

**МИНИСТЕРСТВО ОБРАЗОВАНИЯ И НАУКИ РОССИЙСКОЙ ФЕДЕРАЦИИ**  
**Федеральное государственное автономное образовательное учреждение высшего образования**  
**«Южно-Уральский государственный университет (национальный исследовательский университет)»**  
**Институт естественных и точных наук**  
**Кафедра прикладной математики и программирования**

**Отчет по лабораторной работе № 3**  
**Разработка сверточной нейронной сети**

**по дисциплине**

**«Современные нейросетевые технологии»**

**ЮУрГУ - 01.04.02. 2025. 306/010. Р**

**Руководитель, преподаватель**

\_\_\_\_\_/ Д.М. Кичеев /

« \_\_\_\_ » мая 2025 г.

**Автор**

**студент группы ИЕТН - ЕТ-122**

\_\_\_\_\_/ О.В. Ростова /

« \_\_\_\_ » мая 2025 г.

**Челябинск 2025**

**Цель** настоящей работы состоит в том, чтобы получить базовые навыки работы с одной из библиотек глубокого обучения (Caffe, Torch, TensorFlow или MXNet на выбор) на примере сверточных нейронных сетей.

Выполнение практической работы предполагает решение следующих **задач**:

1. Выбор библиотеки для выполнения практических работ курса.
2. Установка выбранной библиотеки на кластере.
3. Проверка корректности установки библиотеки. Разработка и запуск тестового примера сети, соответствующей логистической регрессии, для решения задачи классификации рукописных цифр набора данных MNIST (пример разобран в лекционных материалах).
4. Выбор практической задачи компьютерного зрения для выполнения практических работ.
5. Разработка программ/скриптов для подготовки тренировочных и тестовых данных в формате, который обрабатывается выбранной библиотекой.
6. Разработка нескольких архитектур полностью связанных нейронных сетей (варьируются количество слоев и виды функций активации на каждом слое) в формате, который принимается выбранной библиотекой.
7. Обучение разработанных глубоких моделей.
8. Тестирование обученных глубоких моделей.
9. Сделать вывод относительно разработанных архитектур.
10. Сделать краткое сравнение скорости и эффективности разработанных сетей относительно полностью связанных нейронных сетей.
11. Подготовка отчета, содержащего минимальный объем информации по каждому этапу выполнения работы.

Выполнение лабораторной работы осуществлено мною в Google Colab на языке программирования Python. Результаты и отчет размещены в личном профиле на GitHub, ссылка: [https://github.com/Olga-1911/Development\\_of\\_a\\_convolutional\\_neural\\_network](https://github.com/Olga-1911/Development_of_a_convolutional_neural_network).

В процессе выполнения лабораторной работы, мною были выполнены следующие шаги:

1. Для выполнения практических работ курса выбрана библиотека TensorFlow с модулем Keras, т.к. TensorFlow уже предустановлен в Google Colab, имеет обширную документацию и поддерживает работу с набором данных MNIST и задачами компьютерного зрения.

Библиотека TensorFlow



Рисунок 1 – Логотип библиотеки TensorFlow

2. TensorFlow версии 2.18.0 предустановлен в Google Colab, установка не потребовалась.

Проверка TensorFlow

Версия TensorFlow: 2.18.0

Рисунок 2 – Визуализация проверки версии TensorFlow

3. Для проверки корректности установки библиотеки, разработан и запущен тестовый пример сети, соответствующей логистической регрессии, для решения задачи классификации рукописных цифр набора данных MNIST с использованием TensorFlow.

Код загружает данные MNIST, нормализует их, создает модель с одним полносвязным слоем и обучает её.

Запуск тестового примера показал, что библиотека установлена корректно, а базовая модель обучается на простом наборе данных.

```

Epoch 1/5
1500/1500 ————— 4s 3ms/step - accuracy: 0.6961 - loss: 1.1990 - val_accuracy: 0.8783 - val_loss: 0.5065
Epoch 2/5
1500/1500 ————— 3s 2ms/step - accuracy: 0.8725 - loss: 0.5089 - val_accuracy: 0.8925 - val_loss: 0.4169
Epoch 3/5
1500/1500 ————— 3s 2ms/step - accuracy: 0.8873 - loss: 0.4334 - val_accuracy: 0.8989 - val_loss: 0.3812
Epoch 4/5
1500/1500 ————— 7s 3ms/step - accuracy: 0.8926 - loss: 0.3989 - val_accuracy: 0.9038 - val_loss: 0.3609
Epoch 5/5
1500/1500 ————— 3s 2ms/step - accuracy: 0.8966 - loss: 0.3780 - val_accuracy: 0.9068 - val_loss: 0.3475
313/313 ————— 1s 2ms/step - accuracy: 0.8910 - loss: 0.3980
Точность на тестовых данных MNIST: 0.9062

