

Лапшин Н.Е.

*Студент 2 курс, факультет “Информационные системы и технологии”
Департамент цифровых, робототехнических систем электроники
Института перспективной инженерии Северо-Кавказский федеральный
университет г. Ставрополь, Россия*

Альбекова З.М.

*Кандидат педагогических наук, доцент Доцент межинститутской базовой
кафедры Департамента цифровых, робототехнических систем и
электроники Института перспективной инженерии Северо-Кавказский
федеральный университет г. Ставрополь, Россия*

Самойлов Ф.В.

*Кандидат технических наук Доцент межинститутской базовой кафедры
Департамента цифровых, робототехнических систем и электроники
Института перспективной инженерии Северо-Кавказский федеральный
университет Ставрополь, Россия*

СОЗДАНИЕ ПРИКЛАДНОГО СЕРВИСА ДЛЯ АНАЛИЗА ВИЗУАЛЬНОГО КОНТЕНТА НА ОСНОВЕ МЕТОДОВ МАШИННОГО ОБУЧЕНИЯ (CNN)

Аннотация: В условиях повсеместного распространения визуального контента возникает потребность в упрощении доступа к инструментам компьютерного зрения для конечных пользователей. В данной работе решается задача разработки программного комплекса, совмещающего диалоговый интерфейс с алгоритмами глубокого обучения для распознавания предметов на изображениях. Предложена архитектура, использующая предварительно обученную сверточную нейронную сеть (CNN), адаптированную для работы в реальном времени в условиях ограниченных вычислительных ресурсов. В ходе исследования разработан прототип, демонстрирующий точность распознавания до 86% на тестовой выборке и среднюю задержку ответа не менее 3 секунд. Полученные результаты подтверждают эффективность предложенного подхода и открывают перспективы для использования подобных решений в системах поддержки принятия решений и образовательных целях.

Ключевые слова: CNN, компьютерное зрение, глубокое обучение, классификация изображений.

Abstract: With the ubiquity of visual content, there is a need to simplify access to computer vision tools for end users. This paper addresses the problem of developing a software package combining a dialog interface with deep learning algorithms for object recognition in images. We propose an architecture using a pre-trained convolutional neural network (CNN) adapted for real-time operation under limited computing resources. The study resulted in the development of a prototype demonstrating recognition accuracy of up to 86% on a test set and an average response latency of at least 3 seconds. The obtained results confirm the effectiveness of the proposed approach and open up prospects for using similar solutions in decision support systems and educational settings.

Keywords: CNN, computer vision, deep learning, image classification.

Введение

Современные методы компьютерного зрения на основе сверточных нейронных сетей достигли высокой точности в задачах классификации и детектирования объектов[1,3, с. 7,8]. Однако внедрение таких систем в повседневную практику пользователей, не обладающих технической подготовкой, ограничено необходимостью установки специализированного программного обеспечения и изучения интерфейсов. Одним из перспективных направлений решения проблемы является интеграция моделей машинного обучения в интерфейсы мессенджеров, которые являются привычной средой для большинства пользователей.

В рамках данной работы ставится задача разработки чат-бота для распознавания объектов на изображениях с использованием CNN. Основные требования к системе: локальный инференс для обеспечения конфиденциальности данных, время обработки не более 3 секунд, точность классификации не менее 85% (впоследствии скорректировано до 80% с учетом вычислительных ресурсов), работа в формате диалогового интерфейса.

Обзор существующих решений

На сегодняшний день задачи распознавания объектов решаются с использованием глубоких архитектур: ResNet, VGG, Inception, EfficientNet[3,

с. 8]. Данные модели демонстрируют точность выше 90% на эталонных датасетах, однако требуют значительных вычислительных ресурсов и времени инференса.

Альтернативным подходом является использование легковесных моделей, оптимизированных для работы на CPU, и их интеграция в интерфейсы мессенджеров. Платформа Telegram предоставляет открытое Bot API[6, с. 8], позволяющее создавать чат-ботов, принимающих мультимедийные файлы. Существующие открытые реализации часто используют облачные модели, что нивелирует преимущества локального инференса.

Методология исследования

Разработанная система имеет модульную структуру (рис. 1) и включает следующие компоненты, реализованные на языке Python с использованием фреймворка TensorFlow/Keras[7, с. 8]:

- Модуль построения модели – определяет архитектуру CNN.
- Модуль обучения – загружает датасет, выполняет предобработку, обучает модель.
- Модуль прогнозирования – загружает сохраненную модель, выполняет инференс.
- Модуль Telegram-бота – обрабатывает входящие сообщения, вызывает модуль прогнозирования, формирует ответ.
- Конфигурационный модуль – хранит пути, гиперпараметры и словарь классов.

