СӘТБАЕВ УНИВЕРСИТЕТІ



SATBAYEV UNIVERSITY

Основы искусственного интеллекта. Лекция 12. Глубокое обучение для компьютерного зрения. Сверточные нейронные сети (CNN), введение в библиотеки TensorFlow/Keras для CV

Преподаватель: **Қасенхан Арай Мейрамбайқызы, доктор PhD, ассоциированный профессор кафедры «Программная инженерия»**

a.kassenkhan@satbayev.university

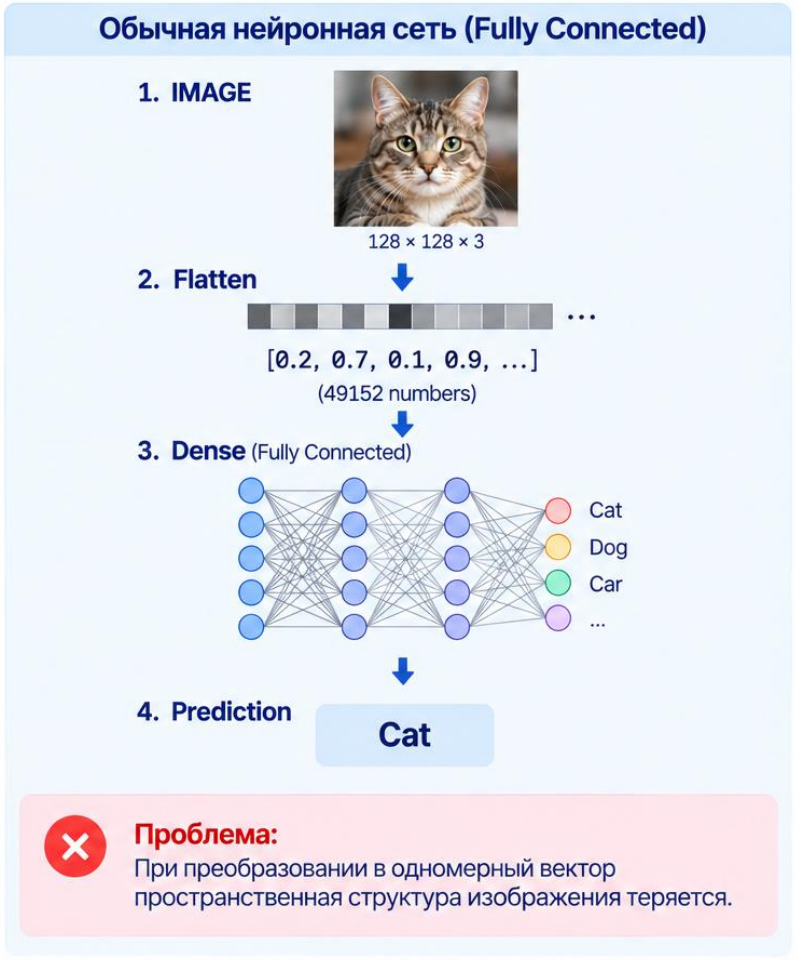
Содержание

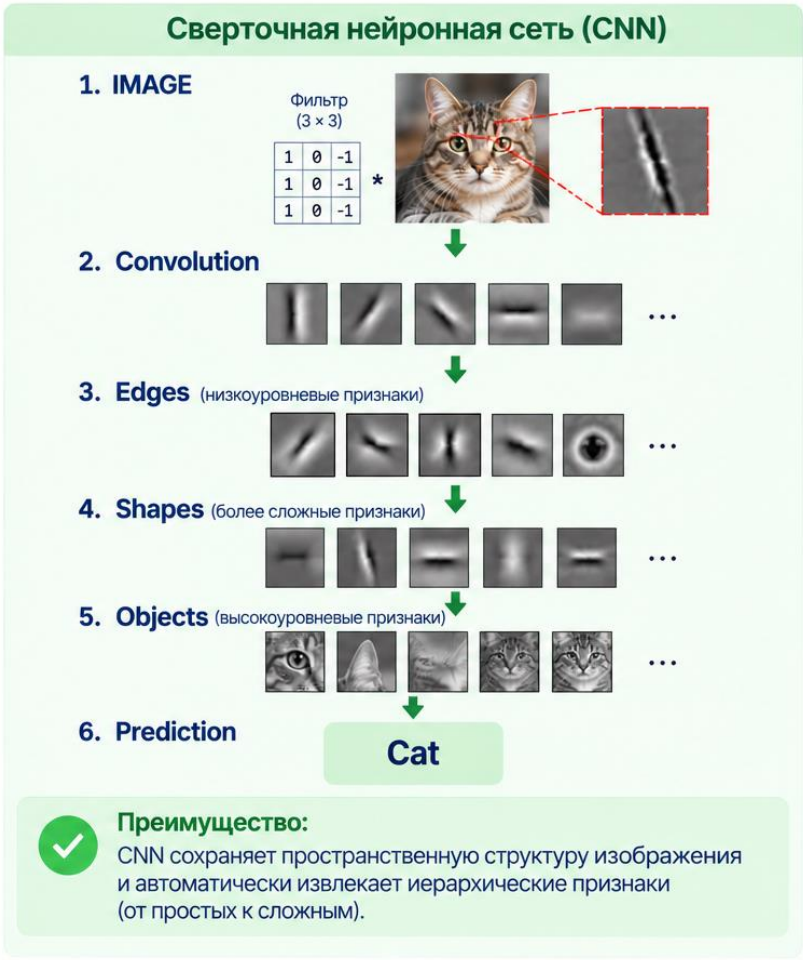
1. Зачем CNN для изображений
2. Как CNN анализирует изображение
3. Convolution
4. Feature Maps
5. Pooling
6. Архитектура CNN
7. CIFAR-10
8. TensorFlow / Keras
9. Создание CNN
10. Обучение
11. Оценка модели
12. Data Augmentation

