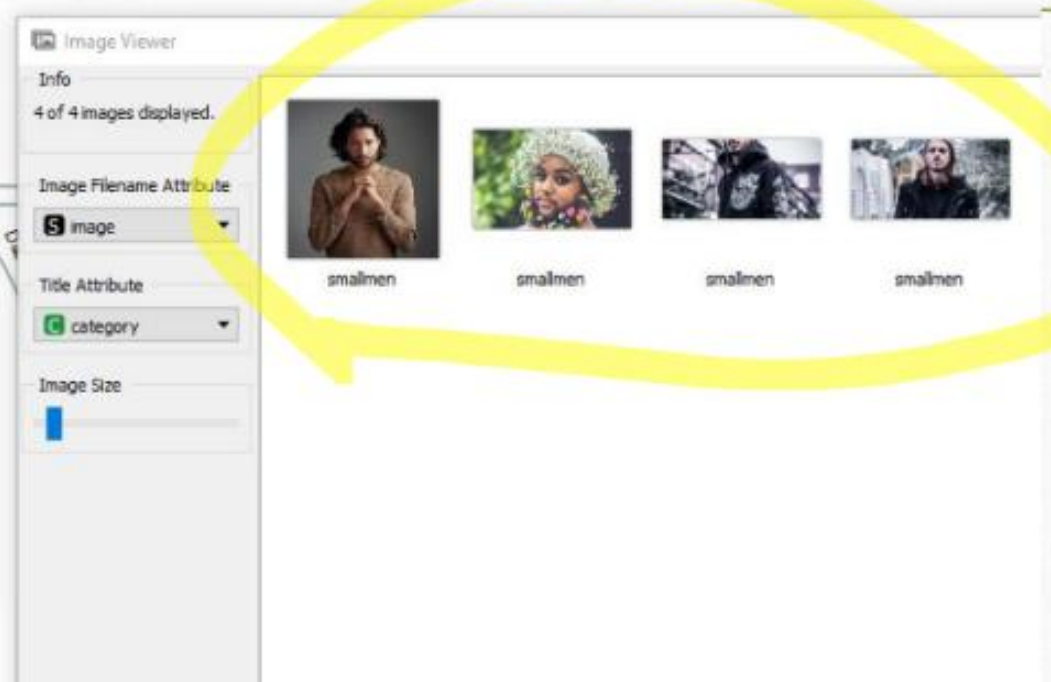
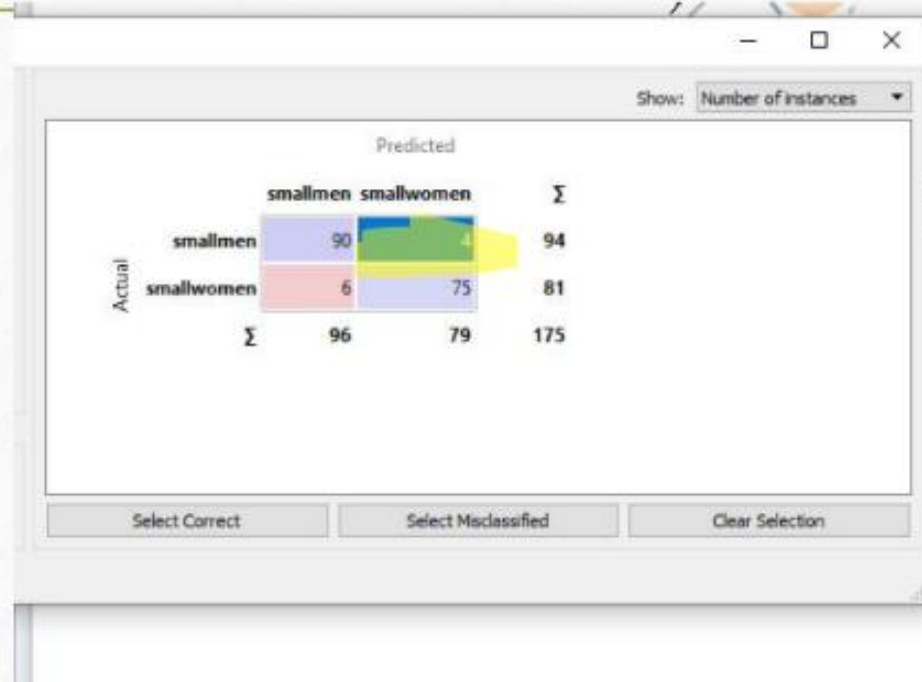
Классификация изображений

