

# Проект по МО

**Приложил титанические усилия для выполнение проекта: Беляев Андрей Евгеньевич**

## Постановка задачи

### Цель проекта

Распознать изображение с картинки и определить к какому из 10 классов она относится.

### Вход

Изображение с одним из 10-ти возможных объекта

### Выход

Имя класса, к которому объект относится

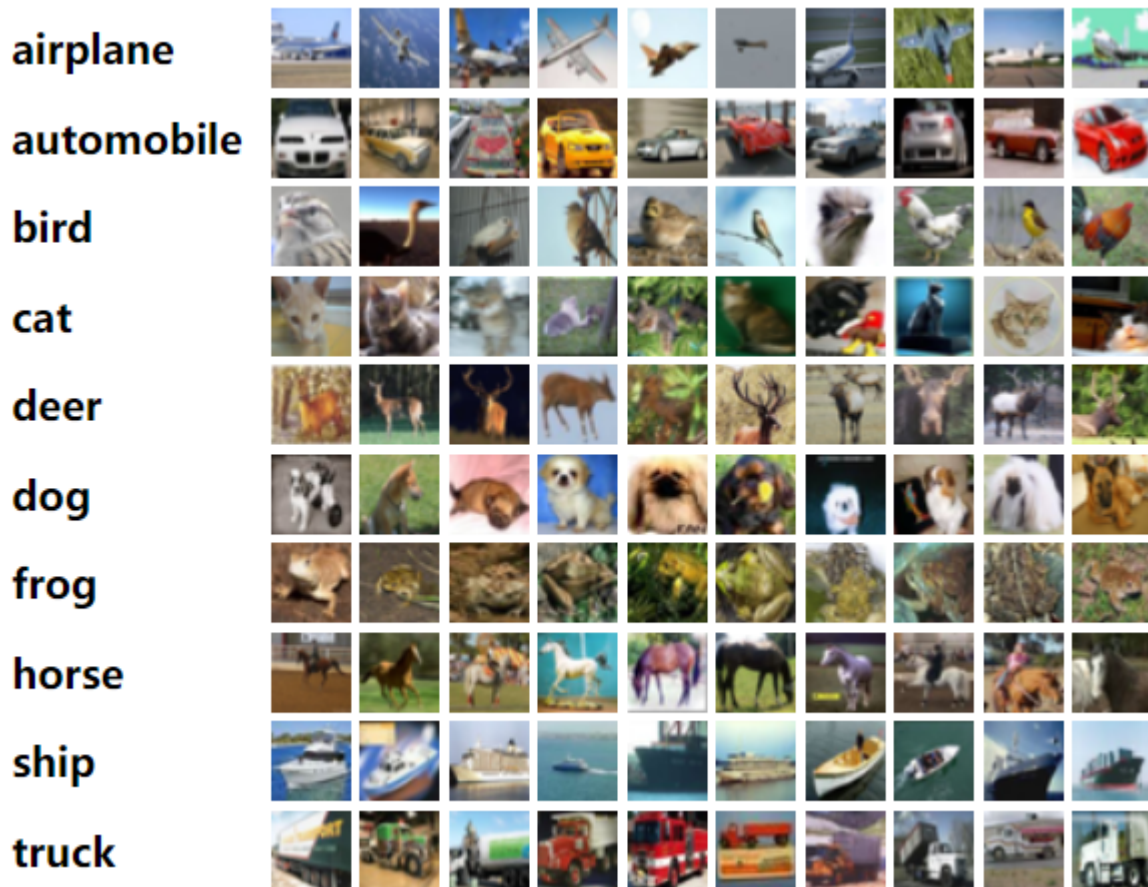
## Общая информация

Распознавание изображения относится к задаче ввода изображения в нейронную сеть и присвоения какой-либо метки для этого изображения. Метка, которую выводит сеть, будет соответствовать заранее определенному классу. Может быть присвоено как сразу несколько классов, так и только один. Если существует всего только один класс, обычно применяется термин «распознавание», тогда как задача распознавания нескольких классов часто называется «классификацией».