По завершению урока Вы будете знать:

1. **Объяснить принцип работы CNN;**
2. **Понимать назначение свёртки и фильтров;**
3. **Объяснить роль pooling;**
4. **Понимать архитектуру CNN;**
5. **Создать простую CNN в Keras;**
6. **Обучить её на изображениях;**
7. **Оценить качество модели;**
8. **Понимать назначение Data Augmentation.**

Почему для изображений нужны CNN?



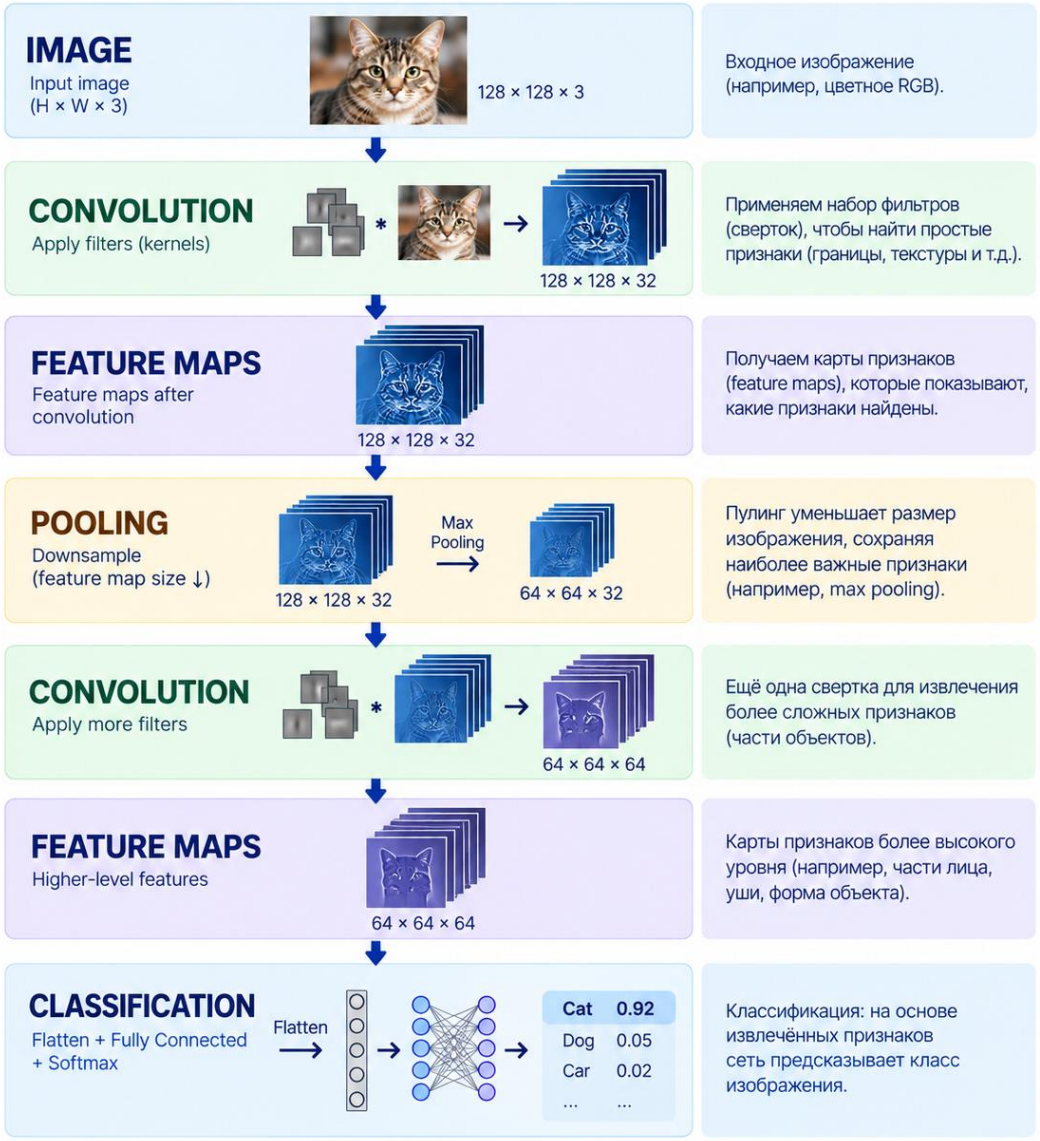


CNN учитывает пространственную структуру изображения и автоматически изучает визуальные признаки.

