МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ РОССИЙСКОЙ ФЕДЕРАЦИИ

ИНФОРМАТИКА И КИБЕРНЕТИКА

2 (40)

Донецк – 2025

УДК 004.3+004.9+004.2+51.7+519.6+519.7

ИНФОРМАТИКА И КИБЕРНЕТИКА, № 2 (40), 2025,
Донецк, ДонНТУ.

Выпуск подготовлен по материалам XVI Международной научно-технической конференции «Информатика, управляющие системы, математическое и компьютерное моделирование – 2025» (ИУСМКМ–2025), проведенной 28 мая 2025 г. в рамках XI Научного форума Донецкой Народной Республики, а также результатам текущей научно-технической деятельности аспирантов, соискателей и научных работников. Статьи посвящены вопросам приоритетных направлений в области компьютерных наук, информатики, кибернетики, вычислительной техники и интеллектуальных систем.

Материалы предназначены для специалистов народного хозяйства, ученых, преподавателей, аспирантов и студентов высших учебных заведений.

Рекомендовано к печати ученым советом ФГБОУ ВО «Донецкий национальный технический университет» Министерства науки и высшего образования РФ. Протокол № 5 от 27 июня 2025 г.

Свидетельство о регистрации СМИ: серия ААА № 000145 от 20.06.2017.

Приказ МОН ДНР № 135 от 01.02.2019 о включении в Перечень рецензируемых научных изданий ВАК ДНР.

Контактный адрес редакции
РФ, ДНР, 283001, г. Донецк, ул. Артема, 58, ФГБОУ ВО «ДонНТУ»,
4-й учебный корпус, к. 36.
Тел.: +7 (856) 301-07-35, +7 (949) 334-89-11
Эл. почта: infcyb.donntu@yandex.ru
Интернет: http://infcyb.donntu.ru

СОДЕРЖАНИЕ

Информатика и вычислительная техника

Компьютерные науки

Информатика и вычислительная техника

УДК 004

Исследование методов прогнозирования добычи нефти

А. А. Сорокин, А. В. Гурко

Санкт-Петербургский Горный Университет императрицы Екатерины II
E-mail: s211920@stud.spmi.ru

Аннотация
В статье анализируются методы прогнозирования добычи нефти с применением нейронных сетей, выполняется компьютерный эксперимент оценки вероятности успешной добычи нефти на основе анализа геофизических параметров. Представлена архитектура нейронной модели, приведены результаты её обучения и тестирования. Направлением дальнейших исследований является расширение перечня входных параметров и обучение модели на других видах алгоритмов машинного обучения для повышения прогностических способностей модели

Введение

В нефтедобыче традиционно применяются методы гидродинамического моделирования, статистического анализа и эмпирические зависимости для прогнозирования дебита скважин и динамики добычи. Однако эти подходы часто ограничены в точности и адаптивности, особенно при учёте сложных геологических условий и временных изменений параметров разработки. В условиях динамично развивающейся нефтегазовой отрасли одной из ключевых задач является обеспечение стабильности и эффективности процессов добычи нефти.

Современные методы машинного обучения предоставляют новые возможности для повышения точности и оперативности прогнозирования. С развитием вычислительных технологий и машинного обучения методы прогнозирования добычи нефти претерпели значительные изменения. Одним из таких методов является использование нейросетевых моделей, которые учитывают сложные взаимосвязи между параметрами работы месторождения. Перспективными являются рекуррентные нейронные сети, такие как LSTM (long short-term memory)[1], которые эффективно моделируют временные зависимости. Замена традиционных подходов методами машинного обучения позволяют значительно повысить точность и оперативность прогнозирования. Например, в [1] исследуется использование искусственного интеллекта для автоматизации процессов разведки и добычи нефти.

В данном исследовании рассматриваются методы, такие как случайный лес и нейронные сети, для повышения точности прогноза и оптимизации процессов бурения. Вершинин В. Е. и Пономарев Р. Ю. показали, что нейросетевые технологии могут быть использованы для прогнозирования работы скважин в условиях нестационарного заводнения, что особенно актуально для оптимизации добычи в реальном времени. Это исследование подтверждает, что нейросетевые модели позволяют значительно сократить время вычислений, что критически важно для оперативного управления процессами [2]. В [3] показано, что нейросетевые модели подходят для анализа временных рядов, что позволяет более точно прогнозировать параметры работы добывающих скважин.

Другим подходом является использование компьютерного моделирования для прогнозирования добычи, например в. патенте № 2794707 [4] предлагается метод, который сочетает геофизические, гидродинамические и тепловые характеристики для более точных прогнозов объема добычи углеводородов. В [5, 6] подчеркивается важность применения машинного обучения для прогнозирования добычи нефти с использованием исторических данных. Использование описанных технологий позволяет не только ускорить процесс анализа, но и более точно предсказать уровни добычи, что является важным для принятия оперативных решений в процессе разработки месторождения. В статье [7] с помощью нейросетевого подхода улучшается точность прогнозов по добыче. В [8] показано, что применение машинного обучения позволяет повысить эффективность бурения, за счет прогнозирования механических нагрузок на оборудование. В работе [9] предложены методики построения математической модели, основанной на экспертных, что позволяет обобщить данные и использовать их для дальнейшего анализа эффективности процессов. Исследователи Негаш

[1] LSTM layer. https://keras.io/api/layers/recurrent_layers/lstm/ (Дата обращения: 13.05.2025)

Б. М. и Ява А. Д. в [10] предлагают использовать нейронные сети для прогнозирования добычи в условиях заводнения скважин. В [11] рассматриваются проблемы и возможности искусственного интеллекта в горнодобывающем секторе промышленности.

Применение интеллектуальной системы прогнозировании добычи нефти

Для выполнения прогнозирования добычи нефти по историческим геофизическим данным представлен новый метод, реализованный в виде интеллектуальной системы, использующей полносвязную нейронную сеть. Выбор архитектуры многослойной нейронной сети прямого распространения для построения интеллектуальной системы прогнозирования добычи нефти обусловлен её способностью моделировать сложные нелинейные зависимости между большим числом входных геофизических параметров и целевым показателем.

В настоящий момент общепринятой схемой обучения нейронной сети является применение нелинейной функций активации ReLU (rectified Linear Unit)[2] во внутренних слоях или сигмоидальной функции[3] на выходе модели. Математически функция активации ReLU определяется следующим образом:

$$ReLU(x) = \max(0, x), \quad (1)$$

где max – функция, возвращающая максимальное значение из двух.

Математически сигмовидная функция активации определяется следующим образом:

$$\sigma(x) = \frac{1}{1+e^{-x}}, \quad (2)$$

где x – входное значение нейрона (суммарное взвешенное значение после линейной комбинации входов), $\sigma(x)$ – выходное значение нейрона, интерпретируемое как вероятность, e – основание натурального логарифма.

Функция используется для преобразования выходного значения нейрона в вероятность, т.е. вероятность того, что входное значение относится к классу 1, если мы работаем с задачей бинарной классификации. Если значение сигмовидной функции близко к 1, то вероятность того, что входное значение относится к классу 1, высока. Если значение близко к 0, то вероятность того, что входное значение относится к классу 1, низкая.

После вычисления значения функции потерь начинается этап распространения ошибки от выходного слоя к входному слою. Алгоритм реализует метод градиентного спуска, позволяя пошагово корректировать значения весов и смещений в слоях сети на основе вычисленной ошибки между фактическим и предсказанным значениями. Обратное распространение включает последовательное применение правила цепного дифференцирования от выходного слоя к входным.

Процесс обновления весов и смещений в нейронной сети представляет собой ключевую часть алгоритма обучения. После того, как вычисляются производные функций потерь, параметры модифицируются в направлении, которое уменьшает значение функции потерь. Этот процесс происходит при помощи оптимизатора Adam (adaptive moment estimation)[4]. Он реализует адаптивное изменение шага обучения для каждого параметра на основе оценки первого и второго моментов градиентов. Алгоритм Adam корректирует скорость обучения для каждого параметра индивидуально на основе первого момента (математического ожидания) и второго момента (дисперсии) градиента. Это позволяет избежать резких колебаний при обучении и ускорить сходимость.

Для решения задачи программирования выбрана нейронная сеть FNN, состоящая из пяти слоев. 128 признаков, выходной признак 1. Также использованы функции активации - сигмоидная и ReLU. Фрагмент программного кода представлен ниже:

```
model = Sequential([
   Dense(128, activation='relu',
input_shape=(X_train.shape[1],)),
   BatchNormalization(),
   Dropout(0.3),
```

Данный слой использует функцию активации согласно формуле (1).

```
Dense(256, activation='relu'),
BatchNormalization(),
Dropout(0.4),
Dense(128, activation='relu'),
BatchNormalization(),
Dropout(0.3),
Dense(64, activation='relu'),
BatchNormalization(),
```

Данный слой использует функцию активации согласно формуле (2).

```
Dense(1, activation='sigmoid') ])
```

Использован оптимизатор Adam.

```
optimizer = Adam(learning_rate=0.001) //
model.compile(optimizer=optimizer, loss='mse',
        metrics=['mae',
tf.keras.metrics.RootMeanSquaredError()])
early_stopping = EarlyStopping(monitor='val_loss',
patience=20,
                restore_best_weights=True)
```

[2] Relu layer. https://keras.io/api/layers/activation_layers/relu/ (Дата обращения: 13.05.2025)

[3] Sigmoid function. https://keras.io/api/layers/activations/ (Дата обращения: 13.05.2025)

[4] Adam. https://keras.io/api/optimizers/adam/ (Дата обращения: 13.05.2025)

Предобработка и анализ данных

Для обучения и тестирования интеллектуальной системы прогнозирования использовался табличный набор данных, предоставленный нефтедобывающей компанией. Представленный набор данных состоит из набора следующих геофизических характеристик: географическая широта и долгота точки исследования, глубина скважины, пористость породы, пластовое давление. Для выполнения прогнозирования по заданному набору характеристик, была добавлена целевая переменная SuccessProb, которая описывает вероятность добычи нефти в рассматриваемой точке. На этапе предобработки реализована визуализация распределений, чтобы предварительно проанализировать потенциальные зависимости между признаками. Визуализация исходных данных представлена на рисунке 1.

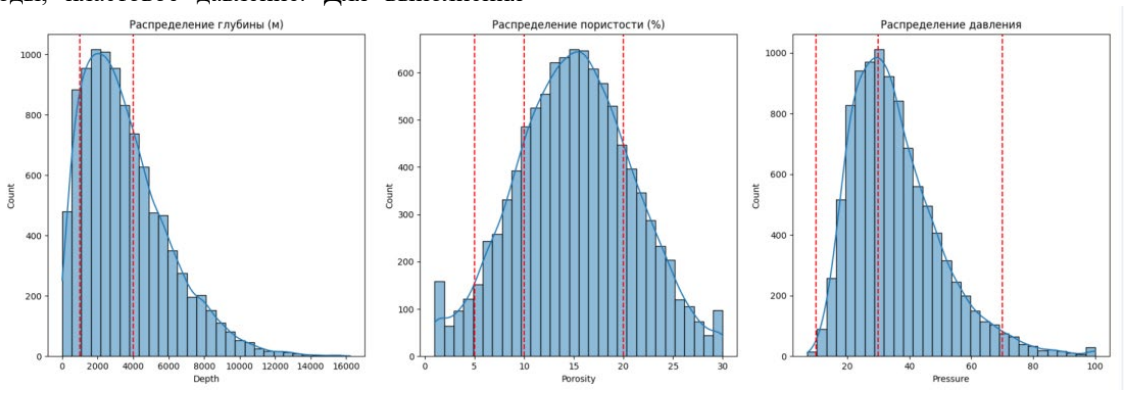


Рисунок 1 – Визуализация исходных данных [материал автора]

Для каждого из параметров были выделены пороговые значения, отражающие геологическую и производственную значимость интервалов.

По пористости выделены границы 5%, 10% и 20%, соответствующие общепринятой классификации пород по степени продуктивности. Это позволяет визуально оценить долю перспективных и неперспективных коллекторов в выборке.

Аналогично, на гистограмме давления пороговые значения в 10, 30 и 70 условных единиц обозначают зоны низкого, среднего и высокого пластового давления, что даёт возможность оценить условия разработки и необходимость использования дополнительных технологий.

На графике распределения глубин с помощью вертикальных линий акцентирован интервал 1000–4000 м., где сосредоточена основная масса данных, что позволяет выделить рабочую глубину основной выборки.

Такое представление данных способствует быстрой интерпретации и выявлению возможных аномалий или закономерностей. Чтобы данные корректно обрабатывались, их необходимо привести к виду [0,1], чтобы они были в едином масштабе. Каждый признак был линейно преобразован по формуле:

$$x_{\text{норм}} = \frac{x - x_{min}}{x_{max} - x_{\min}}, \qquad (3)$$

где x – исходное значение признака, x_{min} и x_{max} – минимальное и максимальное значения признака, $x_{норм}$ – нормализованное значение, лежащее в диапазоне от 0 до 1.

Визуализация нормализации данных на основе формулы (3) представлена на рисунке 2.

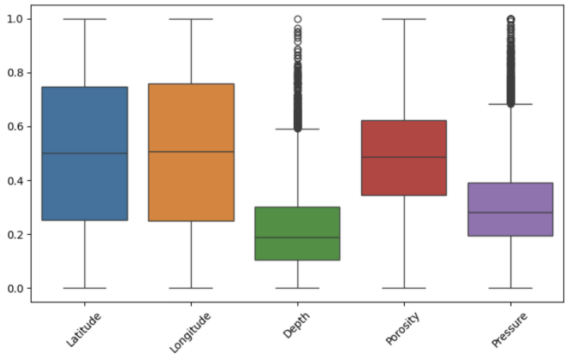


Рисунок 2 – Нормализация исходных данных [материалы авторов]

В процессе создания, обучения и тестирования модели нейронной сети было произведено разделение данных на обучающую (80%) и тестовую (20%) выборки.

После нормализации и разбиения данных на выборки они становятся готовыми для обучения модели нейронной сети.

Визуализация результатов обучения модели

Оценка точности модели производилась на тестовой выборке. На рисунке 3 представлены три ключевые визуализации, отражающие качество работы обученной нейронной сети. Первый график иллюстрирует соответствие между истинными и предсказанными значениями целевой переменной – чем ближе точки к диагонали y = x, тем выше точность модели.

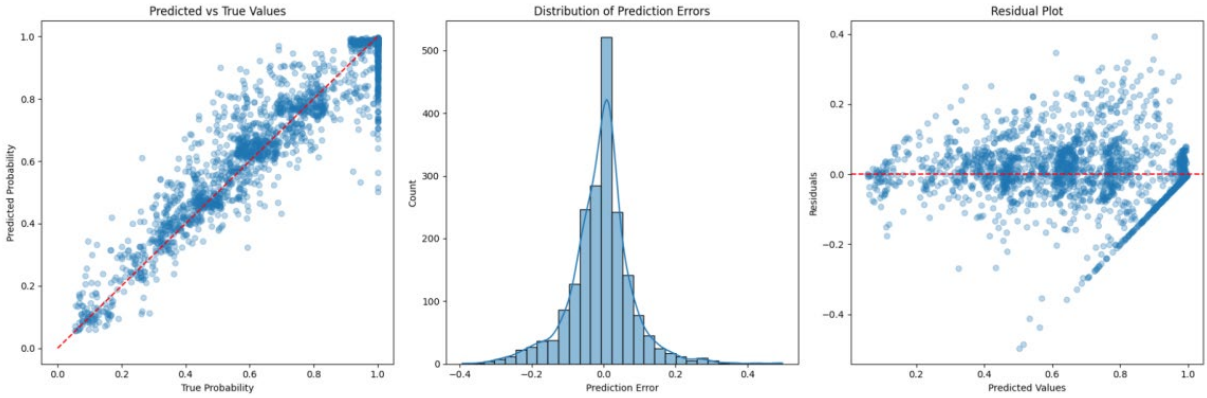


Рисунок 3 – Качество работы обученной нейронной сети [материалы авторов]

Второй график – гистограмма ошибок предсказания – демонстрирует распределение отклонений и позволяет оценить наличие систематических ошибок; её симметричная форма с пиком вблизи нуля указывает на стабильную работу модели. Третий график – график остатков – показывает, насколько равномерно распределены ошибки по диапазону предсказанных значений. Отсутствие ярко выраженного тренда и концентрация точек вдоль нулевой линии говорят о хорошем качестве предсказания без выраженного смещения.

Такая компоновка позволяет в сжатом виде оценить точность модели, характер ошибок и потенциальные проблемы, связанные с пере- или недообучением.

Рисунок 4 демонстрирует взаимосвязь между основными геологическими параметрами (глубина, давление, пористость) и вероятностью успешного бурения, предсказанной нейронной сетью. Все визуализации построены по единому принципу: положение точек на графиках определяется парой признаков, а цвет — предсказанным значением вероятности успеха.

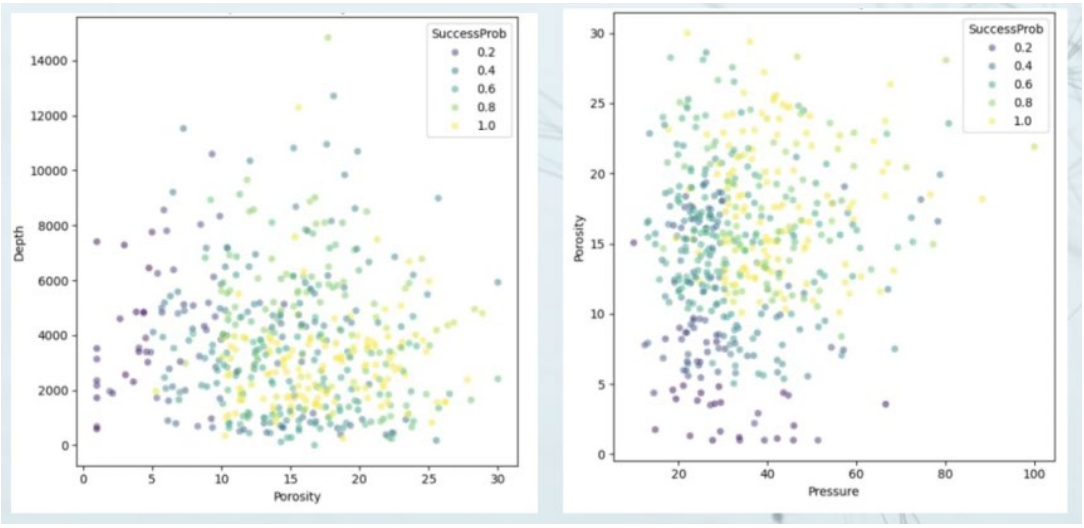


Рисунок 4 – Взаимосвязь геологических параметров [материалы авторов]

Использование цветовой палитры viridis позволяет визуально оценить уровень прогнозируемого успеха: от низкого (фиолетовые и тёмные оттенки) к высокому (жёлто-зелёные оттенки). Такая форма представления делает возможным быстрое выявление благоприятных зон сочетания параметров. Каждая точка соответствует конкретной скважине из выборки (500 случайных наблюдений), что обеспечивает баланс между плотностью информации и наглядностью графика.

Оценка качества модели

Для оценки качества модели используется среднеквадратичная ошибка (MSE). Большее значение среднеквадратического отклонения показывает больший разброс значений в представленном множестве со средней величиной множества; меньшее значение, соответственно, показывает, что значения в множестве сгруппированы вокруг среднего значения. Такой вид оценки качества удобно использовать для

выявления аномалий при обучении и работе модели нейронной сети.

Математически среднеквадратичная ошибка (MSE) определяется следующим образом:

$$MSE = \frac{1}{m}\sum_{i=1}^{m}(\gamma_i - \gamma_{ist_i})^2, \quad (4)$$

где m – число объектов в обучающей выборке, γ_i – истинное значение, γ_{ist} – предсказанное значение моделью.

Дополнительно, для контроля качества работы модели используется средняя абсолютная ошибка MAE и корень из среднеквадратической ошибки RMSE. Средняя абсолютная ошибка рассчитывается как среднее абсолютных разностей между целевыми значением и значением, предсказанным моделью на данном обучающем примере в процессе обучения:

$$MAE = \frac{1}{N}\sum_{i=1}^{m}\left|\gamma_i - \gamma_{ist_i}\right|, \quad (5)$$

где m – число объектов в обучающей выборке, γ_i – истинное значение, γ_{ist} – предсказанное значение моделью.

В отличие от среднеквадратических ошибок, где используется квадрат разности, MAE является линейной оценкой, поэтому вес разностей одинаков независимо от диапазона.

На рисунке 5 представлены графики динамики MSE, MAE согласно формулам (4), (5), которые позволяют оценить стабильность процесса обучения и соотношение ошибок на обучающей и валидационной (контрольной) выборках.

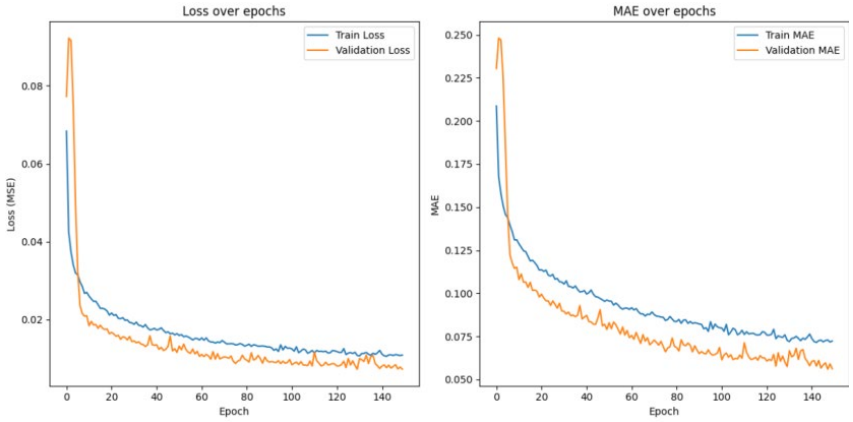


Рисунок 5 – Графики динамики MSE, MAE [материалы авторов]

Практическая реализация и результаты

На рисунке 6 представлена диаграмма географического распределения вероятности, которая служит итоговым продуктом работы нейронной сети.

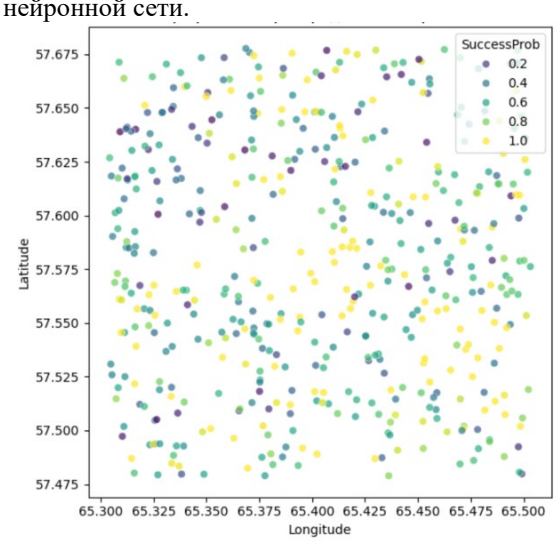


Рисунок 6 – Географическое распределение вероятности [материалы авторов]

На ней отображено визуальное распределение координат точек бурения, где каждая координата окрашена в свой цвет, что говорит о вероятности успеха бурения. На основании географического распределения на ограниченном объеме выборки была разработана интерактивная карта и пользовательский интерфейс для удобства взаимодействия системы и пользователя. На рисунке 7 представлен фрагмент интерактивной карты. На рисунке 8 представлен фрагмент интерактивной карты вблизи. В левом углу карты находится окно градации вероятности успеха добычи в конкретной точке. Основываясь на метриках гистограмм ошибок и графика остатков, можно сделать вывод, что модель имеет небольшие погрешности. Гистограмма ошибок показывает, что погрешности распределены равномерно, что может говорить о стабильности работы модели. График остатков показывает, что отклонение от действительных значений минимально. Несмотря на показатели метрик, для более точной и строгой оценки работы модели необходимо делать дополнительные тесты, в том числе на основе новых исходных данных.

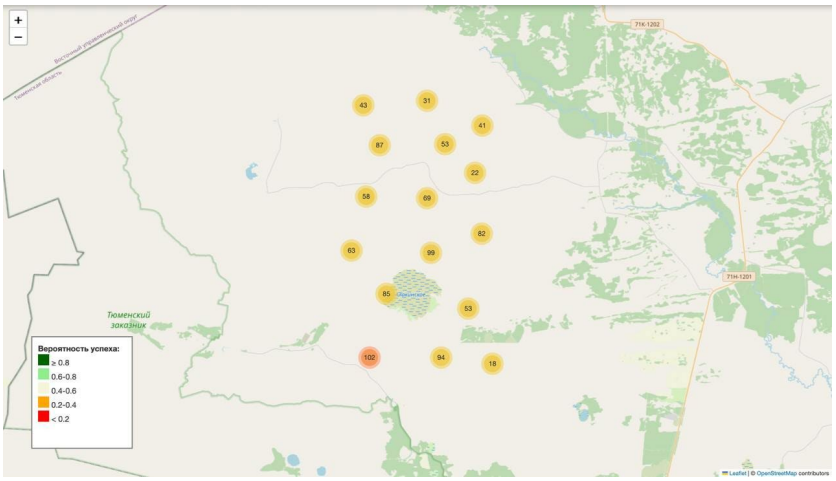


Рисунок 7 – Фрагмент интерактивной карты [материалы авторов]

Рисунок 8 – Приближенный фрагмент интерактивной карты [материалы авторов]

Чем больше будет объем исходных данных, тем больше вероятность научить модель работать корректно, без смещений.

Важным этапом работы модели является анализ географического распределения вероятности успешности добычи нефти. Благодаря визуализации, пользователь может проанализировать более подробно местность, которая учувствует в исследовании. Участки на карте находятся друг от друга не небольшом расстоянии, а это значит, что в условиях реальных исследований, модель может облегчить задачу инженерам в выполнении анализа местности.

Заключение

На основе анализа распределений и визуализации зависимостей были выявлены следующие зависимости:

– при низкой пористости (5%) вероятность успешной добычи резко снижается. Такие породы плохо проницаемы для нефти;

– при средней пористости (10 – 20%) вероятность успеха значительно возрастает, модель воспринимает такие значения как оптимальные;

– при высокой пористости (20%) вероятность также остаётся высокой, но может быть ограничена другими параметрами (например, слишком низким давлением);

– глубины от 1000 до 4000 м являются наиболее благоприятными – на этих уровнях залегают основные продуктивные пласты;

– в оптимальном диапазоне (30 – 70 бар) наблюдается пик успешности – такие условия обеспечивают хорошее движение флюидов.

– чрезмерное давление (70 бар) может оказывать отрицательный эффект, особенно если сопровождается неблагоприятными геологическими условиями.

Для повышения прогностических способностей модели в будущем целесообразно расширить перечень входных параметров и обучить модель на других видах алгоритмов машинного обучения.

Литература

1. Байбаров, Д. А. Оценка продуктивности и экономической эффективности технологий искусственного интеллекта для автоматизации

процессов разведки и добычи нефти и газа / Д. А. Байбаров // XXI век: итоги прошлого и проблемы настоящего плюс. – 2021. – Т. 10, № 3(55). – С. 100-105. – DOI 10.46548/21vek-2021-1055-0019.

2. Вершинин, В. Е. Нейросетевое моделирование Прогнозирование показателей добычи скважин в условиях нестационарного заводнения / В. Е. Вершинин, Р. Ю. Пономарев // Деловой журнал Neftegaz.RU. – 2022. – № 5-6(125-126). – С. 26-32. – EDN ZGEQVB.

3. Нейросетевое прогнозирование входных параметров при добыче нефти / Е. Д. Семенов, М. Я. Брагинский, Д. В. Тараканов, И. Л. Назарова // Вестник кибернетики. – 2023. – Т. 22, № 4. – С. 42-51. – DOI 10.35266/1999-7604-2023-4-6.

4. Патент № 2794707 C1 Российская Федерация, МПК E21B 41/00, E21B 49/00, G01V 99/00. Способ прогнозирования объемов добычи углеводородов из месторождений нефти и газа с использованием компьютерного моделирования: № 2022120987 : заявл. 02.08.2022 : опубл. 24.04.2023 / А. В. Бочкарев, Б. В. Васекин, Н. А. Воробьев [и др.] ; заявитель Общество с ограниченной ответственностью "Страта Солюшенс", Общество с ограниченной ответственностью "Физтех Геосервис".

5. Современные методы применения машинного обучения как инструмента прогнозирования добычи нефти / А. Р. Рустамов, Г. М. Пеньков, Д. Г. Петраков, М. А. Рустамова // Недропользование. – 2024. – Т. 24, № 1. – С. 44-50. – DOI 10.15593/2712-8008/2024.1.6.

6. Standards for Selection of Surfactant Compositions used in Completion and Stimulation Fluids / D. G. Petrakov, A. V. Loseva, N. T. Alikhanov, H. Jafarpour // International Journal of Engineering. – 2023. – Vol. 36, No. 9. – P. 1605-1610. – DOI 10.5829/ije.2023.36.09c.03.

7. Цыпленков, С. В. Нейросетевой подход к ранжированию факторов, влияющих на энергоэффективность добычи нефти / С. В. Цыпленков, Е. Д. Агафонов, Д. И. Цыпленкова // Интеллектуальные системы в производстве. – 2022. – Т. 20, № 1. – С. 22-28. – DOI 10.22213/2410-9304-2022-1-22-28. – EDN KVZQAT.

8. Development of Monitoring and Forecasting Technology Energy Efficiency of Well Drilling Using Mechanical Specific Energy / A. Kunshin, M. Dvoynikov, E. Timashev, V. Starikov // Energies. – 2022. – Vol. 15, No. 19. – P. 7408. – DOI 10.3390/en15197408.

9. Ilyushin, Y.V., Nosova, V.A. (2025). Development of Mathematical Model for Forecasting the Production Rate. International Journal of Engineering, Transactions B: Applications, 38(8), 1749-1757. https://doi.org/10.5829/ije.2025.38.08b.02

10. Negash, B. M. Artificial neural network-based production forecasting for a hydrocarbon reservoir under water injection / B. M. Negash, A. D. Yaw // Petroleum Exploration and Development. – 2020. – Vol. 47, No. 2. – P. 383-392. – DOI 10.1016/s1876-3804(20)60055-6.

11. Колцун, Н. В. Внедрение искусственного интеллекта в горнодобывающем секторе / Н. В. Колцун, А. В. Гурко // Анализ и прогнозирование систем управления в промышленности, на транспорте и в логистике, Санкт-Петербург, 23–25 апреля 2024 года. – СПб.: ООО "Медиапапир", 2024. – С. 223-229.