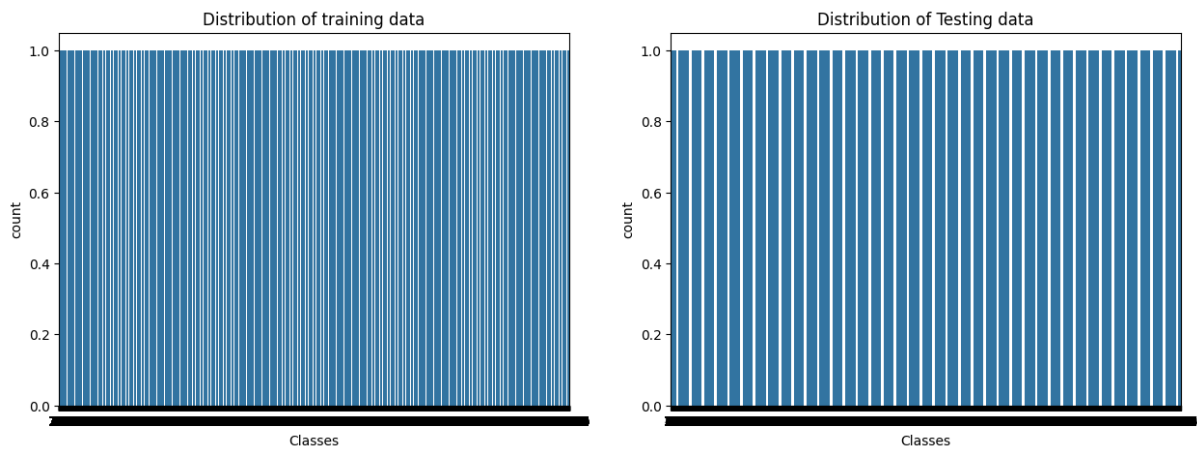
Чтобы выполнить распознавание или классификацию изображений, нейронная сеть должна выполнить извлечение признаков. Признаки - это элементы данных, которые представляют максимальный интерес и которые будут передаваться по нейросети. В конкретном случае распознавания изображений такими признаками являются группы пикселей, такие как линии и точки, которые сеть будет анализировать на наличие паттерна.

# Информация о датасете

Набор данных CIFAR-10 содержит 60 000 цветных изображений размером 32 x 32 пикселя в 3 каналах, разделенных на 10 классов. Каждый класс содержит 6000 изображений. Обучающий набор содержит 50 000 изображений, а тестовый набор — 10 000 изображений.



Задача состоит в том, чтобы распознать ранее невиданные изображения и отнести их к одному из 10 классов.



По результатам небольшого исследования мы видим, что данные распределены равномерно по 10ти классам.

```

Downloading data from https://www.cs.toronto.edu/~kriz/cifar-10-python.tar.gz
170498071/170498071 [=====] - 5s 0us/step
x_train shape: (50000, 32, 32, 3)
y_train shape: (50000, 1)
50000 train samples
10000 test samples

```

Каждый класс содержит ровно 6000 примеров (5000 для обучения и 1000 для тестирования).

## Модель модели 😊

1. Создается модель типа Sequential, который представляет собой линейный стек слоев.
2. Певый блок
  - Добавляется слой Conv2D с 32 фильтрами размером (3, 3) и 'same'.
  - Применяется функция активации ReLU.
  - Добавляется еще один слой Conv2D с 32 фильтрами размером (3, 3).
  - Применяется функция активации ReLU.
  - Добавляется слой MaxPooling2D с пулом размером (2, 2) для уменьшения размерности.

- Добавляется слой Dropout с коэффициентом 0.25 для предотвращения переобучения.