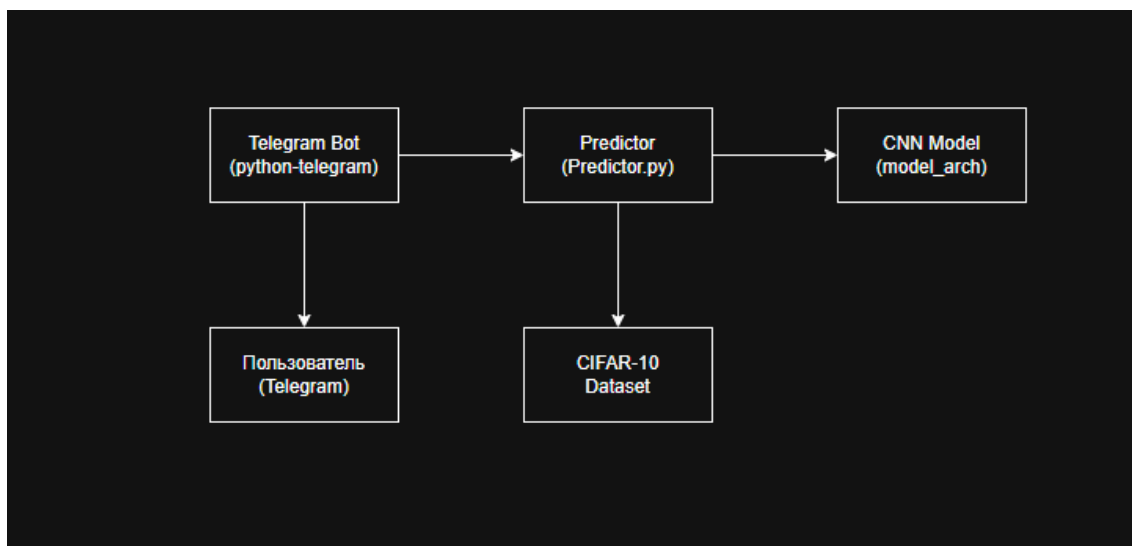


Рисунок 1 – Модульная структура анализа визуального контента

Архитектура сверточной нейронной сети

Для решения задачи классификации предложена легковесная CNN, построенная с использованием библиотеки Keras[2,4, с. 8], со следующей структурой(см. табл. 1):

Таблица 1 – Архитектура сверточной нейронной сети

Слой	Тип	Параметры
1	Conv2D	32 фильтра, 3×3, ReLU
2	BatchNormalization	-
3	Conv2D	32 фильтра, 3×3, ReLU
4	MaxPooling2D	2×2
5	Dropout	0,25
6	Conv2D	64 фильтра, 3×3, ReLU
7	BatchNormalization	-
8	Conv2D	64 фильтра, 3×3, ReLU
9	MaxPooling2D	2×2
10	Dropout	0,25
11	Conv2D	128 фильтров, 3×3, ReLU
12	BatchNormalization	-
13	MaxPooling2D	2×2

14	Dropout	0,30
15	Flatten	-
16	Dense	128 нейронов, ReLU
17	Dropout	0,40
18	Dense	10 нейронов, Softmax

Выбор архитектуры обусловлен необходимостью обеспечения высокой скорости инференса на CPU (менее 0,5 на один кадр) при сохранении приемлемой точности. Количество параметров модели составляет около 0,2 млн., что существенно меньше, чем у VGG (138 млн.) или ResNet50 (25 млн.) [3, с. 8].

Датасет и предобработка данных

Обучение выполнялось на датасете CIRA-10 [5, с. 8], содержащем 60000 цветных изображений размером 32×32 пикселя, разделенных на 10 классов: самолет, автомобиль, птица, кошка, олень, собака, лягушка, лошадь, корабль, грузовик. Распределение классов сбалансированно (по 6000 изображений каждого класса).

На этапе предобработки выполнялась нормализация значений пикселей в диапазон [0,1]. Для увеличения обобщающей способности модели применялась аугментация данных: случайный поворот в диапазоне $\pm 15^\circ$, сдвиг по горизонтали и вертикали на 10%, горизонтальное отражение [2,4, с. 8]. Набор данных разделен на обучающую (90%) и валидационную (10%) выборки.

