

Анализ расходов на основе нейросетевого распознавания чеков

Светлана Мунавировна Куценко¹, Елена Андреевна Салтанаева²

¹Кандидат педагогических наук, доцент кафедры информационных технологий и интеллектуальных систем ФГБОУ ВО «Казанского государственного энергетического университета». E-mail: s.koutsenko@mail.ru

²Кандидат технических наук, доцент кафедры информационных технологий и интеллектуальных систем ФГБОУ ВО «Казанского государственного энергетического университета». E-mail: elena_maister@mail.ru

Аннотация. Прогресс в области нейронных сетей и глубокого обучения открывает возможности для высокоточного автоматического извлечения и анализа данных, даже из сложных документов, таких как кассовые чеки. Внедрение современных цифровых технологий способно значительно упростить ведение личного бюджета и предоставить пользователям ценные инструменты для анализа расходов. Была разработана и обучена сверточная нейронная сеть для классификации товаров, проведена оценка ее качества. Реализованы серверная и клиентская части приложения. Результатом работы является разработанное веб-приложение, которое позволяет пользователям загружать изображения чеков через веб-интерфейс, получать структурированную информацию о покупках (включая детализацию по товарам) и использовать инструменты для анализа и визуализации своих расходов, тем самым решая проблемы трудоемкости и неточности ручного учета.

Ключевые слова: оптическое распознавание символов, сверточная нейронная сеть, библиотеки Python, веб-интерфейс, глубокое обучение

Expense analysis based on neural network receipt recognition

Svetlana Munavirovna Kutsenko¹, Elena Andreevna Saltanaeva²

¹Candidate of Pedagogical Sciences, Associate Professor of the Department of Information Technologies and Intelligent Systems FSBEI of HE «Kazan State Power Engineering University». E-mail: s.koutsenko@mail.ru

²Candidate of Technical Sciences, Associate Professor of the Department of Information Technologies and Intelligent Systems FSBEI of HE «Kazan State Power Engineering University». E-mail: elena_maister@mail.ru

Abstract. Advances in neural networks and deep learning offer opportunities for highly accurate automatic data extraction and analysis, even from complex documents such as cash receipts. The introduction of modern digital technologies can significantly simplify personal budget management and provide users with valuable tools for analyzing costs. A convolutional neural network was developed and trained to classify goods, and its quality was assessed. The server and client parts of the application are implemented. The result of the work is a developed web application that allows users to upload check images via a web interface, obtain structured information about purchases (including product details) and use tools to analyze and visualize their expenses, thereby solving the problems of labor intensity and inaccuracy of manual accounting.

Keywords: optical character recognition, convolutional neural network, Python libraries, web interface, deep learning

Введение

В современном мире эффективное управление личными финансами приобретает все большую значимость для достижения финансовой стабильности и благополучия. Неотъемлемой частью этого процесса является систематическое отслеживание и анализ расходов, что позволяет контролировать личный бюджет, выявлять нецелевые траты, планировать будущие покупки и формировать сбережения. Основой для такого анализа служат данные о совершенных покупках, наиболее полным и первичным источником которых является кассовый чек. Он содержит детализированную информацию: наименования товаров/услуг, их количество, цены, итоговую сумму, дату, время и место транзакции. Традиционные подходы к ведению бюджета, такие как ручные записи или использование электронных таблиц, часто оказываются неэффективными. Они требуют значительных временных затрат, высокой степени самодисциплины и подвержены ошибкам, связанным с человеческим фактором.

Одновременно с этим, несмотря на цифровизацию, значительная часть транзакций в розничной торговле и сфере услуг по-прежнему сопровождается выдачей бумажных кассовых чеков. Эти бумажные чеки, содержащие важную информацию о покупках, часто требуют цифровизации для целей учета, анализа и хранения. Извлечение данных из большого количества чеков вручную представляет собой трудоемкую и рутинную задачу. Кроме того, бумажные чеки со временем выцветают, могут быть повреждены или утеряны, а их хранение требует физического пространства, что еще больше усложняет процесс долгосрочного учета расходов.

Таким образом, неуклонно растущий объем кассовых чеков, как в электронной, так и в бумажной форме, создает острую потребность в разработке и применении эффективных методов их автоматизированной обработки и анализа. Информация, заключенная в чеках, имеет значение не только для потребителей в целях управления личным бюджетом, но и для бизнеса и государственных органов. Ручная обработка такого массива данных становится не просто трудоемкой, но и практически невозможной или экономически нецелесообразной. Это подчеркивает критическую важность автоматизации.

Ключевой технологией, позволяющей автоматизировать извлечение информации из визуальных представлений документов, является оптическое распознавание символов (Optical Character Recognition, OCR). OCR представляет собой процесс преобразования изображений текста (сканированных документов, фотографий) в редактируемый и машиночитаемый текстовый формат [Андреева, Гончаренко, Диас, 2024]. Применение OCR к изображениям кассовых чеков позволяет автоматически "считывать" содержащуюся в них информацию.

Однако кассовые чеки представляют собой сложный объект для стандартных OCR-систем. Во-первых, отсутствие строгой стандартизации приводит к огромному разнообразию форматов, шрифтов, их размеров и взаимного расположения элементов на чеке. Во-вторых, качество печати и изображений часто бывает низким: использование термобумаги, бледная печать, выцветание, а также дефекты изображения при фотографировании (неравномерное освещение, тени, блики, искажения перспективы, складки на бумаге) существенно затрудняют распознавание. В-третьих, сложная структура чека, включающая табличные данные (список товаров с ценами и количеством), логотипы, штрих-коды и прочие нетекстовые элементы, требует не просто распознавания символов, но и понимания семантической структуры документа для корректного извлечения данных.