Сорокин А.А., Гурко А.В. Исследование методов прогнозирования добычи нефти*. В статье анализируются методы прогнозирования добычи нефти с применением нейронных сетей, выполняется компьютерный эксперимент оценки вероятности успешной добычи нефти на основе анализа геофизических параметров. Представлена архитектура нейронной модели, приведены результаты её обучения и тестирования. Направлением дальнейших исследований является расширение перечня входных параметров и обучение модели на других видах алгоритмов машинного обучения для повышения прогностических способностей модели.*

Ключевые слова*: интеллектуальная система, нейронные сети, машинное обучение, прогнозирование добычи нефти, нефтяная отрасль, глубокое обучение, автоматизация, моделирование, искусственный интеллект.*

Sorokin A.A., Gurko A.V. Methods of research of oil production forecasting. *The article considers modern forecasting methods in the oil industry, the use of neural networks in forecasting oil production, and the development of an application solution for automating the assessment of the probability of successful oil production based on the analysis of geophysical parameters. The architecture of the developed model is introduced, and the results of its training and testing are given.*

Keywords*: intelligent system, neural networks, machine learning, oil production forecasting, oil industry, deep learning, automation, modeling, artificial intelligence.*

Статья поступила в редакцию 15.05.2025
Рекомендована к публикации профессором Мальчевой Р. В.

УДК 004.4'2

Экономическое обоснование усовершенствования CVR-системы современными технологиями

Д. А. Алымов, А. В. Боднар, А. Р. Нестеренко
Донецкий национальный технический университет
кафедра программной инженерии
Email: linabykova13@ya.ru

Аннотация
В данной работе рассматривается разработка CRM-системы, ориентированной на массовое тиражирование и интеграцию современных технологий. Выбрана модель жизненного цикла разработки, рассчитаны трудозатраты по модели Фатреллу. Проведен анализ рисков, оценены затраты и показатели эффективности. Разработана структура команды. Полученные результаты демонстрируют возможность создания масштабируемого, экономически оправданного и конкурентоспособного решения для рынка CRM-систем.

Введение

В условиях стремительного роста цифровизации бизнес-процессов потребность в гибких, масштабируемых и безопасных CRM-системах становится одной из ключевых задач для компаний различных сфер деятельности. Классические CRM-решения успешно функционируют на рынке, однако с увеличением объёмов данных, ужесточением требований к защите информации и необходимостью интеллектуального анализа возникает потребность в их усовершенствовании.

Современные технологии — такие как блокчейн, искусственный интеллект (ИИ), облачные хранилища данных, высокопроизводительные протоколы взаимодействия (например, gRPC) — предоставляют широкий спектр инструментов для повышения надёжности, автоматизации и адаптивности CRM-систем. Их применение открывает возможности не только для улучшения безопасности хранения и передачи информации, но и для интеллектуального взаимодействия с клиентами, оптимизации бизнес-аналитики и повышения прозрачности процессов.

Краткий SWOT-анализ

SWOT-анализ позволяет определить сильные и слабые стороны проекта, а также выявить внешние возможности и угрозы, которые могут повлиять на успех CRM-системы на стадии массового тиражирования (табл. 1) [1].

Таблица 1 – SWOT-анализ

Фактор	Описание
Strengths (Сильные стороны)	- Внедрение современных технологий увеличивает конкурентоспособность продукта на рынке. - Возможность интеграции с популярными экосистемами и модулями.
Weaknesses (Слабые стороны)	- Высокие начальные инвестиции на разработку, тестирование и продвижение на рынок. - Риски перегрузки системы функциональностью, что может усложнить её освоение пользователями.
Opportunities (Возможности)	- Рост спроса на модифицированные CRM-решения в эпоху удалённой работы и гибридных моделей управления.
Threats (Угрозы)	- Конкуренция со стороны крупных игроков. - Быстрое устаревание технологий, необходимость регулярных обновлений.

Риски. Риск — это возможное событие, которое может негативно повлиять на проект (по срокам, бюджету, качеству, безопасности и т.д.). Для управления рисками важно оценивать их и приоритизировать, чтобы вовремя принимать меры (см. ф. 1).

$$\text{Оценка риска} = P \times I. \qquad (1)$$

Технические риски (табл.2, 3**).** Первая из возможных внедряемых технологий – блокчейн. Несмотря на безопасность и прозрачность, реализация приватной блокчейн-сети требует устойчивой архитектуры валидаторов, узлов, смарт-контрактов, а также грамотного разграничения прав. Возможны ошибки при реализации логики блоков, риски нарушения

консенсуса, перегрузки сети, недостаточной скорости транзакций и проблем с масштабируемостью хранилищ.

Следующая технология, ИИ, может быть использована в CRM для интеллектуального анализа клиентов, прогнозирования продаж, автоматизации взаимодействия. Однако здесь возникают риски: ошибки в обучении моделей, некачественные или необъективные данные, высокая сложность отладки и интерпретации результатов. Риски в использовании технологии облачных хранилищ касаются прежде всего доступности, защиты данных и зависимости от облачного провайдера. Возможны сбои в работе облачных сервисов, уязвимости в системах разграничения прав, сбои в синхронизации данных между клиентами.

Для оптимизации серверного взаимодействия может быть интегрирована технология gRPC это высокопроизводительный протокол взаимодействия между микросервисами, но он требует высокой квалификации разработчиков. Технические риски включают сложности в отладке и мониторинге межсервисного взаимодействия, уязвимости при прямом открытии портов, возможные трудности в обеспечении совместимости между языками.

Метод FMEA (см. ф. 1) [2].

$$RPN = Severity \times Occurrence \times Detection. \quad (2)$$

RPN > 200 = высокий риск, требует обязательного снижения (например, аудит кода, доп. тесты).

Таблица 2 – Основные технические риски

Технология	**Риск**	**Вероятность (P)**	**Ущерб (I)**	**Оценка (P × I)**
gRPC	Несовместимость между микросервисами	0.6	100,000	60,000
gRPC	Ошибка в реализации протокола	0.4	150,000	60,000
gRPC	Нехватка специалистов по gRPC	0.3	120,000	36,000
Облачные технологии	Утечка данных при неправильной настройке доступа	0.3	250000	75000
Облачные технологии	Резкое удорожание тарифа провайдера	0.4	100000	40000
Блокчейн	Проблемы совместимости с другими системами	0.3	180000	54000
Блокчейн	Ошибки в смарт-контрактах	0.2	300000	60000
Блокчейн	Повышенные издержки из-за перегрузки сети	0.4	120000	48000
ИИ	Ошибочные предсказания модели в критичных блоках	0.5	160000	80000
ИИ	Недостаточная обучающая выборка	0.3	110000	33000

Таблица 3 – Основные технические риски

Технология	Риск	Severity	Occurrence	Detection	**RPN**
gRPC	Несовместимость между микросервисами	6	6	6	216
gRPC	Ошибка в реализации протокола	8	4	7	224
gRPC	Нехватка специалистов по gRPC	5	3	6	90
Облако	Утечка данных при неправильной настройке доступа	9	3	8	216
Облако	Резкое удорожание тарифа провайдера	4	4	5	80
Блокчейн	Проблемы совместимости с другими системами	6	3	5	90
Блокчейн	Ошибки в смарт-контрактах	9	2	8	144
Блокчейн	Повышенные издержки из-за перегрузки сети	5	4	6	120
ИИ	Ошибочные предсказания в критичных блоках	8	5	7	280
ИИ	Недостаточная обучающая выборка	6	3	5	90

Экономические риски (табл. 4). Внедрение блокчейна в CRM связано с высоким порогом начальных инвестиций: разработка или настройка приватной сети, внедрение смарт-контрактов, поддержка распределённых узлов, обучение специалистов и обеспечение соответствующей инфраструктуры.

Разработка и обучение ИИ-моделей требует существенных инвестиций в инфраструктуру, в привлечение специалистов, в сбор и очистку данных. Дополнительно, ИИ-модули требуют регулярного обновления, дообучения, что влечёт постоянные затраты.

Облачные решения часто ассоциируются с снижением затрат, в длительной перспективе они могут приводить к перерасходу бюджета. Это может быть связано с непредсказуемым ростом затрат при масштабировании, со скрытыми расходами на обслуживание, лицензии, премиум-функции.

gRPC требует квалифицированной команды, особенно при построении микросервисной архитектуры. Экономические риски здесь заключаются в необходимости найма специалистов со знанием gRPC. Ошибки на этапе проектирования архитектуры приводят к дорогостоящим переделкам. Также могут потребоваться отдельные расходы на инструменты мониторинга, трассировки, аутентификации и совместимости сервисов.

Таблица 4 – Основные экономические риски

Технология	Риск	Вероятность	Ущерб (руб.)	Оценка риска
ИИ	Перерасход бюджета на обучение моделей	0.5	200000	100000
ИИ	Расходы на инфраструктуру (GPU и др.)	0.3	250000	75000
gRPC	Повышение стоимости интеграции с системами	0.4	180000	72000
gRPC	Простой бизнес-процессов из-за ошибок в вызовах	0.3	150000	45000
gRPC	Перерасход на найм специалистов gRPC	0.2	160000	32000
Облачные технологии	Рост расходов на аренду облака	0.5	220000	110000
Облачные технологии	Плата за хранение дополнительных данных	0.3	140000	42000
Блокчейн	Удорожание поддержки инфраструктуры	0.3	200000	60000
Блокчейн	Недоверие со стороны клиентов и потеря части сегмента	0.2	400000	80000

Организационные риски (табл. 5)**.** Внедрение блокчейна в CRM-систему требует изменения организационных процессов: распределение ответственности между отделами, появление роли администратора цепи, соблюдение процедур валидации данных и их шифрования. Возникают сложности с регулированием доступа, правами модификации и контроля за целостностью записей.

ИИ изменяет распределение задач в команде и взаимодействие с пользователями. Требуется переобучение персонала для работы с новыми инструментами, что требует временных затрат и создает риск непринятия технологии. В процессе эксплуатации может быть сложно установить ответственность за действия системы, если решения частично принимаются ИИ.

Переход на облака требует реорганизации IT-отдела: снижается нагрузка на системных администраторов, но возникает необходимость в облачных инженерах. Необходимость постоянного контроля за расходами и соблюдение политик использования облаков требуют изменения управленческих процессов.

При применении gRPC Проект становится более распределенным, требуется согласование интерфейсов между командами, четкое определение зон ответственности. При отсутствии командного взаимодействия высок риск дублирования функций, несовместимости API, нарушений в коммуникации.

Таблица 5– Основные **организационные** риски

Риск	Вероятность	Ущерб (руб.)	Оценка риска
Конфликты между отделами	0.4	100000	40000
Сопротивление сотрудников изменениям	0.6	150000	90000
Недостаток компетенций	0.5	120000	60000
Высокая текучесть кадров	0.3	180000	54000
Ошибки в управлении сроками и задачами	0.5	140000	70000

Разработка проекта

Обоснование выбора спиральной модели разработки. Для разработки высокотехнологичной CRM-системы, ориентированной на массовое тиражирование и включающей в себя современные технологии, наиболее целесообразным выбором является спиральная модель жизненного цикла программного обеспечения. Данная модель объединяет элементы каскадного и итерационного подходов, при этом основной акцент делает на управление рисками, постепенное развитие и возможность адаптации на каждом витке жизненного цикла проекта.

Каждый цикл спирали охватывает полный набор действий: планирование, анализ, проектирование, реализация, тестирование и оценку. Это позволяет разрабатывать продукт частями.

За счёт регулярной итерации и ревизии каждой версии продукта можно оперативно выявлять технические проблемы, несовместимости или неэффективность отдельных технологий. На каждом этапе осуществляется оценка стоимости, времени, ресурсов, что позволяет оптимизировать проект и избежать перерасхода бюджета, особенно актуально при массовом тиражировании [3].

Определение размера проекта. Одним из наиболее признанных подходов определения размера проекта является методика Р. Фатрелла, базирующаяся на анализе объема функциональности, сложности системы и организационных факторов. Учитывая показатели, проект имеет средний размер.

Таблица 6 – Основные критерии разработки проекта

Критерий	**Оценка для нашего проекта**
Количество строк кода (SLOC)	От 60 000 до 70 000 SLOC
Продолжительность проекта	10 000 человеко-часов
Команда	6 специалистов
Сложность интеграций	Умеренная
Новизна технологий	Высокая
Объем документации	Средний
Объем UI	Средний
Количество модулей	9–10 крупных модулей
Масштаб распространения	Высокий

Человеко-часы, продолжительность, численность команды. Для проекта среднего масштаба, реализующего CRM-систему с использованием блокчейна, ИИ, облачных хранилищ и gRPC, можно применить методику оценки трудоёмкости на основе человеко-часов (ч/ч) и связать её с продолжительностью и размером команды.

Расчётное обоснование (сценарий среднего проекта):

Общий объём проекта: ≈ 10 000 чел-ч.

Продолжительность проекта: 12 месяцев (≈ 48 недель)

Рабочая неделя одного специалиста: 40 ч.

Общая численность команды: 6 чел. (в среднем)

6 чел. × 40 ч/нед × 48 нед = 11 520 ч (что даёт небольшой резерв 10–15%).

Итого: 10 000 ч.

Таблица 7 – Этапы разработки проекта

Этап	Описание	Человеко-часы
Анализ и проектирование	Сбор требований, проектирование архитектуры, диаграммы, UI-прототипы	1 200 ч
Разработка модулей	Основная часть серверной и клиентской логики	4 200 ч
Интеграция и API	Интеграция с внешними сервисами, реализация gRPC и облачного взаимодействия	1 300 ч
Тестирование	Модульное, интеграционное, нагрузочное тестирование	1 200 ч
UI-дизайн и фронтенд	Разработка интерфейса, мобильной и веб-версии	1 000 ч
Документация и DevOps	Документация, CI/CD, деплой, мониторинг	600 ч
Внедрение и сопровождение	Первичная установка, обучение, поддержка	500 ч

Распределение задач между специалистами. Для эффективной реализации проекта потребуется междисциплинарная команда, включающая как технических, так и проектных специалистов (табл. 8).

Таблица 8 – Список сотрудников

Должность	Кол-во	Задачи
Руководитель проекта	1	Координация команды, контроль сроков, работа с заказчиком, управление рисками
Системный архитектор	1	Определение архитектуры, проектирование взаимодействия модулей, безопасность
Backend-разработчики	2	Разработка ядра CRM, реализация бизнес-логики, блокчейн, API (gRPC)
Frontend-разработчик	1	Реализация пользовательского интерфейса, адаптивность, интеграция с backend
ML/AI-инженер	1	Разработка и обучение моделей, внедрение ИИ-модулей в CRM
DevOps-инженер	0.5	Настройка CI/CD, контейнеризация, деплой, мониторинг
Тестировщик	1	Подготовка тест-кейсов, ручное и автоматическое тестирование
Технический писатель	0.5	Подготовка документации

Итог: команда из 6 человек способна реализовать проект CRM-системы за около 10 000 ч / 12 месяцев, при этом обеспечивая модульность, инновационность и возможность масштабирования. Резерв времени и человеко-часов обеспечивает гибкость для адаптации под рынок или заказчика.

Декомпозиция по стадиям. Этап проектирования. На стадии проектирования осуществляется сбор и систематизация требований к системе, как функциональных, так и нефункциональных (надежность, масштабируемость, безопасность). Проводится анализ аналогичных решений и формулируются ключевые отличия. Создаются предварительные технические спецификации, структура базы данных, схема взаимодействия компонентов. Особое внимание уделяется архитектуре системы с учетом распределённого хранения, механизмов идентификации и доступа в блокчейн-среде, а также требованиям к масштабированию через gRPC и облачную инфраструктуру. Разрабатываются интерфейсные прототипы, бизнес-процессы, составляется документация. В рамках спирального подхода проектирование делится на итерации, каждая из которых завершается уточнением архитектурных решений.

Этап программирования. Этап программирования начинается с создания базовой инфраструктуры проекта: структуры каталогов, подключение к облачному окружению, настройка репозитория и CI/CD. Параллельно ведется реализация основных функциональных модулей:

1) ядро CRM (управление контактами, лидами, сделками);
2) модуль идентификации и записи в приватный блокчейн;
3) реализация REST/gRPC-интерфейсов и API для взаимодействия между модулями;
4) внедрение алгоритмов ИИ: обработка лидов, прогнозирование активности клиентов;
5) подключение облачных хранилищ для загрузки и хранения документов;
6) реализация интерфейса на веб- и/или мобильной платформе.

Программирование ведётся по итерационному принципу, позволяя адаптировать код и архитектуру под меняющиеся требования. После каждой итерации код проходит код-ревью и тестирование.

Этап тестирования. Тестирование включает в себя несколько уровней: модульное, интеграционное, системное и приёмочное. В начале разрабатываются тест-кейсы и сценарии на основе функциональных требований. Модульное тестирование позволяет выявить ошибки в логике отдельных компонентов, включая блоки шифрования, алгоритмы машинного обучения, и взаимодействие с API. Далее, при помощи автоматических тестов и ручных проверок, проводится интеграционное тестирование — проверка работы системы при взаимодействии между модулями. Отдельное внимание уделяется нагрузочному тестированию, особенно при взаимодействии через gRPC и при обращениях к блокчейн-модулю. Заключительная фаза — пользовательское тестирование, в ходе которого выявляются ошибки интерфейса, логики и поведенческих паттернов.

Этап интеграции. На этапе интеграции происходит объединение всех ранее протестированных модулей в единую систему. Включается настройка маршрутизации, доступов, удостоверяющих центров (если применимо), шифрования каналов связи. Интеграция охватывает как внутренние сервисы, так и внешние. Настраивается инфраструктура CI/CD для автоматической доставки обновлений, откатов и восстановления в случае ошибок.

Особое внимание уделяется отказоустойчивости и возможности масштабирования системы. Также производится интеграция с аналитикой и логированием, необходимыми для сопровождения и поддержки продукта на этапе внедрения.

Этап внедрения. Финальный этап жизненного цикла проекта — внедрение CRM-системы в рабочую среду. Он включает установку системы на стороне заказчика (или развертывание в облаке), настройку пользовательских ролей и прав доступа, загрузку первоначальных данных, а также обучение персонала. Внедрение сопровождается технической поддержкой, разбором обратной связи и первичной адаптацией под нужды конкретного клиента. Также проводится аудит безопасности и эффективности системы. Разрабатывается и публикуется пользовательская и техническая документация. В сценарии массового тиражирования данный этап дополняется созданием шаблонов развертывания, автоматической регистрации лицензий, а также маркетинговыми материалами и поддержкой версий (релиз-менеджмент) [4].

Диаграмма Ганта. Диаграмма Ганта — это визуальный инструмент планирования, который отображает задачи проекта на временной шкале, показывая их продолжительность, последовательность и зависимости. Она помогает быстро увидеть, какие задачи выполняются в какой момент, какие из них идут параллельно, а какие зависят друг от друга. Далее будет представлена диаграмма (см. рис. 1), спроектированная по представленному плану.

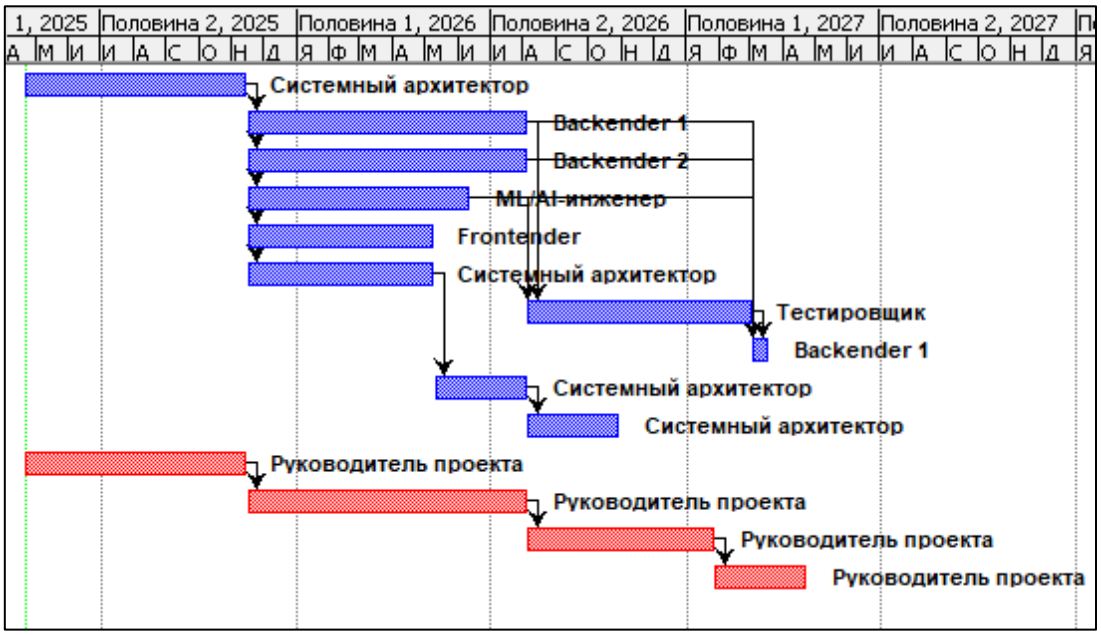


Рисунок 1 – Диаграмма Ганта

LOC, функциональные точки. Оценка объема программного обеспечения может производиться по метрикам LOC (объем кода в строках) и FP (функциональные точки), отражающие функциональность, а не только размер кода [7, 8]. LOC и функциональные точки приведены в табл. 9.

Расчёт затрат и показателей экономической эффективности. Для оценки экономической эффективности используются различные метрики. Рассмотрим одну из самых популярных метрик – ROI, показатель, который используется для оценки эффективности инвестиций. Он помогает определить, насколько прибыльными были вложенные средства, и позволяет сравнить различные инвестиционные возможности. ROI показывает, сколько прибыли можно получить с каждой вложенной единицы капитала.

Таблица 9 – LOC и функциональные точки

Подсистема	LOC (оценка)	Функц. точки
Модуль управления клиентами (CRM Core)	20 000	120
Модуль блокчейн	10 000	65
Модуль ИИ (рекомендации, чат-боты)	15 000	80
Интеграция с облаком (облачное хранилище)	8 000	40
Коммуникации по gRPC между сервисами	5 000	35
Интерфейс администратора и пользователя	12 000	60
Всего	**70 000**	**400+**

Следующий важный показатель – NPV, т.е. это чистая приведенная стоимость, которая используется для оценки привлекательности проекта или инвестиции. Она рассчитывается как разница между приведенной стоимостью всех поступлений от проекта и приведенной стоимостью всех расходов.

CAC — это стоимость привлечения одного нового клиента. Этот показатель является ключевым для оценки эффективности маркетинговых и продажных кампаний. В случае с IT-проектами CAC помогает понять, сколько компания тратит на различные расходы, связанные с привлечением потенциальных клиентов.

Важно, чтобы CAC оставался как можно ниже, иначе компания может столкнуться с проблемами в долгосрочной прибыльности. Для снижения CAC используются различные стратегии, такие как оптимизация каналов привлечения, автоматизация маркетинга или улучшение качества продуктов для повышения лояльности клиентов.

LTV — это показатель, который измеряет общую прибыль, которую компания может ожидать от одного клиента за весь срок его отношений с компанией. Он позволяет понять, сколько стоит удержание клиента и какую прибыль он принесет в течение своего "жизненного цикла". Высокий LTV говорит о том, что клиент будет приносить значительные доходы на протяжении длительного времени.

ARPU — это средний доход, который компания получает от одного пользователя или клиента за определенный период времени, обычно месяц или год. ARPU помогает оценить эффективность монетизации, а также может быть использован для оценки влияния различных маркетинговых и продуктовых стратегий на доходность [5, 6].

Возьмем для примера некоторые основные (табл. 10) и косвенные затраты (табл. 11).

Предположим, что планируемая выручка от подписки на CRM в течение года: 15 000 000 руб., а итоговые затраты: 8 500 000 + 1 500 000 = 10 000 000 руб.

Таблица 10 – Основные затраты

Статья затрат	Сумма (руб.) в год
Зарплаты проектной команды	6 000 000
Серверы и облачные сервисы	800 000
Закупка лицензий и ПО	300 000
Маркетинг, реклама, продвижение	1 200 000
Регистрация, юридические услуги	200 000
Итого	**8 500 000**

Таблица 11 – Дополнительные затраты

Статья затрат	Сумма (руб.) в год
Обучение персонала, документация	400 000
Техподдержка и сопровождение	600 000
Инфраструктура (электричество, офис)	300 000
Внеплановые работы и резервы	200 000
Итого	**1 500 000**

В таком случае показатель ROI = (15 000 000 – 10 000 000) / 10 000 000 = 0.5 или 50% (см. ф. 3) [11].

$$ROI = \frac{(\text{Доход} - \text{Затраты})}{\text{Затраты}}. \quad (3)$$

Далее рассмотрим NPV при таких же показателях сроком на 4 года с ставкой дисконтирования 12% = 0.12. NPV представляет два варианта расчета, где первый предполагает только начальные инвестиции (см. ф. 4), а второй расчитывает с учетом ежегодных расходов на создание и поддержание проекта (см. ф. 5) [12].

$$\text{NPV} = \frac{\sum_{i=1}^{n}(\text{Доход}_i)}{(1+r)^i} - \text{Инвестиции}. \quad (4)$$

$$\text{NPV} = \frac{\sum_{i=1}^{n}(\text{Доход}_i - \text{Расход}_i)}{(1+r)^i}. \quad (5)$$

Учитывая ежегодные расходы, выбираем формулу для расчета (см. ф. 5) и рассчитываем итоговый NPV с промежуточным результированием (табл. 12).

Таблица 12 – Сумма доходов и расходов

Год	Доход	Расход	Чистый доход	Дисконтированный доход
1	15M	10M	5M	4.46M
2	17M	10M	7M	5.58M
3	19M	11M	8M	5.70M
4	21M	12M	9M	5.71M
Итого NPV				**21,45M руб.**

Следующим будет показатель LTV (см. ф. 6), который включает в расчеты показатель ARPU [9]. Предположим, что стоимость подписки приблизительно 1500 руб., а маржа (разница между затратами на производство и выручкой от продажи товаров и услуг) 70 %, при том, что средний жизненный цикл клиента 2 года.

$$\text{LTV} = ARPU \times \text{маржа} \times \text{средний жизненный цикл клиента}. \quad (6)$$

В таком случае LTV = 1500 × 0.7 × 24 (месяца) = 25 200 руб.

Ранее были определены возможные затраты на маркетинг (см. табл. 11). Предположим, что получилось привлечь тысячу новых клиентов. Тогда показатель CAC = 1 200 000 / 1000 = 1 200 руб. (см. ф. 7).

$$\text{CAC} = \frac{\text{Затраты на маркетинг}}{\text{Кол} - \text{во привлеченных клиентов}}. \quad (7)$$

В итоге рассчитаем ключевой показатель эффективности (см. ф. 8). Ключевой показатель эффективности – это отношение пожизненной ценности клиента к стоимости его привлечения. Если этот показатель меньше 1, то происходит потеря денег на каждом клиенте (выручка меньше затрат), если он равен 1, то это является окупаемостью, все что выше этого значения является показателем высокой эффективности [10].

$$\text{Ключевой показатель} = \frac{LTV}{CAC}. \quad (8)$$

Ключевой показатель = 25 200 / 1 200 = 21, что означает большие возможности масштабирования проекта в дальнейшем.

Вывод

В ходе разработки CRM-системы для массового тиражирования был проведен всесторонний анализ различных аспектов проекта, включая оценку рисков, выбор жизненного цикла разработки и расчет экономической эффективности. На основе данных исследований была выбрана спиральная модель разработки, которая позволила минимизировать риски и адаптировать продукт к изменяющимся условиям рынка.

Рынок CRM-систем представляет собой высококонкурентную среду, где крупные игроки предлагают широкий спектр решений. Тем не менее, существует потенциал для создания нового продукта, который будет сочетать лучшие практики существующих решений с уникальными инновациями, ориентированными на потребности малого и среднего бизнеса.

Этапы разработки, включая проектирование, программирование, тестирование и внедрение, были тщательно спланированы с учетом всех возможных затрат, ресурсов и сроков. Важной частью работы было внимание к экономической эффективности проекта.

Расчет прямых и косвенных затрат, а также различных экономических показателей выстроить четкую финансовую модель, которая может быть использована для оценки прибыльности проекта на всех этапах его реализации.

Таким образом, выбранный подход и обоснования для создания новой CRM-системы представляют собой обоснованный и детализированный план, который учитывает все ключевые аспекты — от разработки и тестирования до внедрения и последующего сопровождения. Проект имеет хорошие перспективы для успешной реализации и дальнейшего масштабирования на рынке.

Литература (7)

1. Bensoussan, Babette E. Analysis Without Paralysis: 10 Tools to Make Better Strategic Decisions / Babette E. Bensoussan, Craig S. Fleisher // Published by Pearson, 2008. – 11 с.

2. Levin, Mark A. Improving Product Reliability and Software Quality, 2nd Edition / Mark A. Levin, Ted T. Kalal, Jonathan Rodin // Published by Wiley, 2019. – 6 с.

3. Алексеев, А. Л. Эволюция и анализ моделей жизненного цикла разработки программного обеспечения / А.Л. Алексеев // Международный научный журнал «Вестник науки». – 2024. – №7. – Т. 4. – С. 239-242.

4. Ганеев, Л. И. Этапы разработки программного обеспечения / Л. И. Ганеев, А. М. Султанова, Е. А. Окунева // Новая наука: современное состояние и пути развития. – 2016. – № 11. – С. 145-147.

5. Архипова, Л. И. Управляемый данными маркетинг в цифровой трансформации бизнеса / Л. И. Архипова // Наука и образование: теория и практика. – 2020. – С. 154-160.

6. ROI, ROMI и ещё 25 формул маркетинга [Электронный ресурс] / У. Сулыга – Электрон. дан. – 2023. – Режим доступа: https://takethecake.ru/baza-znaniy/metriki-dlya-marketologa#rec641928836. - Загл. с экрана.

7. Uyanga, S. Prediction for software cost estimation / S. Uyanga, P. Oyunbileg, N. Munkh-tsetseg, CH. Naranmandal // System analysis and mathematical modeling. – 2021. – № 2. – Т. 3. – С. 83-98.

8. Stephen, H. Kan. Bensoussan. Metrics and Models in Software Quality Engineering, Second Edition / H. Kan Stephen // Published by Addison-Wesley Professional, 2002. – 15 с.

9. Никита, Ю. Г. Оценка эффективности управления взаимоотношениями с клиентами / Ю. Г. Никита // Научные дискуссии в области гуманитарных наук. – 2023. – № 11. – С. 261-263.

10. Чекашкина, Н. Р. Ключевые показатели эффективности маркетинга в интернет-среде / Н. Р. Чекашкина, А. А. Микитич // Проблемы и достижения современной науки. – 2023. – С. 157-163.

11. Кербер, Л. С. Применение показателя roi при оценке эффективности корпоративных HR-программ / Л. С. Кербер, А. И. Тихонов // Московский экономический журнал. – 2022. – № 12. – С. 396-407.