```

Рисунок 3 – Результат обучения модели на MNIST

Точность на тестовых данных: 0,9062 это означает, что доля правильно классифицированных изображений на тестовой выборке 90,62%, это говорит о хорошей способности модели распознавать цифры MNIST.

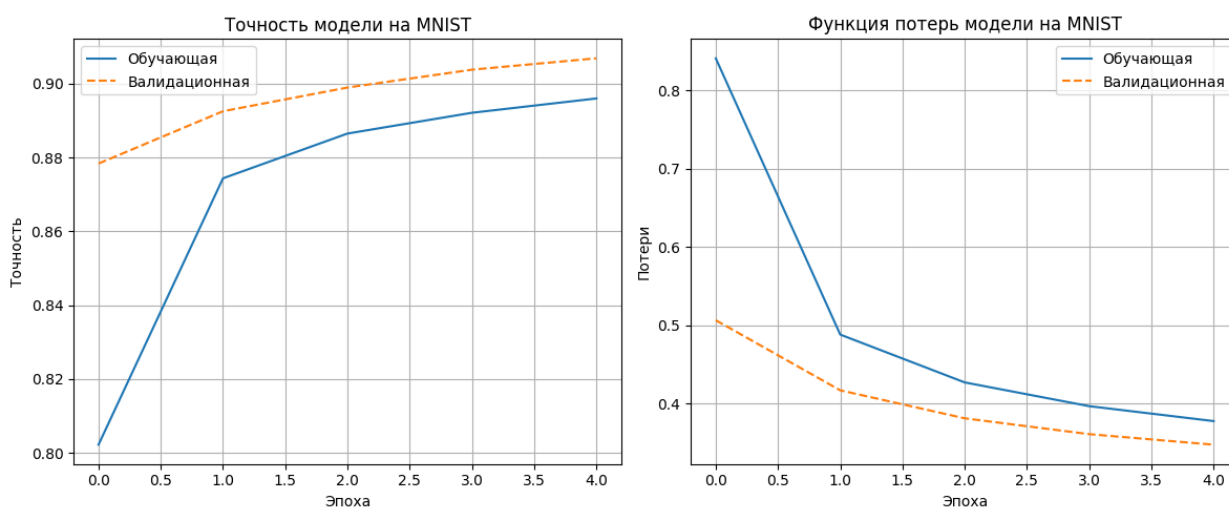


Рисунок 4 – Визуализация графиков точности модели и функции потерь на MNIST

На рисунках 3 и 4 мы видим, как с каждой эпохой точность модели возрастает, причем максимальный темп роста происходит на 2 эпохе (точность достигает 87,25%), в 5-й эпохе точность модели на обучающих данных достигает 89,10%. С функцией потерь – обратная ситуация, во второй эпохе – резкое падение до 41,69%, к 5-й эпохе – 39,80%. На валидационных данных (выделенных из тренировочной выборки) точность выше (более

90%), чем на обучающей выборке; функция потерь ниже (менее 35%), чем на обучающей выборке.

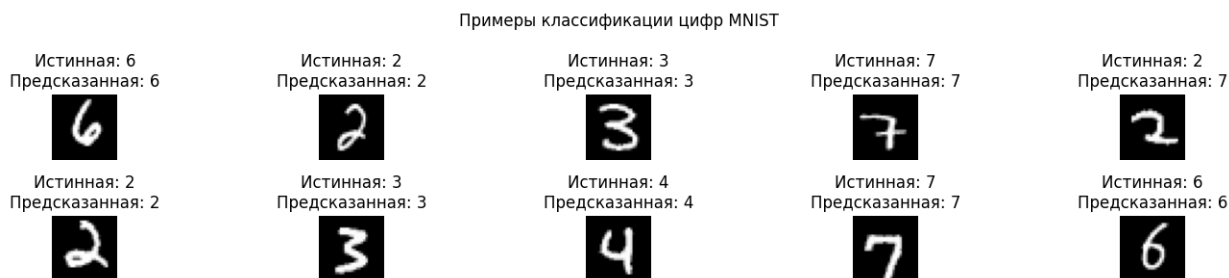


Рисунок 5 – Примеры классификации случайных цифр MNIST

На рисунке 5 представлены примеры классификации случайных цифр MNIST. 9 из 10 цифр предсказаны моделью верно.