Что такое CNN?

CNN (Convolutional Neural Network) — нейронная сеть, специально разработанная для обработки изображений и других данных с пространственной структурой.

CNN автоматически учится находить признаки изображения.



Как CNN изучает изображение?

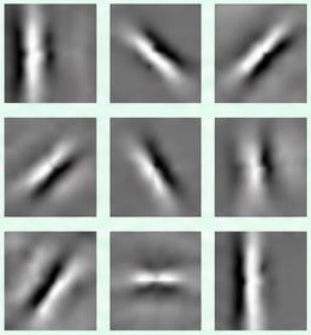
IMAGE
Входное изображение



128 × 128 × 3
(RGB)

Пиксели
(цвета и яркость)

Early Layers
Простые признаки



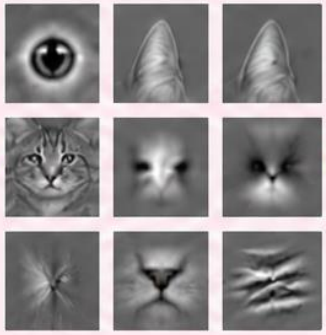
EDGES
Края, линии

Middle Layers
Более сложные признаки



SHAPES / TEXTURES
Формы, текстуры

Deep Layers
Высокоуровневые признаки



OBJECT PARTS
Части объекта
(глаза, уши, нос, усы и т.д.)

OBJECT
Распознанный объект



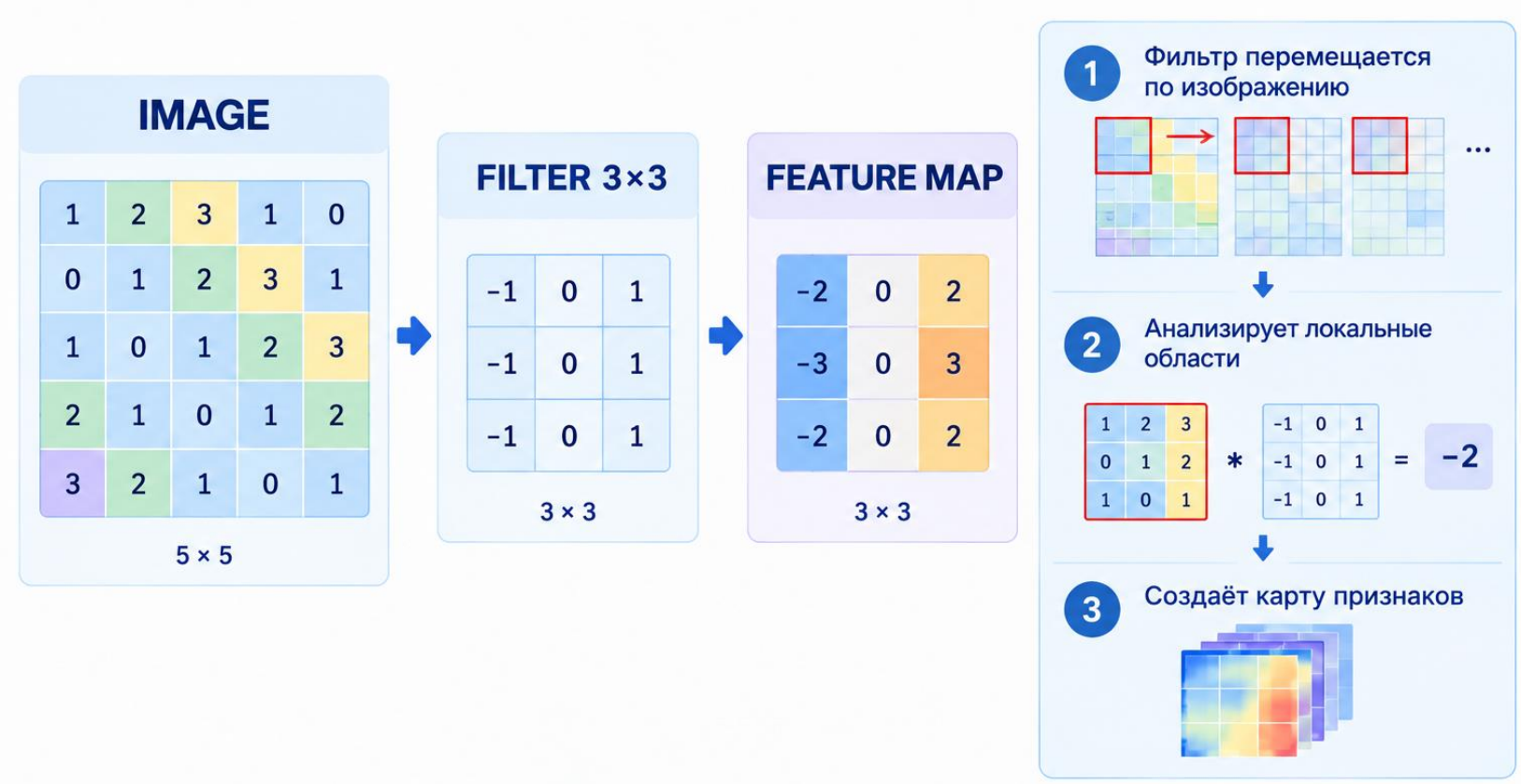
Cat
(класс: кот)