12. Кузьминых, А. Р. Проблемы использования показателя NPV при оценке венчурных инвестиций и способы их решения / А. Р. Кузьминых // Форум молодых ученых, 2019. – №1(29). – С. 440-443.

Алымов Д. А., Боднар А. В., Нестеренко А.Р. Экономическое обоснование усовершенствования CRM-системы современными технологиями. *В данной работе рассматривается разработка CRM-системы, ориентированной на массовое тиражирование и интеграцию современных технологий. Выбрана модель жизненного цикла разработки, рассчитаны трудозатраты по модели Фатреллу. Проведен анализ рисков, оценены затраты и показатели эффективности. Разработана структура команды. Полученные результаты демонстрируют возможность создания масштабируемого, экономически оправданного и конкурентоспособного решения для рынка CRM-систем.*

Ключевые слова: *CRM-система, массовое тиражирование, блокчейн, искусственный интеллект, облачные технологии, gRPC, жизненный цикл разработки, модель Фатрелла, трудозатраты, экономическая эффективность, конкурентоспособность, масштабируемость, бизнес-план.*

Alymov D. A., Bodnar A. V., Nesterenko A. R. Economic Justification of Improving the CRM System with Modern Technologies. *This work explores the development of a CRM system aimed at mass distribution and integration of modern technologies. A suitable software development life cycle model was selected, and labor intensity was calculated using the Putnam (Fatrell) model. Risk analysis was conducted, costs and efficiency indicators were evaluated, and a project team structure was designed. The results demonstrate the feasibility of creating a scalable, economically viable, and competitive solution for the CRM systems market.*

Keywords: *CRM system, mass deployment, blockchain, artificial intelligence, cloud technologies, gRPC, software development life cycle, Fatrell model, labor intensity, economic efficiency, competitiveness, scalability, business plan.*

Статья поступила в редакцию 15.05.2025
Рекомендована к публикации профессором Мальчевой Р. В.

УДК 004.89

Система синтеза видео на основе технологии DeepFake

М.О. Лукащук[*1], С.А. Зори[*2]

[*1] магистрант, Донецкий национальный технический университет,
mikhail.lukashchuk@mail.ru

[*2] д.т.н., доцент, Донецкий национальный технический университет,
ik.ivt.rec@mail.ru, OrcID: 0000-0003-4018-234X, SPIN-код: 3565-6330

Аннотация

В статье представлен краткий обзор существующих методов синтеза видео с использованием технологий DeepFake. На основе анализа современных методов предложен подход к улучшению технологии, направленный на повышение реалистичности и эффективности создаваемых видео, дано описание алгоритмов и моделей, применяемых для генерации видео. Предложенные подходы могут быть перспективными для дальнейших разработок и практического применения в области видеосинтеза и глубокого обучения.

Введение

На сегодняшний день технология DeepFake широко используется для создания реалистичных видео, где изображение лица на видео может быть изменено или заменено лицом другого человека [1]. Несмотря на возможности применения в развлекательной и образовательной сферах, технология DeepFake также вызывает обеспокоенность в отношении безопасности данных и этики. Проблема создания реалистичных видео путем синтеза лиц или замены объектов актуальна не только в связи с вопросами прав собственности и контроля над контентом, но и с точки зрения оптимизации ресурсов и улучшения качества конечного видео.

Анализ современных методов синтеза изображений и применения технологий на основе генеративных состязательных сетей (GAN) рассмотрен в [2]. Согласно этому источнику, наиболее высококачественные результаты синтеза достигаются при использовании GAN [3] и их модификаций, таких как StyleGAN [4] и CycleGAN [5], которые обладают мощным потенциалом в создании фотореалистичных изображений. Эти подходы, основанные на конкурентной системе генератора и дискриминатора, позволяют генерировать изображения, а, следовательно, и видео, которые могут точно имитировать оригинальные.

Под термином DeepFake подразумевается использование алгоритмов, позволяющих синтезировать видео с высокой степенью реализма, сохраняя функции исходного видео, но значительно усложняя процесс его анализа и обратной инженерии. Существующие инструменты для создания DeepFake, такие как DeepFaceLab [6] и FaceSwap [7], активно используют генеративные сети для оптимизации синтеза, хотя их применение требует значительных вычислительных ресурсов и времени.

На данный момент существует множество программных решений, предлагающих инструменты для создания и обработки видео DeepFake, но наиболее производительные и гибкие методы преимущественно представлены в коммерческом формате. Бесплатные решения часто ограничены базовыми функциями, а для достижения оптимальных результатов необходимы значительные вычислительные мощности.

Целью данной работы является демонстрация и улучшение комплекса алгоритмов для синтеза видео с использованием DeepFake, которые обеспечат высокое качество изображения и оптимальную производительность. Такой подход позволит использовать технологию DeepFake в безопасных и полезных приложениях, таких как виртуальная реальность и образовательные проекты.

1 Проектирование системы синтеза видео на основе технологии DeepFake

В качестве прототипа системы синтеза видео на основе DeepFake для апробации разрабатываемых алгоритмов улучшения качества работы технологии предполагается разработка программной системы, которая состоит из нескольких ключевых компонентов, каждый из которых выполняет определённые функции в процессе генерации видео. Для демонстрации взаимодействия компонентов в системе разработана диаграмма компонентов (представлена на рисунке 1). На ней показаны 8 модулей проектируемой системы. Модуль загрузки данных отвечает за загрузку и хранение исходных видео и изображений. Модуль предобработки данных включает в себя функции нормализации, масштабирования и аугментации данных. Модуль обнаружения и отслеживания лиц локализует лица на видео и отслеживает их в течение всего ролика.

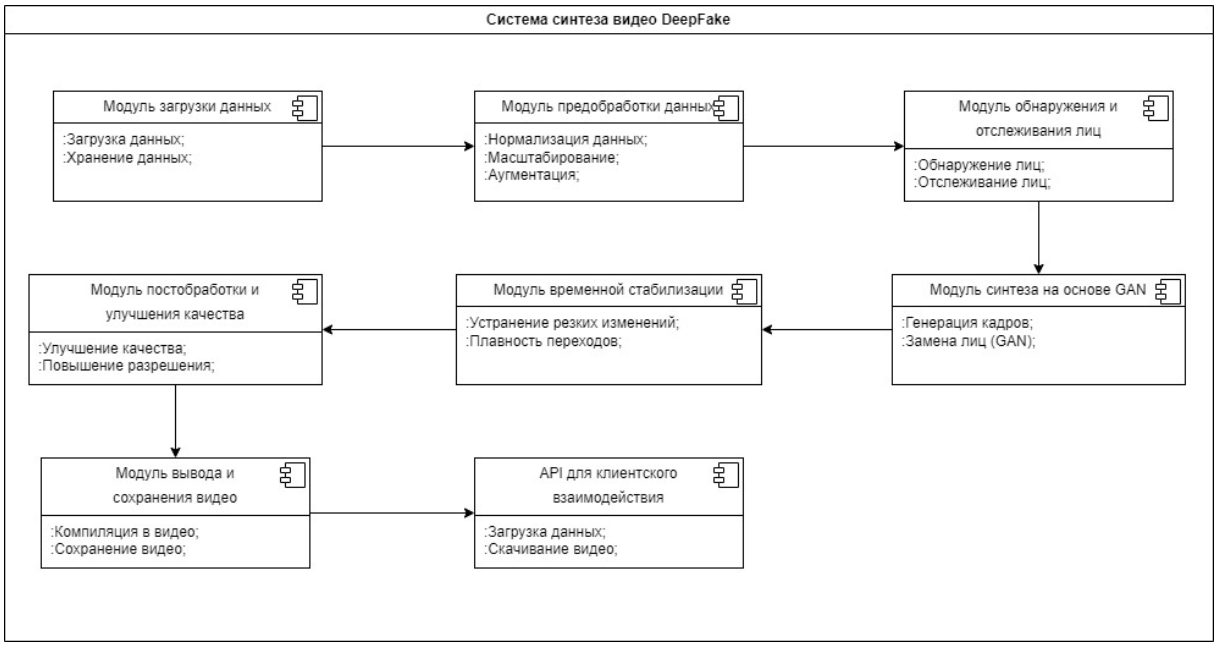


Рисунок 1 – Диаграмма компонентов системы синтеза видео

Модуль синтеза на основе GAN выполняет генерацию новых кадров или замену лиц на основе GAN-алгоритмов. Модуль временной стабилизации устраняет резкие изменения между кадрами для повышения плавности. Модуль постобработки и улучшения качеств: улучшает разрешение и качество изображений. Модуль вывода и сохранения видео компилирует обработанные кадры в видеофайл и сохраняет его. API для клиентского взаимодействия обеспечивает доступ к системе через API, позволяя загружать и скачивать видео.

Процесс синтеза видео включает последовательное выполнение операций по подготовке данных, синтезу и постобработке. На начальном этапе модуль предобработки готовит исходные данные. После этого генеративная сеть выполняет преобразования для синтеза видео. Постобработка стабилизирует изображение, и готовый видеоролик сохраняется. Исходя из спроектированной диаграммы компонентов, была разработана диаграмма последовательности выполнения операций, которая представлена на рис. 2.

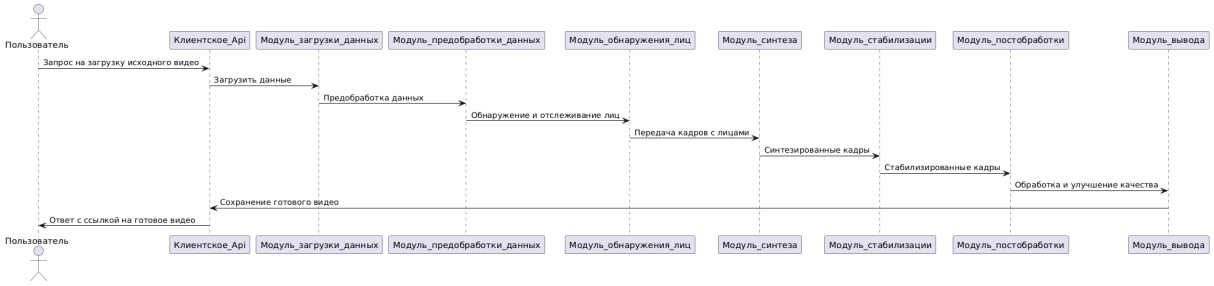


Рисунок 2 – Диаграмма последовательности выполнения операций

Важным элементом системы является физическое расположение основных компонентов системы, включая сервер для вычислений, клиентское устройство пользователя, а также хранилище данных. Для работы системы требуются высокопроизводительные ресурсы, такие как GPU, а также серверные мощности для хранения данных. Основные компоненты системы, включая сервер с GAN-архитектурой, расположены на облачных серверах. Для отображения физического расположения основных элементов системы была разработана диаграмма развёртывания (см. рис. 3).

2 Алгоритм GAN для синтеза видео

Далее будет рассмотрен алгоритм синтеза видео, наиболее актуальным на данный момент является подход с применением технологий на основе генеративных состязательных сетей (GAN). Алгоритм генеративно-состязательных сетей (GAN) для синтеза видео включает два ключевых компонента: генератор и дискриминатор. Генератор обучается создавать реалистичные кадры на основе шума или изображения, а дискриминатор пытается отличить сгенерированные кадры от реальных.

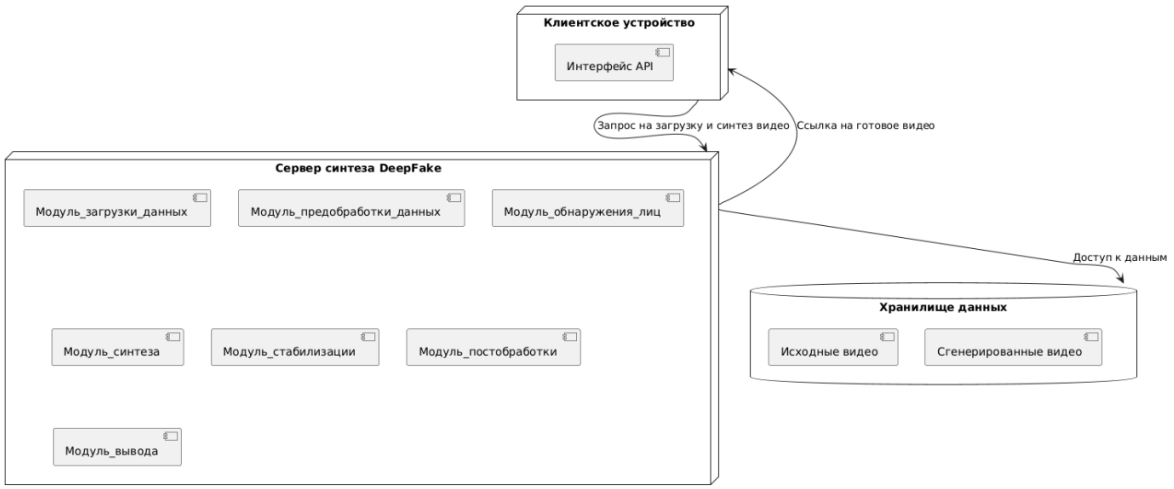


Рисунок 3 – Диаграмма развёртывания

Эти два компонента взаимодействуют друг с другом: генератор учится «обманывать» дискриминатор, создавая все более реалистичные изображения, а дискриминатор старается повысить точность распознавания реальных и сгенерированных кадров.

В контексте алгоритма GAN случайный шум — это вектор случайных чисел, который используется в качестве входных данных для генератора. Это своего рода "платформа" для генерации новых образцов данных, например, изображений или видеокадров. Генератор принимает случайный шум (чаще всего это просто вектор случайных чисел, например, сгенерированных с помощью нормального распределения) и преобразует его в нечто осмысленное, например, «фейковое» изображение или видеокадр. Суть заключается в том, что этот шум не имеет смысла сам по себе, но с помощью нейронной сети генератор обучается создавать из него реалистичные данные. Важно, что случайный шум не представляет собой какие-то заранее определенные данные (например, изображение или видео), а скорее его роль — это создание исходной случайности, на основе которой нейросеть «учится» генерировать осмысленные данные. Каждое изменение случайного шума может привести к созданию разных, но похожих по стилю или содержанию объектов.

Веса — это параметры, которые определяют, как входные данные (в данном случае случайный шум или реальные данные) обрабатываются нейронной сетью, включая как генератор, так и дискриминатор. Генератор использует веса для преобразования случайного шума в изображение или кадр, то есть для того, чтобы сгенерировать изображение, которое будет максимально похожим на реальные данные. Дискриминатор, в свою очередь, использует свои веса для того, чтобы распознать, является ли кадр реальным или сгенерированным.

Для лучшего понимания, алгоритм можно представить в виде диаграммы последовательности (см. рис. 4).

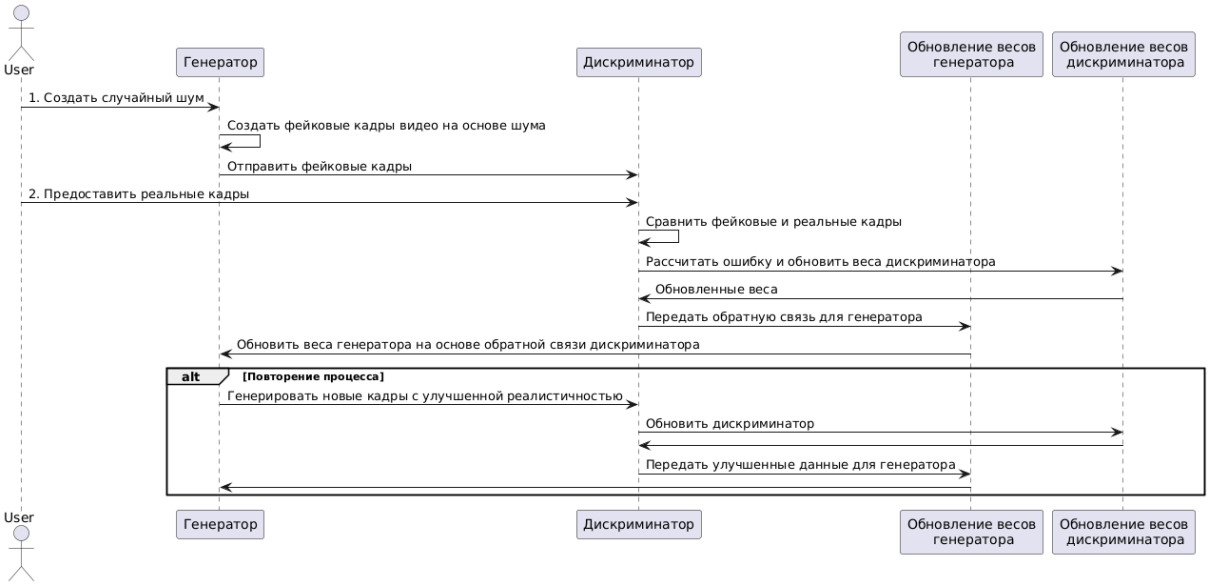


Рисунок 4 – Алгоритм генеративно-состязательных сетей (GAN)

Процесс обучения заключается в том, чтобы на основе ошибок (ошибки дискриминатора, который ошибочно принял фейк за реальное изображение) откорректировать веса сети таким образом, чтобы генератор создавал более реалистичные данные, а дискриминатор мог их лучше распознавать. Для синтеза видео на основе GAN существуют несколько подалгоритмов, ответственных за работу генератора и дискриминатора. Эти подалгоритмы включают обучение генератора, обучение дискриминатора, и постепенное улучшение кадров. Ниже представлены некоторые из них.

2.1 Алгоритм обучения дискриминатора

Дискриминатор обучается на основе двух типов данных: реальных и сгенерированных кадров. Цель дискриминатора — правильно различать эти два типа, минимизируя ошибку классификации. На рисунке 5 представлен псевдокод данного алгоритма.

Далее, при описании функций и алгоритмов, будем опираться на исследование [8] Бинарная кросс-энтропия — это функция потерь, которая используется для оценки качества работы моделей, решающих задачи бинарной классификации.

```
Input: Реальные кадры из обучающего набора данных и сгенерированные кадры от генератора
Output: Обновленные веса дискриминатора

1. Для каждого батча кадров:
    a. Получить реальные кадры X_real и присвоить метки Y_real = 1
    b. Получить сгенерированные кадры X_fake от генератора и присвоить метки Y_fake = 0
    c. Объединить X_real и X_fake в общий обучающий набор X и метки Y
    d. Вычислить предсказание D(X) с использованием дискриминатора
    e. Вычислить ошибку дискриминатора: loss_D = BCE(Y, D(X)),
       где BCE - бинарная кросс-энтропия
    f. Обновить веса дискриминатора, используя градиентный спуск с loss_D
```

Рисунок 5 – Псевдокод алгоритма обучения дискриминатора

Задача бинарной классификации заключается в том, чтобы классифицировать объекты на две категории, например, «истинное» или «ложное», «1» или «0», «реальное» или «сгенерированное». Для одного предсказания функция потерь, основанная на бинарной кросс-энтропии, рассчитывается по формуле 1:

$$L = -(y * \log(\hat{y}) + (1 - y) * \log(1 - \hat{y})) \quad (1)$$

где y — истинное значение (0 или 1);

$\hat{y}$ — предсказанное моделью значение вероятности (от 0 до 1).

Если истинное значение y=1, то функция потерь становится $-\log(\hat{y})$, что означает, что чем ближе $\hat{y}$ к 1, тем меньше значение функции потерь. Цель обучения модели — минимизировать среднее значение бинарной кросс-энтропии для всех обучающих данных, что будет означать, что модель делает правильные предсказания с высокой уверенностью.

Градиентный спуск — это алгоритм оптимизации, который используется для нахождения минимума функции потерь (например, бинарной кросс-энтропии). Цель градиентного спуска — настроить параметры модели (веса), чтобы минимизировать ошибку предсказания. Сначала вычисляется градиент функции потерь по каждому параметру модели. Градиент указывает направление, в котором функция потерь растет быстрее всего. Параметры модели обновляются в направлении, противоположном градиенту. Предыдущие шаги повторяются многократно, пока функция потерь не станет минимальной или почти минимальной.

2.2 Алгоритм обучения генератора

Цель генератора — создать кадры, которые дискриминатор примет за реальные. Обучение генератора происходит по обратной связи, полученной от дискриминатора, который указывает, насколько реалистичными получились кадры.

На рисунке 6 представлен псевдокод данного алгоритма. Генератор обновляет свои параметры, стремясь уменьшить способность дискриминатора отличать реальные данные от фейковых. Его задача — «обманывать» дискриминатор, чтобы тот посчитал фейковые данные реальными. Если генератор успешен, дискриминатор начинает терять способность различать, какие данные — настоящие, а какие — поддельные.

Эта «соревновательная игра» продолжается до тех пор, пока генератор не научится создавать данные, которые выглядят очень правдоподобно, и дискриминатор перестает с лёгкостью их отличать.

2.3 Алгоритм постепенного улучшения кадров

Для создания реалистичного видео важно не только качество отдельных кадров, но и плавность переходов между ними. Алгоритм постепенного улучшения кадров помогает улучшить временную стабильность, чтобы сгенерированные кадры плавно перетекали друг в друга. На рисунке 7 представлен псевдокод данного алгоритма.

```
Input: Случайный шум или изображение-источник
Output: Обновленные веса генератора

1. Для каждого батча:
    a. Сгенерировать фейковые кадры X_fake из случайного шума или изображения с использованием генератора
    b. Получить предсказание дискриминатора D(X_fake) для сгенерированных кадров
    c. Установить целевые метки Y_fake = 1 (поскольку генератор хочет «обмануть» дискриминатор)
    d. Вычислить ошибку генератора: loss_G = BCE(Y_fake, D(X_fake))
    e. Обновить веса генератора, используя градиентный спуск с loss G
```

Рисунок 6 – Алгоритм обучения генератора

```
Input: Сгенерированные кадры X_fake в порядке последовательности
Output: Обновленные кадры X_smooth с уменьшением резких изменений

1. Инициализировать предыдущий кадр как X_prev = X_fake[0]
2. Для каждого кадра X_t в последовательности X_fake:
    a. Вычислить разницу delta = |X_t - X_prev|
    b. Если delta > threshold (порог допустимых изменений):
        - Установить X_t = (X_t + X_prev) / 2, чтобы уменьшить разницу
    c. Обновить X_prev = X_t
3. Вернуть сглаженные кадры X_smooth
```

Рисунок 7 – Алгоритм постепенного улучшения кадров

3 Итоговый алгоритм синтеза видео

На основании описанных выше алгоритмов, можно представить общий алгоритм работы системы, который, исходя из анализа научных работ, является наиболее актуальным на момент написания статьи. Алгоритм представлен на рисунке 8.

Синтез видео с использованием GAN имеет значительные достижения, но также сталкивается с рядом недостатков и сложностей, которые ограничивают его использование и требуют улучшений. В исследовании [9] выделены некоторые недостатки метода:

GAN требует значительных вычислительных ресурсов для обучения, особенно для задач синтеза видео, где каждый кадр должен быть реалистичным и плавно переходить в следующий.

GAN в синтезе видео должен не только создавать реалистичные отдельные кадры, но и обеспечивать плавные, естественные переходы между ними. Часто это приводит к проблемам, как, например, «мерцание» или «дрожание» отдельных частей изображения, что нарушает целостность видео. Такие артефакты становятся особенно заметными при синтезе длинных видео.

Генератор может создавать артефакты, такие как размытые или искаженные участки изображения. Это особенно проблематично в динамичных сценах, где модель не успевает корректно адаптировать каждый кадр.

```
Input: Набор тренировочных данных (видеокадры) или исходное видео
Output: Сгенерированное видео с плавными, реалистичными кадрами

1. Загрузка данных:
    a. Загрузить исходное видео или тренировочный набор кадров.
    b. Разделить видео на отдельные кадры, если исходные данные представляют собой видео.

2. Предобработка данных:
    a. Применить фильтрацию и нормализацию к каждому кадру для улучшения качества.
    b. Использовать метод обнаружения лиц для выделения областей с лицами на кадрах.

3. Инициализация GAN:
    a. Инициализировать веса генератора и дискриминатора случайными значениями.
    b. Установить начальные значения параметров обучения.

4. Обучение GAN:
    Пока ошибка дискриминатора не стабилизируется или не достигнуто максимальное количество эпох:
        a. Обновление дискриминатора:
            - Для каждого батча реальных и сгенерированных кадров:
                i. Подать реальные кадры и установить метки Y_real = 1.
                ii. Сгенерировать фейковые кадры с помощью генератора и установить метки Y_fake = 0.
                iii. Рассчитать ошибку дискриминатора loss_D, используя бинарную кросс-энтропию между предсказаниями и истинными метками.
                iv. Обновить веса дискриминатора с помощью градиентного спуска для минимизации loss_D.

        b. Обновление генератора:
            - Для каждого батча шума или изображения-источника:
                i. Сгенерировать фейковые кадры X_fake.
                ii. Получить предсказания дискриминатора для X_fake.
                iii. Установить метки Y_fake = 1, чтобы "обмануть" дискриминатор.
                iv. Рассчитать ошибку генератора loss_G на основе предсказаний дискриминатора.
                v. Обновить веса генератора для минимизации loss_G.

5. Постобработка кадров (сглаживание кадров):
    a. Инициализировать переменную X_prev как первый сгенерированный кадр.
    b. Для каждого кадра X_t из сгенерированной последовательности:
        i. Вычислить разницу между текущим и предыдущим кадром delta = |X_t - X_prev|.
        ii. Если delta больше допустимого порога:
            - Установить X_t = (X_t + X_prev) / 2, чтобы уменьшить резкие переходы.
        iii. Обновить X_prev = X_t.

6. Создание финального видео:
    a. Объединить сглаженные кадры в видеопоследовательность.
    b. Сохранить видео в заданном формате.

7. Вернуть сгенерированное видео.
```

Рисунок 8 – Общий алгоритм работы системы

Дискриминатор может начать «запоминать» тренировочные данные и терять способность оценивать правдоподобие новых данных, что ухудшает качество синтеза видео.

GAN часто теряют последовательность и правдоподобие в длинных видео. Для реалистичного синтеза длинных видео требуется учитывать взаимосвязь не только между соседними кадрами, но и в более широком временном контексте, что значительно усложняет задачу. Без таких зависимостей модель не может поддерживать стабильное качество по мере увеличения длины видео.

Для улучшения алгоритмов синтеза видео с помощью GAN, можно предложить несколько направлений и конкретных решений, позволяющих повысить качество, стабильность и управляемость результатов.

Вот ключевые подходы:

Включение временных зависимостей позволяет улучшить согласованность кадров, минимизируя артефакты на границах между ними. Для этого в архитектуру GAN добавляются слои, обрабатывающие последовательность кадров, например, рекуррентные нейронные сети [10].

Добавление второго дискриминатора, который проверяет временные зависимости между кадрами, помогает модели лучше понимать, как объекты и фон должны изменяться от кадра к кадру. Один дискриминатор проверяет качество каждого кадра, а второй — их последовательность.

Внедрение атрибутов (например, меток объектов) в генератор для управления содержимым и движением в видео. Это позволяет задать начальные условия, чтобы контролировать вид и поведение объектов, улучшая предсказуемость и управляемость.

Использование промежуточных слоев для повышения разрешения (например, от низкого к высокому) улучшает качество видео, обеспечивая плавность переходов между кадрами и детализацию.

С учётом данных предложений, на рисунке 9 представлен обновлённый алгоритм синтеза видео с использованием технологии GAN.

```
Input: Набор тренировочных данных (видеокадры) или исходное видео
Output: Сгенерированное видео с плавными, реалистичными кадрами

1. Загрузка данных:
   a. Загрузить исходное видео или набор тренировочных кадров.
   b. Если данные представлены как видео, разделить его на отдельные кадры.

2. Предобработка данных:
   a. Применить фильтрацию и нормализацию к каждому кадру.
   b. Использовать обнаружение лиц для выделения областей на кадрах.

3. Инициализация GAN с улучшенными компонентами:
   a. Инициализировать веса генератора, пространственного и временного дискриминаторов.
   b. Добавить рекуррентные нейронные сети в генератор и дискриминатор для обработки последовательности кадров.
   c. Настроить слои многоуровневой генерации, увеличивающие разрешение на каждом этапе.
   d. Установить параметры обучения (скорость обучения, количество эпох и функции потерь).

4. Обучение GAN:
   while ошибка дискриминатора не стабилизируется или не достигнуто максимальное количество эпох:

      a. Обновление дискриминаторов:
         for каждый батч реальных и сгенерированных кадров:
            1. Подать реальные кадры в пространственный дискриминатор и установить метки как реальные.
            2. Использовать временной дискриминатор для проверки последовательностей кадров.
            3. Сгенерировать фейковые кадры с помощью генератора и установить метки как фейковые.
            4. Рассчитать ошибки дискриминаторов и обновить их веса с помощью градиентного спуска.

      b. Обновление генератора:
         for каждый батч случайного шума или начального изображения:
            1. Сгенерировать последовательность кадров, используя RNN в генераторе для учета временной последовательности.
            2. Подать сгенерированные кадры на дискриминаторы для проверки пространственной и временной согласованности.
            3. Установить метки как реальные, чтобы "обмануть" дискриминаторы.
            4. Рассчитать ошибку генератора и обновить его веса для минимизации этой ошибки.

5. Постобработка кадров (сглаживание переходов):
   a. Инициализировать переменную для хранения первого сгенерированного кадра.
   b. for каждый кадр из сгенерированной последовательности:
      1. Вычислить разницу между текущим и предыдущим кадром.
      2. if разница > порог:
         - Усреднить текущий и предыдущий кадры для сглаживания перехода.
      3. Обновить переменную на текущий кадр.

6. Создание финального видео:
   a. Объединить сглаженные кадры в видеопоследовательность.
   b. Сохранить видео в заданном формате.

7. Вернуть сгенерированное видео.
```

Рисунок 9 – Итоговый алгоритм синтеза видео с использованием технологии GAN

Выводы

В работе был проведён системный анализ методов и архитектур систем синтеза видео, в частности, алгоритмов на основе генеративных состязательных сетей (GAN). Основываясь на анализе, были выявлены ключевые проблемы стандартных GAN, такие как низкая согласованность кадров и недостаточная детализация, что может приводить к визуальным артефактам и неестественным переходам между

кадрами. Для устранения этих недостатков предложены несколько усовершенствований, включая: использование рекуррентных нейронных сетей, включение дополнительного временного дискриминатора, многоуровневая генерация с постепенным увеличением разрешения и внедрение управляемых атрибутов.

Эти методы были интегрированы в общую структуру GAN, позволив сформировать улучшенный алгоритм синтеза видео. Предложенные подходы могут быть перспективными для дальнейших разработок в области видеосинтеза и глубокого обучения, а также полезными в практическом применении для задач создания контента, виртуальной реальности и других областей мультимедийных технологий.