4. Для выполнения практических работ я выбрала практическую задачу компьютерного зрения - задачу классификации изображений из набора данных CIFAR-10.

Задача: Многоклассовая классификация изображений CIFAR-10

Набор данных CIFAR-10 (Canadian Institute For Advanced Research) - это коллекция изображений, широко используемых в алгоритмах машинного обучения и компьютерного зрения. Он был разработан исследователями из института CIFAR и состоит из 60 000 цветных изображений размером 32x32 пикселя, разделенных на 10 классов.

Ссылка на набор данных: <https://www.kaggle.com/c/cifar-10>. Также этот набор данных доступен через TensorFlow.

На входе: вектор из 3072 чисел (нормализованные пиксели цветного изображения (RGB) размером 32x32x3).

На выходе: вероятность принадлежности к одному из 10 классов (метка одного из 10 классов: самолет, автомобиль, птица, кошка, олень, собака, лягушка, лошадь, корабль, грузовик).



Рисунок 6 – Примеры изображений CIFAR-10

На рисунке 6 мы видим примеры изображений CIFAR-10.

CIFAR-10 сложнее MNIST, подходит для демонстрации преимуществ CNN.

5. Для подготовки тренировочных и тестовых данных в формате, который обрабатывается выбранной библиотекой, произведено:

- нормализация данных: привела пиксельные значения  $x_{train}$  и  $x_{test}$  к диапазону  $[0, 1]$ ;
- преобразованы метки  $y_{train}$  и  $y_{test}$  в one-hot encoding для 10 классов;
- аугментация: для снижения риска переобучения, она моделирует реальные вариации изображений и повышает точность на тестовых данных;
- выведены размеры выборок: обучающая (50 000: 32, 32, 3), тестовая (10 000: 32, 32, 3);
- построена гистограмма распределения классов в обучающей выборке;
- сгенерированы и визуализированы 5 аугментированных изображений.

6. С целью разработки нескольких архитектур полностью связанных нейронных сетей мною разработаны и настроены 3 архитектуры нейронных сетей для задачи классификации изображений:

- **Модель 1:** полносвязная нейронная сеть (FCNN) с 2 скрытыми слоями: архитектура включает входной слой (Flatten для преобразования

изображений 32x32x3 в вектор), полносвязный слой с 1024 нейронами и активацией ReLU, нормализацию батча, выброс 30% нейронов (Dropout);

- **Модель 2:** полносвязная нейронная сеть (FCNN) с 3 скрытыми слоями: архитектура состоит из входного слоя (Flatten), двух полносвязных слоёв (512 и 128 нейронов с активацией ReLU), каждый из которых сопровождается нормализацией батча и выбросом 30% нейронов;

- **Модель 3:** свёрточная нейронная сеть (CNN) для сравнения с FCNN: включает два блока свёрточных слоёв:

- первый блок: два слоя Conv2D (64 фильтра, ядро 3x3, активация ReLU, padding='same'), нормализация батча, пулинг (MaxPooling2D) и выброс 25%;

- второй блок: два слоя Conv2D (128 фильтров), нормализация батча, пулинг и выброс 25%.

После свёрточных блоков следует полносвязный слой (512 нейронов, ReLU), нормализация батча, выброс 50%.

Все 3 модели имеют выходной слой с 10 нейронами и активацией softmax, модели скомпилированы с оптимизатором Adam, функцией потерь categorical\_crossentropy и метрикой точности (accuracy).

Для каждой модели определена структура, выполнена компиляция и создана визуализация архитектуры.

FCNN подходят для базового анализа, тогда как CNN оптимизирована для обработки изображений благодаря свёрточным слоям. Визуализации подтверждают корректность архитектур.

7. Далее было проведено обучение трёх нейронных сетей для классификации изображений: FCNN с 2 слоями, FCNN с 3 слоями и CNN. Обучение выполнялось в течение 15 эпох с использованием оптимизатора Adam, функции потерь categorical\_crossentropy и метрики точности.