- Загрузка изображений из двух папок: мужчины и женщины
- В одной папке фотографии мужчин, в другой фотографии женщин
имя папки и будет зависимой переменной



Классификация изображений





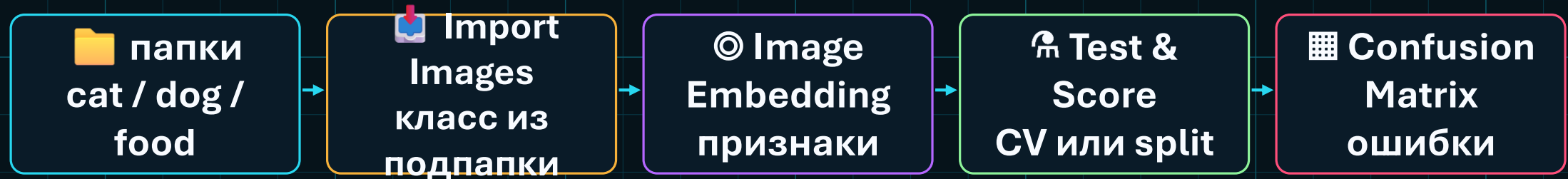
Лучший учебный пример

не “мужчина/женщина по лицу”

классы должны быть понятны и не
чувствительны этически

лучше безопасные классы: кошки /
собаки / еда / транспорт

Workflow классификации изображений



Какие модели сравнить

Logistic Regression — простой сильный baseline

Random Forest — знакомый алгоритм

kNN — естественно для embedding-пространства

SVM — часто силён на embeddings

Метрики

Accuracy — доля верных ответов

Confusion Matrix — кто с кем путается

AUC — если классов два и важен рейтинг уверенности

ошибочные картинки нужно открыть глазами

Вопрос

11.9 Classification

?

Почему высокая ассурасу может скрывать проблему?

Подумайте: какие ошибки может спрятать одна средняя цифра?

Ответ

11.9 Classification



Потому что одна цифра не показывает типы ошибок.

- класс может быть редким
- модель может системно путать два класса
- ошибки могут быть дорогостоящими
- поэтому нужна Confusion Matrix

Ловушка

Не оценивайте модель на тех же картинках, на которых выбирали все настройки

Иначе получаете красивый результат, а не честную проверку.

Совет

После Confusion Matrix открывайте картинки

Смотреть на ошибки глазами — это не слабость, а нормальная часть анализа изображений.

Summary

11.9 Classification

папки могут дать target

Image Embedding даёт признаки

после embedding используем обычные
модели

Confusion Matrix превращает метрику в
расследование ошибок

Проверяем не запоминание слов, а понимание маршрута.

Что умеет современное computer vision

Мы сделали основу. Теперь видно, куда она ведёт.

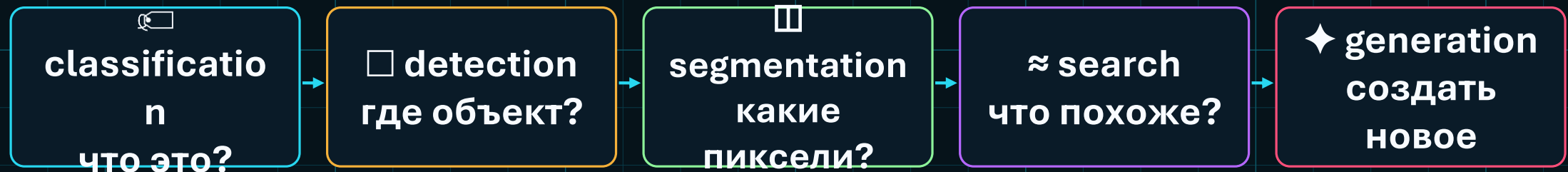
11.10

Сегодня снова: Стоп, проверяем!

Доктор Игорь Клейнер

Большая карта задач vision

Одна картинка — разные вопросы



Vision-language models

модель видит изображение и работает
с текстом

можно получать описание, поиск и
анализ

можно задавать вопросы по картинке

но проверка всё равно нужна

Генерация и deepfake

модели умеют создавать
реалистичные изображения

возникают вопросы этики, авторства и
проверки

фото перестаёт быть автоматическим
доказательством

Вопрос

11.10 Финал

?

Что сегодня важнее: нейросеть или представление данных?

Ответ будет не названием алгоритма, а объяснением всей цепочки.