Простые признаки
(низкий уровень)

Сложные признаки
(высокий уровень)

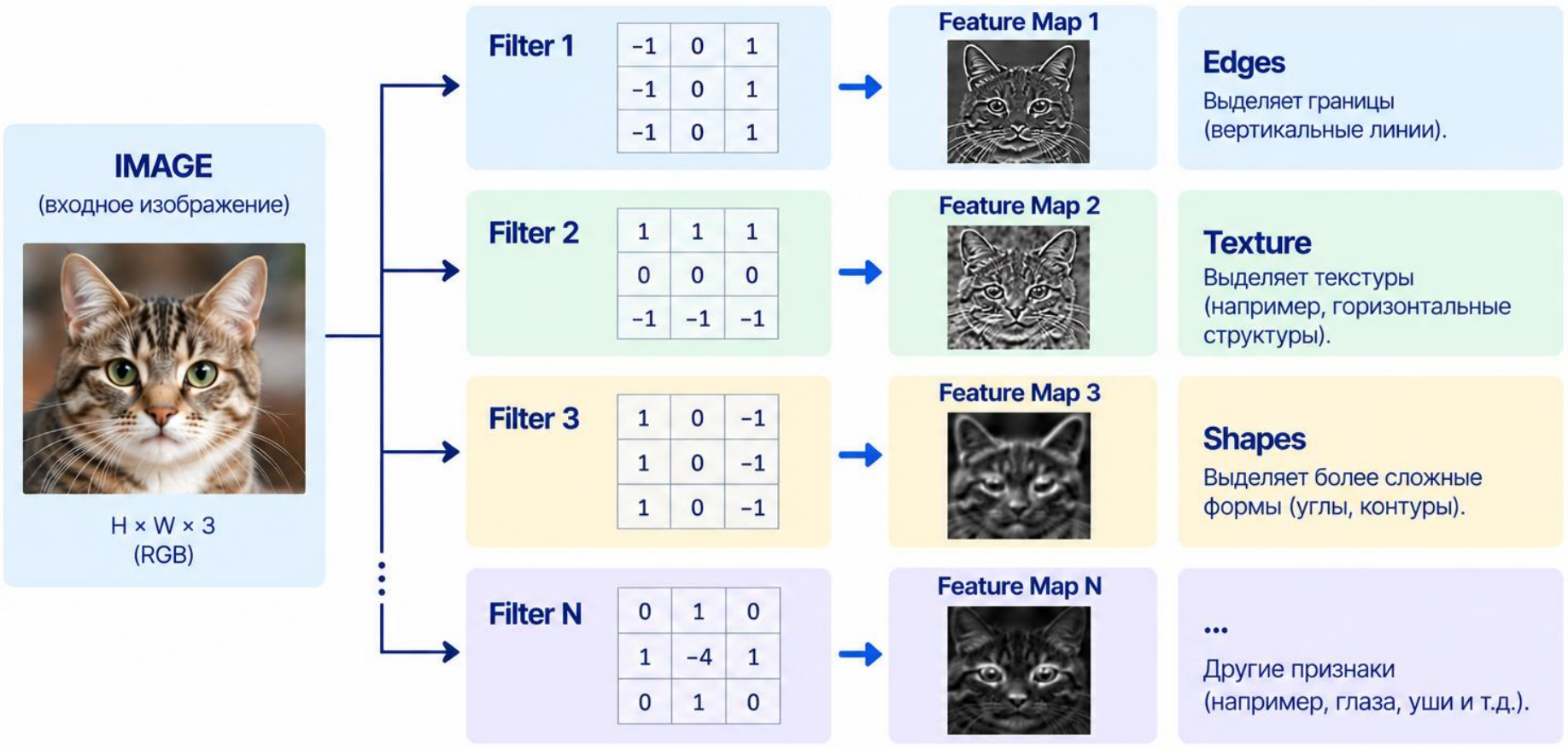
Чем глубже слой, тем более сложные признаки может представлять сеть.

Что такое свёртка?