Литература

1. Deepfake: краткая история появления и нюансы работы технологии [Электронный ресурс]. – Режим доступа: https://habr.com/ru/companies/neuronet/articles/592119/. – Загл. с экрана.

2. Узких, Г. Ю. Применение генеративно-состязательных сетей (GAN) в обработке изображений / Г. Ю. Узких // Вестник науки. – 2024. – Т. 4, № 8 (77). – С. 182-185.

3. Generative adversarial network [Электронный ресурс] // Википедия. – Режим доступа: https://en.wikipedia.org/wiki/Generative_adversarial_network. – Загл. с экрана.

4. StyleGAN: Revolutionizing AI-Driven Image Creation [Электронный ресурс]. – Режим доступа: https://www.simplilearn.com/tutorials/generative-ai-tutorial/stylegan. – Загл. с экрана.

5. Cycle Generative Adversarial Network (CycleGAN) [Электронный ресурс]. – Режим доступа: https://www.geeksforgeeks.org/cycle-generative-adversarial-network-cyclegan-2/. – Загл. с экрана.

6. DeepFaceLab - AI-Powered Face Manipulation and Editing [Электронный ресурс]. – Режим доступа: https://perchance-ai.vercel.app/free-ai-tools/deepfacelab. – Загл. с экрана.

7. Face Swap Video [Электронный ресурс]. – Режим доступа: https://faceswapvideo.ai/. – Загл. с экрана.

8. Великанов, М. С. Нейросетевой алгоритм поиска областей открытия/закрытия в видеопоследовательностях / М. С. Великанов, А. Б. Анзина, С. В. Лаврушкин, Д. С. Ватолин // International Journal of Open Information Technologies. – 2020. – Т. 8, № 3. – С. 55-62.

9. Аверченков, А. В. Анализ и применение генеративно-состязательных сетей для получения изображения высокого качества / А. В. Аверченков, А. А. Андросов, Ю. А. Малахов // Эргодизайн. – 2020. – № 4. – С. 167-175.

10. Рекуррентная нейронная сеть (RNN): виды, обучение, примеры [Электронный ресурс]. – Режим доступа: https://neurohive.io/ru/osnovy-data-science/rekurrentnye-nejronnye-seti/. – Загл. с экрана.

Лукащук М.О., Зори С.А. Система синтеза видео на основе технологии DeepFake. *В статье представлен краткий обзор существующих методов синтеза видео с использованием технологий DeepFake. На основе анализа современных методов предложен подход к улучшению технологии, направленный на повышение реалистичности и эффективности создаваемых видео, дано описание алгоритмов и моделей, применяемых для генерации видео. Предложенные подходы могут быть перспективными для дальнейших разработок и практического применения в области видеосинтеза и глубокого обучения.*

Ключевые слова: *DeepFake, синтез видео, генеративные модели, GAN, StyleGAN, алгоритмы, машинное обучение, нейронные сети*

Lukashchuk M.O., Zori S.A. Video Synthesis Methods Based on DeepFake Technology. *This article provides a brief overview of existing video synthesis methods utilizing DeepFake technology. Based on the analysis of current methods, an approach to improving the technology is proposed, aimed at increasing the realism and efficiency of the generated videos, and the algorithms and models used for video generation are described. The proposed approaches may be promising for further development and practical application in the field of video synthesis and deep learning*.

Keywords: *DeepFake, video synthesis, generative models, GAN, StyleGAN, algorithms, machine learning, neural networks*

Статья поступила в редакцию 17.05.2025
Рекомендована к публикации профессором Мальчевой Р. В.

УДК 004.93

Автономная навигация мобильного робота

И. В. Савицкая, А. П. Семёнова
Донецкий национальный технический университет
i.v.savitskaya@mail.ru, nastena-semenova19@rambler.ru

Аннотация
В работе представлено исследование и практическая реализация алгоритмов компьютерного зрения для автономного прохождения трассы роботом в рамках первого Кубка России по спортивному программированию в дисциплине "программирование робототехники". Основное внимание уделено разработке алгоритма навигации колесного робота по сложной трассе с различными типами покрытий (трава, лед, земля) и геометрическими особенностями (S-образные участки, "змейки").

Введение

В этом году Кубок России по спортивному программированию в дисциплине «программирование робототехники» проводился впервые. Отборочный этап соревнования проходил в формате онлайн на платформе симулятора «1Т МИР». Необходимо было создать решение на основе машинного зрения и нейронных сетей, обеспечивающее автономное прохождение роботизированной машиной трассы с различными препятствиями и типами покрытий.

Компания «1Т» более 20 лет выполняет ИТ-проекты для крупных государственных и бизнес-заказчиков, развивает образовательное направление и создаёт собственные цифровые продукты, включая симулятор «1Т МИР», функционирующий на импортонезависимом программном движке. С 2023 года на базе платформы «1Т МИР» прошли обучение свыше 10 000 слушателей, а также организованы масштабные соревнования, включая чемпионат «Новая высота» на площадке Калужского Технопарка профессионального образования и «Фестиваль пилотирования БЛА» в партнёрстве с УГТУ.

Для достижения цели проекта необходимо:

1. Изучить существующие полигоны и особенности их прохождения.
2. Выполнить анализ предоставленного организаторами полигона.
3. Проанализировать существующие алгоритмы автономного ориентирования мобильного робота.

Разработать ПО для автоматического прохождения трассы.

Анализ полигонов

На соревнованиях по робототехнике предлагаются полигоны (трассы) различных размеров и форм [1]:

– линейные трассы – представляют собой замкнутые или разомкнутые линии с поворотами и различными участками, по которым должен перемещаться робот;

– игровые поля – специальные площадки для тренировок и соревнований, где роботы выполняют конкретные задания;

– городские среды – имитируют дорожную инфраструктуру: движение по улицам, пересечение перекрёстков, реакцию на внешние факторы;

– специализированные сценарии – моделируют уникальные условия (например, зоопарк, ферму, космическое пространство);

– лабиринты – полигоны с трёхмерными бортиками, ограничивающими траекторию движения робота;

– лестница – робот должен пройти замкнутый путь по специальной лестнице;

– слалом по линии – задача: пройти s-образную трассу (чёрная линия) за минимальное время, избегая кеглей-препятствий;

– кегельринг и сумо – робот должен за наиболее короткое время вытолкнуть кегли или противника за пределы круга (ринга);

– дорога – робот должен преодолеть маршрут без столкновений с другими роботами-участниками.

Ключевой фактор для робота – не только ширина, но и форма линии. Среди линейных трасс выделяют [2]:

– «Simple» (см. рис. 1а) – простая трасса для демонстрации базовых алгоритмов;

– «Гантеля» (см. рис. 1б) – универсальное поле для тренировок и соревнований разного уровня;

– Полигон Политехнического музея (см. рис. 1в) – позволяет развивать высокие скорости (свыше 1 м/с);

– «Зародыш» (см. рис. 1г) – трасса эмбрионоподобной формы с поворотами малого радиуса;

– «Истукан» (см. рис. 1д) – содержит сложные участки, включая «прямые змейки».

Наиболее сложными для прохождения являются S-образные участки трассы («змейки») и «Прямые змейки» (например, на полигоне «Истукан»). При определённых углах риск потери трассы резко возрастает, а некоторые роботы на определённых скоростях неспособны их преодолеть [3].

Обзор платформы для соревнований

Стартовая точка при загрузке симулятора [4] имеет вид, представленный на рисунке 2. Управление роботом осуществляется в двух режимах: ручном и автоматическом. Вид управления указан в левом верхнем углу.

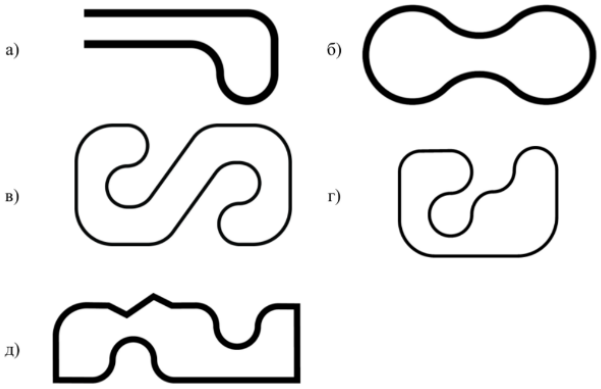


Рисунок 1 – Виды линейных трасс

Рисунок 2 – Режим автоматического управления роботом

Переключение между режимами происходит при нажатии клавиши «T». Перемещение машины на стартовую позицию осуществляется клавишей «R». Данная подсказка размещена в левом нижнем углу. По центру в верхней части экрана через наклонную черту отображается два показателя fps – количество кадров в секунду и fp – интенсивность светового потока. А в правом нижнем углу указано состояние подключения робота к автоматическому управлению.

При прохождении трассы в ручном режиме для изменения доступны множество параметров робота, появляющихся в правой части экрана при нажатии на клавишу «Q» (см. рис. 3). Приведение в движение роботизированной тележки происходит клавишами WASD. Кроме вышеперечисленных параметров в ручном режиме доступны две кнопки: «Скачать снимки с камер» и «Скачать autopilot.zip». Содержимое файлов при скачивании снимков с камер представлено на рисунке 4 и содержит изображение робота с 5 статичных камер и камеры, следящей за машинкой.

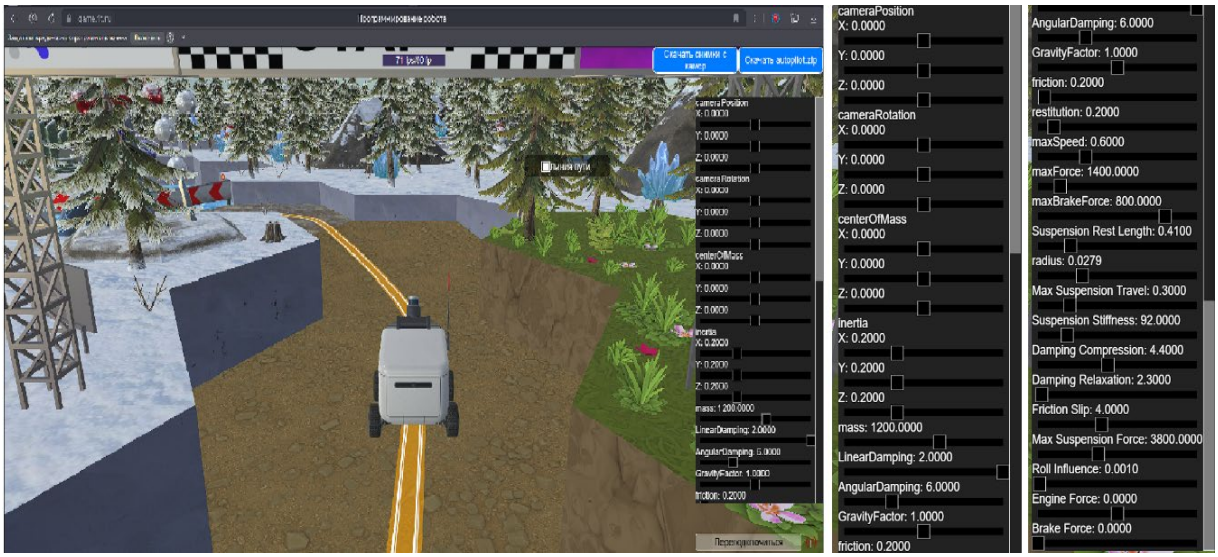


Рисунок 3 – Доступные для изменения параметры робота

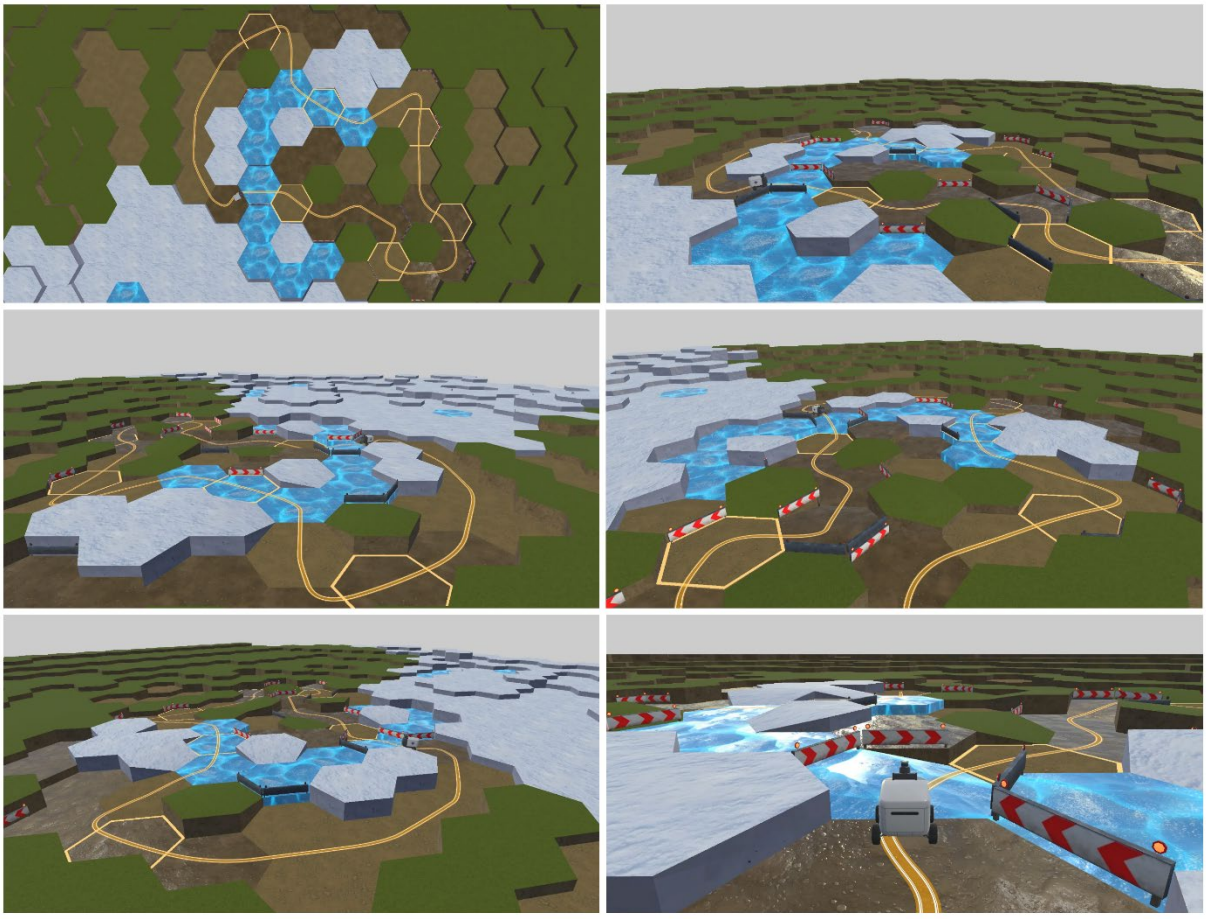


Рисунок 4 – Доступные снимки с камер

Обзор программы

При нажатии на кнопку «Скачать autopilot.zip» происходит загрузка проекта на языке python, реализующего примеры применения управляющих роботом команд. После распаковки архива autopilot.zip появляются 4 папки с функциями управления роботом и файлы. Содержание папок с файлами:

– algorithm – модули алгоритма работы робота;

– connection – модули, реализующие подключение к роботу;

– services – модули для создания log-файла;

– vehicle – модули непосредственного управления машинкой.

Отдельные файлы:

– requirements.txt – установка зависимостей;

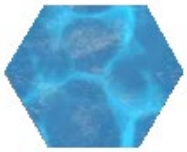
– readme.md – инструкция;

– config.py – модуль конфигурации;

– main.py – основная программа для запуска автоматического управления роботом.

Задача, стоявшая перед участниками соревнований, заключалась в том, чтобы роботизированная машинка проехала всю трассу без съезда с линии и посетила все контрольные точки (так называемые чек-поинты). Вид трассы с камер представлен на рисунке 4. Если внимательно рассмотреть виды с разных камер, то можно заметить, что трасса представляла собой шестиугольные ячейки в виде сот. Кроме того, ячейки могли располагаться под различными углами, имитирующими горку. Ячейки с ярко желтой обводкой представляли собой контрольные точки. На каждую ячейку-соту была наложена текстура. Всего на трассе представлено 5 текстур (таблица 1).

Таблица 1 - Виды текстур

Внешний вид	Описание
	зеленая трава; проблем в движении «тележки» нет
	сине-голубой лед; возможен занос «тележки» на поворотах
	темно коричневая земля, покрытая льдом; у «тележки» возникают проблемы с подъемом на горку
	белый снег; проблем в движении «тележки» нет, т.к. частью трассы не является
	светло коричневая земля; проблем в движении «тележки» нет

Непосредственное управление движущими элементами машинки-робота в автоматическом режиме осуществляется в файле vehicle_control.py, находящемся в папке vehicle. Организаторами Кубка было предоставлено всего две управляющие команды:

1. setMotorPower(right, left) – подает управляющие сигналы на сервоприводы колес и имеет два параметра. Первый параметр – мощность вращения правых колес в процентах, второй – мощность вращения левых колес в процентах. Если значение фактического параметра положительное, то колесо вращается по часовой стрелке, если отрицательное – то против часовой стрелки.
2. rotate(angle) – метод поворота робота относительно текущего положения. Отрицательные значения параметра angle будут поворачивать робота по часовой стрелке, положительные – против часовой стрелки.

Анализ современных алгоритмов компьютерного зрения

Компьютерное зрение является междисциплинарной областью исследований, объединяющую методы цифровой обработки изображений, машинного обучения и искусственного интеллекта. Оно представляет собой ключевую технологию современной цифровой трансформации, находящую широкое применение в таких стратегически важных областях, как робототехника, промышленная автоматизация, медицинская диагностика и интеллектуальные системы безопасности. Его внедрение позволяет существенно расширить функциональные возможности автоматизированных систем за счет реализации сложных визуально-аналитических функций.

Основная задача компьютерного зрения заключается в разработке алгоритмов автоматического извлечения и интерпретации значимой информации из визуальных данных. Фундаментальные задачи включают:

– распознавание и классификацию объектов в статических изображениях и видеопотоках;

– детектирование и точную пространственную локализацию объектов интереса;

– семантическую и инстанс-сегментацию изображений;

– трекинг объектов в динамических сценах;

– оценку пространственной ориентации и положения объектов;

– оптическое распознавание текстовой информации;

– анализ и интерпретацию динамики сцены.

Особое значение в рамках компьютерного зрения занимают задачи трехмерной реконструкции сцены по двумерным изображениям и семантической интерпретации визуального контента. Решение этих задач требует комплексного подхода, сочетающего

методы цифровой обработки изображений, машинного обучения и вычислительной геометрии. Современные алгоритмы позволяют не только идентифицировать объекты, но и восстанавливать их пространственные характеристики, анализировать взаимное расположение и динамику изменений, что открывает новые перспективы для создания интеллектуальных систем принятия решений.

Современная система компьютерного зрения включает в себя несколько ключевых компонентов:

1. Оптическая подсистема – состоит из одной или нескольких видеокамер, системы освещения и оптических фильтров.
2. Аппаратная платформа обработки данных – включает центральный процессор (CPU), графический процессор (GPU), специализированные ускорители (FPGA, ASIC).
3. Программное обеспечение – состоит из алгоритмов предварительной обработки и методов анализа изображений, решающих правил.
4. Интерфейс передачи данных – включает проводные и беспроводные каналы связи и протоколы обмена данными.

В рамках решения задачи визуальной идентификации объектов существует множество различных подходов. Применительно к конкретной задаче следования по линии, ключевым аспектом становится надежное выделение целевой линии на контрастном фоне, где может использоваться как классическая схема черной линии на белом фоне, так и инверсный вариант с белой линией на черном фоне.

Типичный алгоритм компьютерного зрения для решения подобных задач предполагает последовательную обработку изображения через несколько взаимосвязанных этапов (см. рис. 2).

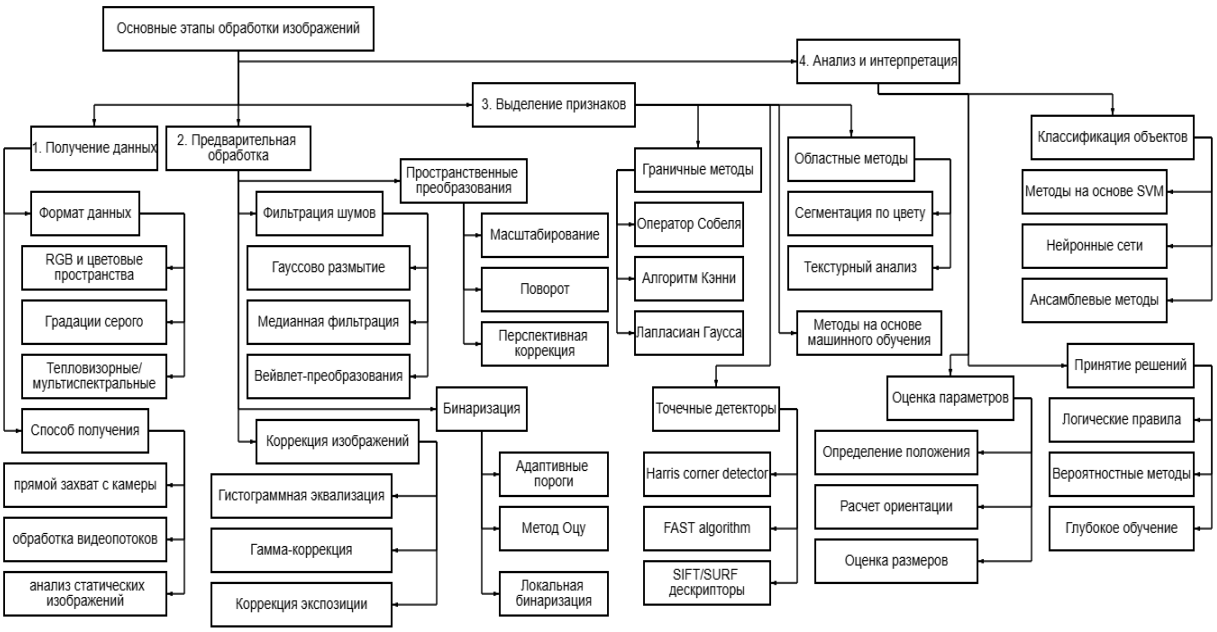


Рисунок 5 – Основные этапы обработки изображений

Первоначально система получает исходное изображение, после чего выполняет его предварительную обработку для улучшения качества и выделения значимых признаков. Затем следует этап детектирования ключевых элементов изображения, таких как границы, линии или характерные точки. На заключительной стадии происходит анализ и интерпретация выделенных признаков для принятия решенияКаждый из этих этапов может реализовываться различными методами в зависимости от конкретных условий задачи, характеристик оборудования и требований к точности и быстродействию системы. Важно отметить, что эффективность работы алгоритма в значительной степени зависит от корректного выбора и настройки методов обработки на каждом этапе, а также от их согласованного взаимодействия в рамках единого конвейера обработки изображений.

Предлагаемое решение

В качестве средств реализации выбран язык Python и библиотека OpenCV. Python в сочетании с OpenCV представляет собой оптимальное решение для обработки изображений, обеспечивая высокую производительность за счёт C++-ядра и удобство разработки благодаря лаконичному синтаксису Python и интеграции с NumPy/SciPy. OpenCV

поддерживает широкий спектр методов - от классических алгоритмов фильтрации и детектирования до современных нейросетевых моделей (YOLO, SSD) через DNN-модуль, сохраняя кроссплатформенность (Windows/Linux/macOS, CPU/GPU) и масштабируемость (OpenCL/CUDA). Ключевыми преимуществами являются быстрое прототипирование, обширная документация, активное сообщество и регулярные обновления, что делает эту связку универсальным инструментом как для исследований, так и для промышленных решений. Таким образом, OpenCV предоставляет оптимальный баланс между производительностью, удобством разработки и доступностью ресурсов для обучения.

Робот определяет направление движения ориентируясь на снимки с камер. Всего предоставлено 5 камер стационарных и 1 следующая за роботом. Для автономной работы робота необходимо, чтобы на используемом изображение осталась видна только линия пути. Рассмотрим алгоритм детектирования линии трассы с использованием OpenCV:

Шаг 1. Сбор входные данных. На вход алгоритма подается изображение с камеры (цветное). Также учитываются параметры линии, такие как цвет, ширина и контрастность относительно фона.

Шаг 2. Предварительная обработка изображения. Изображение конвертируется в цветовое пространство HSV с помощью cvtColor (images, cv2.COLOR_BGR2HSV). Применяется фильтр-маска для отсечения цветов окружающего пространства. Для фильтрации изображения используется функция inRange (hsv_ images, lower_line, upper_line). Первый параметр – изменяемое изображение, а второй и третий – левая и правая граница пропускаемого цвета.

Затем переводим полученное изображение в градации серого для лучшего выделения линии с помощью cvtColor(img, cv2.COLOR_BGR2GRAY). Далее выполняется бинаризация изображения с помощью threshold(gray_images, 1, 255, cv2.THRESH_BINARY), после чего применяется размытие Гаусса для уменьшения шумов и артефактов изображения GaussianBlur(result_line, (3, 3), 0).

Шаг 3. Выделение контуров линии. С помощью функции поиска контуров findContours(blurred, cv2.RETR_EXTERNAL, cv2.CHAIN_APPROX_SIMPLE) находятся все контуры на бинаризованном изображении. Для устранения мелких шумов выполняется фильтрация контуров по минимальной площади, что позволяет оставить только значимые элементы, соответствующие линии трассы.

Шаг 4. Определение центральной линии. Из оставшихся контуров выбираем наибольший по площади, который считается основной линией трассы с помощью функции max(filtered_contours, key=cv2.contourArea). Для этого контура выполняется аппроксимация для упрощения его формы arcLength(largest_contour, True), строится ограничивающий прямоугольник минимальной площади и вычисляется центр масс контура, который принимается за центральную точку линии.

Шаг 5. Определение угла отклонения. На основе положения центра линии относительно центра робота вычисляется угол отклонения «тежки» от трассы. Этот угол используется в системе управления: если отклонение превышает пороговое значение в 5 градусов вправо или влево, роботу подается команда на соответствующий поворот, в противном случае - команда движения прямо.

Шаг 6. Визуализация (опционально). Для отладки и демонстрации работы алгоритма может выполняться визуализация: на исходное изображение наносятся контуры обнаруженной линии и маркер центра робота. Результат отображается в отдельном окне до нажатия клавиши (см. рис. 6).

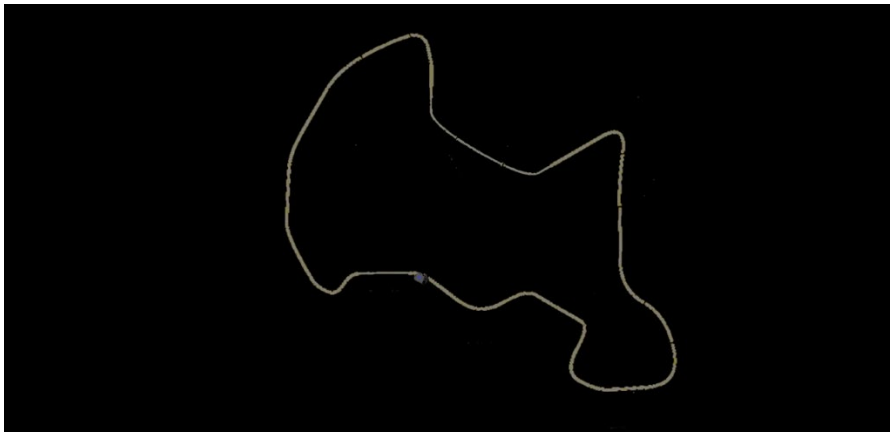


Рисунок 6 – Обработанный рисунок

Заключение

Проведенное исследование и практическая реализация алгоритмов компьютерного зрения для автономного прохождения трассы роботом в рамках Кубка России по спортивному программированию позволило убедиться в эффективности выбранного подхода. Использование связки Python + OpenCV доказало свою эффективность для задач автономной навигации. Предложенный алгоритм детектирования линии трассы, основанный на цветовой сегментации, бинаризации и контурном анализе, обеспечил устойчивость к различным типам покрытий (трава, лед, снег), точное определение центра линии даже при сложной геометрии трассы (S-образные участки, "змейки"), реальное время обработки (~15-20 FPS).

Литература

1. Трассы соревнований. [Электронный ресурс]. – Режим доступа: https://myrobot.ru/sport/index.php?n=Reglaments.Fields

2. Мартыненко Ю.Г. Динамика мобильных роботов. // Соросовский образовательный журнал. – 2000. – Т.6. – №5. – С. 110-116.

3. Мартыненко, Ю.Г. Управление движением мобильных колесных роботов // Фундаментальная и прикладная математика. - 2005. – №8. – С. 29-80.

4. Ларкин Е.В. Трассировка движения мобильного робота по пересеченной местности / Е.В. Ларкин, Ву Зуй Нгхиа // Известия ТулГУ. Технические науки. – 2012. – №8. – С.257-260.

5. Рядчиков И.В. Создание робота автономного движения по линии / И.В. Рядчиков, С.Г. Синица, Б.О. Брагин [и др.]. // Технические науки: проблемы и перспективы : материалы III Междунар. науч. конф. (г. Санкт-Петербург, июль 2015 г.). – Санкт-Петербург : Свое издательство, 2015. — С. 19-25.

6. Лазарев М.В. Управление движением мобильного робота по различным трассам на основе принципа редукции теории нечеткости / М.В. Лазарев, В.Г. Ежкова // Вестник Белгородского государственного технологического университета им. В.Г. Шухова. – 2024. – №.1. – С. 95-103.

7. Орехов С.Ю. Исследование методики планирования движений мобильного робота в известной среде / С.Ю. Орехов, Р.А. Багдошвили, В.О. Копылов // StudNet. – 2022. – №6. – С.7073-7084.

8. Власов, С.М., Бойков В.И., Быстров С.В., Григорьев В.В. Бесконтактные средства локальной ориентации роботов. – СПб: Университет ИТМО, 2017. – 169с.

9. Чулкова, А.С. Разработка алгоритма автономного движения робота в условиях смоделированной городской среды / А.С. Чулкова, Я.А. Карташов, А.А. Шарапов // Сборник материалов Международной научной конференции «Интерэкспо ГЕО-Сибирь 2024». – Т.7. – №2. – *С. 208-212*

10. Шапиро, Л. Компьютерное зрение / Л. Шапиро, Д. Стокман. – 5-е изд. – Москва: Лаборатория знаний, 2024. – 762 с.