Ответ

11.10 Финал



Нейросеть важна именно потому, что строит представление.

- без хорошего представления модель слепа
- embedding делает картинку строкой таблицы
- дальше снова нужны проверки, метрики и человеческий смысл

Summary

11.10 Финал

компьютер получает изображение как числа

ручные признаки помогают, но плохо масштабируются

нейросеть учит признаки по примерам

embedding превращает картинку в таблицу

Orange позволяет применить знакомые методы к изображениям

Проверяем не запоминание слов, а понимание маршрута.

Мини-квиз: что проверяем?

Быстрый устный опрос перед финалом

1. RGB-изображение
три числовые матрицы: R, G, B

2. Ручной признак
гипотеза человека: яркость, цвет, форма

3. Embedding
вектор признаков, который построила сеть

4. Similarity search
ищем близкие векторы и проверяем глазами

5. Confusion Matrix
смотрим, какие ошибки делает модель



Проверяем понимание маршрута, а не запоминание слов.

Совет от преподавателя

Главная фраза урока

Картинка становится данными только после того, как мы выбрали способ её представить.

Что дальше?

Как превратить урок в собственный маленький проект

1. Собрать папки изображений — каждая папка может стать классом

2. Import Images → Image Embedding — получить таблицу признаков

3. Проверить похожие изображения — Neighbors и визуальная проверка

4. Построить классификацию — Test & Score → Confusion Matrix

5. Следующая большая тема — preprocessing и честные pipelines

Источники и опора урока

**старые слайды Игоря Клейнера:
занятие 7.1–7.5, 2020**

ImageNet и AlexNet как историческая
точка перехода к deep learning vision

Orange Data Mining: Import Images, Image
Viewer, Image Embedding, Image Grid

Godec et al. 2019: image analytics through
visual programming

ХОРОШИЕ
КНИГИ
РАСШИРЯЮТ
ГОРИЗОНТЫ

Читать
значит
думать
дальше

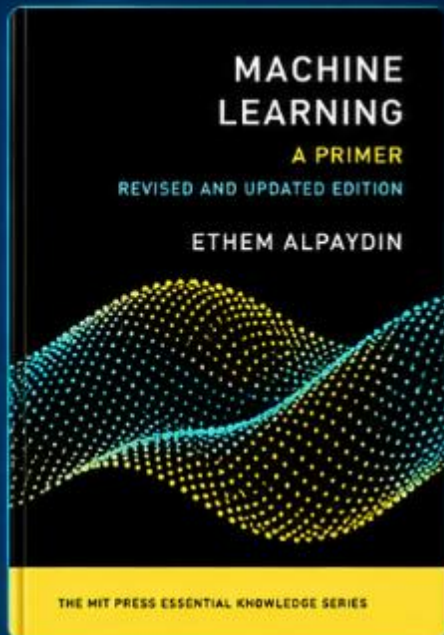
Уголок библиофила

MIT Press Essential Knowledge

Небольшие
книжки —
большие
идеи

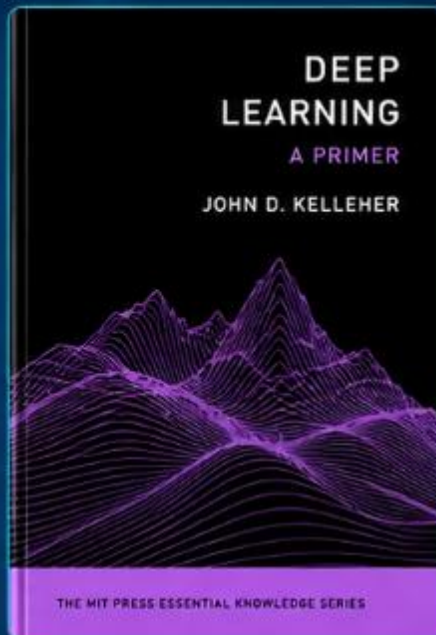
УЧИТЬСЯ
ИССЛЕДОВАТЬ
СОЗДАВАТЬ
БУДУЩЕЕ

1 Machine Learning



Хороший вход
в машинное
обучение:
базовые идеи,
алгоритмы,
примеры и место
нейронных сетей
в общей картине.

2 Deep Learning



Доступное
введение
в современное
deep learning:
архитектуры,
backpropagation,
примеры
и ключевые
применения.

3 О серии



The MIT Press
Essential Knowledge

Серия MIT Press Essential Knowledge — компактные, понятные книги, которые объясняют сложные темы без лишней воды.