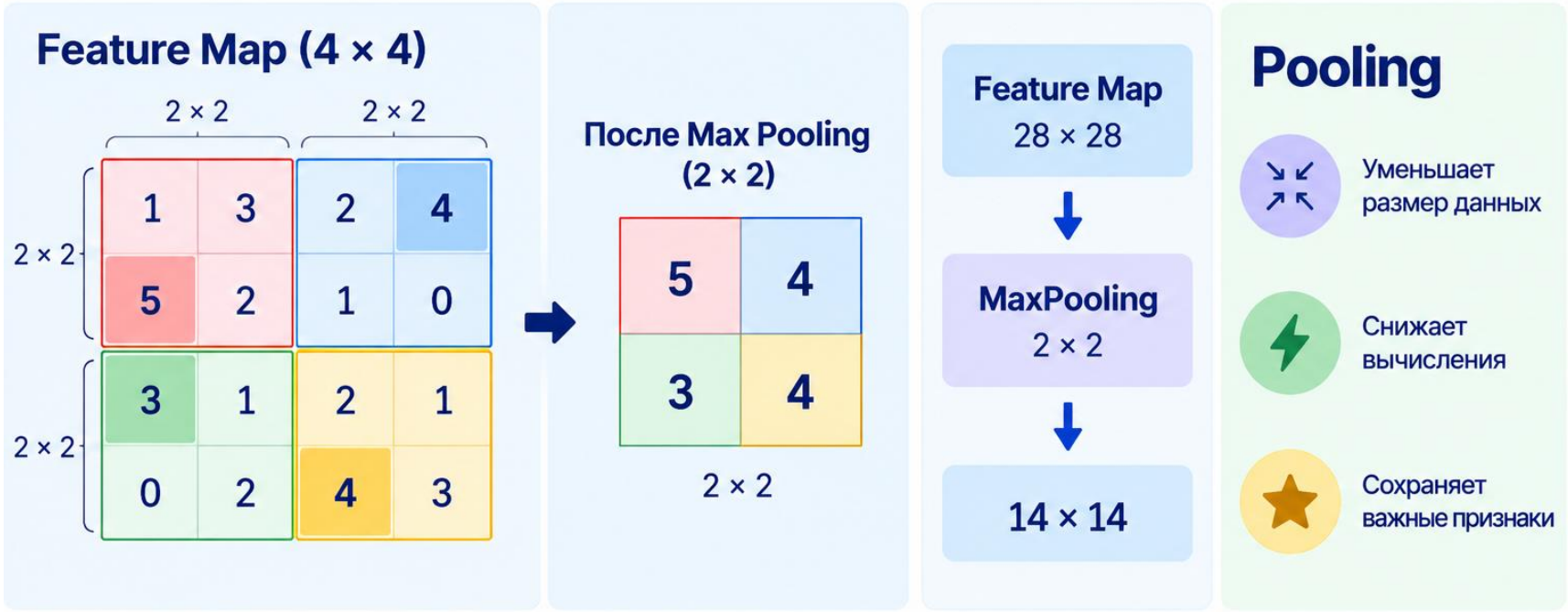
Convolution = поиск локальных признаков изображения.