11. Компьютерное зрение с OpenCV и Python: практическое руководство [Электронный ресурс]. – Режим доступа: https://www.litres.ru/book/inzhener-33334081/komputernoe-zrenie-s-opencv-i-python-prakticheskoe-rukov-71881402/

Савицкая И.В., Семёнова А.П. Автономная навигация мобильного робота. *В работе представлено исследование и практическая реализация алгоритмов компьютерного зрения для автономного прохождения трассы роботом в рамках первого Кубка России по спортивному программированию в дисциплине "программирование робототехники". Основное внимание уделено разработке алгоритма навигации колесного робота по сложной трассе с различными типами покрытий (трава, лед, земля) и геометрическими особенностями (S-образные участки, "змейки").*

Ключевые слова: *робототехника, соревнования, OpenCV, компьютерное зрение, обработка изображений.*

Savitskaya I.V., Semenova A.P. Autonomous navigation of a mobile robot. *The paper presents a study and practical implementation of computer vision algorithms for autonomous track running by a robot within the framework of the first Russian Cup in sports programming in the discipline "robotics programming". The main attention is paid to the development of an algorithm for navigating a wheeled robot along a complex track with various types of surfaces (grass, ice, earth) and geometric features (S-shaped sections, "snakes").*

Key words*: robotics, competitions, OpenCV, computer vision, image processing.*

Статья поступила в редакцию 18.05.2025
Рекомендована к публикации профессором Павлышом В. Н.

УДК 004.93

Сравнительный анализ методов поиска ключевых точек для распознавания лица

А.П. Семёнова, А.В. Левкина, Е.В. Радевич, Е.О. Сухорукова
Донецкий национальный технический университет
nastena-semenova19@rambler.ru, lewkina.anastasya@yandex.ru,
radevich_katerina@mail.ru

Аннотация
В данной статье проводится анализ методов сопоставления изображений, основанных на ключевых точках, таких как SIFT, SURF, FAST/FREAK, BRISK, KAZE и ORB. Рассматриваются их преимущества и недостатки, а также эффективность в различных условиях, включая обработку изображений. Эксперименты показывают, что алгоритмы успешно справляются с размытием и аффинными преобразованиями. Выбор метода критически важен для достижения высоких результатов в задачах компьютерного зрения, что подчеркивает необходимость учета полученных результатов при выборе алгоритмов для реальных приложений.

Введение

Сопоставление изображений является ключевым аспектом множества задач в области компьютерного зрения, включая восстановление трехмерных структур, SLAM, обнаружение изменений, совмещение изображений, робототехнику, распознавание лиц и эмоций [1-4]. Для решения этой задачи часто применяются ключевые точки, которые представляют собой пиксели с уникальными характеристиками, позволяющими идентифицировать и классифицировать изображения [5-7].

Ключевые точки окружены областями с уникальными отличиями от соседних пикселей, что позволяет создавать их описания в виде векторов признаков или дескрипторов. Элементы дескриптора могут включать модули и направления градиентов, вычисляемых в окрестности ключевой точки. После обнаружения и описания ключевых точек данные могут быть использованы для сопоставления с ключевыми точками на других изображениях [7].

Существует множество алгоритмов для обнаружения и описания ключевых точек, каждый из которых имеет свои особенности и эффективность.

В данной работе рассматриваются следующие методы: SIFT (Scale Invariant Feature Transform), SURF (Speeded-Up Robust Features), FAST/FREAK (Features from Accelerated Segment Test / Fast Retina Keypoint), BRISK (Binary Robust Invariant Scalable Keypoints), KAZE и ORB (Oriented FAST and Rotated BRIEF Features). Все они доступны в расширении Computer Vision Toolbox для MATLAB.

Метод обнаружения ключевых точек FAST/ FREAK

Метод FAST [8] разработанный Эдвардом Ростеном и Т. Драммондом в 2006 году, включает несколько этапов:

1) Определение порога интенсивности. На первом этапе устанавливается пороговое значение интенсивности пикселей изображения, которое используется для классификации пикселей как ключевых точек;

2) Выявление кандидатов на ключевые точки. Алгоритм анализирует интенсивность соседних пикселей (обычно 16 пикселей, расположенных по окружности вокруг анализируемого пикселя), чтобы определить, может ли данный пиксель стать потенциальной ключевой точкой;

3) Ускорение процесса обнаружения. Для повышения скорости обнаружения разработчики метода предложили сначала проверять только четыре пикселя, расположенные на осях Ox и Oy относительно потенциальной ключевой точки. Если хотя бы три из этих пикселей превышают установленный порог интенсивности, выполняется полная проверка по всем 16 точкам. В противном случае пиксель помечается как «не ключевой»;

4) Определение ключевых точек. Пиксель считается ключевой точкой, если он и все 16 его соседей либо ярче порога интенсивности, либо темнее его на заданное значение.

FAST отличается высокой вычислительной эффективностью и подходит для систем реального времени. Существуют усовершенствованные версии, такие как FAST-9, FAST-12, FAST-AGAST и FAST-ER.

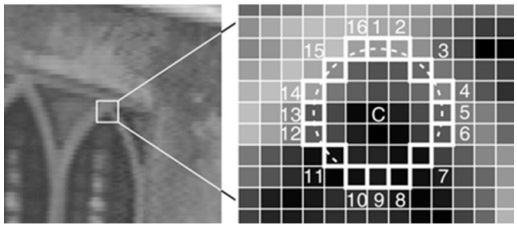


Рисунок 1 – Иллюстрация к детектору Fast

1) FAST-9 и FAST-12. Эти модификации используют 9 или 12 точек окружности вместо 16, что позволяет увеличить скорость выполнения алгоритма;

2) FAST-AGAST – это улучшенная версия метода, использующая алгоритм AGAST (Adaptive and Generic Accelerated Segment Test), который адаптирует пороговое значение для определения ключевых точек в зависимости от уровня яркости окружающих пикселей, что повышает устойчивость к изменениям освещения;

3) FAST-ER (Enhanced Repeatability). Данная версия разработана для повышения устойчивости к шумам и улучшения повторяемости обнаруженных ключевых точек. FAST-ER достигает этого за счет улучшения процесса детектирования с использованием дополнительных шагов, таких как фильтрация и выбор наилучших точек.

Поскольку метод FAST является исключительно детектором ключевых точек, для их описания необходимо использовать другие алгоритмы. Одним из часто применяемых дескрипторов является FREAK (Fast Retina Keypoint), разработанный Александром Алахи, Рафаэлем Ортизом и Пьером Вандергейнстом. FREAK предназначен для быстрого и компактного описания ключевых точек, что делает его подходящим для работы в реальном времени и с большими объемами данных.

Одним из часто применяемых дескрипторов является FREAK (Fast Retina Keypoint), разработанный Александром Алахи, Рафаэлем Ортизом и Пьером Вандергейнстом. FREAK предназначен для быстрого и компактного описания ключевых точек, что делает его подходящим для работы в реальном времени и с большими объемами данных. Дескриптор FREAK является бинарным и имеет длину 512 бит.

Данный метод описания ключевой точки основывается на особенностях зрительной системы человека, в частности, на структуре сетчатки глаза. FREAK использует сетчаточную сетку выборки с круглой формой (см. рис. 2), где плотность точек выше в центре. Для соответствия модели сетчатки алгоритм применяет различные размеры ядер (радиусы) для каждой точки выборки.

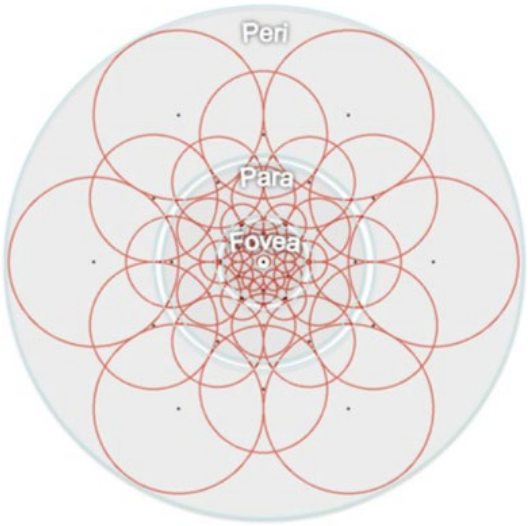


Рисунок 2 – Схема выборки фрагментов ключевой точки, имитирующая рецептивные структуры сетчатки

Уникальность метода заключается в экспоненциальном изменении размеров и перекрытии рецептивных полей. Каждый круг на иллюстрации представляет собой стандартные отклонения гауссовских ядер, примененных к соответствующим точкам выборки.

Метод обнаружения ключевых точек SIFT

Метод SIFT [9] решает проблемы, связанные с вращением, аффинными преобразованиями и изменением интенсивности. Ключевые точки обнаруживаются с помощью каскадной фильтрации и масштабирования пространства, что позволяет находить устойчивые местоположения ключевых точек.

Функция масштабирования является сверткой гауссовой функции переменного масштаба $L(x, y, \sigma)$ с входным изображением $I(x, y)$:

$$L(x, y, \sigma) = G(x, y, \sigma) * I(x, y) = \\ = \frac{1}{2\pi\sigma^2} e^{\frac{x^2+y^2}{2\sigma^2}} * I(x, y). \quad (1)$$

Для эффективного выявления устойчивых местоположений ключевых точек в масштабном пространстве находятся экстремумы функции:

$$D(x, y, \sigma) = (G(x, y, k\sigma) - G(x, y, \sigma)) * I(x, y) = \\ = L(x, y, \sigma) - L(x, y, \sigma). \quad (2)$$

При этом каждая точка выборки сравнивается с восьмью своими соседями на текущем изображении и девятью соседями в шкале сверху и снизу (рис. 3, заимствован из [9]).

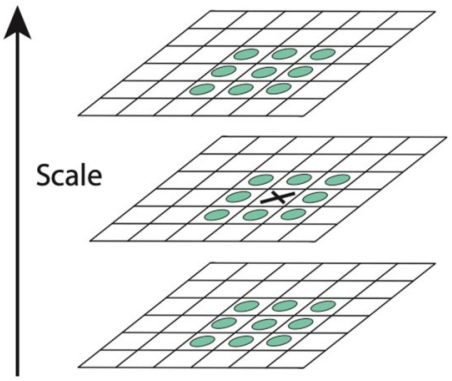


Рисунок 3 – Визуализация сравнения значений ключевой точки с соседними точками в областях 3×3 в текущем и соседних масштабах.

Ключевой точка считается только в том случае, если ее значение больше всех этих соседей или меньше их всех.

Гистограмма ориентации, состоящая из 36 интервалов, охватывающих полный круг в 360 градусов, формируется на основе направления градиента в окрестности ключевой точки. В итоге локальная область вокруг ключевой точки делится на 16 подрегионов размером 4×4, при этом в каждом подрегионе содержится по 8 значений гистограммы ориентации. Таким образом, вектор дескриптора ключевой точки включает 128 значений признаков (16×8).

Метод обнаружения ключевых точек SURF

Метод SURF [10], разработанный Гербертом Бейем, Андреасом Эссемом, Тинне Туйтелаарс и Люком Ван Гулом, является усовершенствованной версией SIFT, предлагая более быстрый и эффективный подход к обнаружению ключевых точек с использованием интегрального представления изображений.

Для обнаружения ключевых точек в методе SURF используется сравнение гессиана $\det H(I)$ функция яркостей $I(x,y),...,x=\overline{1,M}$; $y=\overline{1,N}$:

$$\det H(I) = I_{xx}\cdot I_{yy} - (I_{xy})^2, \qquad (3)$$

где I_{xx}, I_{yy} – частные производные второго порядка по переменным x и y соответственно, I_{xy} – смешанная частная производная второго порядка функции яркости изображения $I(x,y)$.

Для получения оценок названных частных производных как один из возможных вариантов используются матричные маски A_{xx}, A_{yy}, A_{xy}:

$$A_{xx} = \begin{pmatrix} 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 1 & 1 & 0 & -1 & -1 & -1 & 0 \\ 0 & 1 & 1 & 1 & 0 & -1 & -1 & -1 & 0 \\ 1 & 1 & 1 & 1 & 0 & -1 & 1 & 1 & 1 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & -1 & -1 & -1 & 0 & 1 & 1 & 1 & 1 \\ 0 & -1 & -1 & -1 & 0 & 1 & 1 & 1 & 1 \\ 0 & -1 & -1 & -1 & 0 & 1 & 1 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \end{pmatrix}$$

$$A_{xy} = \begin{pmatrix} 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 1 & 1 & 1 & -2 & -2 & -2 & 1 & 1 & 1 \\ 1 & 1 & 1 & -2 & -2 & -2 & 1 & 1 & 1 \\ 1 & 1 & 1 & -2 & -2 & -2 & 1 & 1 & 1 \\ 1 & 1 & 1 & -2 & -2 & -2 & 1 & 1 & 1 \\ 1 & 1 & 1 & -2 & -2 & -2 & 1 & 1 & 1 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \end{pmatrix}$$

Маска A_{yy} получается транспонированием матрицы A_{xx}. Значение гессиана $\det H(I)$ с использованием приведенных масок находится как свертка:

$$\det H(I) = (A_{xx} * I)\cdot(A_{yy} * I) - (0{,}9\cdot A_{xy} * I)^2. \quad (4)$$

Формула, применяемая на практике (4), отличается от (1) наличием корректировочного коэффициента в виде множителя 0,9, который снижает вклад в оценку гессиана смешанной второй производной. Гессиан сохраняет инвариантность к изменениям яркости изображения и поворотам, однако не является инвариантным к изменениям масштаба. Для учета масштабного эффекта применяются фильтры (маски) различных размеров для оценки гессиана. Чтобы сократить количество используемых фильтров, вводятся октавы, в каждой из которых размер фильтра изменяется с определенным шагом, при этом размеры фильтров в соседних октавах перекрываются. Пиксель считается ключевым, если гессиан в нем имеет экстремум как по отношению к соседним пикселям в данной октаве, так и в соседних нижней и верхней октавах того же размера (3×3). Для ускорения вычислений сумм яркостей пикселей в соответствующей прямоугольной области изображения используется интегральное представление изображений.

На втором этапе метода SURF происходит вычисление дескрипторов. Для формирования дескриптора используются анализируемое изображение и набор ключевых точек, определенных на исходном изображении в первом этапе работы алгоритма. Результатом

работы дескриптора является набор векторов признаков. Вектор признаков ключевой точки (пикселя) (i_0, j_0) представляет собой набор функций (признаков) $f_k(w(i_0, j_0)), \quad k = 1, K,$ описывающих характерные особенности данной ключевой точки в её окрестности $w(i_0, j_0)$ Эти признаки формируются на основе информации об интенсивности, цвете и текстуре данной точки.

Дескриптор каждой ключевой точки в методе SURF содержит 64 (в других вариантах 128 чисел) [10]. Основная задача дескриптора – определить направление, в котором модуль градиента $|\nabla I| = \left(\left(\frac{\partial I}{\partial x} \right)^2 + \left(\frac{\partial I}{\partial y} \right)^2 \right)^{1/2}$ функция яркости принимает максимальное значение.

Метод обнаружения ключевых точек KAZE

Метод KAZE [11] представляет собой усовершенствованную версию SIFT, разработанную Пабло Фернандесом Алькантариллом и его коллегами. Этот метод направлен на повышение точности и устойчивости к различным видам временных и геометрических трансформаций, таким как поворот, масштабирование, изменения освещения, шум и размытие, а также на работу с изображениями, имеющими различные структуры и фоны. KAZE применяет адаптивный детектор порога для поиска локальных максимумов значений гессиана в пространстве и масштабе.

Чтобы избежать размытия краев и потери деталей в пространстве линейного масштаба, созданного с использованием гауссовского фильтра, формируется пространство нелинейного масштаба с помощью нелинейного диффузионного фильтра. Это пространство строится с применением методов аддитивного операторного расщепления (AOS) и диффузии с переменной проводимостью. Для выявления значимых точек вычисляется нормализованное значение определителя гессиана на различных уровнях масштаба.

$$\det H(I) = \sigma^2 (I_{xx} \cdot I_{yy} - I_{xy}{}^2), \qquad (5)$$

где I_{xx}, I_{yy} – производные второго порядка по горизонтали и вертикали соответственно, а I_{xy} – смешанная производная второго порядка. Каждый экстремум ищется в прямоугольном окне размером $\sigma_i \times \sigma_i$ для текущего i, верхнего $i+1$, нижнего $i-1$, отфильтрованных изображений.

Для ускорения поиска экстремумов результаты вычислений проверяются сначала в окне размером 3×3 пикселя, чтобы быстрее отбросить минимальные ответы.

В конечном итоге положение ключевой точки оценивается с субпиксельной точностью с использованием метода, предложенного в [14]. Набор производных первого и второго порядка аппроксимируется с помощью фильтров Шарра размером 3×3 с различными шагами производной σ_i. Производные второго порядка аппроксимируются с использованием последовательных фильтров Шарра [13]. Применение двоичного дескриптора значительно ускоряет процесс описания признаков в нелинейном масштабном пространстве. Дескриптор KAZE инвариантен к изменениям масштаба и вращению, а также требует минимального объема для хранения.

Метод обнаружения ключевых точек BRISK

Метод BRISK объединяет методы поиска и описания ключевых точек. Для поиска он использует усовершенствованный алгоритм FAST и вычисляет максимумы не только в плоскости изображения, но и в масштабном пространстве [12].

Пирамида масштабного пространства создается путем понижающей дискретизации исходного изображения c_0 с использованием 4 октав c_i и 4 внутриоктав d_i, каждая из которых располагается между слоями c_i и c_{i+1} (см. рис. 4, заимствованном из [12]). Детекторы FAST 9–16 применяются на каждой октаве и внутриоктаве отдельно для выявления потенциальных ключевых точек. Для достижения инвариантности к масштабу предлагается выбирать точку с максимальным значением интенсивности.

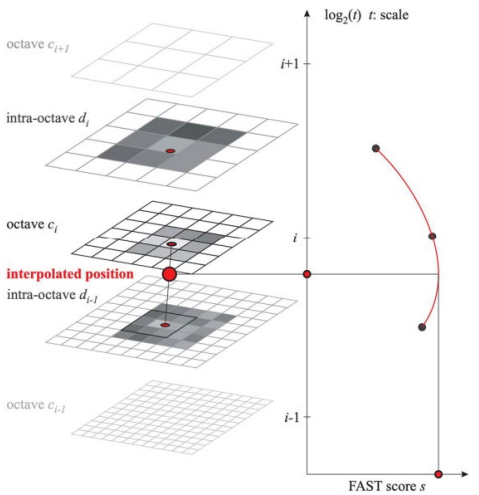


Рисунок 4 – Ключевая точка идентифицируется в октаве путем сравнения 8 пикселей окрестности c_i, а также соответствующих участков соседних слоев выше и ниже

Метод обнаружения ключевых точек ORB

Метод ORB [13] представляет собой комбинацию детектора FAST и усовершенствованного дескриптора BRIEF. Он демонстрирует высокую производительность благодаря использованию бинарного дескриптора, что делает его подходящим для применения в системах компьютерного зрения в реальном времени. В модифицированной версии детектора FAST, используемой в ORB, алгоритм анализирует центральный пиксель и 12 пикселей, расположенных в кольце радиусом 3 пикселя. Пиксель считается ключевым, если существует набор из 9 или более пикселей, все яркостные различия между которыми превышают или не достигают определенного порога. Улучшение BRIEF заключается в добавлении инвариантности к ориентации в дескрипторе. Однако время сопоставления ключевых точек в методе ORB значительно выше по сравнению с другими методами, что ограничивает его применение для обработки изображений в реальном времени.

Экспериментальные исследования

Сравнение исследуемых методов проводилось на основе данных, полученных экспериментальным путем с использованием платформы программирования и числовых вычислений MATLAB (версия 23.2.0.2485118 (R2023b) Update 6) и расширения Computer Vision Toolbox. Вычисления выполнялись на компьютере Apple MacBook Pro с процессором Apple M1 Max и 32 ГБ общей оперативной и видеопамяти. Операционная система – macOS Sonoma версии 14.4.1 (23E224).

В исследовании использовался открытый набор данных Oxford, который включает 8 групп изображений, каждая из которых подвергнута одному из типов преобразований, таких как поворот, масштабирование, изменение угла съемки, размытие, изменение освещения и сжатие. Каждая группа содержит по 6 изображений с различной степенью искажений (см. табл. 1).

Таблица 1 - Группы набора данных Oxford.

Название группы	Тип преобразования
graff	Изменение угла съемки
wall	Изменение угла съемки
boat	Масштабирование и вращение
bark	Масштабирование и вращение
bikes	Размытие
trees	Размытие
ubc	Сжатие (наличие шумов)
leuven	Изменение освещения

Все сравниваемые алгоритмы входят в состав расширения Computer Vision Toolbox в виде отдельных функций, и их вызов происходил с настройками по умолчанию.

Таблица 2 - Среднее количество обнаруженных ключевых точек.

Название группы	SIRF	SURF	FAST	BRISK	KAZE	ORB
graff	3247	1775	1123	3249	8084	13868
wall	8961	2308	3051	4742	13652	47618
boat	5360	1907	2768	5389	8593	27397
bark	5708	1817	1322	2580	5660	26150
bikes	1197	610	150	545	4965	4543
trees	7922	3355	4563	7873	14486	44549
ubc	4848	1610	2220	3937	6460	25946
leuven	1605	639	756	13335	4486	7276

Оценка эффективности работы алгоритмов обнаружения ключевых точек проводилась на основе двух показателей: количества найденных ключевых точек и времени обработки одного изображения. В таблице 2 представлены усредненные результаты по количеству обнаруженных ключевых точек, сгруппированные по категориям изображений.

Поскольку скорость обработки изображений не зависит от их типа, результаты по этому показателю можно сравнивать в общем, без разбивки на группы (см. рис. 5).

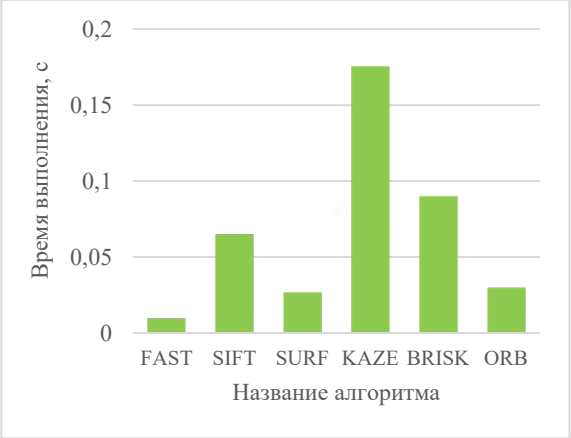


Рисунок 5 – Среднее время работы алгоритмов обнаружения ключевых точек для обработки одного изображения

Из таблицы 2 видно, что алгоритмы SURF и FAST в среднем обнаружили меньше ключевых точек по сравнению с другими методами. Алгоритмы SIFT и BRISK выявили примерно в два раза больше ключевых точек, а алгоритм ORB продемонстрировал значительно большее количество (в среднем в 3–10 раз) обнаруженных ключевых точек.

Однако эффективность метода не может быть оценена исключительно по количеству найденных ключевых точек. Эти данные помогают понять, как по-разному работают

алгоритмы. Следует отметить, что алгоритм FAST показал наихудшие результаты в группе "bikes" и не обнаружил ключевых точек на нескольких изображениях с сильным размытием. В то же время остальные алгоритмы, хотя и выявили значительно меньше ключевых точек, все же обеспечили достаточные результаты для сопоставления изображений. Похожая зависимость между количеством обнаруженных ключевых точек и уровнем размытия наблюдается и в группе "trees", хотя в этом случае все алгоритмы нашли некоторое количество точек.

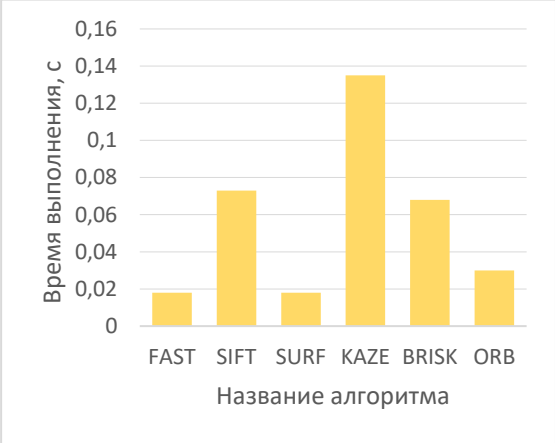


Рисунок 6 – Среднее время работы алгоритмов обнаружения ключевых точек для обработки одного изображения

Время работы алгоритмов значительно варьируется, и здесь можно выделить явного лидера. Метод FAST показал наилучшие результаты, в среднем затрачивая 6 миллисекунд на обработку одного изображения. Алгоритмы SURF и ORB также продемонстрировали хорошие результаты со средним временем 20 и 25 миллисекунд соответственно. Остальные алгоритмы потребовали значительно больше времени на вычисления, причем наихудший результат показал KAZE – 174 миллисекунды.

Для оценки эффективности дескрипторов использовались время работы алгоритма (рисунок 6), время сопоставления ключевых точек двух изображений (рисунок 7) и показатель корректности сопоставления "Inlier Ratio" (рисунок 8).

Из рисунка 6 видно, что среднее время работы алгоритмов-дескрипторов аналогично изменениям времени работы детекторов. KAZE снова выполняется дольше всех остальных алгоритмов, в то время как FREAK (дескриптор для точек, обнаруженных методом FAST), SURF и ORB показывают максимальную скорость выполнения.

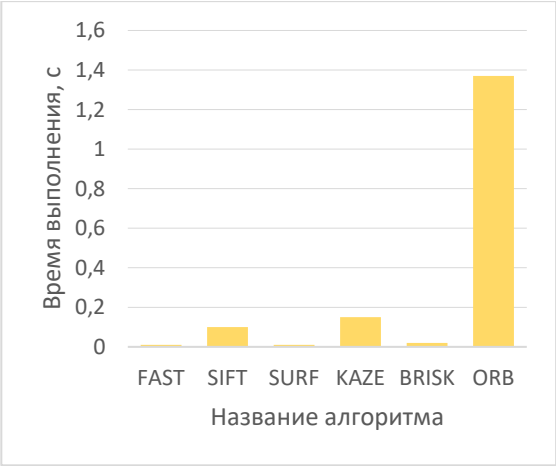


Рисунок 7 – Среднее время сопоставления ключевых точек для двух изображений

Тем не менее, время, затраченное на сопоставление, нивелирует преимущества в скорости обнаружения и описания ключевых точек метода ORB. В результате эксперимента, включавшего попарное сопоставление каждого изображения внутри группы, среднее время сопоставления с использованием этого метода составило 1,33 секунды, что в десятки раз дольше по сравнению с другими методами. Это, вероятно, связано с большим количеством сопоставляемых точек.

С точки зрения времени сопоставления наилучшие результаты продемонстрировали алгоритмы SURF, FREAK и BRISK, затратившие 12, 13 и 24 миллисекунды соответственно.

Для оценки качества обнаруженных и описанных ключевых точек использовался параметр Inlier Ratio, алгоритм вычисления которого включает следующие шаги:

1. Вычисление фундаментальной матрицы на основе сопоставленных точек.
2. Определение эпиполярных линий для каждого из двух изображений.
3. Поиск ближайших точек на эпиполярных линиях к точкам изображения.
4. Оценка расстояния между найденными ближайшими точками и исходными парами точек на изображениях.
5. Установление порогового расстояния, ниже которого сопоставление точек считается корректным.
6. Подсчет количества «inliers» – пар точек, расстояние между которыми меньше установленного порога.
7. Вычисление соотношения количества «inliers» к общему числу пар точек.

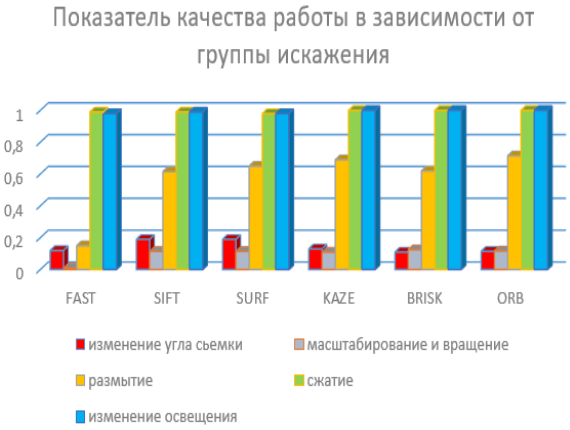


Рисунок 8 – Качество работы алгоритма в зависимости от типа преобразований изображений губы

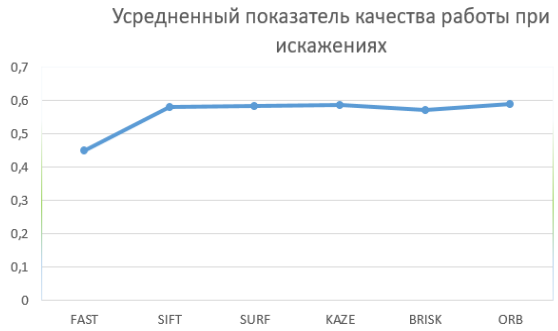


Рисунок 9 – Усредненный показатель качества работы алгоритма при искажениях на изображении

Чем ближе значение Inlier Ratio к единице, тем более точным считается сопоставление. Из рисунка 8 видно, что все алгоритмы успешно справились с задачами обработки изображений, подвергшихся сжатию и изменению угла освещения. Также наблюдается высокая корректность обработки изображений с размытием. При этом стоит отметить, что все алгоритмы, кроме FREAK, продемонстрировали высокий уровень работы с размытыми изображениями, испытывая трудности только с максимальным уровнем размытия. В то время как FREAK смог корректно сопоставить только изображения с минимальным размытием.

Аналогичное отставание алгоритма FREAK наблюдается и при работе с изображениями, подвергшимися масштабированию и вращению. Остальные алгоритмы показали стабильные результаты на равном уровне.

По итогам эксперимента видно, что обработка изображений, подверженных аффинным преобразованиям, представляет собой более сложную задачу для всех исследуемых алгоритмов. Это указывает на то, что метод сопоставления изображений с использованием ключевых точек более эффективен при незначительных преобразованиях данного типа, например, при обработке кадров видеопотока или аэрофотосъемки.

Заключение

Подводя итог, можно заключить:

- алгоритмы SURF и FAST в среднем обнаружили меньше ключевых точек, чем SIFT и BRISK;
- ORB продемонстрировал значительно большее количество обнаруженных ключевых точек;
- метод FAST показал наилучшие результаты по времени обработки (6 мс), в то время как KAZE потребовал 174 мс;
-время сопоставления наилучших результатов показали SURF, FREAK и BRISK.

Проведенный анализ позволяет учитывать полученные результаты при выборе метода для реальных приложений. Рассмотренные алгоритмы являются стандартом в области компьютерного зрения. Эксперимент выявил особенности каждого алгоритма, что поможет в дальнейшем выборе подходящих методов для различных задач компьютерного зрения.

Литература

1. Семенова, А. П. Область применения алгоритма распознавания эмоций в информационных технологиях / А. П. Семенова, А. С. Миненко, Т. В. Ванжа // Цифровой регион: опыт, компетенции, проекты: сборник статей Международной научно-практической онференции (г. Брянск, 30 ноября 2018 г.) [Электронный ресурс]. – Брянск: Брян. гос. инженерно-технол. ун-т., 2018. – С. 443-446.