Традиционные методы OCR, основанные на шаблонном сопоставлении или ручном проектировании признаков для последующей классификации, демонстрируют ограниченную эффективность при работе с такими сложными и вариативными документами, как кассовые чеки. Их адаптивность к разнообразию шрифтов и устойчивость к искажениям недостаточны, что приводит к частым ошибкам распознавания.

Прорыв в решении этих проблем связан с активным развитием нейронных сетей и методов глубокого обучения. Нейросетевые подходы к OCR обладают значительными

преимуществами [Воробьев, 2025]. Благодаря способности обучаться на больших массивах данных, они демонстрируют высокую адаптивность к различным стилям оформления чеков и вариациям шрифтов. Архитектуры глубокого обучения, в частности сверточные нейронные сети (CNN), известны своей устойчивостью к шумам и способностью эффективно извлекать информативные признаки даже из изображений низкого качества. Важнейшим достоинством является автоматическое извлечение иерархических признаков, что избавляет от необходимости их ручной разработки. Для задач обработки последовательностей, каковой является текст, успешно применяются рекуррентные нейронные сети (RNN) и их более совершенные варианты (LSTM, GRU), а также комбинированные сверточно-рекуррентные архитектуры (CRNN), позволяющие учитывать контекст при распознавании [Терлецкий, Терлецкая, 2023].

На рынке представлен широкий спектр программных решений, предназначенных для помощи пользователям в управлении личными финансами и анализе структуры расходов. Существующие решения предлагают разные механизмы работы с данными о расходах. Методы, основанные на данных ФНС (через QR-код или прямой доступ), обеспечивают высокую точность, но зависят от наличия QR-кода или предварительной регистрации контактных данных. Банковские приложения удобны для безналичных операций, но не дают полной детализации и не учитывают наличные. Приложения, полагающиеся исключительно на OCR-распознавание изображений, часто сталкиваются с проблемами точности.

В этой связи сохраняется потребность в разработке решения, которое бы специализировалось на высокоточном извлечении данных именно из изображений кассовых чеков, используя для этого продвинутое нейросетевые технологии.

Методы и решение

Оптимальным вариантом реализации системы анализа расходов является веб-приложение, что позволит значительно упростить его использование и поддержку. Такое веб-приложение могло бы эффективно обрабатывать чеки даже при отсутствии читаемого QR-кода или регистрации в системе ФНС, предоставляя пользователям удобный инструмент для анализа расходов с детализацией до товарных позиций через доступный веб-интерфейс.

С клиентской стороны при такой реализации пользователю необходим только стандартный веб-браузер и подключение к сети Интернет, без необходимости установки специализированного программного обеспечения на каждое устройство. Такой подход позволяет легко получать доступ к приложению с различных платформ, упрощает процесс обновления и снижает затраты на поддержку клиентских устройств. Интерфейс, с которым взаимодействует пользователь, формируется с использованием технологий HTML для разметки структуры, CSS для визуального оформления и JavaScript для обеспечения интерактивности и динамического обновления данных без перезагрузки страницы [Куценко, Салтанаева, 2025]. Клиент отправляет HTTP-запросы на сервер для выполнения различных действий, таких как аутентификация, загрузка файлов (изображений чеков), запрос данных для отображения, например, истории покупок или аналитики, и получение ответов от сервера. Для ускорения разработки пользовательского интерфейса и обеспечения его адаптивности на различных устройствах используется CSS-фреймворк Bootstrap 5 [Bootstrap, 2025]. Bootstrap предоставляет готовую сетку для верстки, набор стилизованных компонентов и утилиты, что позволяет сосредоточиться на функциональности, а не на базовой стилизации [Адаптивная вёрстка сайта, 2025]. Для визуализации аналитических данных на дашборде пользователя выбрана библиотека Chart.js [Библиотека Chart.js для WEB, 2025].

Серверная часть приложения, реализованная на платформе FastAPI, выполняет роль центрального узла, координирующего всю работу системы. Она содержит основную бизнес-логику приложения. Сервер обрабатывает входящие HTTP-запросы от клиента, предоставляет программный интерфейс (API) для обмена данными, осуществляет аутентификацию и

авторизацию пользователей, управляет их сессиями. Важной функцией сервера является взаимодействие с базой данных для сохранения, извлечения и модификации информации. Кроме того, сервер интегрирует и управляет работой модуля машинного обучения, ответственного за обработку чеков. В некоторых случаях сервер также генерирует HTML-страницы для отображения клиенту, используя систему шаблонов.

Для реализации веб-приложения был выбран язык программирования Python, имеющий множество библиотек как для веб-разработки, так и машинного обучения. В качестве веб-фреймворка для Python был выбран FastAPI [Веб-фреймворк FastAPI, 2025]. В отличие от более традиционных фреймворков, FastAPI построен на основе современных стандартов ASGI, что обеспечивает высокую производительность благодаря поддержке асинхронных операций. Встроенная валидация данных с использованием Pydantic схем повышает надежность приложения, а система внедрения зависимостей делает код более модульным и тестируемым [Шафорост, Д. В. Власов, 2023].

