

*Стеценко Алексей Иванович, студент магистратуры, Карагандинский*

*Технический Университет имени Абылкаса Сагинова, МОиАДМ-21,*

*e-mail: [mrgreen348@gmail.com](mailto:mrgreen348@gmail.com)*

*Смагулова Асемгуль Сериковна, доцент, к.т.н., Карагандинский Технический*

*Университет имени Абылкаса Сагинова, ФИТ,*

*e-mail: [assemgul\\_work@mail.ru](mailto:assemgul_work@mail.ru)*

## **РАЗРАБОТКА НЕЙРОННОЙ СЕТИ ДЛЯ РАСПОЗНАВАНИЯ ИЗОБРАЖЕНИЙ НА БАЗЕ TENSORFLOW**

**Аннотация:** Цель статьи заключается в разработке нейронной сети для распознавания изображений на базе фреймворка TensorFlow. В статье рассмотрены основные понятия необходимые для работы с нейросетями. Особое внимание уделяется построению архитектуры нейросети и визуализации предсказаний.

**Ключевые слова:** Классификация изображений, анализ нейросетей, архитектура нейросетей, свёрточные нейронные сети, свёрточные слои.

**Annotation:** The purpose of the article is to develop a neural network for studying images based on the TensorFlow framework. The article discusses the basic concepts for working with neural networks. Particular attention is paid to the architecture of neural networks and premature predictions.

**Keywords:** Image classification, neural network analysis, neural network architecture, convolutional neural networks, convolutional layers.

### **Введение**

В последнее время прослеживается рост популярности устройств дополненной реальности, умного дома и беспилотных транспортных средств.

Данный рост вызвал острую потребность в алгоритмах семантической сегментации (или визуального понимания сцены), которые могут работать в режиме реального времени на мобильных устройствах с низким энергопотреблением. В последние годы доступность больших наборов, данных и вычислительно мощных машин помогла глубоким сверточным нейронным сетям превзойти производительность многих обычных алгоритмов компьютерного зрения [1]. Сверточные нейронные сети (CNN) стали наиболее мощным методом классификации изображений и сегментации изображений. CNN можно постоянно обучать и улучшать по мере увеличения количества аннотированных обучающих выборок. Целью нашей работы являлась разработка нейронной сети для распознавания изображений на базе фреймворка TensorFlow.

### **Разработка нейронной сети**

Для разработки нейросети на языке Python требуются несколько библиотек, включая NumPy, Matplotlib и Keras. Процесс создания нейросети начинается с подготовки данных, для чего мы используем библиотеку TensorFlow. С помощью TensorFlow мы загружаем данные и разделяем их на обучающую и тестовую выборки. Изображения использованы из датасета Fashion MNIST, который содержит около 70 000 изображений различных типов одежды в разрешении 28 на 28 пикселей. В данном наборе данных представлены 10 типов вещей. Примеры изображений одежды из датасета можно увидеть на рисунке 1.