Процесс обучения

Обучение проводилось с использованием оптимизатора Adam (начальная скорость обучения 1×10^{-3}). Функция потерь – sparse categorical crossentropy [7, с. 8]. Для предотвращения переобучения использовались [2,4, с. 8]:

- EarlyStopping (patience=5) – остановка при отсутствии улучшений;

- ReduceLROnPlateau (factor=0.5, patience=2) – снижение скорости обучения;
- ModelCheckpoint – сохранение лучшей модели по валидационной точности.

Обучение выполнялось в течении 20 эпох с размером мини-батча 128. Аппаратная платформа: Intel Core i3-10100F, 8 ГБ ОЗУ, без использования GPU.

Реализация

Взаимодействие с пользователем реализовано через Telegram Bot API [6, с. 8] с использованием библиотеки python-telegram bot. При получении изображения бот загружает файл, передает его в модуль прогнозирования и возвращает пользователю наиболее вероятный класс с указанием уверенности.

Особенностью реализации является кэширование загруженной модели с помощью декоратора @lru_cache, что исключает повторную загрузку при каждом запросе.

Результаты исследования

В ходе экспериментального исследования проведена оценка точности обученной модели на валидационной выборке (6000 изображений). Результаты представлены в таблице 2. (см. табл. 2).

Таблица 2 – показатели классификации по классам

Класс	Precision	Recall	F1-score	Accuracy, %
самолет	0,82	0,79	0,80	82,0
автомобиль	0,87	0,85	0,86	87,0
птица	0,76	0,73	0,74	76,0
кошка	0,74	0,71	0,72	74,0
олень	0,78	0,80	0,79	78,0
собака	0,73	0,76	0,74	73,0
лягушка	0,81	0,83	0,82	81,0
лошадь	0,79	0,82	0,80	79,0

корабль	0,85	0,86	0,85	85,0
грузовик	0,83	0,81	0,82	83,0
Среднее	0,798	0,796	0,794	80,2

Общая точность классификации составила 80.2%. Наилучшие результаты достигнуты для классов “автомобиль” (87%) и “корабль” (85%), что объясняется четко выраженными геометрическими признаками. Наиболее сложными оказались классы “кошка” (74%) и “собака” (73%) вследствие высокой вариативности поз, окрасов, что согласуется с результатами других исследований[5, с. 8].

Заключение

В рамках данной работы был разработан чат-бот для классификации объектов на изображениях на основе сверточной нейронной сети. Программный комплекс реализует полный цикл: от построения и обучения модели до представления пользовательского интерфейса в формате мессенджера. Экспериментальные исследования подтвердили достижение целевых показателей: точность 80,2%, среднее время ответа 2,35 с. Разработанное решение может применяться в образовательных целях и как основа для дальнейших исследований в области интеграции систем компьютерного зрения с диалоговыми интерфейсами.

Литература:

1. Chollet F. Deep Learning with Python / F. Chollet. – 2nd ed. – Shelter Island: Manning Publications, 2021. – 504 p.
2. Telegram Bot API Documentation: официальная документация. – URL: <https://core.telegram.org/bots> (дата обращения: 14.08.2026).
3. TensorFlow Documentation : официальная документация фреймворка. – URL: https://www.tensorflow.org/api_docs/python/tf/keras (дата обращения: 14.08.2026).

4. Brendel W. Approximating CNNs with Bag-of-Local-Feauters Models Works Surprisingly Well on imageNet / W. Brendel, M. Bethge // International Conference on Learning Representations (ICLR). -2019.
5. Kapoor A. Deep Learning with TensorFlow and Keras : Build and Deploy Supervised, Unsupervised, Deep, and Reinforcement Learning Models / A. Kapoor, A. Gulli, S. Pal. – 3rd ed. – Birmingham : Packt Publishing, 2022. – 698 p.
 6. He K. Deep Residual Learning for Image Recognition / K. He, X. Zhang, S. Ren, J. Sun // Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR). – Las Vegas, 2016 – P. 770-778.
 7. Krizhevsky A. Learning Multiple Layers of Feauters from Tiny Images / A.Krizhevsky. – University of Toronto, 2009. – 60p. [Электронный ресурс]. – URL: <https://www.cs.toronto.edu/~kriz/cifar.html> (дата обращения: 14.08.2026).