Для взаимодействия с базой данных была выбрана библиотека SQLAlchemy в связке с асинхронным драйвером asyncpg. SQLAlchemy является мощной и гибкой Object-Relational Mapper (ORM), позволяющей работать с базой данных, используя объектно-ориентированный подход. Ее поддержка асинхронных операций необходима для полноценного использования возможностей FastAPI. В качестве СУБД была выбрана PostgreSQL. База данных, функционирующая под управлением СУБД PostgreSQL, служит хранилищем всей персистентной информации приложения. В ней содержатся сведения о зарегистрированных пользователях, их персональные данные и настройки, история совершенных покупок с детализацией по товарным позициям, а также необходимые справочники, такие как категории товаров и чеков, и мастер-справочник товаров. Серверная часть взаимодействует с базой данных посредством ORM-инструментария.

Особым компонентом серверной части является модуль машинного обучения (ML-пайплайн). Его задача – автоматизировать извлечение и первичную обработку данных из изображений кассовых чеков. Этот модуль включает подсистему оптического распознавания символов для преобразования изображения в текст, модуль парсинга для извлечения структурированных полей (дата, сумма, магазин, товары) из текста с помощью регулярных выражений, и нейросетевую модель для автоматической классификации извлеченных товарных позиций по предопределенным категориям. Для реализации компонента машинного обучения, отвечающего за классификацию товарных позиций, был выбран фреймворк глубокого обучения PyTorch. Его тесная интеграция с Python упрощает встраивание ML-моделей в веб-приложение [Yuxi (Hayden) Liu, 2020]. Для предобработки русскоязычных текстовых данных были выбраны стандартные и широко используемые библиотеки NLTK (Natural Language Toolkit) для токенизации и Morpho3 для лемматизации, что позволяет привести текстовые данные к нормализованному виду перед подачей в модель [Deshmukh , 2022, Пальмов, Салихов , 2024].

Задача извлечения текста из изображений чеков решается с помощью Tesseract OCR [Елхимова, Канев, 2022]. Интеграция этой системы в Python относительно проста, что позволяет быстро реализовать базовую функциональность OCR в приложении.

Процесс взаимодействия компонентов можно проиллюстрировать на примере обработки загруженного чека. Пользователь через браузер загружает изображение. Клиент отправляет файл на сервер по API. Сервер передает изображение ML-пайплайну. Пайплайн выполняет шаги OCR, парсинга и классификации, возвращая структурированные данные серверу. Сервер передает эти данные клиенту для отображения в интерфейсе редактирования. Пользователь проверяет, при необходимости корректирует данные, и нажимает кнопку сохранения. Клиент отправляет подтвержденные данные на сервер. Сервер выполняет финальную валидацию и сохраняет информацию о покупке и ее позициях в базу данных.

Ключевой особенностью предлагаемого приложения является использование методов машинного обучения для автоматизации извлечения и категоризации данных из кассовых чеков. Проектирование этого компонента включало определение последовательности обработки данных (ML-пайплайна) и выбор архитектуры нейронной сети для решения задачи классификации товаров.

Весь процесс автоматической обработки изображения чека состоит из нескольких последовательных этапов. Первым шагом является извлечение текстовой информации из предоставленного изображения. Для этой цели используется система оптического распознавания символов Tesseract. Она принимает на вход изображение чека и возвращает распознанный текст в виде единой строки или набора строк. Полученный сырой текст далее передается на этап парсинга. Задача парсера – проанализировать неструктурированный текст и извлечь из него семантически значимые поля: дату и время покупки, наименование магазина, итоговую сумму чека, и, что наиболее важно, список товарных позиций. Для каждой позиции парсер стремится выделить наименование товара, его количество, цену за единицу и общую стоимость строки. На данном этапе проектирования для реализации парсера был выбран подход, основанный на использовании регулярных выражений, настроенных на типичные шаблоны представления информации в кассовых чеках. После извлечения списка товарных позиций возникает задача их автоматической категоризации. Ручное назначение категории для каждого товара в каждом чеке было бы трудоемким для пользователя. Поэтому был спроектирован нейросетевой классификатор, основной задачей которого является предсказание наиболее вероятной категории товара (из предопределенного справочника категорий товаров) на основе его текстового наименования, полученного на этапе парсинга.

Входными данными для классификатора служит текстовое наименование товара. Перед подачей в нейронную сеть текст проходит стадию предобработки, включающую токенизацию (разбиение на слова), лемматизацию (приведение слов к начальной форме с использованием морфологического анализатора), удаление неинформативных стоп-слов и преобразование очищенной последовательности токенов в числовое представление (вектор или последовательность индексов), понятное для нейронной сети.

В качестве архитектуры нейросетевого классификатора была выбрана сверточная нейронная сеть [Куценко, Салтанаева, 2023]. Данный выбор обусловлен способностью CNN эффективно извлекать локальные признаки (комбинации слов, n-граммы) из последовательностей, что хорошо подходит для анализа относительно коротких текстов, какими являются наименования товаров. Спроектированная архитектура (см. рис. 1) включает входной слой Embedding, который преобразует индексы слов в плотные векторные представления. Далее следуют несколько параллельных блоков, каждый из которых содержит сверточный слой (Conv1d) с различными размерами фильтров, позволяющие захватывать признаки разного масштаба, за которым следуют слой батч-нормализации BatchNorm1d и функция активации ReLU. Результаты сверток агрегируются с помощью слоев субдискретизации (MaxPool1d) для уменьшения размерности и выделения наиболее значимых признаков. Полученные векторы признаков конкатенируются, подаются на полносвязный скрытый слой с батч-нормализацией и функций активации ReLU. Полученные данные проходят через слой регуляризации (Dropout) для предотвращения переобучения и подаются на выходной полносвязный слой (Linear), который генерирует оценки принадлежности товара к каждому из возможных классов категорий. Выходом классификатора является идентификатор категории с наивысшей оценкой.