### 3. Второй блок

- Аналогично первому блоку, но с увеличенным количеством фильтров (64)

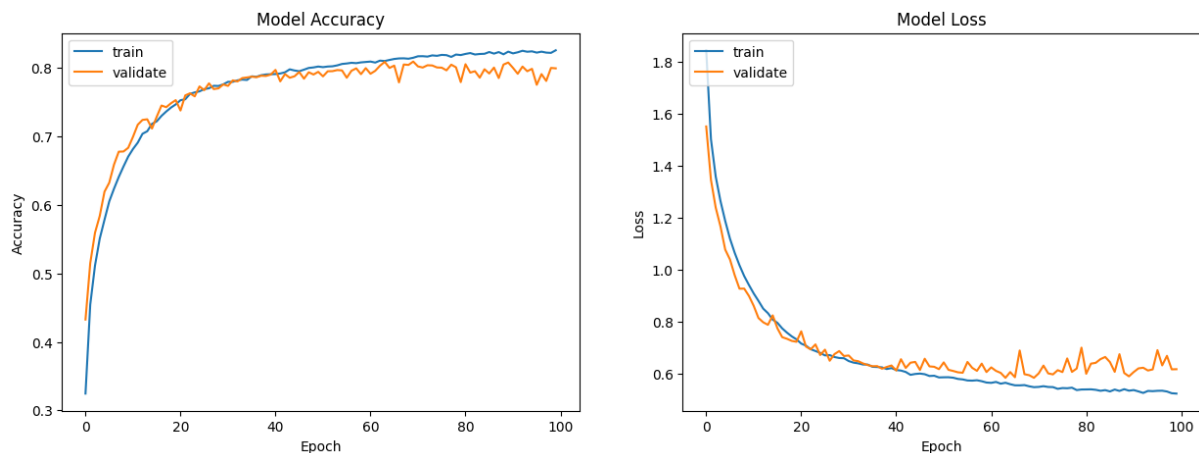
### 4. FLATTEN $\Rightarrow$ DENSE $\Rightarrow$ RELU $\Rightarrow$ DROPOUT

- Выполняется операция Flatten для преобразования выхода сверточных слоев в одномерный вектор.
- Применяется функция активации ReLU.
- Добавляется полносвязный слой Dense с 512 нейронами.
- Добавляется слой Dropout с коэффициентом 0.5.

### 5. Softmax классификатор

- Добавляется полносвязный слой Dense с количеством нейронов, равным числу классов (num\_classes).
- Применяется функция активации Softmax для получения вероятностного распределения по классам.