2. Миненко, А. С. Анализ эмоционального состояния человека по фотографическим изображениям / А. С. Миненко, А. П. Семенова // Информатика, управляющие системы, математическое и компьютерное моделирование (ИУСМКМ – 2019). Материалы X Международной научно-технической конференции в рамках V Международного Научного форума Донецкой Народной Республики «Инновационные перспективы Донбасса». – Донецк: ДонНТУ, 2019. – С. 123-126.

3. Левкина, А. В. Область применения систем распознавания эмоций / А. В. Левкина, Е. В. Радевич, А. П. Семёнова // Донбасс будущего глазами молодых ученых: сб. материалов науч.-техн. конф. для студ., асп. и мол. уч. – Донецк: ДонНТУ, 2024. – С. 158-162.

4. Семёнова, А. П. Анализ мимических выражений для задачи распознавания эмоций / А. П. Семёнова, В. Н. Павлыш // Проблемы искусственного интеллекта, 2020. - № 4 (19). - С. 69-79.

5. Миненко, А. С. Формальная модель эмоций / А. С. Миненко, А. П. Семенова // Проблемы искусственного интеллекта, 2018. - №3(10). - С. 84-93.

6. Семенова, А. П. Математическая модель эмоций // Международная научно-техническая конференция молодых ученых БГТУ им. В.Г. Шухова. Материалы национальной конференции с международным участием. – Белгород, 2019. – С. 4584-4587.

7. Семёнова, А. П. Поиск ключевых точек лица для задачи распознавания эмоций/ А. П. Семёнова, В. Н. Павлыш // Информатика и кибернетика, 2021. - № 1-2 (23-24). - С. 59-64.

8. Rosten, E. Machine learning for high-speed corner detection / E. Rosten, T. Drummond // Computer Vision – ECCV 2006. – PP. 430-443.

9. Lowe, D.G. Distinctive Image Features from Scale-Invariant Keypoints / D.G. Lowe // International Journal of Computer Vision, 2004. –№ 60. – PP. 91-110.

10. Bay, H. SURF: speeded up robust features / H. Bay, T. Tinne, V.G. Luc // Computer Vision and Image Understanding, 2008. – № 3(110). – PP. 346-359.

11. Alcantarilla, P. F. KAZE Features / P. F. Alcantarilla, A. Bartoli, A. J. Davison // Computer Vision – ECCV 2012. – PP. 214-227.

12. Leutenegger, S. BRISK: binary robust invariant scalable keypoints / S. Leutenegger, M. Chli, R. Y. Siegwart // The IEEE International Conference on Computer Vision (ICCV). – November 2011, Barcelona, Spain. – PP. 2548-2555.

13. Rublee, E. ORB: an efficient alternative to SIFT or SURF / E. Rublee [et al.] // The IEEE International Conference on Computer Vision (ICCV). – November 2011, Barcelona, Spain. – PP. 2564-2571.

14. Brown, M. Invariant features from interest point groups / M. Brown, D. Lowe // In: British Machine Vision Conf., BMVC, Cardiff, UK (2002).

Семёнова А.П., Левкина А.В., Радевич Е.В., Сухорукова Е.О. Сравнительный анализ методов поиска ключевых точек для распознавания лица. *В данной статье проводится анализ методов сопоставления изображений, основанных на ключевых точках, таких как SIFT, SURF, FAST/FREAK, BRISK, KAZE и ORB. Рассматриваются их преимущества и недостатки, а также эффективность в различных условиях, включая обработку изображений. Эксперименты показывают, что алгоритмы успешно справляются с размытием и аффинными преобразованиями. Выбор метода критически важен для достижения высоких результатов в задачах компьютерного зрения, что подчеркивает необходимость учета полученных результатов при выборе алгоритмов для реальных приложений.*

Ключевые слова: *ключевые точки лица, контурная модель, распознавание образов.*

Semenova A.P., Levkina A.V., Radevich E.V., Sukhorukova E.O. Comparative analysis of key point search methods for face recognition. *This article analyzes image matching methods based on key points such as SIFT, SURF, FAST/FREAK, BRISK, KAZE and ORB. Their advantages and disadvantages, as well as their effectiveness in various conditions, including image processing, are considered. Experiments show that algorithms successfully cope with blurring and affine transformations. The choice of method is critically important for achieving high results in computer vision tasks, which emphasizes the need to take into account the results obtained when choosing algorithms for real-world applications.*

Key words*: key points of the face, contour model, image recognition.*

Статья поступила в редакцию 18.05.2025
Рекомендована к публикации профессором Павлышом В. Н.

УДК 004.8:339.138

Экономическая эффективность автоматической модерации с помощью искусственного интеллекта в e-commerce

А.Р. Ястребов[*1], А.В. Боднар[*2]

[*1] магистрант, Донецкий национальный технический университет,
yastrebov_sasha98@mail.ru

[*2] к.э.н., доцент кафедры, Донецкий национальный технический университет,
linabykova@ya.ru

Аннотация

В статье рассматриваются теоретические и практические аспекты модерации контента в электронной коммерции, включая её роль в обеспечении соответствия законодательным, этическим и коммерческим требованиям. Особое внимание уделяется преимуществам автоматической модерации на основе искусственного интеллекта (ИИ) по сравнению с ручными методами, а также её влиянию на экономическую эффективность, скорость обработки данных и качество пользовательского опыта.

Введение

С ростом электронной коммерции объёмы пользовательского контента стремительно увеличиваются. По данным Grand View Research, в 2023 году мировой рынок услуг по модерации контента оценивался в 9,67 млрд долларов и прогнозируется его рост до 22,78 млрд долларов к 2030 году [1].

Ручная модерация становится всё менее эффективной: средняя стоимость услуг по модерации составляет растет, при этом человеческий фактор приводит к ошибкам, что может повлечь за собой финансовые и репутационные потери.

Настоящая статья посвящена теоретическому анализу экономической эффективности внедрения автоматизированной модерации на базе ИИ в e-commerce и её потенциалу для оптимизации бизнес-процессов. Выбор ИИ в качестве решения обусловлен его способностью эффективно решать задачи, которые ранее требовали значительных человеческих ресурсов.

Теоретические аспекты модерации в e-commerce

Модерация в электронной коммерции это проверка информации о товарах с целью соответствия законодательным, этическим и коммерческим требованиям платформы. При работе с карточкой товара основное внимание уделяется изображениям, описаниям, отзывам и пользовательскому контенту, влияющему на принятие решения покупателями и на общую репутацию торговой площадки.

Ключевыми задачами модерации являются:

- защита пользователей от недостоверной или вредоносной информации;
- соблюдение законодательства;
- обеспечение честности и прозрачности взаимодействия между продавцом и покупателем;
- повышение качества пользовательского опыта и доверия к платформе;

Без эффективной модерации площадка сталкивается с рисками размещения запрещенных товаров или вводящей в заблуждение информации, что может привести к юридическим разбирательствам, оттоку пользователей и ухудшению позиций на рынке.

Модерация напрямую влияет на удовлетворенность пользователей и эффективность коммерческих операций, покупатели ожидают, что информация о продукции будет достоверной, полной и актуальной.

Недостаточная или некачественная модерация контента может привести к существенным финансовым потерям для компаний:

- ***Штрафы и санкции***. В ЕС компании могут быть оштрафованы на сумму до 6% от годового оборота за нарушение норм модерации контента, как предусмотрено в Digital Services Act;
- ***Потеря клиентов***. Платформы с низким уровнем модерации теряют доверие пользователей, что приводит к снижению пользовательской базы и доходов;
- ***Репутационные риски***. Скандалы, связанные с недостаточной модерацией, могут нанести долгосрочный ущерб репутации компании, влияя на её финансовые результаты;
- ***Убытки от мошенничества в электронной коммерции***. По оценкам, 3% от общего дохода электронной коммерции теряется ежегодно из-за мошенничества [2].

Сравнительный анализ стандартов модерации, представлен в таблице 1.

Таблица 1 - Сравнение стандартов модерации в ЕС, США и РФ

Страна	Уровень модерации	Особенности
РФ	Средний	Платформы обязаны удалять запрещённый контент по требованию Роскомнадзора. Возможны блокировки и уголовная ответственность за несоблюдение требований.
ЕС	Высокий	Принятие Закона о цифровых услугах, предусматривающего штрафы до 6% от глобального оборота за несоблюдение требований. Платформы обязаны активно удалять незаконный контент.
США	Низкий	Платформы имеют широкую защиту от ответственности за пользовательский контент, что снижает стимулы для активной модерации.
Китай	Высокий	Китай придерживается системы строгой ответственности, в которой онлайн-платформы должны «активно отслеживать, фильтровать и удалять контент в соответствии с законодательством. Неспособность сделать это подвергает посредника риску штрафов, уголовной ответственности и отзыва лицензий на бизнес или СМИ

Требования к модерации

Современные торговые платформы предъявляют к процессу модерации высокие требования, определяемые как внутренними стандартами качества, так и внешними нормативными актами. Основными параметрами, по которым оценивается эффективность модерации, являются скорость обработки, точность принятия решений и соответствие контента нормативным требованиям.

Высокая конкуренция и стремительно меняющийся спрос факторы, из-за которых время вывода товаров является очень важным фактором. Продавцы ожидают что их товар будет доступен в максимально кратчайшие сроки после загрузки. Согласно исследованию DataReportal, задержка с публикацией товара даже на 24–48 часов может негативно сказаться на продажах, особенно при участии в акциях или сезонных распродажах.

Также, размещаемый контент должен соблюдать закон о рекламе и защите прав потребителей, запрет на продажу определенных категорий товаров, этические стандарты.

В разных странах действуют различные нормативы: в ЕС это директивы по защите прав потребителей, в России — требования Роскомнадзора, а также законы о защите информации. Глобальный характер работы многих платформа делает соблюдение норм многослойной задачей, требующей постоянного обновления политики модерации и мониторинга новых требований.

Таким образом, эффективная система модерации должна быть одновременно быстрой, точной и юридически корректной, что требует как организационных решений, так и технологических инноваций — в том числе с использованием искусственного интеллекта.

Несмотря на привычность и относительную прозрачность ручной модерации, она обладает рядом существенных ограничений, особенно в условиях стремительного роста объемов контента в e-commerce.

Основные проблемы ручного подхода связаны с высокой стоимостью, рисками, обусловленными человеческим фактором, а также ограниченной масштабируемостью. Допускаемые модераторами ошибки неизбежны независимо от квалификации. Усталость, субъективность суждений, нехватка времени и неясность регламентов приводит к некорректной модерации.

На некоторых платформах, например Ozon, фиксируются доплаты и штрафы в зависимости от уровня ошибок: при превышении 4% бонусы не начисляются, а при достижении 7% и выше — полностью отменяются. Такая зависимость указывает на системную проблему: даже при мотивации персонала сложно обеспечить стабильно высокое качество при высоких объёмах задач.

Затраты на ручную модерацию складываются из расходов на персонал, инфраструктуру и управление. Таким образом, при масштабировании платформы затраты на персонал возрастают линейно, что ограничивает экономическую эффективность модели.

Ручная модерация плохо масштабируется: увеличение количества контента требует пропорционального роста численности сотрудников. Это создает узкие места при пиковых нагрузках (например, в праздничные периоды), снижает скорость обработки и увеличивает время выхода товара на рынок. Кроме того, найм, обучение и удержание модераторов являются затратными процессами, особенно в условиях высокой текучести кадров, типичной для этой профессии.

Сравнение модерации контента ИИ и человеком предоставлены в таблице 2.

Таблица 2 – Сравнение ИИ и человека при модерации контента

Показатель	ИИ	Ручная модерация
Скорость	Обработка в реальном времени, позволяющая анализировать миллионы единиц контента в минуту.	В среднем 2–5 минут на одну единицу контента.
Точность	Высокая точность в структурированных задачах.	Лучше справляется с контекстом и нюансами, такими как сарказм и культурные особенности
Объемы обработки	Способен обрабатывать огромные объемы данных без снижения производительности.	Ограничен физическими возможностями и численностью команды модераторов
Затраты	Высокие начальные инвестиции, но низкие операционные расходы при масштабирова-нии.	Постоянные затраты на оплату труда, обучение и управление персоналом.
Психологическая нагрузка	Отсутствует.	Модераторы подвержены стрессу и эмоциональному выгоранию.

В совокупности эти факторы делают ручную модерацию уязвимой моделью в условиях современной электронной торговли. Это обосновывает интерес к более эффективным и масштабируемым решениям, включая автоматизацию на основе искусственного интеллекта.

Возможности искусственного интеллекта в автоматической модерации

Искусственный интеллект может сыграть ключевую роль в трансформации процессов модерации контента на платформах электронной коммерции. Основным преимуществом можно выделить обработку больших объёмов данных с высокой скоростью и последовательностью, минимизируя человеческие ошибки и обеспечивает масштабируемость.

Искусственный интеллект может выполнять различные функции в рамках модерации:

- классификация и фильтрация изображений и текстов;
- распознавание запрещённых товаров и нарушений политик платформы;
- обнаружение спама, манипулятивных описаний, агрессивного или оскорбительного контента;
- автоматическое обновление правил модерации на основе новых нормативов или рыночных условий.

Современные модели, основанные на машинном обучении и глубоком обучении, особенно нейросети архитектуры BERT и GPT, показывают высокие результаты в задачах анализа естественного языка (NLP), позволяя автоматически интерпретировать и оценивать смысловое содержание карточек товаров, описаний и отзывов. Например, в задачах по выявлению вредоносного контента точность моделей может достигать 94%, а полнота — 95%.

Современные технологии автоматической модерации опираются на достижения в области искусственного интеллекта, включая нейросетевые архитектуры, обработку естественного языка (NLP) и компьютерное зрение.

Глубокие нейросети лежат в основе большинства современных систем автоматической модерации.

Сети типа Convolutional Neural Networks (CNN) широко применяются для анализа изображений и визуального контента, обеспечивая высокую точность в распознавании объектов, логотипов, запрещённых товаров и сцен насилия. Например, CNN-модели успешно применяются на платформах вроде YouTube и Facebook для фильтрации видео- и фотоматериалов на предмет нарушений.

Для анализа текстов — описаний товаров, отзывов, названий и других текстовых элементов — применяются модели на базе трансформеров, такие как BERT (Bidirectional Encoder Representations from Transformers), RoBERTa, GPT и их производные.

Они способны выявлять скрытые смыслы, токсичность, ненормативную лексику, манипулятивные формулировки и даже элементы мошенничества в пользовательском контенте. Кроме того, эти модели адаптируются под специфику языка и контекста e-commerce, включая понимание товарной терминологии и структуры карточек товаров.

Методы компьютерного зрения

применяются для автоматического анализа визуального представления товара: соответствие изображения заголовку, наличие запрещённых объектов, неприемлемый фон, водяные знаки, качество изображения. Такие методы особенно важны для модерации контента, который не всегда может быть адекватно оценен по одному лишь тексту. На крупных площадках — таких как Amazon и Taobao — ИИ-системы на базе компьютерного зрения ежедневно анализируют миллионы изображений с высокой точностью классификации.

Использование моделей BERT, GPT и CNN стало основой для создания высокоэффективных систем автоматической модерации в электронной коммерции. Эти модели обеспечивают широкий функционал — от анализа текстов и изображений до выявления нарушений и прогнозирования поведения пользователей.

Модель BERT (Bidirectional Encoder Representations from Transformers), разработанная Google, продемонстрировала высокую эффективность в задачах понимания естественного языка, таких как классификация, извлечение сущностей, определение тональности и контекстной фильтрации. В e-commerce BERT и его модификации, например RoBERTa или DistilBERT, применяются для анализа описаний товаров на предмет соответствия правилам, выявление агрессивных, токсичных или манипулятивных отзывов и для классификации карточек товаров по категориям.

Модели семейства GPT (Generative Pre-trained Transformer), особенно GPT-3 и GPT-4, демонстрируют способность к генерации и интерпретации сложных текстов, что делает их полезными в задачах автоматического редактирования карточек товаров, выявления скрытых нарушений или несоответствий между текстом и изображением. Кроме того, GPT может применяться для распознавания запрещенной или вводящей в заблуждение информации и для построения диалоговых систем для взаимодействия с модераторами.

Сверточные нейронные сети (CNN) являются стандартом де-факто в задачах компьютерного зрения. В e-commerce их применяют для автоматического анализа фотографий товара, выявления запретных изображений, оценки качества изображения.

Оценка эффективности моделей искусственного интеллекта в задачах модерации осуществляется на основе набора ключевых метрик:

- accuracy (точность) — доля правильно классифицированных объектов от общего числа. Используется для оценки общей надёжности модели;
- recall (полнота) — показатель, отражающий способность модели выявлять все релевантные случаи, особенно важен при обнаружении нарушающего контента;
- F1-score — гармоническое среднее между precision и recall; даёт сбалансированную оценку, особенно полезную при несбалансированных классах.

Согласно данным отраслевых исследований, модели на базе архитектуры BERT-CNN достигают следующих показателей при выявлении чувствительного и нежелательного контента:

- точность (accuracy) — до 94 %;
- полнота (recall) — до 95 %;
- F1-score — 94 % [3].

Эти значения указывают на то, что современные ИИ-системы способны заменять или дополнять ручную модерацию без серьёзных потерь в качестве, особенно в задачах первичной фильтрации.

Экономические выгоды от внедрения ИИ на примере кейсов некоторых компаний:

- ***Unitary***. Система ИИ обрабатывает до 26 миллионов видео в день, обслуживая более 100 миллионов пользователей. Использование Amazon EKS и Karpenter позволило эффективно масштабировать инфраструктуру и сократить время запуска контейнеров на 80 % что привело к снижению общих затрат на 50–70% в течение 18 месяцев [4];
- ***123RF***. Внедрение генеративного ИИ на базе Amazon Bedrock и моделей Anthropic Claude 3 позволило автоматизировать перевод и модерацию контента, значительно повысив эффективность процессов и снизив затраты на перевод контента на 95 % и ускорение проверки контента до 100 раз [5];
- ***Ozon***. Разработка собственной системы модерации на основе машинного обучения позволила Ozon автоматизировать проверку новых позиций, сократив время обработки карточки до 3 минут. Это повысило скорость размещения товаров и снизило нагрузку на команду модераторов и сократило время модерации карточек товаров в 10 раз [6];
- ***Wildberries***. Wildberries предотвратила более 94 тысяч фейковых заказов благодаря обновленной системе ИИ-модерации. Wildberries эффективно выявляет подозрительные действия и предотвращает их, что способствует снижению числа мошеннических заказов и повышению доверия покупателей [7].

Таким образом, показатели точности ИИ-моделей уже достигли уровня, достаточного для коммерческого применения в крупных проектах.

Дальнейшее развитие технологий и дообучение моделей позволяют наращивать эффективность, снижать число ложных срабатываний и адаптироваться к новым типам угроз.

Потенциал автоматизации модерации: экономический взгляд

Одним из ключевых преимуществ внедрения искусственного интеллекта в процессы модерации электронной коммерции является его высокая степень автоматизации.

Такая степень автоматизации обусловлена несколькими технологическими возможностями:

- модели ИИ обрабатывают текст, изображения и видео в режиме реального времени;
- нейросети способны выявлять нарушения политик, даже если они выражены неявно;
- система может самообучаться на новых данных и адаптироваться к изменяющимся правилам.

Реализация автоматической модерации позволяет платформам:

- сократить время проверки карточек с часов и дней до минут или секунд;
- уменьшить объём ручной работы, перераспределив ресурсы модераторов на более сложные или спорные кейсы;
- уменьшает выгорание сотрудников;
- позволяет сосредоточить внимание на сложных и нестандартных кейсах;
- снизить операционные затраты, включая расходы на персонал, обучение, контроль качества и исправление ошибок.

Чтобы измерить прибыльность вложений проведем расчёт ROI при внедрении ИИ в модерацию контента.

По данным Dream Job, средняя зарплата модератора в России в 2025 году составляет 39 тыс. руб. в месяц. Для средних компаний затраты на внедрение ИИ могут составлять от 5 до 10 миллионов рублей, включая разработку, интеграцию и обучение персонала, для расчётов возьмем 7 миллионов рублей.

Число модераторов до внедрения ИИ возьмем 100, а после внедрения сократим их число до 20. Так как ИИ покрывает примерно 90 % работы, достаточно оставить около 10–20 % персонала, чтобы закрыть "человеческую часть". Выполним расчеты.

Затраты на персонал в год (до внедрения ИИ):

$$100 * 39\,000 * 12 = 46\,800\,000 \text{ руб.}$$

Затраты на персонал в год (после внедрения ИИ):

$$20 * 39\,000 * 12 = 9\,360\,000 \text{ руб.}$$

Экономия в год за счет внедрения технологий ИИ:

$$46\,800\,000 - 9\,360\,000 = 37\,440\,000 \text{ руб.}$$

ROI:

$$ROI = \left(\frac{37\,440\,000 - 7\,000000}{7\,000000}\right) \times 100 = \left(\frac{30\,440\,000}{7\,000000}\right) \times 100 = 434,85.$$

ROI = 434,85 % — это означает, что инвестиции в ИИ-модерацию окупаются более чем в четыре раза уже в первый год эксплуатации.

Кроме того, ИИ способен ускорять цикл обновления правил — если в ручной системе внедрение новых регламентов занимает недели, то в автоматизированной — часы или даже минуты при наличии подходящего API и обучающей выборки.

В результате ускоряется:

- обновление товарного ассортимента;
- участие продавцов в маркетинговых кампаниях;
- оборачиваемость товаров на платформе.

Описание предлагаемого инновационного решения

В рамках теоретического обоснования экономической эффективности автоматической модерации можно предложить концептуальную архитектуру ИИ-системы, предназначенную для обработки товарного контента на онлайн-платформах.

Основное внимание уделяется функциональной структуре, возможности интеграции и логике работы.

Архитектура автоматической модерации строится по модульному принципу и включает следующие ключевые компоненты:

1) ***Модуль предварительной обработки.*** Этот модуль отвечает за нормализацию входящих данных (текста, изображений, метаданных), их классификацию и сегментацию.

2) ***Модуль обработки текстов (NLP).*** Модуль реализован на основе моделей трансформерного типа (например, BERT или GPT), анализирует наименование товара, описание, теги, пользовательские отзывы.

3) ***Модуль компьютерного зрения (CV).*** Применяет сверточные нейронные сети (CNN) для анализа изображений на предмет запрещённого контента, соответствия заголовку, качества фото.

4) ***Бизнес-логика и правила.*** Слой, объединяющий выводы моделей с внутренними правилами платформы, нормативными актами и региональными ограничениями.

5) ***Интерфейс взаимодействия с модератором.*** Позволяет оператору подтверждать или отклонять спорные случаи, контролировать работу системы и вносить корректировки.

Интеграция системы в действующую инфраструктуру маркетплейса или интернет-

магазина возможна через API-интерфейсы. Это позволяет обрабатывать заявки на публикацию товара в режиме реального времени, интегрироваться с внутренними системами, получать обратную связь от операторов для дообучения моделей и автоматически обновлять правила и фильтры при изменении регламентов или законодательства.

Система должна поддерживать режим самообучения на основе обратной связи и логов модерации, что обеспечивает повышение точности с течением времени.

Таким образом, предлагаемое решение представляет собой универсальный модерационный модуль, способный адаптироваться к различным платформам и обеспечивать высокое качество обработки контента при снижении затрат.

Риски и ограничения автоматической модерации

ИИ-системы могут неправильно классифицировать товары, что приводит к удалению допустимого контента или пропуску запрещённого. Такие ошибки могут быть вызваны недостаточным качеством обучающих данных или ограничениями алгоритмов. Например, предвзятость в обучающих данных может привести к дискриминации определённых групп товаров или продавцов.

Также ИИ испытывает трудности с интерпретацией контекста, что особенно важно при модерации контента, содержащего сарказм, культурные особенности или политический подтекст. Это может привести к ошибочной блокировке или пропуску контента, нарушающего правила.

Кроме того, ошибки модерации могут повлечь за собой юридические последствия, особенно в юрисдикциях с жёстким регулированием ИИ. Например, в Европейском союзе действует Регламент об искусственном интеллекте, который классифицирует системы ИИ по уровню риска и устанавливает требования к их использованию. Несоблюдение этих требований может привести к штрафам и другим санкциям.

Для поддержания актуальности и точности ИИ-систем требуется регулярное дообучение на новых данных. Это связано с изменениями в правилах модерации, появлением новых типов контента и эволюцией пользовательского поведения. Без своевременного обновления модели могут устаревать, что увеличивает вероятность ошибок.

Не стоит забывать об юридической ответственности за ошибки ИИ, это актуальная и сложная проблема. В различных юрисдикциях подходы к определению ответственности за действия ИИ различаются, что требует особого внимания со стороны компании.

В России ИИ не признаётся самостоятельным субъектом права, а рассматривается как инструмент, используемый человеком или организацией. Согласно статье 1095 Гражданского кодекса РФ, ответственность за вред, причинённый использованием ИИ, несёт владелец или изготовитель системы. Это означает, что компании, применяющие ИИ для модерации товаров, обязаны обеспечить надёжность и безопасность таких систем, а также нести ответственность за их действия.

В Европейском союзе действует Регламент об искусственном интеллекте (AI Act), который классифицирует ИИ-системы по уровням риска и устанавливает соответствующие требования к их использованию. Системы, относящиеся к высокому риску, такие как ИИ для модерации контента, подлежат строгому контролю и обязательной сертификации. Компании, нарушающие положения AI Act, могут быть оштрафованы на сумму до €35 млн или 7% от годового оборота, в зависимости от того, какая сумма выше.

Можно выделить следующие стратегии минимизации рисков:

- ***Комбинированный подход к модерации***. Совмещение автоматической модерации с ручной позволяет компенсировать недостатки ИИ. Человеческие модераторы могут проверять спорные случаи и обеспечивать более точную оценку контента;
- ***Регулярное обновление и обучение моделей***. Постоянное обновление алгоритмов и обучение моделей на новых данных помогает уменьшить предвзятость и повысить точность модерации;
- ***Обучение персонала.*** Обучение сотрудников принципам этичного и правомерного использования ИИ;
- ***Прозрачность и возможность апелляции***. Предоставление пользователям информации о причинах блокировки контента и возможности обжалования решений способствует повышению доверия к системе модерации;
- ***Внедрение этических принципов.*** Разработка и соблюдение этических стандартов в процессе модерации помогают обеспечить справедливость и уважение к правам пользователей.
- ***Юридическая поддержка***. Привлечение юридических специалистов для анализа соответствия ИИ-систем действующему законодательству.

Прогноз экономической эффективности внедрения

Можно выделить следующие аспекты эффективности внедрения ИИ в модерацию товара:

1) ***Сокращение затрат на персонал.*** Внедрение ИИ позволяет автоматизировать рутинные задачи модерации, что снижает необходимость в большом количестве сотрудников. По данным Hanston, использование ИИ в модерации контента может снизить нагрузку на персонал на 50 %, позволяя перераспределить ресурсы на более сложные задачи, требующие человеческого вмешательства.

2) ***Повышение скорости и точности модерации.*** ИИ-системы способны обрабатывать большие объемы данных в реальном времени, обеспечивая быструю и точную модерацию контента. Это сокращает время реакции на нарушения и уменьшает количество ошибок, что в свою очередь снижает затраты, связанные с повторной проверкой и корректировкой контента.

3) ***Экономия на обучении и адаптации.*** Современные ИИ-модели обладают способностью к самообучению и адаптации к новым правилам и требованиям. Это снижает необходимость в частом переобучении моделей и связанных с этим затрат. Кроме того, использование методов посттренировочных улучшений позволяет повысить производительность моделей без значительных дополнительных инвестиций.

4) ***Автоматизация процессов и снижение операционных расходов.*** Автоматизация процессов модерации с помощью ИИ снижает операционные расходы, связанные с ручной проверкой контента. Это включает в себя сокращение затрат на оборудование, программное обеспечение и другие ресурсы, необходимые для поддержки ручной модерации. Кроме того, автоматизация позволяет обеспечить круглосуточную модерацию без дополнительных затрат на оплату сверхурочной работы или найм дополнительного персонала.

5) ***Повышение эффективности и конкурентоспособности.*** Внедрение ИИ в процессы модерации способствует повышению общей эффективности бизнеса, позволяя быстрее реагировать на изменения и обеспечивать высокий уровень качества обслуживания клиентов. Это, в свою очередь, повышает конкурентоспособность компании на рынке и способствует росту доходов.

6) ***Мгновенная модерация контента.*** ИИ-системы способны анализировать и модерировать контент в реальном времени, что позволяет немедленно публиковать или отклонять пользовательские материалы. Это особенно актуально для платформ с большим объемом пользовательского контента, где ручная модерация может занимать значительное время. Например, решения, подобные Utopia AI Moderator, обеспечивают автоматическую модерацию текстов и изображений, сокращая время обработки до миллисекунд.

7) ***Автоматизация процессов разработки и запуска продуктов.*** ИИ помогает ускорить этапы разработки продукта, включая анализ рынка, прогнозирование трендов и генерацию контента. Это позволяет быстрее переходить от идеи к запуску, снижая время вывода продукта на рынок. Компании, такие как Pimberly, используют ИИ для оптимизации процессов разработки, что способствует более быстрому запуску продуктов.

8) ***Ускорение вывода товаров на рынок.*** ИИ-системы способны обрабатывать и модерировать контент в реальном времени, что сокращает время вывода товаров на рынок. Harvard Business Review отмечает, что использование ИИ-инструментов может сократить время производства контента до 60%, позволяя компаниям быстрее реагировать на рыночные изменения.

Согласно исследованию Aisera, компании, внедряющие ИИ-решения, могут ожидать окупаемость инвестиций в течение 6–9 месяцев, в зависимости от масштаба внедрения и конкретных сценариев использования [8].

Эффективность и сроки окупаемости внедрения ИИ в модерацию контента могут значительно различаться в зависимости от размера и ресурсов компании. Исследование Capterra показывает, что малые и средние предприятия уже получают значительные выгоды от внедрения ИИ в ключевые бизнес-функции, включая управление проектами и обслуживание клиентов.

Таким образом, при правильной реализации и интеграции ИИ в процессы модерации контента, компании могут рассчитывать на окупаемость инвестиций в течение 6–12 месяцев, благодаря значительному снижению затрат, повышению эффективности и ускорению бизнес-процессов.

Перспективы использования ИИ-модерации

Интеграция искусственного интеллекта (ИИ) в процессы модерации контента приносит компаниям не только краткосрочные выгоды, но и значительные долгосрочные преимущества. Ключевые из них — повышение доверия пользователей, снижение количества жалоб и соблюдение нормативных требований.

ИИ-системы обеспечивают более объективную и последовательную модерацию контента, что способствует укреплению доверия пользователей к платформе. Исследования показывают, что пользователи социальных сетей могут доверять ИИ так же, как и человеческим модераторам, особенно если ИИ демонстрирует точность и беспристрастность в выявлении вредоносного контента.