Таким образом, общий ML-пайплайн обработки чека представляет собой последовательность: изображение чека поступает на вход OCR-системы (Tesseract), извлеченный текст обрабатывается парсером на основе регулярных выражений, который выделяет структурированные данные, включая список товарных позиций. Затем наименование каждой товарной позиции проходит предобработку и подается на вход нейросетевого

классификатора, который предсказывает категорию товара. Опционально, после этого может производиться поиск соответствующего товара в мастер-справочнике. Результатом работы всего пайплайна является структурированный объект с данными чека и категоризованным списком товаров, который передается основной логике бэкенд-приложения для дальнейшего использования.

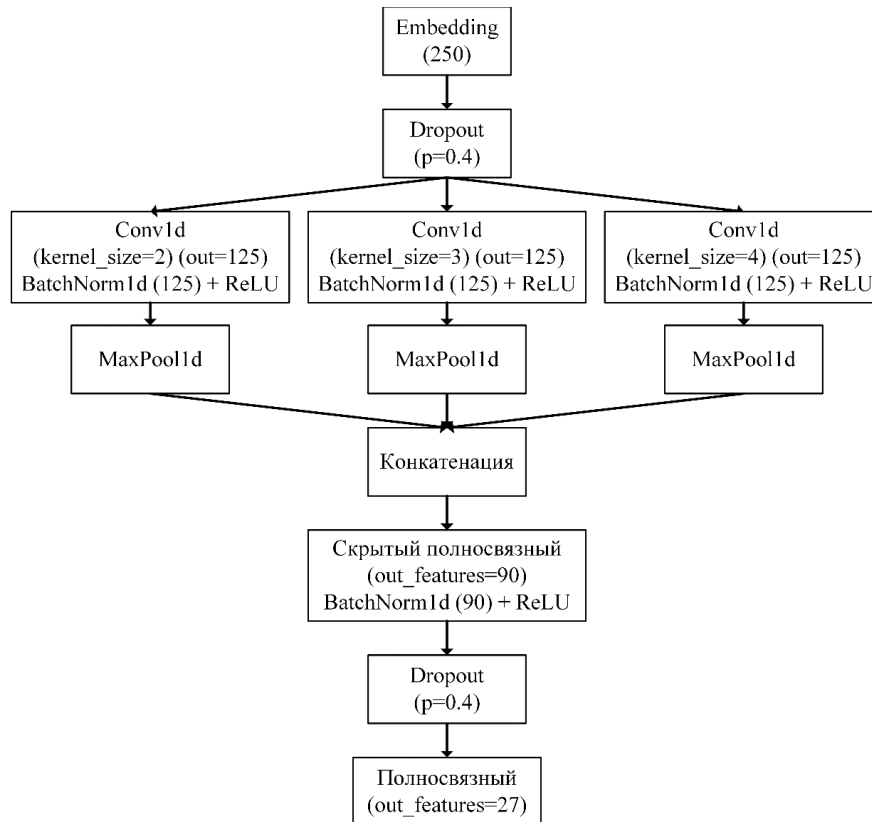


Рис. 1. Архитектура нейронной сети

У разработанного программного обеспечения пользовательский интерфейс представлен в виде набора веб-страниц, отображаемых в браузере пользователя. Интерфейс разрабатывается с использованием стандартных веб-технологий: языка разметки HTML для определения структуры контента, формального языка описания внешнего вида документа CSS для стилизации элементов, а также языка программирования JavaScript для реализации интерактивности, динамического обновления данных и взаимодействия с серверным API.

При проектировании интерфейса необходимо учесть основные принципы создания качественных пользовательских интерфейсов. Особое внимание было уделено предоставлению пользователю контроля над системой, например, через возможность редактированию автоматически распознанных данных чека перед их окончательным сохранением. Принцип уменьшения нагрузки на пользователя реализован за счет автоматизации извлечения данных из чеков и использования интуитивно понятных элементов управления, таких как выпадающие списки для выбора категорий, вместо того чтобы требовать от пользователя полного ручного ввода. Необходимо обеспечить совместимость и последовательность дизайна и поведения элементов управления на всех страницах приложения для предсказуемости взаимодействия. Использовать визуальные подсказки, такие как индикаторы загрузки или выделение полей с

ошибками, а также информативные сообщения об успехе или неудаче операций, для обеспечения непрерывной обратной связи с пользователем.

Элементом интеллектуальной составляющей приложения является нейросетевой модуль, предназначенный для автоматической классификации наименований товарных позиций, извлеченных из кассовых чеков, по predetermined категориям. Разработка данного модуля включала несколько этапов: предобработку текстовых данных, формирование словаря, проектирование и реализацию архитектуры нейронной сети. Процесс предобработки текстовых данных включал: приведение к нижнему регистру, удаление неалфавитных символов, токенизацию, лемматизацию. Это позволило подготовить наименования товаров для дальнейшей обработки. На основе очищенных лемм был создан словарь, отображающий уникальные слова в числовые индексы.

Для решения задачи классификации была спроектирована и реализована сверточная нейронная сеть с использованием PyTorch, что обусловлено способностью CNN эффективно извлекать локальные признаки из последовательностей, что хорошо подходит для анализа коротких текстов, каким являются наименования товаров [Madani, 2023]. Архитектура модели включает слой вложений (Embedding) для получения векторных представлений слов, несколько параллельных 1D сверточных слоев с разными размерами ядер для выявления n-грамм, BatchNorm1d и функцией активации ReLU, слой глобального максимального пулинга, скрытый полносвязный слой с Dropout для регуляризации и полносвязный выходной слой для получения оценок принадлежности к категориям.