## Исследования

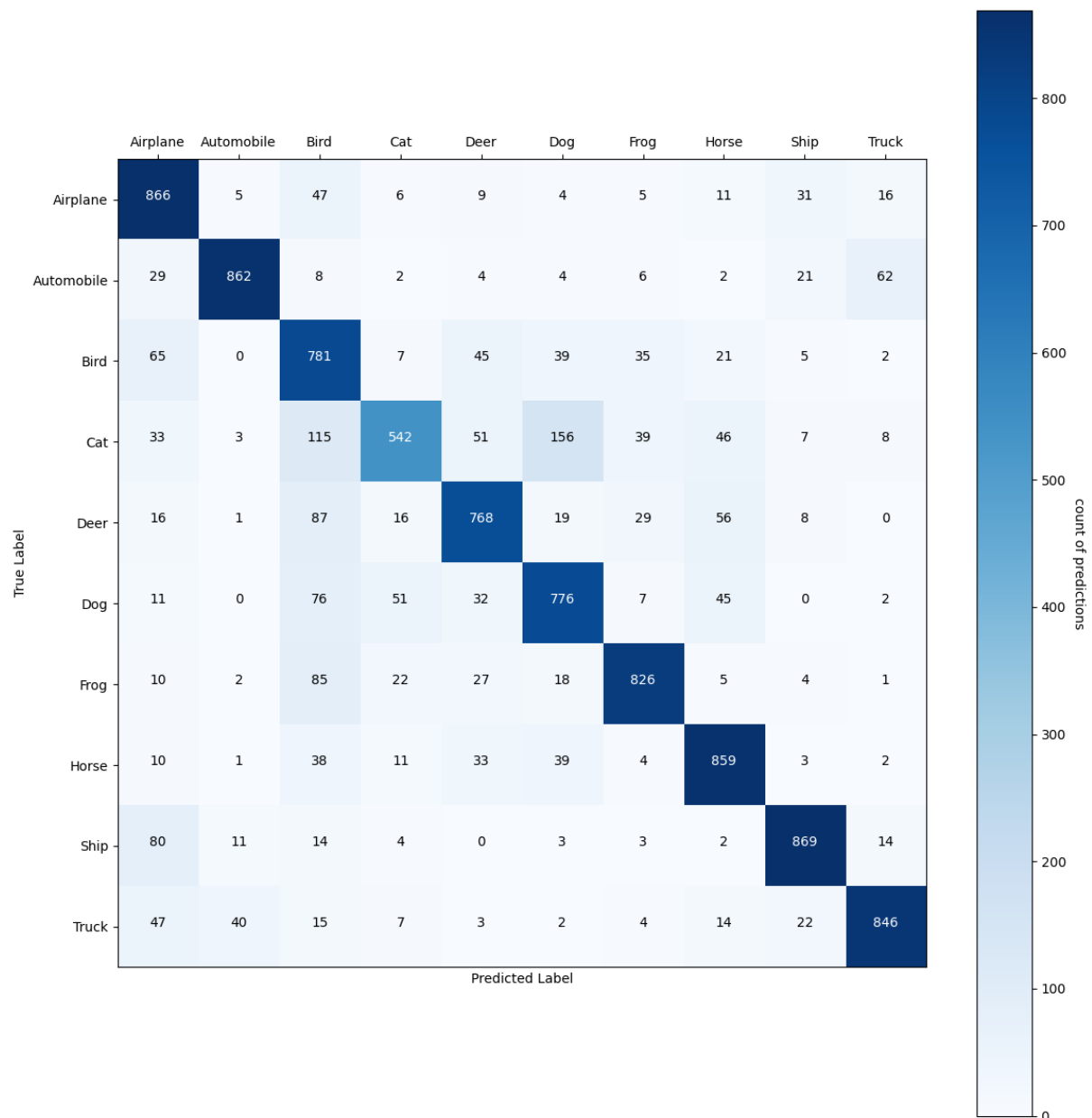


Исходя из графика точности и графика потерь делаем вывод, что модель обучается успешно.

После 60 эпох точность нашей модели особо не увеличивается. Но наша модель не переобучается.

Давайте посмотрим в чем заключаются ошибки. Грубо говоря имеется две гипотезы, что точка роста находится в датасете/в построении модели.

Составим матрицу соответствия!



По вертикали обозначены реальный классы, к которым относятся фотографии. По горизонтали предсказание модели, при поступлении на вход такой же фотографии.

	precision	recall	f1-score	support
0	0.74	0.87	0.80	1000
1	0.93	0.86	0.90	1000
2	0.62	0.78	0.69	1000
3	0.81	0.54	0.65	1000
4	0.79	0.77	0.78	1000
5	0.73	0.78	0.75	1000
6	0.86	0.83	0.84	1000
7	0.81	0.86	0.83	1000
8	0.90	0.87	0.88	1000
9	0.89	0.85	0.87	1000
accuracy			0.80	10000
macro avg	0.81	0.80	0.80	10000
weighted avg	0.81	0.80	0.80	10000

Здесь мы видим 4 метрики для каждого класса:

- Точность (Precision): доля верно предсказанных положительных наблюдений от общего числа предсказанных положительных наблюдений.
- Полнота (Recall): доля верно предсказанных положительных наблюдений от общего числа истинных положительных наблюдений.
- F1-мера (F1-score): гармоническое среднее точности и полноты.
- Поддержка (Support): количество появлений каждого класса в истинных метках.

Из этого исследования мы видим, что кошки являются самыми проблемными для определения.

Давайте посмотрим примеры предиктов.