Filters u Feature Maps

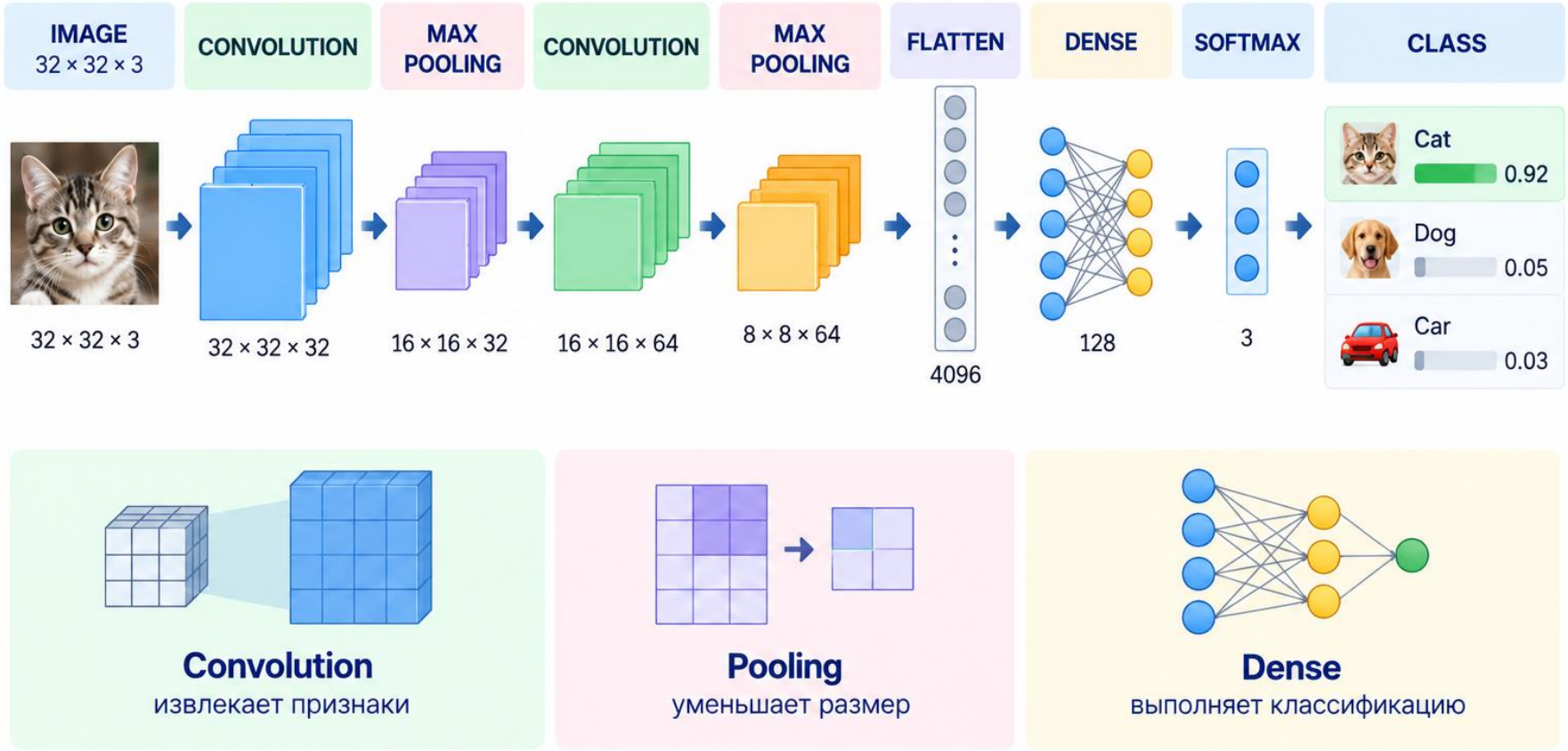


Во время обучения CNN сама настраивает значения фильтров.

Pooling


Feature Map (4 × 4)
2 × 2
2 × 2
2 × 2
2 × 2
1 3 2 4
5 2 1 0
3 1 2 1
0 2 4 3
После Max Pooling (2 × 2)
5 4
3 4
2 × 2
Feature Map
28 × 28
MaxPooling
2 × 2
14 × 14
Pooling
Уменьшает размер данных
Снижает вычисления
Сохраняет важные признаки

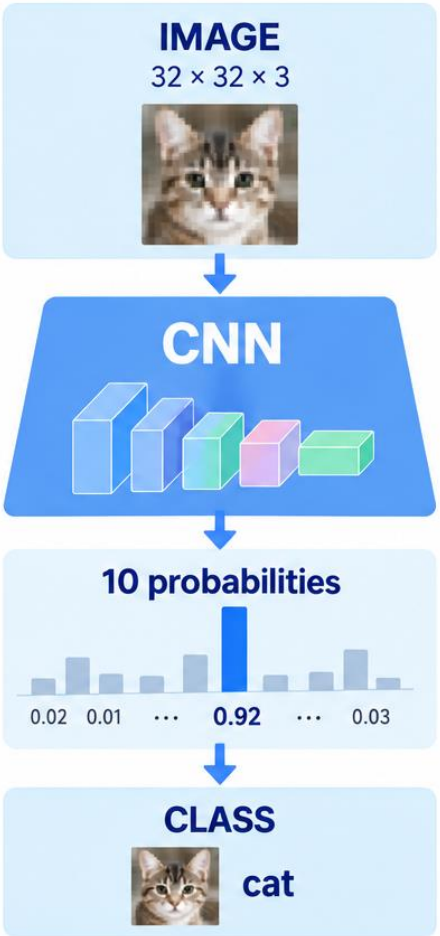
Архитектура CNN



Практическая задача: CIFAR-10

Обучить CNN определять класс изображения.