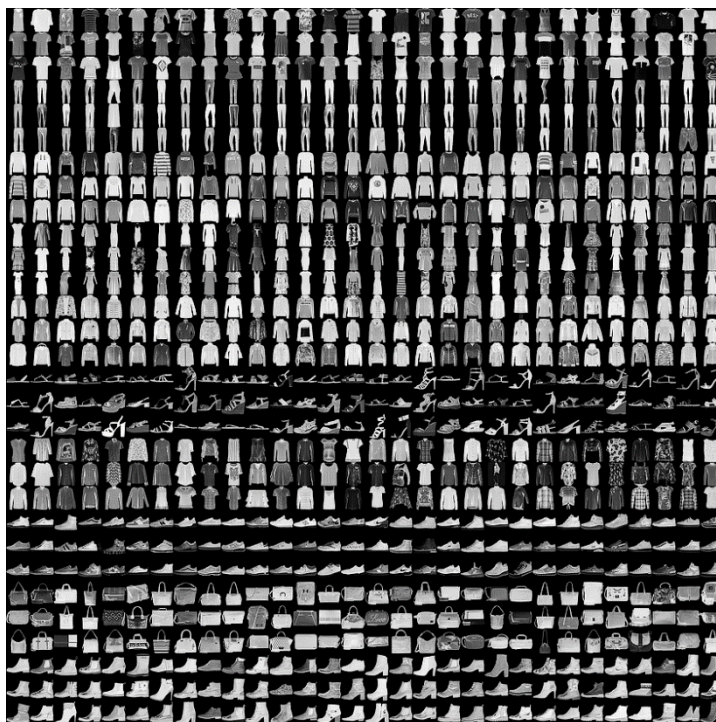


Рис. 1. Пример изображений из датасета

Изображения, которые содержатся в датасете, будут распределены на две группы. В первую группу будет включено 60 000 изображений, которые будут использованы для обучения нейросети. Оставшиеся изображения будут использованы для проверки точности распознавания. Для работы с набором данных в библиотеке TensorFlow необходимо импортировать данные в проект при помощи команды `load_data`. После загрузки данных будут созданы массивы: `train_images`, `train_labels`, `test_images`, `test_labels`. Массивы `train_images` и `train_labels` будут использованы для обучения модели нейросети, в то время как другие массивы будут использоваться для оценки точности распознавания.

Каждый из массивов, представляет собой набор изображений, представленных в виде значений пикселей, которые находятся в диапазоне от 0 до 255. Каждая категория одежды имеет соответствующую метку, которая состоит из массива целых чисел в диапазоне от 0 до 9. Так как в наборе данных не включены имена меток их необходимо прописать вручную. Имена меток отображены в таблице 1.

Таблица 1 - Имена меток

Метка	Класс
0	T-shirt (Футболка)
1	Trouser (Брюки)
2	Pullover (Свитер)
3	Dress (Платье)
4	Coat (Пальто)
5	Sandal (Сандали)
6	Shirt (Рубашка)
7	Sneaker (Кроссовки)
8	Bag (Сумка)
9	Ankle boot (Ботильоны)

Перед построением нейросети необходимо выполнить предварительную обработку данных. С помощью деления на 255 мы масштабируем значения пикселей в диапазон от 0 до 1. Настройка слоев нейросети также является важным этапом при разработке нейронной сети. Этот процесс включает в себя объединение простых слоев, что является основной частью глубокого обучения. Большая часть слоев, содержат параметры, которые изучаются в процессе обучения [2]. На рисунке 2 приведен код, описывающий настройку слоев нейросети, который необходим для достижения оптимальной производительности сети.

```
#настройка слоев модели
model = keras.Sequential([
    keras.layers.Flatten(input_shape=(28, 28)),
    keras.layers.Dense(128, activation=tf.nn.relu),
    keras.layers.Dense(10, activation=tf.nn.softmax)
])
```

Рис. 2. Код настройки слоев нейросети

Один из слоев модели, известный как flatten-слой, изменяет тип данных из двумерного массива на одномерный с 784 пикселями. Этот слой не содержит обучаемых параметров, поскольку он просто изменяет формат данных. Следующие слои это 2 dense-слоя. Первый из них состоит из 128 нейронов, а второй 10 узлов, определяющие массив из десяти оценок. Сумма всех оценок равна единице. Путем оценки каждого узла можно определить вероятность

принадлежности изображения к одному из классов объектов.

Обучение модели нейросети проходит через несколько этапов. Сначала данные для обучения загружаются из массивов с изображениями и метками. Затем начинается процесс обучения, где устанавливаются связи между изображениями и соответствующими метками. Далее, на основе тестового набора данных, генерируются прогнозы и проверяется точность меток.