Обучение нейросетевого классификатора проводилось на датасете из CSV-файла, содержащего 10339 пар "наименование товара – ID категории". В датасете представлено 27 уникальных категорий. Данные прошли предобработку, описанную ранее.

Для объективной оценки датасет был разделен на обучающую (6616 примеров), валидационную (1655 примеров) и тестовую (2068 примеров) выборки со стратификацией по классам.

Процесс обучения включал использование функции потерь CrossEntropyLoss, подходящей для многоклассовой классификации, и оптимизатора Adam со скоростью обучения 0.001. Обучение проводилось в течение 20 эпох. Для повышения обобщающей способности модели и уменьшения переобучения к обучающей выборке применялась онлайн-аугментация данных: с вероятностью 30% из наименования товара удалялось от одного до 10% слов, с вероятностью 30% два случайных слова в наименовании товара менялись местами. Аугментация применялась только к обучающим данным непосредственно в процессе загрузки батчей, что позволило модели на каждой эпохе видеть слегка измененные версии одних и тех же данных.

На каждой эпохе модель обучалась на аугментированной обучающей выборке и обновляла свои веса для минимизации функции потерь, после чего ее производительность оценивалась на валидационной выборке. По результатам обучения можно построить графики функции потерь и точности. На рисунке 2(а) представлена динамика изменения функции потерь. Значения функции потерь на обучающей выборке (синяя кривая) снизились с примерно 2.87 до 0.99. На валидационной выборке (оранжевая кривая) потери также стабильно уменьшались, достигнув минимального значения около 1.10 на 19-й эпохе, после чего незначительно выросли, что указывает на начало переобучения модели. Для предотвращения переобучения сохранялась лучшая модель, показавшая наименьшее значение функции потерь на валидационной выборке.

Динамика изменения точности представлена на рисунке 2(б). Точность на обучающей выборке (синяя кривая) увеличивалась с каждой эпохой, достигая к концу обучения значения 70%. Точность на валидационной выборке (оранжевая кривая) также показывала рост, стабилизировавшись на значениях в районе 67%. Разрыв между точностью на обучающей и валидационной выборках также является индикатором переобучения, что подтверждает важность использования валидационной выборки для финальной модели.

Итерационный процесс обучения, сопровождающийся обновлением весов и мониторингом производительности на валидационном наборе данных, позволил получить модель. Несмотря на проявление признаков переобучения, модель была сохранена на своей оптимальной итерации для последующей независимой оценки на тестовой выборке.

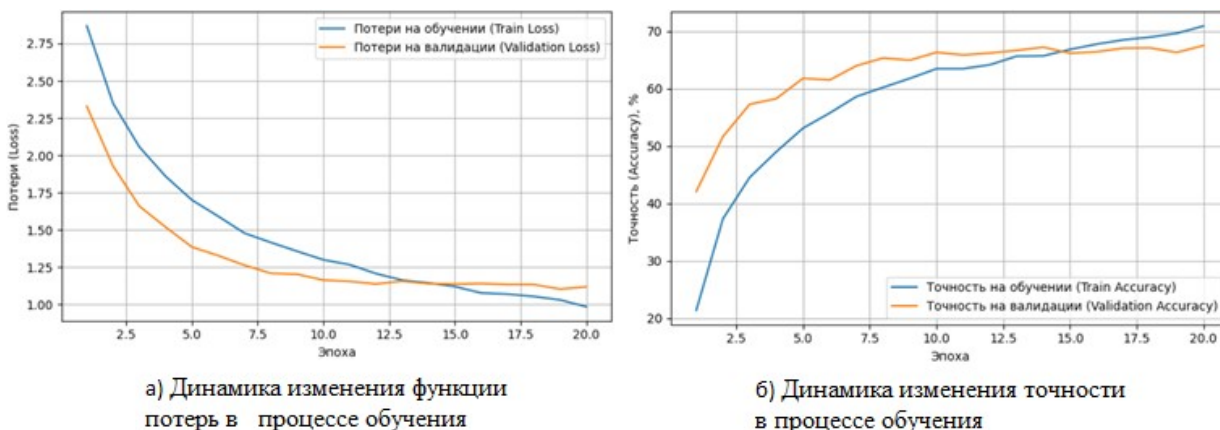


Рис. 2. Динамики изменения функций
 а) функция потерь в процессе обучения, б) точность в процессе обучения

После завершения обучения была проведена оценка качества классификатора на отложенной тестовой выборке данных. Для оценки использовались стандартные метрики качества многоклассовой классификации: общая точность (Accuracy), Precision, Recall и F1-score для каждого класса. Общая точность на тестовом наборе, состоящем из 2068 примеров, составила 69.58%. Детальный отчет по классификации представлен в таблице. Из отчета видно, что категории 1 и 4 показывают относительно высокую F1-меру. Это свидетельствует о том, что данные категории товаров имеют достаточно характерные и отличимые наименования.