Загружаем CIFAR-10

```
import tensorflow as tf
from tensorflow import keras
from tensorflow.keras import layers

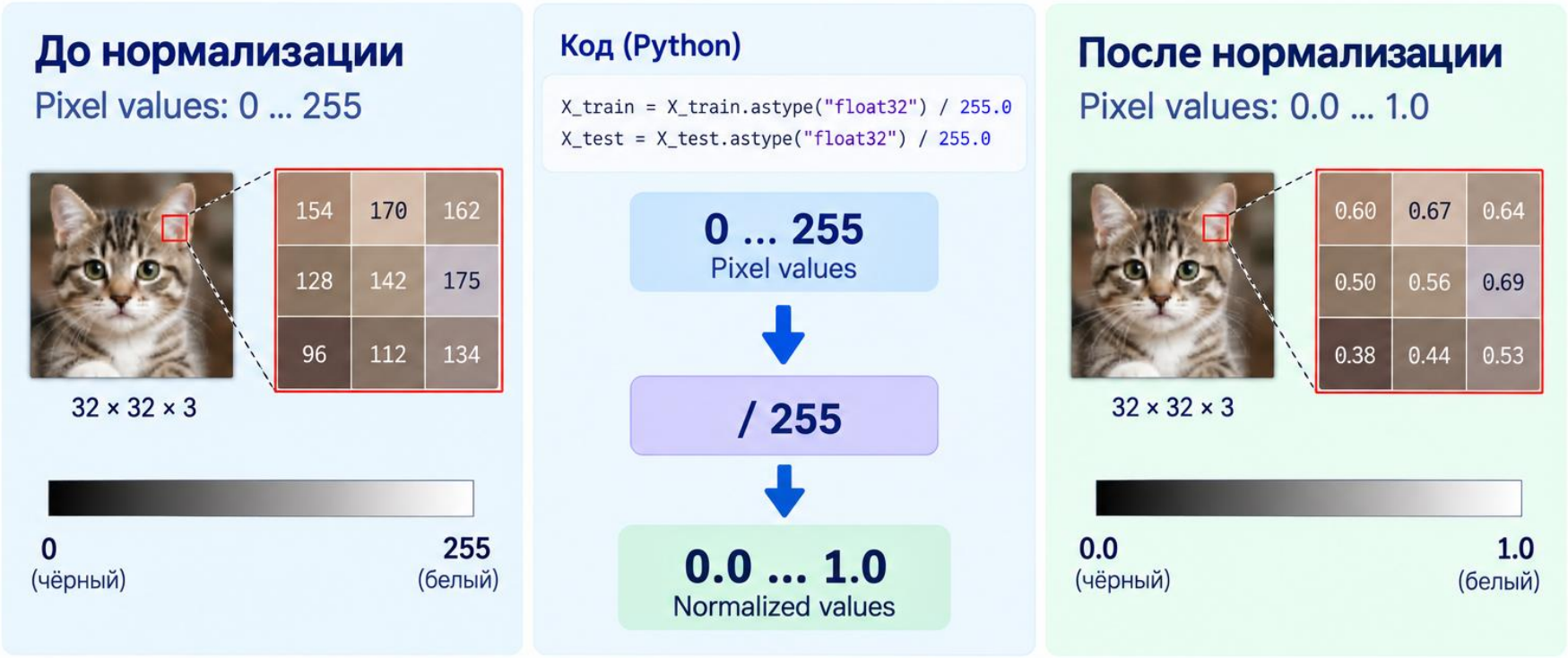
(X_train, y_train), (X_test, y_test) = \
    keras.datasets.cifar10.load_data()

print(X_train.shape)
print(X_test.shape)
```

```
(50000, 32, 32, 3)
(10000, 32, 32, 3)
```



Нормализация изображений



Нормализация помогает сделать обучение нейронной сети более стабильным.

Создаём CNN

```
model = keras.Sequential([
    keras.Input(shape=(32, 32, 3)),

    layers.Conv2D(
        32, (3, 3),
        activation="relu"
    ),
    layers.MaxPooling2D((2, 2)),

    layers.Conv2D(
        64, (3, 3),
        activation="relu"
    ),
    layers.MaxPooling2D((2, 2)),

    layers.Flatten(),
    layers.Dense(64, activation="relu"),
    layers.Dense(10, activation="softmax")
])
```



Compile: как будет обучаться модель?

```
model.compile(
    optimizer="adam",
    loss="sparse_categorical_crossentropy",
    metrics=["accuracy"]
)
```

Optimizer

Adam

Как обновлять веса?

Adam использует адаптивную скорость обучения для эффективного обновления весов.

Loss

Sparse Categorical Crossentropy

Насколько модель ошибается?

Измеряет разницу между предсказанным распределением вероятностей и истинным классом (для целочисленных меток).

Metric

Accuracy

Какую долю изображений классифицировали правильно?

Показывает процент верных предсказаний на данных.

Процесс обучения (один шаг)



Image → Model (CNN) → Prediction (0.1 0.2 0.6 0.1) Predicted probabilities → Loss: Compute error → Backpropagation: Compute gradients → Update weights: Adjust parameters

Repeat for many batches / epochs

Обучаем и оцениваем CNN

```
history = model.fit(
    X_train,
    y_train,
    epochs=10,
    validation_split=0.2
)

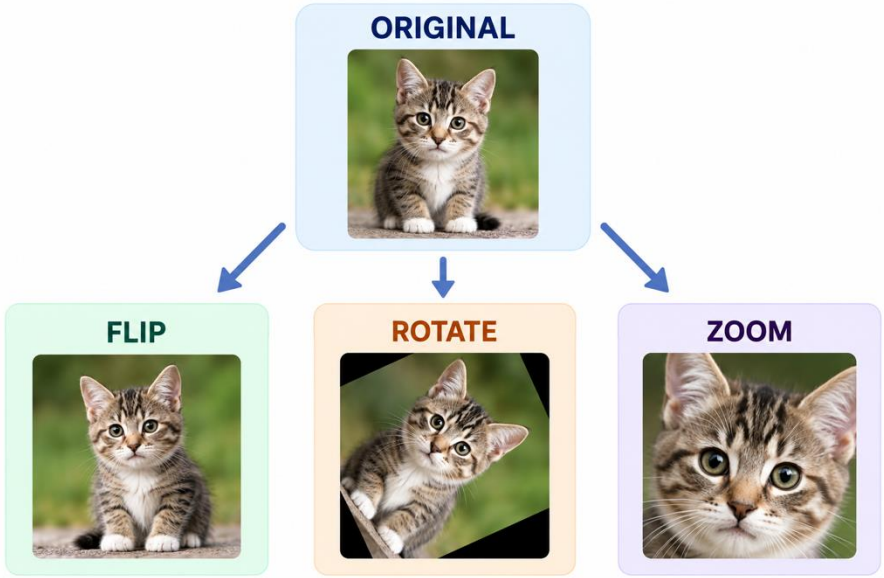
# После обучения:
test_loss, test_acc = model.evaluate(
    X_test,
    y_test
)

print("Test accuracy:", test_acc)
```