Актуальные
темы



Кратко
и по делу



Надёжные
авторы

Лучшие
алгоритмы
начинаются
с хороших книг



Почему это подходит нашему курсу?
Книжки короткие, понятные и помогают
связать практику с общей картиной.

БОЛЬШЕ
ЗНАНИЙ
—
БОЛЬШЕ
ВОЗМОЖНОСТЕЙ

ХОРОШИЕ
КНИГИ
РАСШИРЯЮТ
ГОРИЗОНТЫ

Читать
значит
думать
дальше

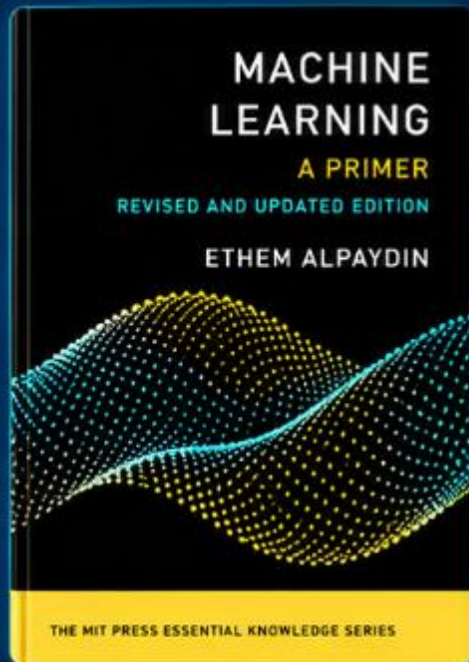
Уголок библиофила

Machine Learning — о чём эта книга?

Больше
знаний
больше
возможностей

УЧИСЬ
ИССЛЕДУЙ
РАЗВИВАЙСЯ
СОЗДАВАЙ

1 Что внутри



- что такое машинное обучение
- основные типы задач
- базовые алгоритмы
- как данные превращаются в модель
- где в общей картине находятся нейронные сети

2 Почему это полезно



даёт понятную
общую карту темы



объясняет идеи
без лишней
математики



помогает связать
теорию с практикой
курса



хорошо подходит
перед более глубоким
разговором
о нейросетях

3 Для кого



для студентов,
которые только
входят в тему



для тех, кто хочет
увидеть всю картину
целиком



для тех, кому нужен
мостик от данных
к моделям

Лучшие
идеи
начинаются
с хороших книг



Хорошая книга помогает увидеть лес,
а не только отдельные деревья.

БОЛЬШЕ
КНИГ
БОЛЬШЕ
ИДЕЙ

ХОРОШИЕ
КНИГИ
РАСШИРЯЮТ
ГОРИЗОНТЫ

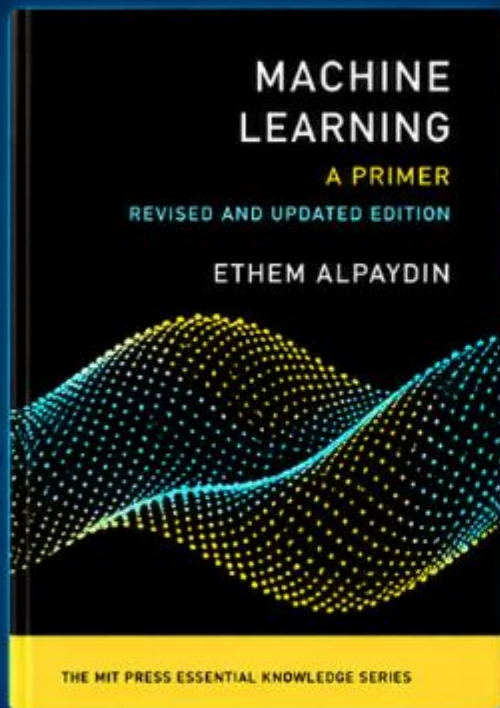
Читать
значит
думать
дальше

Уголок библиофила

Machine Learning — чем книга полезна студентам?

Небольшие
книжки —
большие
идеи

УЧИТЬСЯ
ИССЛЕДОВАТЬ
СОЗДАВАТЬ
БУДУЩЕЕ



Классика, которая
структурирует мысли

Лучшие
алгоритмы
начинаются
с хороших книг

1 Для нашего курса



Книга помогает собрать **общую карту** машинного обучения: задачи, модели, обучение, оценка качества.