Таблица

Отчет классификации

Категория	Precision	Recall	F1-score
CatID_1	0.96	1.00	0.98
CatID_2	0.90	0.87	0.89
CatID_3	0.68	0.80	0.73
CatID_4	0.90	1.00	0.95
CatID_5	0.28	0.21	0.24
CatID_6	0.56	0.37	0.44
CatID_7	0.76	0.91	0.83
CatID_8	0.95	0.83	0.89
CatID_9	0.52	0.49	0.50
CatID_10	0.86	0.85	0.86
CatID_11	0.63	0.67	0.65
CatID_12	0.49	0.54	0.51
CatID_13	0.79	0.67	0.72
CatID_14	0.84	0.88	0.86
CatID_15	0.84	0.84	0.84
CatID_16	0.70	0.77	0.73
CatID_17	0.56	0.43	0.48
CatID_18	0.57	0.73	0.64
CatID_19	0.82	0.66	0.73

Категория	Precision	Recall	F1-score
CatID 20	0.64	0.52	0.57
CatID 21	0.65	0.73	0.69
CatID 22	0.63	0.59	0.61
CatID 23	0.87	0.91	0.89
CatID 24	0.73	0.71	0.72
CatID 25	0.72	0.72	0.72
CatID 26	0.72	0.67	0.70
CatID 27	0.56	0.60	0.58

Пользовательский интерфейс программного обеспечения представляет собой набор HTML страниц, отображаемых в браузере. При первом входе в приложение неавторизованный пользователь видит приветственную страницу, содержащую краткое описание возможностей системы и призыв к регистрации или входу в аккаунт для доступа к основной функциональности.

После успешного прохождения регистрации и входа для авторизованного пользователя главной страницей является информационная панель (см. рис. 3).

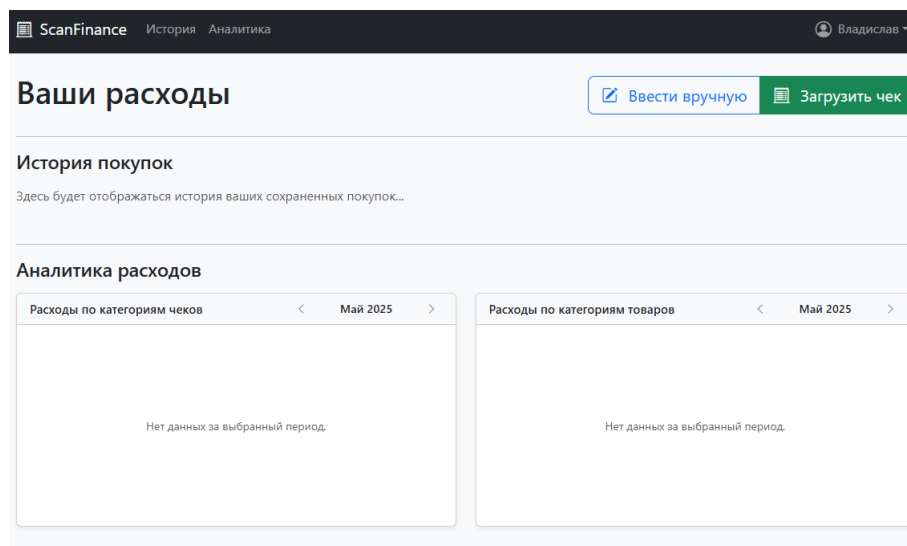


Рис. 3. Главная страница


Основная функция приложения – добавление данных о покупках. На данной странице пользователю доступны основные элементы управления: кнопки «Загрузить чек» и «Ввести вручную» для добавления новых данных о расходах. При нажатии кнопки «Загрузить чек» на странице открывается модальное окно загрузки.

После загрузки и автоматической обработки изображения пользователю отображается интерфейс с результатами распознавания (см. рис. 4). В левой части показано превью загруженного изображения чека (с возможностью увеличения по клику). В правой части – поля с распознанными данными: дата, магазин, итоговая сумма и категория чека, выбираемая из списка. Ниже расположена таблица с товарными позициями, где каждая строка соответствует одному товару из чека.

Пользователь имеет возможность редактировать все распознанные данные: изменить дату, магазин, сумму, выбрать категория чека. Пользователь может удалять неверно распознанные строки или добавлять новые с помощью кнопки «Добавить товар»

Второй способ добавления данных о покупке – вручную. При нажатии кнопки «Ввести вручную» открывается аналогичный интерфейс, но все поля изначально пустые, и пользователь сам заполняет информацию о покупке. При этом также доступна возможность прикрепить изображение чека.

Предпросмотр (клик для увеличения):



Дата: 02.12.2018 17:40

Магазин: Пятёрочка

Категория чека: Выберите категорию...

Итого: 55.59

Товары в чеке

Наименование	Товар (Поиск)	Кол-во	Цена	Сумма товара	Категория
2111189 К.Ц.Мол.ул/паст. 1,5% 970		1,000	49,99	49,99	Сладости / Снеки
3467778 ЛИБ.Напит. ябл/вишн/чер д/д1,93		1,000	82,99	82,99	Сладости / Снеки
3648084 Томат сливовидный 600г		1,000	139,99	139,99	Сладости / Снеки
3498026 ПЮБ.Нап.АПЕЛ.МАНГО дет.мак1,93Э		1,000	82,99	82,99	Сладости / Снеки
3442578 PICNIC Батончик BIG 76г		1,000	42,80	42,80	Сладости / Снеки
3442578 PICNIC Батончик BIG 76г		1,000	42,80	42,80	Сладости / Снеки
3277399 РАВА Найон.ПРОВАНС .67X ana 400г		1,000	64,00	64,00	Сладости / Снеки
2111189 К.Ц.Мол.ул/паст. 1,5% 970мл i		1,000	500,90	500,90	Сладости / Снеки

+ Добавить товар

✔ Подтвердить и сохранить
✕ Отмена

Рис. 4. Интерфейс отображения и редактирования распознанного чека

Все сохраненные покупки доступны на странице истории, и отображаются в виде списка, отсортированного по дате, от новых к старым. Каждая запись в списке содержит основную информацию: название магазина, дате и время, итоговую сумму и категорию чека.