Epoch
один полный проход по обучающим данным
Validation
проверка качества во время обучения
Test
финальная проверка на данных, которые модель не использовала для обучения

Как улучшить модель? Data Augmentation

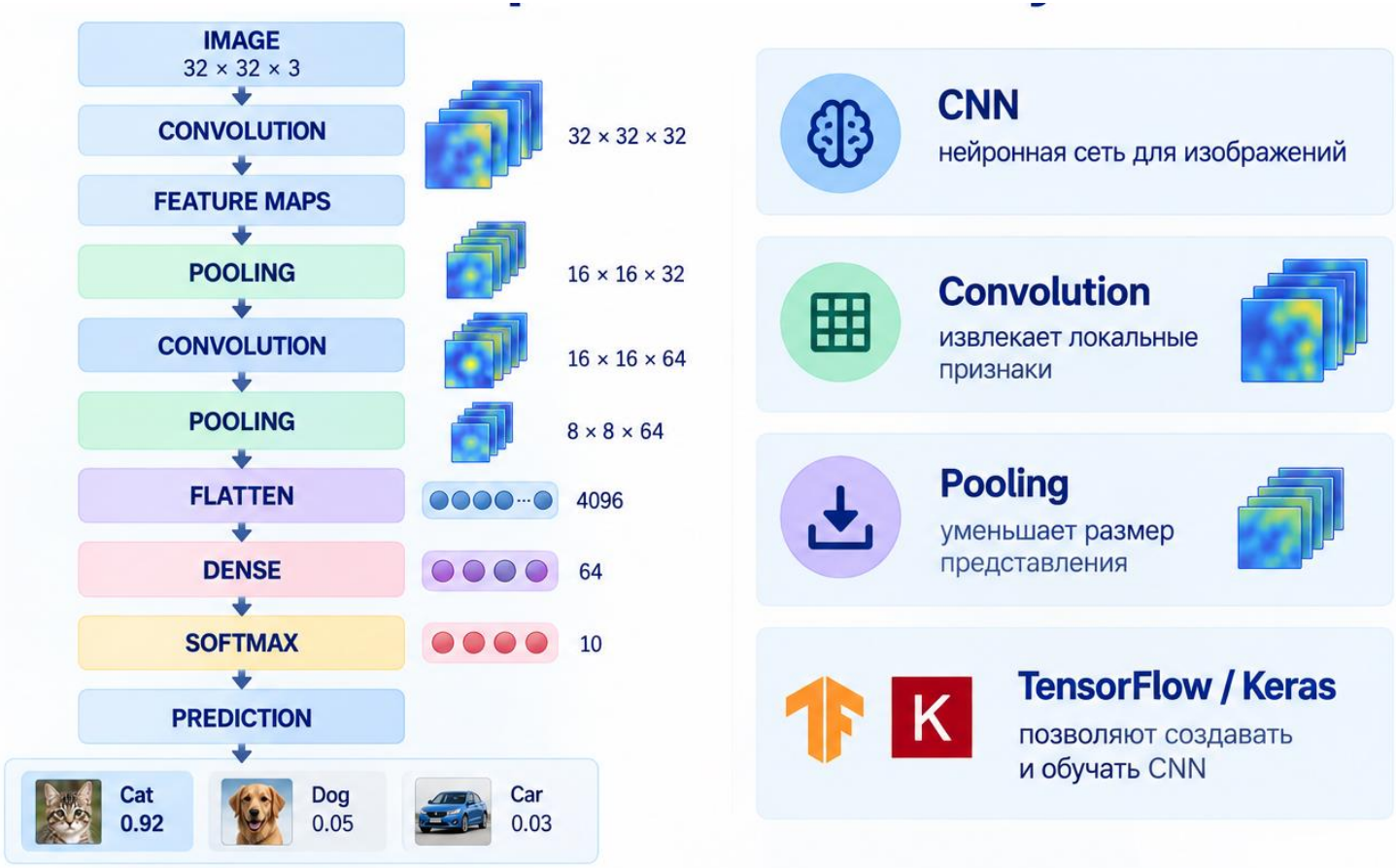
```
data_augmentation = keras.Sequential([
    layers.RandomFlip("horizontal"),
    layers.RandomRotation(0.1),
    layers.RandomZoom(0.1)
])
```



Одно изображение → несколько немного разных вариантов.

Data Augmentation помогает модели лучше обобщать и снижает риск переобучения.

Заключение



CNN учится переходить от пикселей к признакам, а от признаков — к распознаванию объекта.