Автоматизированная модерация позволяет оперативно выявлять и устранять нежелательный контент, что снижает количество пользовательских жалоб и улучшает общее качество взаимодействия с платформой. Это способствует созданию безопасной и комфортной среды для пользователей.

Современные ИИ-системы развиваются в сторону предиктивной модерации, позволяя анализировать тенденции и прошлые данные для предсказания потенциальных проблем с контентом. Это обеспечивает возможность принятия проактивных мер до появления нарушений, создавая более безопасную цифровую среду.

Несмотря на рост автоматизации, человеческий фактор остается критически важным. Будущие системы модерации будут сочетать ИИ и человеческий надзор, где алгоритмы выделяют потенциально проблемный контент, а модераторы принимают окончательные решения, обеспечивая баланс между эффективностью и этичностью.

Развитие ИИ позволяет обрабатывать контент на различных языках и в разных форматах (текст, изображения, видео), что особенно важно для глобальных платформ. Это обеспечивает единые стандарты модерации независимо от региона и типа контента.

Выводы

ИИ-модерация позволяет существенно сократить расходы на ручной труд. Автоматические системы способны обрабатывать большие объемы данных круглосуточно и без участия человека, что уменьшает потребность в крупных штатах модераторов.

ИИ обрабатывает и классифицирует товары за доли секунды, в отличие от ручной модерации, которая может занимать часы или дни. Это приводит к ускоренному выводу товаров на рынок, что особенно критично в быстро меняющихся нишах.

Внедрение автоматической модерации с применением ИИ — это не только технологический шаг вперёд, но и экономически целесообразное решение, обеспечивающее снижение затрат, ускорение процессов, рост качества обслуживания и быструю окупаемость инвестиций. Это подтверждает её стратегическое значение для устойчивого развития цифровых торговых платформ.

Для небольших объёмов контента затраты на полное внедрение ИИ могут быть значительными, а окупаемость — длительной (6–18 месяцев).

Для крупных объёмов контента автоматическая модерация позволяет существенно снизить операционные расходы и повысить производительность. Компании, внедрившие ИИ, отмечают повышение прибыли и улучшение производительности.

Таким образом, автоматическая модерация с применением ИИ даёт бизнесу мощные инструменты для повышения эффективности и снижения издержек, но требует взвешенного подхода с учётом рисков, необходимости контроля и постоянной оптимизации.

Литература

1. Content Moderation Services Market Report [Электронный ресурс] – Режим доступа: https://www.grandviewresearch.com/industry-analysis/content-moderation-services-market-report – Загл. с экрана.

2. Global Fraud Report 2024 [Электронный ресурс] – Режим доступа: https://www.cybersource.com/content/dam/documents/campaign/fraud-report/global-fraud-report-2024.pdf – Загл. с экрана.

3. Применение искусственного интеллекта в бизнесе [Электронный ресурс] – Режим доступа: https://www.sciencedirect.com/org/science/article/pii/S1546221821001314 – Загл. с экрана.

4. Unitary на AWS EKS: автоматизация модерации с ИИ [Электронный ресурс] – Режим доступа: https://aws.amazon.com/ru/solutions/case-studies/unitary-eks-case-study/ – Загл. с экрана.

5. Результаты внедрения ИИ в 123RF [Электронный ресурс] – Режим доступа: https://aws.amazon.com/ru/solutions/case-studies/123rf-2024/ – Загл. с экрана

6. OZON ускорил модерацию товаров [Электронный ресурс] – Режим доступа: https://www.cnews.ru/news/line/2023-03-20_ozon_uskoril_moderatsiyu_tovarov – Загл. с экрана.

7. Новости Wildberries, Ozon, Яндекс Маркет, Мегамаркет: обзор недели [Электронный ресурс] – Режим доступа: https://selsup.ru/blog/novosti-vajldberriz-ozon-yandeks-market-megamarket-obzor-za-nedelyu-s-30-sentyabrya-po-4-oktyabrya/ – Загл. с экрана.

8. ROI with AI [Электронный ресурс] – Режим доступа: https://aisera.com/blog/roi-with-ai/ – Загл. с экрана.

9. Online Content Regulation: An International Comparison [Электронный ресурс] – Режим доступа: https://studentbriefs.law.gwu.edu/ilpb/2021/12/08/online-content-regulation-an-international-comparison/ – Загл. с экрана.

10. Automated Content Moderation vs Manual Moderation [Электронный ресурс] – Режим доступа: https://www.cometchat.com/blog/automated-content-moderation-vs-manual-moderation – Загл. с экрана.

Ястребов А.Р., Боднар А.В. Экономическая эффективность автоматической модерации с помощью искусственного интеллекта в e-commerce. *В статье рассматриваются теоретические и практические аспекты модерации контента в электронной коммерции, включая её роль в обеспечении соответствия законодательным, этическим и коммерческим требованиям. Особое внимание уделяется преимуществам автоматической модерации на основе искусственного интеллекта (ИИ) по сравнению с ручными методами, а также её влиянию на экономическую эффективность, скорость обработки данных и качество пользовательского опыта.*

Ключевые слова: *модерация контента, электронная коммерция, искусственный интеллект, автоматизация, NLP, компьютерное зрение, экономическая эффективность, пользовательский опыт, законодательные требования, ручная модерация.*

Yastrebov A.R., Bodnar A.V. Economic efficiency of automatic moderation using artificial intelligence in e-commerce. *The article discusses the theoretical and practical aspects of content moderation in e-commerce*, including its role in ensuring compliance with legal, ethical and commercial requirements. *Special attention is paid to the advantages of automatic moderation based on artificial intelligence (AI) compared to manual methods, as well as its impact on economic efficiency, data processing speed and user experience quality.*

Key words: *content moderation, e-commerce, artificial intelligence, economic efficiency, manual moderation.*

Статья поступила в редакцию 25.05.2025
Рекомендована к публикации профессором Мальчевой Р. В.

УДК 004.415.2

Анализ подходов к разработке и оптимизации микросервисных систем

А. Е. Колесников, Т. В. Мартыненко, Е. А. Шуватова
Донецкий национальный технический университет
E-mail: lepy0ha@yandex.ru

Аннотация
В статье рассматриваются архитектурные принципы и подходы к разработке и оптимизации микросервисных систем. Приведён анализ промышленных реализаций, выделены общие закономерности. Особое внимание уделено межсервисному взаимодействию, отказоустойчивости, наблюдаемости и инструментам повышения производительности. Микросервисный подход открывает значительные возможности для масштабируемости и гибкости систем, но требует аккуратного и обоснованного применения. Его эффективность проявляется там, где архитектурная сложность оправдана бизнес-потребностями и поддерживается зрелыми процессами разработки и эксплуатации.

Введение

Современная цифровая трансформация предъявляет новые требования к архитектуре программных систем. Монолитные приложения постепенно уступают место микросервисным архитектурам, обеспечивающим большую гибкость, масштабируемость и скорость разработки. Однако переход к микросервисам сопряжён с техническими и организационными вызовами, что требует внимательного анализа подходов к их разработке и оптимизации [2].

Актуальность темы обусловлена усложнением бизнес-процессов, активным внедрением облачных технологий и накопленным опытом первых внедрений, выявившим как преимущества, так и скрытые риски микросервисов. Цель исследования — комплексный анализ современных методов построения и оптимизации микросервисных систем, включая технические и организационные аспекты, а также обоснование целесообразности перехода к такой архитектуре.

Методология включает анализ успешных кейсов, систематизацию практик и выявление закономерностей, полезных для архитектурных решений. Работа охватывает ключевые характеристики микросервисов, этапы оптимизации и рекомендации по архитектурному планированию с учётом эксплуатационных требований.

Практическая значимость исследования заключается в том, что представленные выводы и рекомендации основаны на реальном опыте, что делает их полезными для архитекторов и технических руководителей, принимающих стратегические решения в области ИТ-разработки.

Сравнительный анализ готовых микросервисных решений

В современных условиях разработчики имеют возможность выбирать между созданием собственной микросервисной платформы с нуля и использованием готовых решений от ведущих вендоров. Каждый из этих подходов имеет свои преимущества и ограничения, которые необходимо учитывать при проектировании архитектуры.

Коммерческие платформы (такие как Red Hat OpenShift, Pivotal Cloud Foundry или IBM Cloud Private) предлагают комплексные решения для развертывания и управления микросервисами [3]. Эти платформы предоставляют встроенные механизмы оркестрации, мониторинга, балансировки нагрузки и безопасности, что значительно ускоряет процесс внедрения. Основное преимущество — снижение операционных расходов за счет использования предварительно настроенных и протестированных компонентов. Однако такие решения часто обладают ограниченной гибкостью и могут накладывать дополнительные лицензионные расходы.

Open-source экосистемы (Kubernetes в сочетании с Istio, Prometheus и другими инструментами) предлагают более гибкий подход [6]. Этот вариант позволяет создать полностью кастомизируемую платформу, идеально соответствующую конкретным требованиям проекта. Сообщество разработчиков постоянно совершенствует эти инструменты, предлагая новые функции и исправления. Недостатком является необходимость значительных внутренних экспертиз для поддержки и интеграции различных компонентов.

Сервисы облачных провайдеров (AWS ECS/EKS, Azure Service Fabric, Google Kubernetes

Engine) занимают промежуточное положение. Они сочетают преимущества управляемых сервисов с возможностью использования открытых стандартов. Облачные решения особенно эффективны для компаний, уже использующих соответствующую инфраструктуру, так как обеспечивают глубокую интеграцию с другими сервисами провайдера. Однако они могут создавать vendor lock-in и увеличивать долгосрочные эксплуатационные расходы.

При выборе между этими вариантами следует учитывать несколько ключевых факторов:

— требования к времени вывода на рынок (готовые решения быстрее);

— наличие внутренней экспертизы (open-source требует больше знаний);

— бюджетные ограничения (облачные сервисы могут быть дороже в долгосрочной перспективе);

— потребности в кастомизации (готовые платформы менее гибки);

— требования к безопасности и соответствию стандартам.

Практика показывает, что крупные предприятия часто выбирают гибридный подход, комбинируя управляемые сервисы для базовой инфраструктуры с кастомными решениями для специфических бизнес-процессов. Стартапы и средний бизнес обычно отдают предпочтение облачным решениям из-за их простоты развертывания. Наиболее технически зрелые организации создают собственные платформы на базе open-source инструментов, что обеспечивает максимальную гибкость и контроль.

Архитектура микросервисных систем и подходы к разработке

Эксплуатация микросервисной архитектуры требует принципиально нового подхода к управлению жизненным циклом приложения. В отличие от монолитных систем, где основной акцент делался на стабильности версий, в микросервисной экосистеме ключевое значение приобретает управление постоянными изменениями. Это обусловлено самой природой распределенной архитектуры, где десятки или даже сотни сервисов могут независимо эволюционировать с разной скоростью.

Центральное место в эксплуатационной модели занимают практики непрерывной интеграции и доставки (CI/CD), которые становятся не просто желательными, а абсолютно необходимыми. Современные инструменты CI/CD эволюционировали для работы со сложными микросервисными топологиями, предлагая возможности канареечного развертывания (canary releases) и blue-green деплойментов. Эти механизмы позволяют постепенно вводить изменения в эксплуатацию, минимизируя потенциальное воздействие на конечных пользователей.

Особую сложность представляет обеспечение наблюдаемости (observability) распределенной системы. Традиционный мониторинг, ориентированный на отдельные сервисы, уступает место комплексным решениям, способным отслеживать транзакции через границы множества сервисов. Современные инструменты типа распределенного трейсинга (distributed tracing) и лог-агрегации стали неотъемлемой частью микросервисного стека, позволяя инженерам понимать поведение системы в целом, а не только ее отдельных компонентов [10].

Хаотическое тестирование (Chaos Engineering) превратилось из экзотической практики в стандартную процедуру поддержания надежности. В условиях, когда отказы отдельных компонентов становятся не исключением, а ожидаемым событием, важно заранее проверять, как система ведет себя в различных сценариях сбоев. Такие инструменты как Chaos Monkey или Gremlin позволяют моделировать различные сценарии отказов в контролируемых условиях [5].

Не менее важным аспектом является управление конфигурацией и секретами в распределенной среде. Централизованные хранилища конфигураций, такие как HashiCorp Consul или Spring Cloud Config, помогают поддерживать согласованность настроек между множеством сервисов, обеспечивая при этом необходимый уровень безопасности [4]. Особое внимание уделяется механизмам ротации секретов и управления доступом, так как компрометация одного сервиса в распределенной системе может иметь каскадные последствия.

Промышленные реализации микросервисных архитектур

Современные технологические гиганты демонстрируют разнообразные подходы к реализации микросервисных архитектур, адаптированные под специфику их бизнес-моделей и технических требований.

Netflix, являясь пионером микросервисного подхода, разработала высокоэффективную систему, основанную на принципах отказоустойчивости и глобального масштабирования. Их архитектура сочетает функциональную декомпозицию с продвинутыми механизмами кэширования и уникальными инструментами тестирования на устойчивость, такими как Chaos Monkey [8].

Uber реализовал инновационную георасределённую архитектуру, где ключевые сервисы дублируются в различных регионах мира. Это решение обеспечивает беспрецедентную отказоустойчивость и низкую

задержку для пользователей в любой точке земного шара, но требует сложных механизмов синхронизации данных.

Amazon применил микросервисный подход для создания событийно-ориентированной экосистемы, где каждый сервис соответствует конкретной бизнес-способности. Их архитектура демонстрирует выдающуюся масштабируемость, хотя и сталкивается с вызовами обеспечения транзакционной согласованности.

Spotify разработал уникальную организационно-техническую модель, где структура микросервисов зеркалирует организацию команд разработки (сквады и трибы). Этот подход обеспечивает высокую скорость внедрения изменений, но требует тщательного контроля за границами сервисов [3].

Alibaba представляет пример строго стандартизированной микросервисной экосистемы с жёстким разделением компонентов. Их решение, построенное вокруг Service Mesh (Istio), обеспечивает предсказуемость и управляемость системы в условиях экстремальных нагрузок [9].

Таблица 1 - Сравнительная таблица характеристик

Критерий	Netflix	Uber	Amazon	Spotify	Alibaba
Метод разработки	Функциональная декомпозиция, независимые сервисы на Java/Python	Геокластерная архитектура, gRPC для межсервисного взаимодействия	Декомпозиция по бизнес-способностям, event-driven (SQS/SNS)	Организационная декомпозиция (сквады/ трибы), REST API	Гибридный подход (REST + Events), строгое разделение данных
Оптимизация производительности	Кэширование (EVCache), асинхронная обработка (RxJava), Chaos Engineering (Chaos Monkey)	Шардирование данных по регионам, кэширование в Redis	Автомасштабирование (EC2 Auto Scaling), оптимизация запросов (DynamoDB)	Ленивая загрузка данных, edge-кэширование	Service Mesh (Istio), оптимизация запросов через CDN
Управление данными	Polyglot persistence (Cassandra, MySQL), репликация между регионами	Глобально распределённая БД (Schemaless), Event Sourcing	Разделение БД по сервисам, CQRS-подход	Гибридный подход: общие БД для связанных сервисов	Строгое «DB per service», распределённые транзакции через Seata
Безопасность	Centralized Auth (Zuul), mTLS для внутренних сервисов	Геоизолированные сервисы, ролевая модель доступа	IAM-политики, шифрование KMS	OAuth 2.0, изоляция окружений	Глубокая защита на уровне Service Mesh, аппаратное шифрование
Мониторинг и observability	Hystrix для отказоустойчивости, Atlas для метрик	Jaeger для трейсинга, Prometheus + Grafana	CloudWatch, X-Ray для трейсинга	Логирование через ELK, кастомные дашборды	SkyWalking (APM), интеграция с Prometheus
DevOps-практики	Spinnaker для CD, канареечные развёртывания	Инфраструктура как код (Terraform), feature flags	AWS CodePipeline, blue/green-деплойменты	GitOps-подход, постепенный rollout	Внутренние PaaS-решения, автоматическое тестирование сбоев

Эти реализации демонстрируют, что не существует универсального "идеального" подхода к микросервисной архитектуре - каждая компания разрабатывает решения, оптимальные для её конкретных требований и масштабов. Однако все они разделяют общие принципы декомпозиции, автономности сервисов и автоматизированного управления инфраструктурой.

Разработка обобщённого алгоритма оптимизации микросервисных систем

Оптимизация микросервисной архитектуры — это комплексный процесс, который требует последовательных и взаимосвязанных шагов. Обобщённый алгоритм включает четыре ключевых этапа, направленных на поэтапное улучшение состояния системы: от анализа текущего положения до финальной настройки инфраструктуры.

Этап 1. Диагностика текущего состояния

Первым шагом в алгоритме является диагностика системы, которая играет роль отправной точки для всех последующих действий. На этом этапе команда формирует целостное представление о структуре микросервисной архитектуры, фиксирует существующие зависимости и исследует поведение сервисов под нагрузкой.

Ключевая задача — выявить узкие места, перегруженные участки, неэффективные вызовы и скрытые зависимости, которые мешают масштабированию или вызывают сбои. Диагностика охватывает как технические параметры (производительность, ресурсоёмкость), так и логические аспекты взаимодействия компонентов. Использование инструментов мониторинга и трассировки позволяет получить данные, необходимые для принятия решений на следующих этапах.

Этап 2. Оптимизация взаимодействия между сервисами

На основе результатов диагностики проводится анализ того, как именно микросервисы обмениваются данными и координируют свои действия. Основная цель этого этапа — минимизировать задержки, снизить сетевую нагрузку и повысить устойчивость системы к сбоям.

Оптимизация может включать переход на более эффективные протоколы взаимодействия, сокращение количества синхронных вызовов, внедрение асинхронных паттернов и пересмотр логики передачи данных. Также рассматриваются подходы к уменьшению избыточной связанности между компонентами, что способствует повышению гибкости и управляемости архитектуры [7].

Этап 3. Оптимизация работы с данными

Следующий этап касается одного из наиболее ресурсоёмких элементов микросервисной архитектуры — работы с данными. Здесь важно не только ускорить выполнение запросов и снизить нагрузку на базу данных, но и пересмотреть подход к проектированию моделей хранения.

Оптимизация охватывает улучшение качества SQL-запросов, внедрение стратегий кэширования, а также выбор подходящей базы данных для каждой бизнес-задачи — будь то документоориентированная, колоночная или графовая модель. Дополнительно рассматриваются методы масштабирования хранилищ, такие как репликация и шардирование, с учётом компромиссов между доступностью и согласованностью данных.

Этап 4. Инфраструктурные улучшения

Заключительный этап алгоритма направлен на повышение надёжности, масштабируемости и безопасности всей системы, а также на ускорение процессов вывода новых версий в эксплуатацию.

Реализуется гибкое масштабирование, адаптированное к текущей нагрузке, повышается устойчивость к отказам за счёт применения шаблонов отказоустойчивости, таких как ограничители, повторные попытки и изоляция ресурсов. Особое внимание уделяется безопасности внутреннего взаимодействия и автоматизации процессов развертывания [8]. Это позволяет не только оперативно реагировать на сбои, но и снижать риски при выпуске обновлений.

Таким образом, предложенный алгоритм охватывает все критически важные аспекты функционирования микросервисных систем. Его применение обеспечивает не только устранение текущих проблем, но и формирует устойчивую основу для их дальнейшего развития и масштабирования.

Заключение

Разработка и оптимизация микросервисных систем представляет собой многоуровневую задачу, требующую комплексного подхода.

Как показал проведённый анализ, успешная реализация микросервисной архитектуры возможна только при сочетании технической зрелости, осознанного проектирования и готовности к организационным изменениям.

Предложенный обобщённый алгоритм оптимизации систематизирует ключевые этапы — от диагностики и улучшения межсервисного взаимодействия до оптимизации работы с данными и инфраструктурных решений. Он основан на обобщении практик ведущих технологических компаний и может служить опорой при принятии архитектурных решений в условиях реальных проектов.

Микросервисный подход открывает значительные возможности для масштабируемости и гибкости систем, но требует аккуратного и обоснованного применения. Его эффективность проявляется там, где архитектурная сложность оправдана бизнес-потребностями и поддерживается зрелыми процессами разработки и эксплуатации.

Литература

1. Шуватова, Е. А. Анализ методов определения эффективной архитектуры программных систем / Е. А. Шуватова, Т. В. Мартыненко // Информатика, управляющие системы, математическое и компьютерное моделирование (ИУСМКМ-2024) : XV Международная научно-техническая конференция в рамках X Международного Научного форума Донецкой Народной Республики, Донецк, 29–30 мая 2024 года. – Донецк: Донецкий национальный технический университет, 2024. – С. 125-130. – EDN ZXRUAI.

2. Ньюмен, С. Микросервисы. Паттерны разработки и рефакторинга [Текст] / С. Ньюмен; пер. с англ. А. Киселева. - СПб.: Питер, 2021. - 352 с.

3. Ричардсон, К. Микросервисы. Паттерны проектирования [Текст] / К. Ричардсон; пер. с англ. Д. Воронкова. - М.: Вильямс, 2020. - 592 с.

4. Вольф, К., Эйнсворт, Р. Микросервисы в действии [Текст] / К. Вольф, Р. Эйнсворт; пер. с англ. П. Петрова. - М.: ДМК Пресс, 2019. - 384 с.

5. Жемеров, Д. Kubernetes в действии [Текст] / Д. Жемеров. - СПб.: Питер, 2021. - 512 с.

6. Клеппман, М. Высоконагруженные приложения. Программирование, масштабирование, поддержка [Текст] / М. Клеппман; пер. с англ. И. Иванова. - СПб.: Питер, 2020. - 608 с.

7. Гребенюк, С. А. Архитектура микросервисов: теория и практика [Текст] / С. А. Гребенюк. – М.: ДМК Пресс, 2021. – 312 с.

8. Ляпунов, С. В. DevOps и микросервисы: внедрение и сопровождение [Текст] / С. В. Ляпунов. – СПб.: Питер, 2022. – 288 с.

9. Microsoft Docs. Руководство по микросервисной архитектуре [Электронный ресурс]. - Режим доступа: https://docs.microsoft.com/ru-ru/azure/architecture/guide/architecture-styles/microservices (дата обращения: 14.04.2025).

10. Habr. Практический опыт перехода на микросервисы [Электронный ресурс]. - Режим доступа: https://habr.com/ru/company/oleg-bunin/blog/522396/ (дата обращения: 14.04.2025).

Колесников А. Е., Мартыненко Т. В. , Шуватова Е. А. Анализ подходов к разработке и оптимизации микросервисных систем. *В статье рассматриваются архитектурные принципы и подходы к разработке и оптимизации микросервисных систем. Приведён анализ промышленных реализаций, выделены общие закономерности. Особое внимание уделено межсервисному взаимодействию, отказоустойчивости, наблюдаемости и инструментам повышения производительности. Микросервисный подход открывает значительные возможности для масштабируемости и гибкости систем, но требует аккуратного и обоснованного применения. Его эффективность проявляется там, где архитектурная сложность оправдана бизнес-потребностями и поддерживается зрелыми процессами разработки и эксплуатации.*

Ключевые слова: *микросервисная архитектура, распределенные системы, оптимизация производительности, межсервисное взаимодействие, DevOps, Kubernetes, service mesh, observability*

Kolesnikov A. E., Martynenko T. V., Shuvatova E. A. Analysis of approaches to the development and optimization of microservice systems. *The article explores architectural principles and approaches to the development and optimization of microservice systems. Industrial implementations are analyzed, revealing patterns. Special attention is paid to inter-service communication, fault tolerance, observability, and performance tools. The microservice approach opens up significant opportunities for system scalability and flexibility, but requires careful and well-reasoned application. Its effectiveness is evident where architectural complexity is justified by business needs and supported by mature development and operational processes.*

Keywords: *microservice architecture, distributed systems, performance optimization, inter-service communication, DevOps, Kubernetes, service mesh, observability*

Статья поступила в редакцию 25.05.2025
Рекомендована к публикации доцентом Приваловым М. В.

УДК 004.8 : 004.932

Использование технологий ИИ в проектировании элементов компьютерных систем

А. А. Койбаш, Р. В. Мальчева
E-mail: mr.koibash@yandex.ru, raisa.malcheva@yandex.ru

Аннотация
В статье выполнен обзор основных технологий искусственного интеллекта, которые применяются при проектировании компьютерных систем. Рассмотрены перспективы их развития. В рамках исследования разработан интеллектуальный ассистент для помощи инженерам-проектировщикам. Описаны выбор основных компонентов, таких как система управления базами данных, модель ИИ, аппаратные ускорители, а также реализация прототипа ассистента. Намечены направления для будущих исследований.

Введение

Технологии искусственного интеллекта (ИИ) находят свое применений в различных областях, таких как компьютерная графика [1, 2] и многие другие. В настоящее время системы автоматизированного проектирования (САПР) также претерпевают значительные изменения благодаря интеграции технологий искусственного интеллекта. Эта тенденция обусловлена растущей сложностью проектируемых систем и необходимостью повышения эффективности процесса проектирования. Анализ современного состояния исследований в данной области показывает активное развитие различных направлений применения искусственного интеллекта: от экспертных систем до глубокого обучения. Особый интерес представляет изучение возможностей использования этих технологий для автоматизации различных этапов проектирования и улучшения качества принимаемых решений.

В России планируется создать систему ИИ для проектирования цифровых микросхем. Проект рассчитан на период с 2024 по 2026 гг. и направлен на разработку «интеллектуальной» САПР цифровых сверхбольших интегральных схем (СБИС) [3]. Проект рассчитан на период с 2024 по 2026 гг. и направлен на разработку «интеллектуальной» САПР СБИС. В основе разработки будет лежать открытый маршрут OpenLane, который уже успешно применяется в реальных проектах. Ожидается, что новая система позволит улучшить характеристики СБИС примерно на 10–20 % и повысить эффективность процесса разработки. Проект также предполагает оптимизацию энергопотребления и производительности, разработку моделей машинного обучения, поддержку средств проектирования отечественных технологических процессов и создание центров компетенции для внедрения и технической поддержки.

Искусственный интеллект в проектировании

Сверточные нейронные сети (CNN) успешно применяются для анализа топологии печатных плат и проектирования интегральных схем [4]. Рекуррентные нейронные сети (RNN) и их современные варианты, такие как LSTM и GRU, показывают эффективность в моделировании временных последовательностей и прогнозировании поведения сложных систем [5]. Особое значение приобретают методы обучения с подкреплением (Reinforcement Learning), позволяющие системам ИИ обучаться оптимальным стратегиям проектирования через взаимодействие с виртуальной средой. Данный подход особенно перспективен для решения задач оптимизации архитектуры и параметров проектируемых систем.

Применение нейронных сетей в проектировании компьютерных систем представляет собой отдельное перспективное направление. Современные архитектуры нейронных сетей позволяют решать широкий спектр задач, от оптимизации топологии до предсказания характеристик проектируемых систем [6]. Глубокие нейронные сети демонстрируют особую эффективность в задачах:

- автоматической генерации схемотехнических решений;
- оптимизации размещения компонентов;
- прогнозирования тепловых и электрических характеристик;
- анализа надежности проектируемых систем.

Особый интерес представляют генеративно-состязательные сети (GAN), способные создавать новые варианты проектных решений на основе обучающих данных. Такие

сети могут генерировать множество альтернативных вариантов топологии или архитектуры системы, удовлетворяющих заданным ограничениям.

Появление больших языковых моделей (Large Language Models, LLM) открыло новые возможности в области автоматизированного проектирования. В отличие от традиционных систем автоматизации, LLM способны понимать и генерировать естественноязыковые описания технических решений, что существенно упрощает взаимодействие между проектировщиком и системой [7].

Современные языковые модели, такие как GPT, BERT и их производные, демонстрируют способность к пониманию контекста и генерации осмысленных рекомендаций по проектированию [8]. Особенно важным является их способность к обобщению накопленных знаний и применению их в новых контекстах. Это позволяет использовать LLM для:

- анализа технических требований и спецификаций;
- генерации вариантов проектных решений;
- документирования процесса проектирования;
- верификации соответствия проектных решений требованиям;
-оптимизации существующих проектов.

Важной особенностью применения LLM в проектировании является возможность их интеграции с существующими САПР через специализированные программные интерфейсы. Это позволяет создавать гибридные системы, сочетающие преимущества традиционных инструментов автоматизации с возможностями искусственного интеллекта.

Применение промпт-инжиниринга. Промпт-инжиниринг представляет собой новое направление в области применения языковых моделей, особенно актуальное для задач проектирования компьютерных систем. Данный подход заключается в разработке специализированных запросов (промптов), позволяющих получить от языковой модели максимально релевантные и точные результаты.

В контексте проектирования элементов компьютерных систем промпт-инжиниринг включает несколько ключевых аспектов. Прежде всего, это разработка структурированных шаблонов запросов, учитывающих специфику предметной области. Такие шаблоны должны содержать всю необходимую контекстную информацию и четкие инструкции для языковой модели.

Особое значение приобретает итеративный подход к формированию промптов, когда каждый последующий запрос учитывает результаты предыдущих взаимодействий с моделью. Это позволяет постепенно уточнять и оптимизировать проектные решения. Например, при проектировании интерфейса системы промпты могут последовательно уточнять требования к производительности, энергопотреблению и физическим параметрам компонентов.

Важным аспектом является также разработка системы валидации получаемых результатов. Это может включать как автоматическую проверку на соответствие техническим ограничениям, так и механизмы экспертной оценки генерируемых решений.

Интеграция технологий ИИ в процесс проектирования

Эффективное применение технологий искусственного интеллекта в проектировании требует их глубокой интеграции с существующими процессами и инструментами. Важным аспектом является разработка методологии, позволяющей органично сочетать традиционные подходы с возможностями ИИ.

Современные подходы к такой интеграции предполагают создание многоуровневых систем, где различные технологии ИИ применяются на разных этапах проектирования [9]. На концептуальном уровне могут использоваться языковые модели для анализа требований и генерации архитектурных решений. На уровне детального проектирования применяются специализированные нейронные сети для оптимизации конкретных параметров и характеристик. Особое внимание уделяется созданию интерфейсов взаимодействия между различными компонентами системы проектирования. Это включает разработку стандартизированных форматов данных, протоколов обмена информацией и механизмов синхронизации различных инструментов проектирования [10, 11].

Важным аспектом является также обеспечение прослеживаемости и объяснимости решений, принимаемых системами ИИ. Это необходимо как для валидации проектных решений, так и для соответствия нормативным требованиям в области проектирования критически важных систем.

Перспективы развития технологий ИИ в проектировании КС

Дальнейшее развитие технологий искусственного интеллекта в проектировании компьютерных систем связано с несколькими ключевыми направлениями. Прежде всего, это совершенствование существующих моделей и алгоритмов, повышение их точности и эффективности. Особое внимание уделяется развитию методов, позволяющих работать с

ограниченными наборами данных, что особенно актуально для специализированных областей проектирования [9].