Для FCNN с 2 и 3 слоями обучение проводилось на нормализованных обучающих данных (x\_train, y\_train) с размером батча 64. Для CNN

применялась аугментация данных через ImageDataGenerator для повышения обобщающей способности. Все модели оценивались на тестовой выборке ( $x_{test}$ ,  $y_{test}$ ) после каждой эпохи. Время обучения и точность на тестовых данных сохранены для анализа.

		Время обучения	Тестовая точность
Модель 1	FCNN с 2 слоями	1056,31 сек.	43,07%
Модель 2	FCNN с 3 слоями	861,18 сек.	47,96%
Модель 3	CNN	14881,19 сек.	84,05%

CNN показала наилучшую точность благодаря свёрточным слоям, эффективно извлекающим пространственные признаки. FCNN с 3 слоями превзошла FCNN с 2 слоями за счёт большей глубины, но обе уступают CNN.

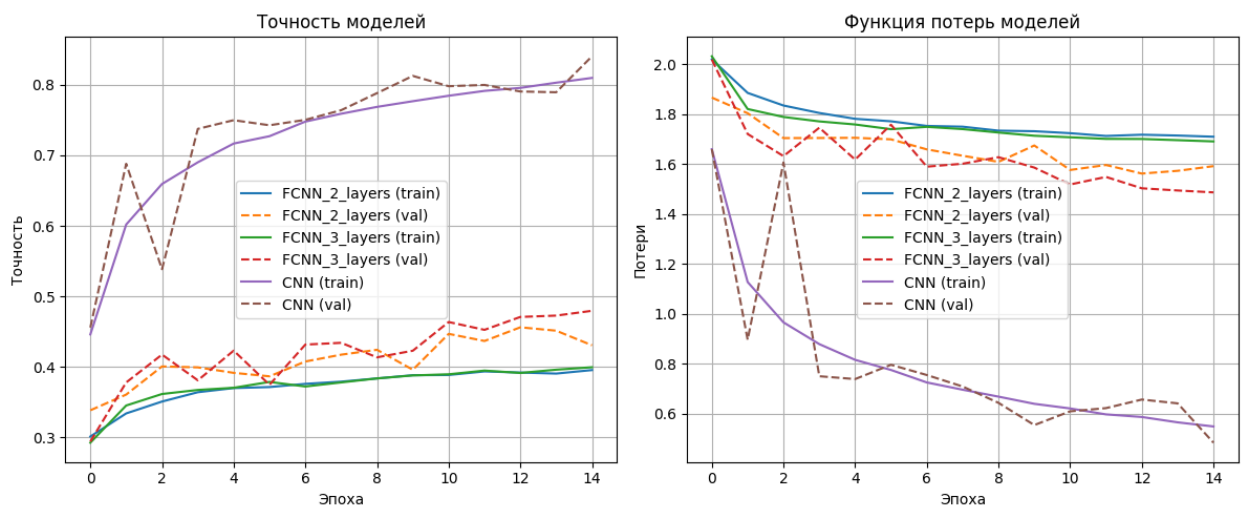


Рисунок 7 – Графики зависимости точности и функции потерь от номера эпохи для обучающей и валидационной выборок

Построены графики зависимости точности и функции потерь от номера эпохи для обучающей и валидационной выборок (рисунок 7). CNN демонстрирует устойчивый рост точности (до 81,27% на обучении, 84,05% на тесте) и снижение потерь (до 0,5416 на обучении, 0,4841 на тесте). FCNN показывают меньшую точность и признаки переобучения (разрыв между тренировочными и валидационными метриками).



CNN значительно превосходит FCNN по точности (84,05% против 43,07% и 47,96%), что подтверждает её пригодность для задач классификации изображений. Длительное время обучения CNN (14881,19 сек.) обусловлено сложностью модели и аугментацией данных. FCNN с 3 слоями показала лучший результат, чем с 2 слоями, но обе модели менее эффективны. Графики метрик иллюстрируют динамику обучения и помогают выявить переобучение у FCNN.

8. Проведено тестирование трёх нейронных сетей (FCNN с 2 слоями, FCNN с 3 слоями, CNN) на тестовой выборке. Выполнен анализ ошибок классификации с помощью матриц ошибок и визуализированы примеры предсказаний моделей.