Изображенные на главной странице аналитические диаграммы интерактивны (см. рис. 5): с помощью встроенного навигатора по месяцам пользователь может просматривать статистику за различные периоды.

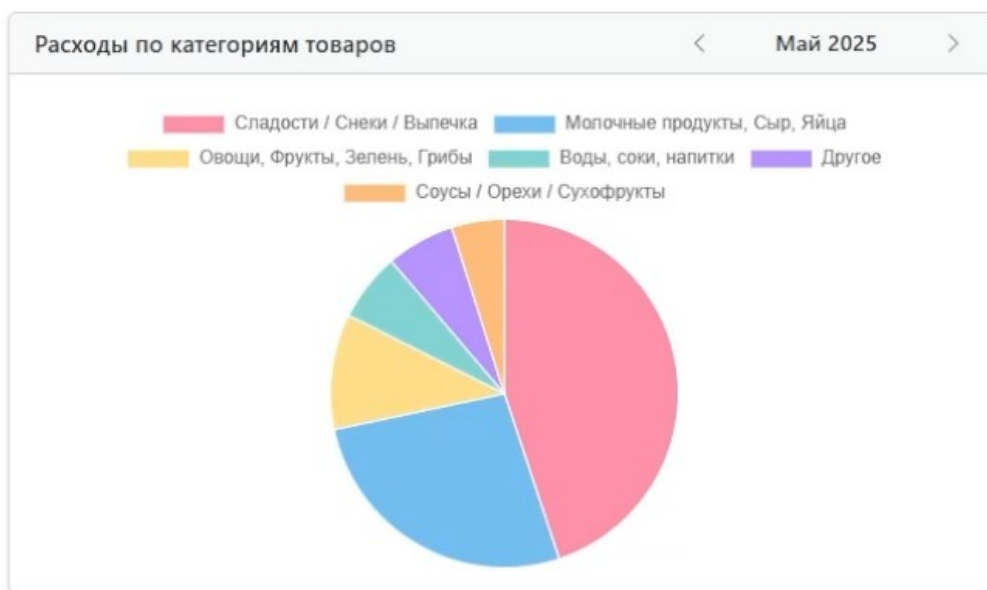


Рис. 5. Аналитическая диаграмма «Расходы по категориям товаров»

Заключение

Таким образом, эффективное управление личными финансами играет ключевую роль в достижении финансовой стабильности, а детальный учет расходов является его неотъемлемой частью.

В связи с этим разработка веб-приложения, использующего нейросетевое распознавание чеков для анализа личных финансов, является актуальной задачей, решающей проблемы ручного учета и отвечающей современным потребностям пользователей. Ключевой особенностью разработанного веб-приложения является использование современных нейросетевых методов OCR для максимально точного и автоматизированного извлечения данных из изображений кассовых чеков.

Веб-приложение предоставляет пользователям следующий основной функционал:

- управление профилем и настройками отображения информации;
- загрузка изображений кассовых чека с использованием интерфейса drag-and-drop или стандартного выбора файла;
- автоматическое извлечение данных из чека;
- автоматическая классификация товарных позиций по категории с использованием обученной нейросетевой модели;
- возможность редактирования распознанных данных;
- просмотр истории совершенных покупок с возможностью детализации каждой записи;
- визуализация структуры расходов с помощью интерактивных диаграмм с возможностью навигации по периодам.

Веб-приложение является готовым к использованию инструментом, который автоматизирует процесс учета личных расходов, предоставляет пользователям удобные средства для анализа своей финансовой активности. Развитие проекта может включать расширение аналитических возможностей и интеграцию с другими финансовыми сервисами.

Литература

1. Адаптивная верстка сайта. URL: <https://html5book.ru/adaptivnaya-vyorstka-sayta/> (дата обращения: 26.04.2025). Воробьев, В. А. Исследование возможностей технологий нейросетевой обработки графических данных с различной детализацией / В. А.
2. Андреева, Н. Н. Оптическое распознавание символов с применением машинного обучения в области проектирования / Н. Н. Андреева, Н. А. Гончаренко, М. А. Диас // Автоматизация и информатизация ТЭК. – 2024. – № 5(610). – С. 36-40.
3. Библиотека Chart.js для WEB. URL: <https://java-online.ru/chartjs.xhtml> (дата обращения: 26.10.2025).
4. Веб-фреймворк FastAPI. URL: <https://fastapi.tiangolo.com/> (дата обращения: 26.10.2025).
5. Воробьев // Инновационная наука. – 2025. – № 6-1. – С. 58-64.
6. Елхимова, И. С. Оптическое распознавание символов с помощью Tesseract / И. С. Елхимова, А. И. Канев // Тенденции развития науки и образования. – 2022. – № 92-10. – С. 71-75. – DOI 10.18411/trnio-12-2022-480.
7. Куценко, С. М. Веб-приложение как средство оптимизации бизнес-процессов в организациях / С. М. Куценко, Е. А. Салтанаева // Экономика и предпринимательство. – 2025. – № 1(174). – С. 1008-1015. – DOI 10.34925/EIP.2024.174.1.180.
8. Куценко, С. М. Выбор инструментов построения нейронной сети / С. М. Куценко, Е. А. Салтанаева // Научно-технический вестник Поволжья. – 2023. – № 12. – С. 313-315.
9. Пальмов, С. В. Сравнительный анализ библиотек PUMORPHY3 и PUMYSTEM3 / С. В. Пальмов, Р. Р. Салихов // Наука и бизнес: пути развития. – 2024. – № 6(156). – С. 45-49.