2 Особенно ценно



Она **даёт интуицию**. После неё легче понимать, зачем нужны признаки, train/test, регуляризация и выбор модели.

3 Связь с нейросетями



Нейронные сети здесь показаны как часть большого семейства методов. Это полезно: студенты видят **не магию, а место метода в общей системе**.

4 Как читать



Можно читать выборочно: сначала обзор и базовые идеи, потом главы, которые поддерживают темы наших уроков.



Совет от Игоря

Если тема кажется новой и большой, **начните с общей карты**. Потом детали ложатся гораздо легче.



Сначала общая картина, потом **специализация**.

ЗНАНИЯ
СЕГОДНЯ
СВЕТЛОЕ
ЗАВТРА

ХОРОШИЕ
КНИГИ
РАСШИРЯЮТ
ГОРИЗОНТЫ

Читать
значит
думать
дальше

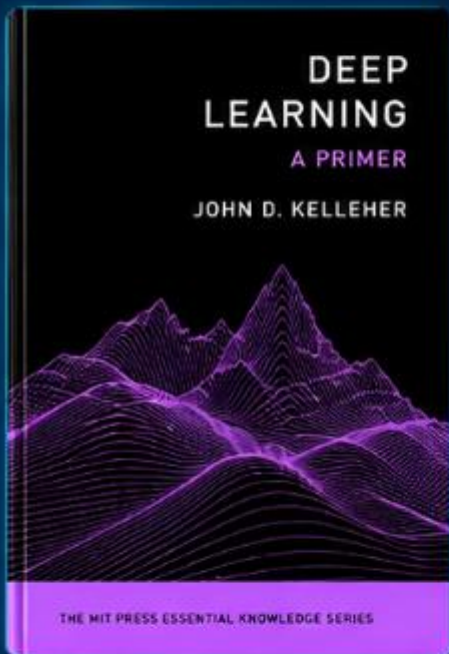
Уголок библиофила

Deep Learning – о чём эта книга?

Небольшие
книжки –
большие
идеи

УЧИСЬ
ИССЛЕДУЙ
СОЗДАВАЙ
БУДУЩЕЕ

1 Ключевая идея



Deep learning
учит модели
самим находить
сложные
представления
в данных.

Особенно хорошо
это работает
для изображений,
текста и звука.

2 Что разбирается



архитектуры
нейросетей



градиентный спуск



backpropagation



важные примеры
применения



сильные стороны
и ограничения

3 Почему книга удобна



короткая
и обзорная



понятна как первое
системное знакомство



не теряется связь
с реальными задачами

Лучшие
алгоритмы
начинаются
с хороших книг



Не только что умеют нейросети,
но и почему они работают.

БОЛЬШЕ
ЗНАНИЙ
—
БОЛЬШЕ
ВОЗМОЖНОСТЕЙ

ХОРОШИЕ
КНИГИ
РАСШИРЯЮТ
ГОРИЗОНТЫ

Читать
значит
думать
дальше

Уголок библиофила

Deep Learning — чем книга полезна студентам курса?

Небольшие
книги —
большие
идеи

УЧИСЬ
ИССЛЕДУЙ
СОЗДАВАЙ
БУДУЩЕЕ

DEEP LEARNING

A PRIMER

JOHN D. KELLEHER

THE MIT PRESS ESSENTIAL KNOWLEDGE SERIES

Небольшая книга
— большие идеи

Лучшие
алгоритмы
начинаются
с хороших книг

1

После наших уроков



Книга помогает перейти от интуитивного знакомства с нейросетями к более **цельному пониманию**: что происходит внутри модели и как она учится.

2

Что станет понятнее



Почему **нужны слои**, как **рождаются признаки** внутри сети, зачем нужен **backpropagation** и почему **большое количество данных** так важно.

3

Для практики



После этой книги легче обсуждать **computer vision**, **embeddings**, **transfer learning** и современные AI-системы без ощущения чёрного ящика.

4

Ограничения и здравый смысл



Важно, что книга не только хвалит deep learning, но и помогает **увидеть трудности, ограничения и реальные вызовы**.



Игорь

Совет от Игоря

Читайте эту книгу **как навигатор**: не обязательно запоминать всё сразу, главное — **понять логику**.

Понимание
важнее
заучивания



Deep learning становится ближе, когда видна его логика.

БОЛЬШЕ
ЗНАНИЙ
—
БОЛЬШЕ
ВОЗМОЖНОСТЕЙ