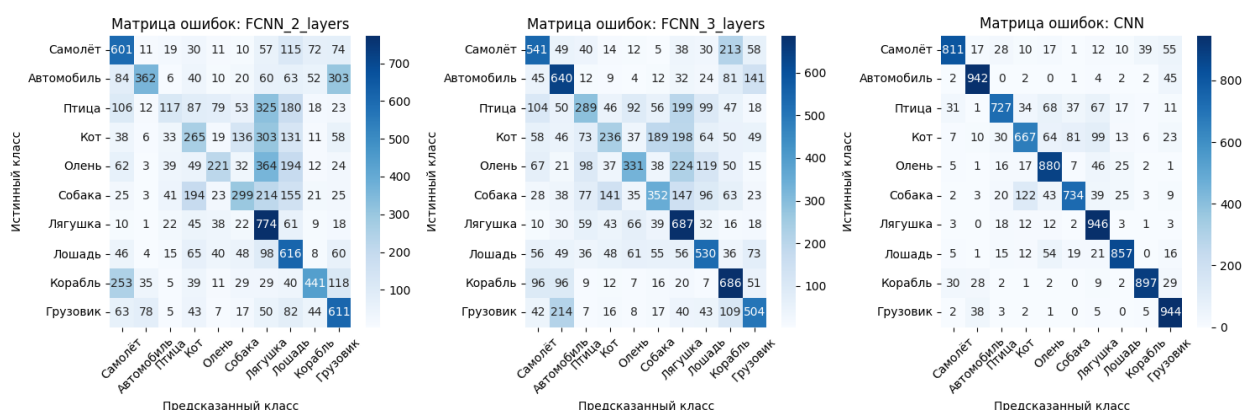


Рисунок 8 – Матрицы ошибок для 3-х моделей

Для каждой модели получены предсказания на тестовой выборке ( $x_{test}$ ), преобразованы в классы ( $\text{argmax}$ ) и сравнены с истинными метками. Построены матрицы ошибок с использованием библиотеки seaborn для анализа распределения правильных и ошибочных предсказаний по 10 классам (рисунок 8). Матрицы показывают, что CNN имеет больше правильных предсказаний (диагональные элементы), тогда как FCNN чаще ошибаются, особенно на схожих классах.