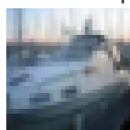
True: Cat  
Predict: Cat



True: Ship  
Predict: Ship



True: Ship  
Predict: Ship



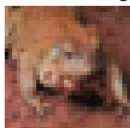
True: Airplane  
Predict: Airplane



True: Frog  
Predict: Frog



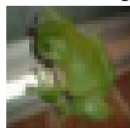
True: Frog  
Predict: Frog



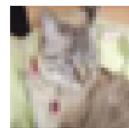
True: Automobile  
Predict: Automobile



True: Frog  
Predict: Frog



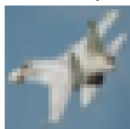
True: Cat  
Predict: Cat



True: Automobile  
Predict: Automobile



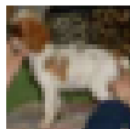
True: Airplane  
Predict: Airplane



True: Truck  
Predict: Truck



True: Dog  
Predict: Horse



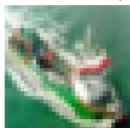
True: Horse  
Predict: Horse



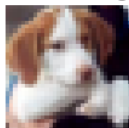
True: Truck  
Predict: Truck



True: Ship  
Predict: Ship



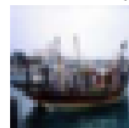
True: Dog  
Predict: Dog



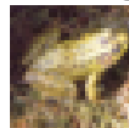
True: Horse  
Predict: Horse



True: Ship  
Predict: Ship



True: Frog  
Predict: Frog



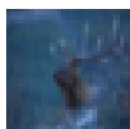
True: Horse  
Predict: Horse



True: Airplane  
Predict: Airplane



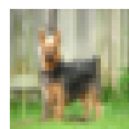
True: Deer  
Predict: Deer



True: Truck  
Predict: Truck

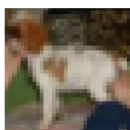


True: Dog  
Predict: Deer

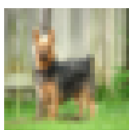


Почти все предсказания являются правильными. Давайте посмотрим в каких предиктах была допущена ошибка. Для этого выведем несколько ошибочных предиктов!

True: Dog  
Predicted: Horse



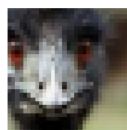
True: Dog  
Predicted: Deer



True: Deer  
Predicted: Bird



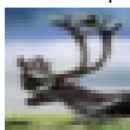
True: Bird  
Predicted: Cat



True: Automobile  
Predicted: Truck



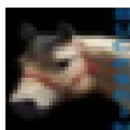
True: Deer  
Predicted: Airplane



True: Airplane  
Predicted: Horse



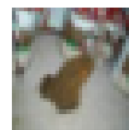
True: Horse  
Predicted: Cat



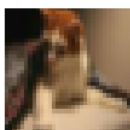
True: Deer  
Predicted: Dog



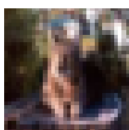
True: Frog  
Predicted: Dog



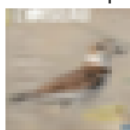
True: Cat  
Predicted: Dog



True: Cat  
Predicted: Dog



True: Bird  
Predicted: Airplane



True: Truck  
Predicted: Airplane



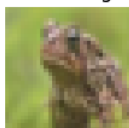
True: Cat  
Predicted: Dog



Сложно определить какой-то один фактор из-за которого может быть ошибка. Возможно основной проблемой является не естественный фон для класса и необычный ракурс.

Далее предлагается исследовать 10 наиболее далеких от истинны на языке вероятности модели.

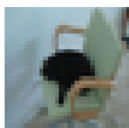
Predicted: Bird  
True: Frog



Predicted: Frog  
True: Cat



Predicted: Bird  
True: Cat



Predicted: Ship  
True: Automobile



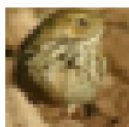
Predicted: Automobile  
True: Truck



Predicted: Horse  
True: Bird



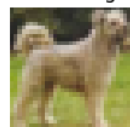
Predicted: Frog  
True: Bird



Predicted: Airplane  
True: Cat



Predicted: Horse  
True: Dog



Predicted: Deer  
True: Cat



Тут, честно говоря, для меня не все фотографии являются легко различимыми.

Также на этих фотографиях мы можем убедиться в правильности гипотез по входным данным.



## Ссылки

<https://github.com/Jedykqoo/ml>