Для начала обучения используется метод `fit`. Во время тренировки, отображаются значения потерь и точности модели, а после завершения этого этапа, модель проходит проверку на тестовом наборе данных, используя метод `evaluate()`. Рисунок 3 показывает заметную разницу между точностью модели на тестовой и тренировочной выборках, что свидетельствует о переобучении. Переобучение — это неблагоприятный эффект, возникающий в процессе обучения алгоритма, когда он формирует прогнозы, слишком точно соответствующие конкретному набору данных, что делает их непригодными для применения на других данных или в будущих условиях [4].

```
313/313 [=====] - 1s 2ms/step - loss: 0.3301 - accuracy: 0.8814  
Test accuracy: 0.8813999891281128
```

Рис. 3. Разница между точностью модели на тестовой и тренировочной выборках

С помощью метода `predict` можно использовать модель для предсказания класса изображений в тестовом наборе. Как видно на рисунке 4, нейронная сеть успешно определяет метку для каждого изображения в данном тестовом наборе.

```
313/313 [=====] - 0s 2ms/step  
array([1.1001471e-06, 4.3868590e-07, 9.2458885e-09, 8.2666429e-10,  
1.4754295e-07, 3.1831979e-03, 1.6810850e-07, 1.1802188e-02,  
3.6634569e-07, 9.8501241e-01], dtype=float32)
```

Рис. 4. Процесс создания предсказания

Данная модель выражает свою «уверенность» в том, что каждый образец соответствует одному из 10 типов одежды. Для этого используется массив из 10 чисел, где наибольшее значение соответствует классу «Ботильоны». Чтобы

наглядно представить соответствие между предсказаниями модели и классом изображения, необходима специальная функция. Визуализация осуществляется при помощи синего цвета для правильных меток и красного цвета для ошибочных меток прогноза. На рисунке 5 продемонстрированы примеры использования предсказаний для разных классов одежды.

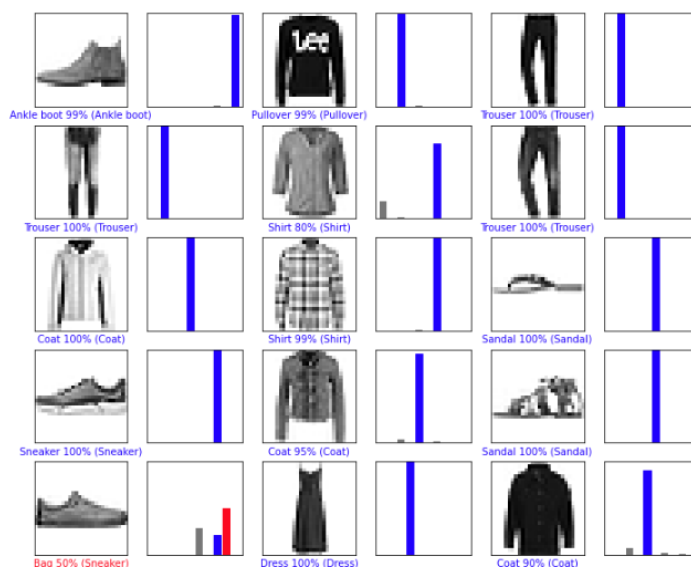


Рис. 5. Визуализация предсказаний нейросети для разных классов одежды

## Заключение

В данной работе был рассмотрен процесс обучения и тестирования искусственной нейронной сети на основе TensorFlow для распознавания изображений. Исходным набором данных был выбран Fashion MNIST. По результатам исследования была получена точность модели около 0,88 (или 88%), что свидетельствует о ее высокой эффективности. Таким образом, данное исследование может быть полезным для разработки систем распознавания изображений на базе искусственных нейронных сетей.

## Библиографический список:

1. Паттанаяк С. Глубокое обучение и TensorFlow для профессионалов – М.: Диалектика, 2019. – 480 с.
2. Аггарвал Ч. Нейронные сети и глубокое обучение: учебный курс –

М.: Диалектика-Вильямс, 2020. 752 с.

3. Омеляненко Я. Эволюционные нейросети на языке Python – М.: ДМК Пресс, 2020. – 310 с.

4. Потапов А. Автоматический анализ изображений и распознавание образов – М.: LAP Lambert Academic Publishing, 2017. – 292 с.