10. Терлецкий, А. С. Нейронные сети и искусственный интеллект: Основы нейронных сетей на языке Python / А. С. Терлецкий, Е. С. Терлецкая. – Липецк: Липецкого государственного педагогического университета имени П.П. Семенова-Тян-Шанского, 2023. – 76 с.
11. Шафорост, Н. В. Развертывание модели машинного обучения с использованием современного фреймворка FastaPI / Н. В. Шафорост, Д. В. Власов // Современное образование: традиции и инновации. – 2023. – № 3. – С. 102-106. – DOI 10.51623/23132027_2303_102.
12. Bootstrap · Самая популярная HTML, CSS и JS библиотека. URL: <https://getbootstrap.su/> (дата обращения: 26.10.2025).
13. Deshmukh, B. Text summarization using Python NLTK / B. Deshmukh // International Journal of Advanced Research. – 2022. – Vol. 10, No. 06. – P. 202-209. – DOI 10.21474/ijar01/14876.
14. Madani A. 2023. Debugging Machine Learning Models with Python. Develop high-performance, low bias, and explainable machine learning and deep learning models. Birmingham: Packt Publishing Ltd., 344 p.
15. Yuxi (Hayden) Liu. 2020. Python Machine Learning By Example. Third Edition. Build intelligent systems using Python, Tensor Flow 2, PyTorch, and scikit-learn. Birmingham: Packt Publishing Ltd., 526 p.

References inCyrillics

1. Adaptivnaya vyorstka sajta. URL: <https://html5book.ru/adaptivnaya-vyorstka-sajta/> (data obrashcheniya: 26.04.2025). Vorob'ev, V. A. Issledovanie vozmozhnostej tekhnologij nejrosetevoj obrabotki graficheskikh dannyh s razlichnoj detalizaciej / V. A.
2. Andreeva, N. N. Opticheskoe raspoznavanie simvolov s primeneniem mashinnogo obucheniya v oblasti proektirovaniya / N. N. Andreeva, N. A. Goncharenko, M. A. Dias // Avtomatizaciya i informatizaciya TEK. – 2024. – № 5(610). – S. 36-40.
3. Biblioteka Chart.js dlya WEB. URL: <https://java-online.ru/chartjs.xhtml> (data obrashcheniya: 26.10.2025).
4. Veb-frejmvork FastAPI. URL: <https://fastapi.tiangolo.com/> (data obrashcheniya: 26.10.2025).
5. Vorob'ev // Innovacionnaya nauka. – 2025. – № 6-1. – S. 58-64.
6. Elhimova, I. S. Opticheskoe raspoznavanie simvolov s pomoshch'yu Tesseract / I. S. Elhimova, A. I. Kanev // Tendencii razvitiya nauki i obrazovaniya. – 2022. – № 92-10. – S. 71-75. – DOI 10.18411/trnio-12-2022-480.
7. Kucenko, S. M. Veb-prilozhenie kak sredstvo optimizacii biznes-processov v organizacijah / S. M. Kucenko, E. A. Saltanaeva // Ekonomika i predprinimatel'stvo. – 2025. – № 1(174). – S. 1008-1015. – DOI 10.34925/EIP.2024.174.1.180.
8. Kucenko, S. M. Vybor instrumentov postroeniya nejronnoj seti / S. M. Kucenko, E. A. Saltanaeva // Nauchno-tekhnicheskij vestnik Povolzh'ya. – 2023. – № 12. – S. 313-315.
9. Pal'mov, S. V. Sravnitel'nyj analiz bibliotek PYMORPHY3 i PYMYSTEM3 / S. V. Pal'mov, R. R. Salihov // Nauka i biznes: puti razvitiya. – 2024. – № 6(156). – S. 45-49.
10. Terleckij, A. S. Nejronnye seti i iskusstvennyj intellekt: Osnovy nejronnyh setej na yazyke Python / A. S. Terleckij, E. S. Terleckaya. – Lipeck: Lipeckogo gosudarstvennogo pedagogicheskogo universiteta imeni P.P. Semanova-Tyan-Shanskogo, 2023. – 76 s.
11. Shaforost, N. V. Razvertyvanie modeli mashinnogo obucheniya s ispol'zovaniem sovremennogo frejmvorka FastaPI / N. V. Shaforost, D. V. Vlasov // Sovremennoe obrazovanie: tradicii i innovacii. – 2023. – № 3. – S. 102-106. – DOI 10.51623/23132027_2303_102.

12. Bootstrap · Samaya populyarnaya HTML, CSS i JS biblioteka. URL: <https://getbootstrap.su/> (data obrashcheniya: 26.10.2025).
13. Deshmukh, B. Text summarization using Python NLTK / B. Deshmukh // International Journal of Advanced Research. – 2022. – Vol. 10, No. 06. – P. 202-209. – DOI 10.21474/ijar01/14876.
14. Madani A. 2023. Debugging Machine Learning Models with Python. Develop high-performance, low bias, and explainable machine learning and deep learning models. Birmingham: Packt Publishing Ltd., 344 p.
15. Yuxi (Hayden) Liu. 2020. Python Machine Learning By Example. Third Edition. Build intelligent systems using Python, Tensor Flow 2, PyTorch, and scikit-learn. Birmingham: Packt Publishing Ltd., 526 p.