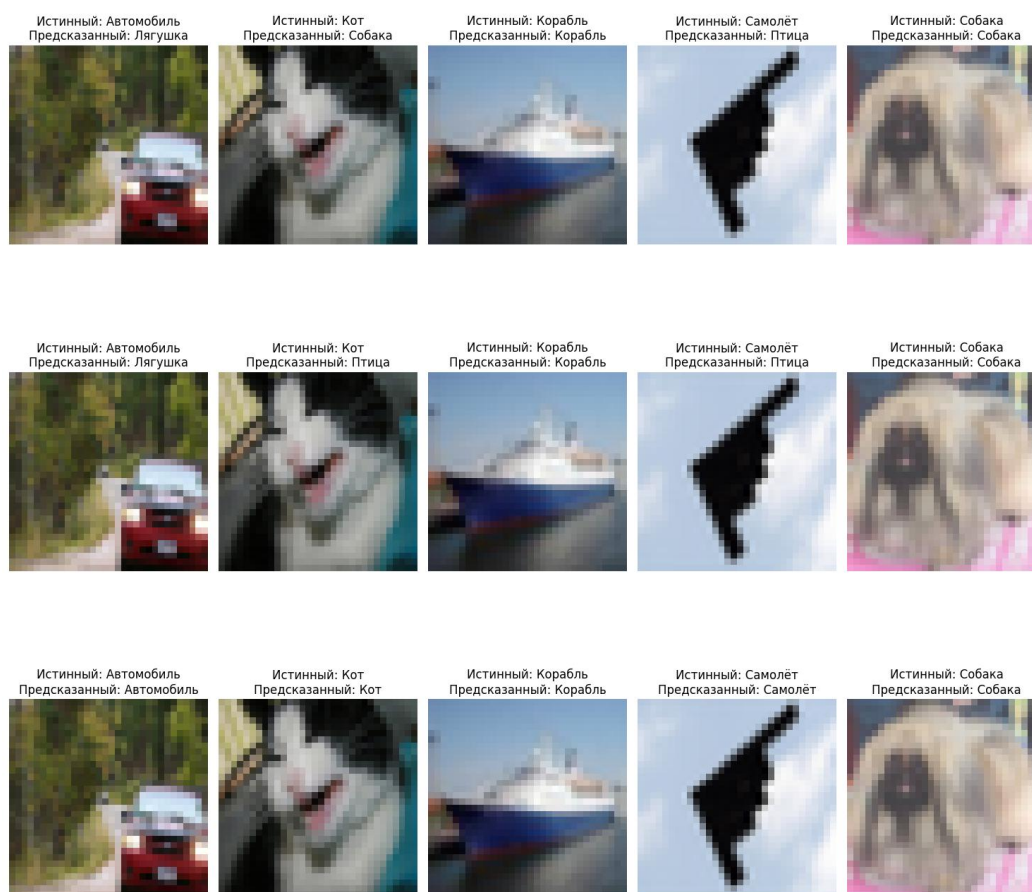


Рисунок 9 – Визуализация тестирования 3-х моделей

Случайно выбраны 5 тестовых изображений, для которых получены предсказания всех моделей. Визуализированы изображения с истинными и предсказанными метками, что позволяет качественно оценить работу моделей (рисунок 9). CNN демонстрирует более точные предсказания, тогда как FCNN чаще допускают ошибки, что согласуется с их меньшей тестовой точностью

9. Относительно разработанных архитектур можно сделать следующие выводы.

Разработанные архитектуры нейронных сетей - полносвязные нейронные сети (FCNN) с 2 и 3 слоями и свёрточная нейронная сеть (CNN) - продемонстрировали различную эффективность в задаче классификации изображений. Основные выводы:

- **Эффективность моделей:** CNN показала наилучшую тестовую точность (84,05%), значительно превосходя FCNN с 2 слоями (43,07%) и FCNN с 3 слоями (47,96%). Это обусловлено использованием свёрточных слоёв, которые эффективно извлекают пространственные признаки изображений, такие как текстуры и формы, в отличие от полносвязных слоёв FCNN, ограниченных в обработке двумерных данных.

- **Сравнение FCNN:** FCNN с 3 слоями оказалась более точной (47,96%), чем FCNN с 2 слоями (43,07%), благодаря большей глубине и дополнительному слою, что позволило лучше моделировать сложные зависимости. Однако обе FCNN показали низкую производительность из-за отсутствия свёрточных операций, что подтверждается матрицами ошибок, где они чаще путали классы с похожими визуальными признаками.

- **Регуляризация и обучение:** все модели включали нормализацию батча и выброс нейронов (dropout), что снизило риск переобучения. Аугментация данных, применённая только для CNN, способствовала её высокой обобщающей способности. Графики метрик показали, что CNN устойчиво улучшала точность и снижала потери, тогда как FCNN демонстрировали признаки переобучения (расхождение между тренировочными и валидационными метриками).

- **Вычислительная сложность:** CNN требовала значительно больше времени на обучение (14881,19 сек.) и тестирование (46 сек.) по сравнению с FCNN (1056,31 сек. и 3 сек. для FCNN с 2 слоями, 861,18 сек. и 2 сек. для FCNN с 3 слоями). Это связано с большим числом параметров и сложностью свёрточных операций, что может быть ограничением без использования GPU.

- **Практическая применимость:** CNN является предпочтительной архитектурой для задач классификации изображений благодаря высокой точности и способности обрабатывать пространственные данные. FCNN могут использоваться для простых задач или в качестве базовых моделей, но их эффективность ограничена для сложных датасетов, таких как CIFAR-10.

10. Исходя из проделанной работы, можно сравнить скорость и эффективность разработанных сетей относительно полностью связанных нейронных сетей. CNN превосходит FCNN по эффективности (84,05% против 43,07-47,96%), но требует значительно больше времени на обучение и тестирование. FCNN обеспечивают более высокую скорость, но их низкая точность ограничивает применимость для сложных задач классификации изображений.

11. Подготовлен отчет, содержащий минимальный объем информации по каждому этапу выполнения работы.

Таким образом, в процессе выполнения лабораторной работы мною были разработаны, обучены и протестированы три нейронные сети для классификации изображений на датасете CIFAR-10: FCNN с 2 слоями (точность 43,07%), FCNN с 3 слоями (47,96%) и CNN (84,05%). Выполнены нормализация данных, one-hot encoding, аугментация, анализ распределения классов, визуализация архитектур, метрик обучения и матриц ошибок. CNN показала наилучшую эффективность благодаря свёрточным слоям, но требует больше времени на обучение (14881,19 сек.) по сравнению с FCNN (861,18 – 1056,31 сек.). FCNN обеспечивают высокую скорость, но ограничены в точности. CNN предпочтительна для задач классификации изображений, тогда как FCNN подходят для простых задач, требующих дальнейшей оптимизации.