Одним из наиболее перспективных направлений является развитие интеллектуальных ассистентов, способных предоставлять экспертную поддержку в процессе проектирования. Такие системы, основанные на современных языковых моделях, могут не только давать рекомендации по техническим решениям, но и обучаться на основе накопленного опыта взаимодействия с инженерами-проектировщиками, постоянно повышая качество своих рекомендаций.

Перспективным направлением является также разработка гибридных систем, сочетающих различные подходы к искусственному интеллекту. Интеграция интеллектуальных ассистентов с традиционными САПР и специализированными нейронными сетями может обеспечить более полное покрытие задач проектирования и повысить надежность принимаемых решений.

Развитие технологий федеративного обучения без необходимости обмена данных между клиентскими устройствами и сервером может способствовать созданию распределенных систем проектирования, где различные организации могут совместно использовать накопленный опыт без прямого обмена конфиденциальными данными. Это особенно важно для развития интеллектуальных ассистентов, которые могут обогащать свою базу знаний, сохраняя при этом конфиденциальность проектных решений отдельных организаций.

Разработка интеллектуального ассистента

В рамках исследования разработан интеллектуальный ассистент для помощи инженерам. При создании прототипа определены и реализованы следующие ключевые требования: интуитивно понятное пользовательское взаимодействие, максимальное использование потенциала современных больших языковых моделей, а также широкий потенциал применения – возможность использования ассистента как в связке с существующими системами автоматизированного проектирования, так и в качестве самостоятельного инструмента консультационной поддержки

Архитектура предлагаемого решения

В рамках исследования разработан интеллектуальный ассистент системы автоматизированного проектирования, реализованный в виде программного комплекса с использованием API мессенджера Telegram.

Система построена на принципах объектно-ориентированного программирования. Основой системы является модульный принцип организации программного кода. Использована классическая «слойная» архитектура, в основе которой лежит принцип различной ответственности для каждого из участков программы. Схема такой архитектуры изображена на рисунке 1.

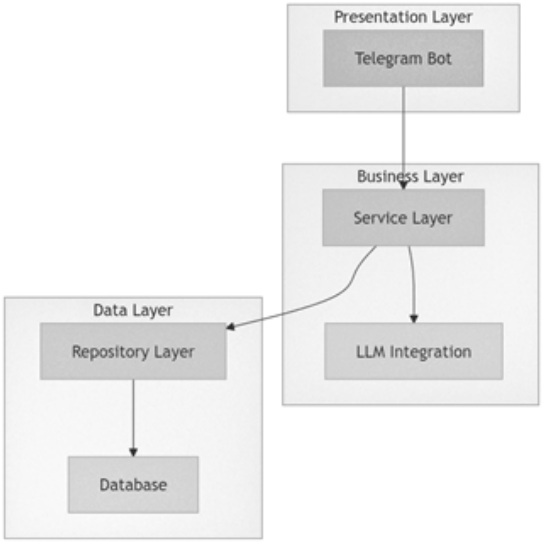


Рисунок 1 – Архитектура приложения

Центральным элементом выступает модуль Telegram Bot, обеспечивающий взаимодействие с пользователем. Он реализованный через API Telegram и является слоем презентации (точкой входа). Данный модуль обеспечивает прием пользовательских запросов и отправку ответов системы.

Второй ключевой компонент – модуль обработки естественного языка, построенный на базе современной языковой модели. Этот модуль отвечает за анализ пользовательских запросов и генерацию релевантных ответов в контексте автоматизированного проектирования.

Система хранения данных, основанная на реляционной базе данных, обеспечивает долговременное сохранение и восстановление пользовательских сессий (рисунок 2).

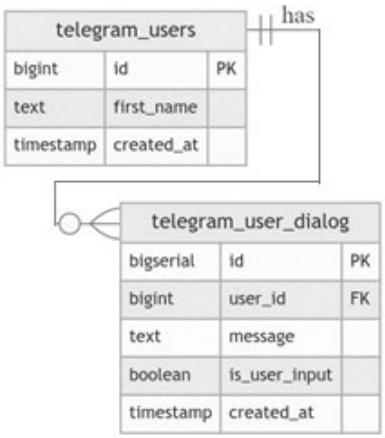


Рисунок 2 – Таблицы базы данных

В рамках прототипа необходимо и достаточно наличие двух таблиц: пользователей (telegram_users) и пользовательских диалогов (telegram_user_dialog) с целью обеспечения возможности учёта контекста предыдущих сообщений интеллектуальным ассистентом.

В рамках программного комплекса используется паттерн Service Locator, который предоставляет единую точку доступа ко всем зависимостям. Благодаря этому, любой класс имеет возможность получить доступ к централизованному реестру всех сервисов приложения. Данный подход обеспечивает гибкость конфигурации и масштабируемость решения.

Также для приложения разработана подсистема логирования и мониторинга, позволяющая отслеживать работу всех компонентов системы и обеспечивающая возможность анализа эффективности её работы.

Выбор технологий

При разработке системы особое внимание было уделено выбору технологического стека. В качестве основного языка программирования выбран Python, что обусловлено его широкими возможностями в области обработки естественного языка и машинного обучения, а также развитой экосистемой библиотек для разработки программных систем.

Для работы с языковой моделью применяется библиотека llama.cpp, обеспечивающая эффективное использование вычислительных ресурсов и гибкую настройку параметров модели. Данный выбор обоснован возможностью оптимизации работы модели под различные устройства и поддержкой широкого спектра форматов взаимодействия.

Взаимодействие с пользователем организовано через программный интерфейс Telegram посредством библиотеки telebot, что обеспечивает надежную и масштабируемую платформу для обмена сообщениями. Данное решение позволяет абстрагироваться от низкоуровневых особенностей сетевого взаимодействия и сосредоточиться на логике работы системы.

Разработка системы промптов

В рамках исследования разработана специализированная система промптов, учитывающая специфику задач автоматизированного проектирования. Базовый системный промпт определяет роль виртуального ассистента и задает основные параметры его поведения в контексте решения проектных задач.

Особое внимание уделено механизму поддержания контекста диалога, что достигается путем сохранения и анализа истории взаимодействия с пользователем. Система промптов включает также временные метки и контекстную информацию, что позволяет генерировать более релевантные и актуальные ответы.

Посредством использования системного промпта языковой модели заданы основные директивы: консультирование по техническим вопросам, анализ проектных решений, предоставление рекомендаций, помощь в поиске ошибок. К системному промпту в ходе каждого взаимодействия добавляется текущее время, чтобы ассистент мог учитывать этот параметр в ходе диалога.

Реализация прототипа

Для наглядной демонстрации работы интеллектуального ассистента создан прототип программного продукта на локальном компьютере. В процессе реализации были заложены удобные архитектурные паттерны, дающие возможность дальнейшего расширения. Система протестирована на условиях, максимально приближенных к реальным.

Описание программной реализации

Программная реализация системы выполнена в соответствии с принципами объектно-ориентированного программирования и современными практиками разработки программного обеспечения. Диаграмма классов приложения изображена на рисунке 3. Центральным элементом реализации является класс TelegramBot, включающий в себя использование функционала взаимодействия с Telegram API. Данный класс обеспечивает основную функциональность системы и координирует работу остальных компонентов.

Взаимодействие с языковой моделью реализовано через специализированный класс Llm, инкапсулирующий логику работы с моделью и предоставляющий высокоуровневый интерфейс для генерации ответов. Конфигурация модели осуществляется через внешний конфигурационный файл, что обеспечивает гибкость настройки параметров без необходимости модификации программного кода. В системе реализован механизм логирования всех взаимодействий, что позволяет отслеживать работу системы и анализировать качество генерируемых ответов. Данный механизм является критически важным для дальнейшего совершенствования системы и адаптации её работы под конкретные задачи проектирования.

В качестве системы управления базами данных (СУБД) выбрана PostgreSQL, что обусловлено рядом существенных технических и эксплуатационных преимуществ данной СУБД. PostgreSQL представляет собой объектно-реляционную систему управления базами данных. Ключевым фактором выбора PostgreSQL является её архитектурная надёжность,

обеспечивающая высокий уровень целостности данных благодаря встроенным механизмам репликации и продвинутым системам резервного копирования. Существенное значение имеет также и производительность системы, достигаемая за счёт эффективной обработки конкурентных запросов. В контексте разрабатываемой системы особую значимость имеет возможность интеграции PostgreSQL с различными языками программирования и платформами, что обеспечивает необходимую гибкость при реализации программных компонентов. СУБД развёрнута на удалённом сервере с операционной системой Ubuntu 22.04. Подключение доступно по внешнему ip-адресу для любого вычислительного устройства. Схема базы данных реализована согласно выбранной структуре в виде двух таблиц (рисунок 4).

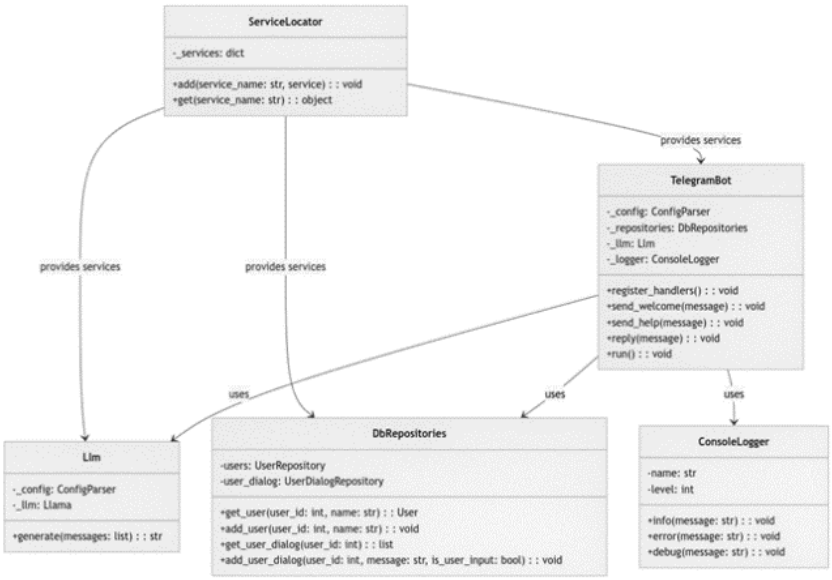


Рисунок 3 – Диаграмма классов (основная архитектура приложения)

```
aicad=# \dt
              List of relations
 Schema |         Name         | Type  |   Owner
--------+----------------------+-------+-----------
 public | telegram_user_dialog | table | aicad_dba
 public | telegram_users       | table | aicad_dba
(2 rows)

aicad=# \d telegram_users;
                                  Table "public.telegram_users"
   Column   |            Type             | Collation | Nullable |      Default
------------+-----------------------------+-----------+----------+--------------------
 id         | bigint                      |           | not null |
 first_name | text                        |           | not null |
 created_at | timestamp without time zone |           | not null | CURRENT_TIMESTAMP
Indexes:
    "telegram_users_pkey" PRIMARY KEY, btree (id)
Referenced by:
    TABLE "telegram_user_dialog" CONSTRAINT "telegram_user_dialog_user_id_fkey" FOREIGN KEY (user_id) REFERENCES telegr
am_users(id) ON UPDATE CASCADE ON DELETE CASCADE

aicad=# \d telegram_user_dialog;
                                        Table "public.telegram_user_dialog"
    Column     |            Type             | Collation | Nullable |                    Default
---------------+-----------------------------+-----------+----------+-----------------------------------------------
 id            | bigint                      |           | not null | nextval('telegram_user_dialog_id_seq'::regclass)
 user_id       | bigint                      |           |          |
 message       | text                        |           | not null |
 is_user_input | boolean                     |           | not null |
 created_at    | timestamp without time zone |           | not null | CURRENT_TIMESTAMP
Indexes:
    "telegram_user_dialog_pkey" PRIMARY KEY, btree (id)
Foreign-key constraints:
    "telegram_user_dialog_user_id_fkey" FOREIGN KEY (user_id) REFERENCES telegram_users(id) ON UPDATE CASCADE ON DELETE
 CASCADE
```

Рисунок 4 – Диаграмма классов (основная архитектура приложения)

Выбор модели ИИ

В качестве большой языковой модели выбран Qwen 2.5 с 7 миллиардами параметров. Данная нейронная сеть разработана исследовательской командой компании Alibaba и представляет собой модель с открытым исходным кодом. Такой выбор был сделан по нескольким причинам. Прежде всего, модель демонстрирует впечатляющие способности в работе с русским языком, что критически важно для русскоязычных пользователей САПР-систем. В отличие от многих других конкурентов с открытым исходным кодом, показывающих посредственные результаты при работе с русским языком, Qwen 2.5 обеспечивает естественное и грамотное общение, включая корректное использование технической терминологии в области проектирования и инженерных расчётов.

Другим существенным преимуществом является компактный размер модели – 7 миллиардов параметров представляют собой разумный компромисс между

производительностью и требованиями к вычислительным ресурсам. Это позволяет развернуть модель на относительно доступном оборудовании, сохраняя при этом высокое качество ответов по техническим вопросам. Кроме того, Qwen 2.5 7B обладает хорошими показателями в работе с контекстом и демонстрирует способность удерживать в памяти детали предыдущих взаимодействий, что особенно важно при обсуждении сложных технических задач и многоэтапных процессов проектирования.

Для разработки прототипа использована квантизованная версия данной модели. Её размер составляет 3.3 гигабайта, что позволяет эффективно использовать аппаратную акселерацию вычислений.

Аппаратное ускорение

В целях оптимизации производительности системы и минимизации времени отклика при обработке запросов задействован графический ускоритель (GPU) в качестве специализированного аппаратного акселератора для выполнения матричных вычислений нейронной сети. Данное архитектурное решение обосновано тем, что современные графические процессоры обладают значительным преимуществом перед центральными процессорами (CPU) в задачах параллельной обработки данных, характерных для работы нейронных сетей.

В рамках реализации системы библиотека llama.cpp была скомпилирована с поддержкой технологии CUDA (Compute Unified Device Architecture) от компании NVIDIA. CUDA представляет собой программно-аппаратную архитектуру параллельных вычислений, которая позволяет существенно увеличить вычислительную производительность за счет использования графических процессоров NVIDIA. Данная технология обеспечивает прямой доступ к набору инструкций графического процессора и управление его памятью, что позволяет добиться эффективной параллельности вычислительных операций.

Интеграция компонентов

Интеграция компонентов системы осуществляется через механизм инверсии зависимостей, что обеспечивает слабую связанность модулей и возможность их независимого тестирования и модификации. Взаимодействие между компонентами организовано через четко определенные интерфейсы, что минимизирует влияние изменений в одном модуле на работу других частей системы.

Особое внимание уделено организации хранения данных. Реализованная система репозиториев обеспечивает унифицированный доступ к данным пользователей и истории взаимодействий. Такой подход позволяет абстрагироваться от конкретной реализации хранилища данных и обеспечивает возможность масштабирования системы.

Тестирование системы

В ходе тестовой эксплуатации система продемонстрировала способность эффективно решать различные задачи в области автоматизированного проектирования.

Типичный сценарий использования включает следующие этапы взаимодействия:

- инициацию диалога;
- уточнение требований;
- генерацию проектного решения;
- обсуждение альтернатив и детализацию выбранного варианта.

При работе с запросами система демонстрирует способность к контекстному анализу и генерации структурированных ответов, учитывающих специфику предметной области. Ответы системы включают не только конкретные технические решения, но и обоснование принятых решений, что особенно важно в контексте проектирования технических систем.

Тестирование разработанной системы проводилось в несколько этапов. На первом этапе осуществлялось модульное тестирование отдельных компонентов, что позволило верифицировать корректность их работы в изоляции. Далее проводилось интеграционное тестирование, направленное на проверку корректности взаимодействия между компонентами системы.

Особое внимание было уделено тестированию качества генерируемых ответов. Для этого был разработаны тестовые сценарии, охватывающих различные аспекты проектирования компьютерных систем. Результаты тестирования показали высокую релевантность генерируемых ответов и способность системы адаптироваться к различным типам запросов.

Выводы

В данной статье рассмотрены основные технологии ИИ, которые используют исследователи и практики при проектировании элементов компьютерных систем.

В качестве примера описана разработка ассистента инженера-проектировщика, включающая выбор основных компонентов, таких как система управления базами данных, модель ИИ, аппаратные ускорители, а также реализацию прототипа ассистента.

Направлением дальнейших исследований является разработка САПР для демонстрации применения технологий ИИ в учебном процессе при подготовке инженеров-проектировщиков компьютерных систем.

Литература

1. Мулявин, Д. Е. Расширение возможностей систем генерации изображений путем использования нейронных сетей / Д. Е. Мулявин, Р. В. Мальчева, А. А. Койбаш // Информатика и кибернетика. – Донецк: ДонНТУ, 2024. - № 3(37). – С. 13-18.

2. Волгушева, А. И. Применение сверточных нейронных сетей для распознавания объектов на изображении. / А. И. Волгушева, Р. В. Мальчева // Информатика и кибернетика. – Донецк: ДонНТУ, 2024. - № 3(37). – С. 32-38.

3. В России учат ИИ проектировать цифровые микросхемы [Электронный ресурс] – UML: https://www.cnews.ru/news/top/2024-07-15_uchenye_iz_rossii_nauchat_iskusstvennyj

4. Peter, S. M. Lithographic Hotspot Detection Using CNN / S. M. Peter, K. L. Nisha, M. S. Arun Sankar // IEEE Recent Advances in Intelligent Computational Systems (RAICS), 2024. - С. 1-5.

5. Faraji, A. A Hybrid Approach Based on Recurrent Neural Network for Macromodeling of Nonlinear Electronic Circuits / A. Faraji, S. A. Sadrossadat, M. Yazdian-Dehkordi, M. Nabavi, Y. Savaria // IEEE Access, 2022. - №10. - С. 127996-128006.

6. Савостин, Д. А. Будущие перспективы автоматизированного проектирования (САПР) с точки зрения искусственного интеллекта и 3D-печати / Д. А. Савостин, Е. О. Кириченко, А. О. Шаранов // Известия ТулГУ. Технические науки, 2023.- №2.

7. Балашев, А. В. Сравнительный анализ эффективности различных моделей машинного обучения в задачах генерации контента / А. В. Балашев, М. В. Ступина // Молодой исследователь Дона, 2024. - №3.

8. Gupta, P. Generative AI: A systematic review using topic modelling techniques / P. Gupta, B. Ding, C. Guan, D. Ding // Data and Information Management, 2024. - №2. - 100066.

9. Luukko M. UTILIZATION OF ARTIFICIAL INTELLIGENCE IN ELECTRONICS DESIGN. – 2024.

10. Li, Y. Large Language Models for Manufacturing / Y. Li, H. Zhao, H. Jiang, Y. Pan, Z. Liu and others // arXiv:2410.21418 [cs.AI]. 2024. Available at: https://arxiv.org/abs/2410.21418.

11. Chen, G. Intelligent OPC Engineer Assistant for Semiconductor Manufacturing / G. Chen, H. Y. Yang, H. Ren, B. Yu // arXiv:2408.12775v1 [cs.AI]. 2024. Available at: https://arxiv.org/abs/2408.12775.

Койбаш А. А., Мальчева Р. В. Использование технологий ИИ в проектировании элементов компьютерных систем. *В статье выполнен обзор основных технологий искусственного интеллекта, которые применяются при проектировании компьютерных систем. Рассмотрены перспективы их развития. В рамках исследования разработан интеллектуальный ассистент для помощи инженерам-проектировщикам. Описаны выбор основных компонентов, таких как система управления базами данных, модель ИИ, аппаратные ускорители, а также реализация прототипа ассистента. Намечены направления для будущих исследований.*

Ключевые слова: *САПР, искусственный интеллект, ассистент, прототип, тестирование*

Koibash A. A., Malcheva R. V. The use of AI technologies in the design of computer system components*. The article provides an overview of the main AI technologies used in the design of computer systems. The prospects for their development are discussed. As part of the research, an intelligent assistant has been developed to assist design engineers. The selection of key components, such as a database management system, an AI model, and hardware accelerators, as well as the implementation of the assistant prototype, are described. Directions for future research are outlined.*

Keywords*: CAD, AI, assistant, prototype, testing*

Статья поступила в редакцию 08.06.2025
Рекомендована к публикации профессором Зори С. А.

Компьютерные науки

УДК 004.942

Анализ публикационной активности студентов и построение структурной модели движения бакалавров и магистров по группам

А.Э. Ульяненко
Донецкий национальный технический университет
e-mail: wtfskilledoy1@yandex.ru

Аннотация
Рассмотрен вклад студенческих публикаций, как один из показателей эффективности научной деятельности вузов. Выявлена корреляция между количеством студенческих публикаций и роста позиций университетов в рейтинге, обнаружена большая дифференциация, как в разрезе образовательных институтов, так и в разрезе ученых степеней. Была построена структурная модель переходов студентов между кластерами, в рамках их публикационной активности, необходимая для дальнейшего создания математической модели прогнозирования.

Введение

В условиях глобализации высшего образования наукометрические показатели становятся ключевым критерием оценки эффективности научной деятельности университетов. В данном исследовании рассмотрен вклад студенческих публикаций в наукометрию вузов на основе анализа данных международных рейтингов, индексов цитирования и программ государственного финансирования. Стратегическое вовлечение студентов в публикационную деятельность способствует росту позиций университетов в международных рейтингах QS, THE и увеличению индекса Хирша научных коллективов на 18-25 %, а также повышает шансы на получение грантовой поддержки.

Анализ публикационной активности

Современные университеты функционируют в условиях жесткой конкуренции за научный авторитет, финансирование и кадровые ресурсы. В этом контексте публикационная активность становится критически важным показателем, влияющим на:

– позиционирование вуза в международных рейтингах;

– объемы государственного и корпоративного финансирования;

– формирование научных школ.

Кроме того, министерством науки и высшего образования Российской Федерации проводится ежегодный мониторинг, целью которого является – контроль эффективности образовательной деятельности вузов. Порядок проведения данного мониторинга предусматривает 8 пороговых значений показателей эффективности, в том числе это касается и научной деятельности. В случае, если более 4 показателей окажутся ниже данных значений, высшее учебное заведение будет лишено права на осуществление своей образовательной деятельности или будет вынуждено реформироваться.

Традиционно анализ публикационной активности фокусируется на показателях преподавателей и аспирантов, тогда как вклад студентов остается недостаточно изученным. С каждым годом все больше внимания уделяется вовлечению студентов в научную деятельность и приходу талантливых молодых людей в сферу исследований. Ученые активно исследуют этот феномен, выдвигают теории и ведут дискуссии, а практика подтверждает его важность для образовательных и научных результатов [1]. Данное исследование направлено на заполнение этого пробела.

Для отображения создания эффективной модели выращивания научных кадров, сочетающей глобальный опыт с национальной спецификой, рассмотрим Китай, который является мировым лидером в области научных исследований. Он представляет особый интерес с точки зрения организации студенческой научной работы. Эффективность этой системы объясняется двумя основными факторами: адаптацией лучших международных практик (европейских, американских и японских) и сохранением национальных образовательных традиций [2].

Значительные изменения в данной сфере стали возможны благодаря ряду ключевых реформ:

- 1978 год – проведение «Реформы Дэн Сяопина», заложившей основы модернизации образования;

– 2005 год – переломный момент, связанный с заявлением премьера Госсовета КНР Вэнь Цзябао о критической важности подготовки творчески мыслящих молодых ученых для технологического и социально-экономического прогресса;
– 2009 год – запуск «Плана Джомолунгма» (программы индивидуального сопровождения одаренных студентов с первых курсов), направленного на поддержку талантливой молодежи в фундаментальных науках.

Данные меры позволили Китаю создать эффективную модель «выращивания» научных кадров, сочетающую глобальный опыт с национальной спецификой [3]. Вовлеченность китайских студентов в науку и подходы, способствующие этому, представлены на табл. 1.

Таблица 1 - Подходы, влияющие на вовлеченность студентов Китая в науку

Университет	Наименование программы/плана подготовки талантливых студентов в рамках научной деятельности	Особенности программы/плана	Специфика организации научной и инновационной работы
Университет Фудань	«План подготовки талантливых студентов Цинхуа Сюетан	Организация 6 специальных групп лучших студентов по математике, информатике, механике, науке о жизни, физике, химии под руководством известных профессоров.	Для повышения мотивации студентов к НИР создан курс «Дорога к научным исследованиям».
Нанкинский университет	«План Вандао»	В центре внимания - развитие электронных образовательных ресурсов, включая специализированные онлайн-платформы и курсы с открытым доступом.	В образовательный процесс внедрены инновационные методики: интерактивные формы обучения, обмен знаниями, а также междисциплинарные семинары с участием студентов всех уровней подготовки - от бакалавров до аспирантов. При этом бакалавры приступают к самостоятельной научно-исследовательской работе уже на втором семестре второго курса.
Университет Цинхуа	«Экспериментальная программа подготовки талантливых студентов по фундаментальным наукам» (Институт Куан Ямин)	Создание индивидуальных учебных планов для студентов в соответствие с их научными интересами и учебными достижениями.	Студенты имеют возможность формировать индивидуальную образовательную траекторию, включая в свой учебный план предметы из других курсов и направлений. Обязательные дисциплины могут быть заменены элективными курсами. Кроме того, бакалавры получают доступ к изучению магистерских и аспирантских программ, а также дисциплин других специальностей.

Далее был проведен анализ публикационной активности на примере некоторых российских вузов. показан рост ключевых показателей – индекса Хирша (ΔH) и количества публикаций в Scopus и WoS (ΔSW) за период с начала 2017 по конец 2018 года. Результаты анализа свидетельствуют о том, что некоторые институты (ИФ, ИПИСН, ИПП, ИФКСИТ, ИЛГИСН) в первую очередь стремились увеличить число публикаций в

международных базах данных, тогда как ФТИ сосредоточился на росте обоих показателей. В остальных институтах более заметен прирост публикаций в РИНЦ, однако это не обязательно означает, что они сделали их приоритетом, поскольку изначально имели значительно более высокие показатели по сравнению с другими [4].

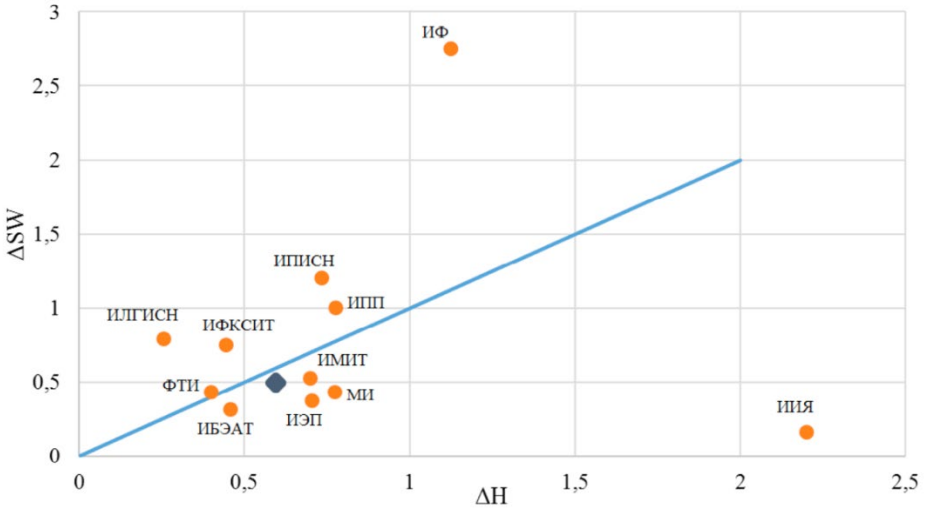


Рисунок 1 – Прирост основных показателей индекса Хирша (ΔH) и числа публикаций в Scopus и WoS (ΔSW) за период с 2016 по 2018 гг. включительно

В результате анализа публикационной активности обнаружена большая дифференциация, как в разрезе образовательных институтов, так и в разрезе ученых степеней.

Для оценки влияния студенческих публикаций использованы:

– Библиометрический анализ данных Scopus и WoS за 2015-2023 гг. по 20 ведущим российским университетам;
– Корреляционное исследование связи между количеством студенческих публикаций, динамикой в рейтингах QS/THE, объемом грантового финансирования;
– Кейс-стади программ поддержки студенческих публикаций в МФТИ, НИУ ВШЭ, ТГУ.

Как показал анализ, существует устойчивая зависимость ($r=0.72$, $p<0.05$) между активностью студентов в публикационной деятельности и ростом университетов в рейтинге QS (критерий «Citation per Faculty»). В частности, вузы с программами поддержки студенческих исследований (например, «Студент-исследователь» в НИУ ВШЭ) показывают прирост публикаций на 12–18 % в год и увеличение цитирования на 1,3–1,8 пункта ежегодно.

Публикации, написанные в соавторстве преподавателей и студентов, цитируются на 23% чаще в первые два года по сравнению с работами без участия студентов. Кроме того, такие публикации способствуют росту h-index научных коллективов на 0,5–0,7 пункта ежегодно.

Согласно информации об организации НИРС, представленной на сайте Тюменского индустриального университета (ТИУ), можно выделить следующие особенности: руководство НИРС осуществляют профессоры и преподаватели, сотрудники научных учреждений института и аспиранты; в университете ответственные за научную работу, курирующие организацию НИРС на кафедре или в научном учреждении института. Для раскрытия научно-творческого потенциала студентов в ТИУ существует Студенческое научное общество, которое является общественным, добровольным, самостоятельным, постоянно действующим научным объединением студентов и аспирантов, участвующих в научно-исследовательской, проектно-конструкторской и инновационной деятельности [5].

Помимо публикационной активности, была также оценена и грантовая эффективность, поскольку она также отражает студенческую вовлеченность в научную деятельность. Критерии оценки грантовой активности исследователей можно классифицировать по четырем ключевым категориям (в рамках студенческих грантов):

– Персональные характеристики (Возрастные показатели, наличие ученой степени и звания, занимаемая профессиональная позиция;
– Научно-исследовательские показатели (уровень исследовательской и проектной квалификации, степень вовлеченности в научное сообщество, показатели

академической мобильности, динамика публикационной активности;

– Грантовая компетентность (информированность о грантовых возможностях, готовность к конкурсному участию, опыт индивидуальной и коллективной грантовой деятельности, эффективность участия в грантовых конкурсах, практическая результативность научных исследований);
– Мотивационные факторы (презентация научных достижений, перспективы межвузовского сотрудничества, профессиональный престиж, финансовая заинтересованность.

Данная классификация позволяет комплексно оценивать грантовую активность с учетом многофакторной природы исследовательской деятельности [6-7].

Вузы с развитыми системами публикаций (топ 20 из Scopus/WoS) на 27% чаще выигрывают гранты РФФИ, получают на 15-20% больше финансирования по программе «Приоритет-2030».

Структурная модель переходов студентов

На основе проведенного выше анализа разрабатывается математическая модель движения бакалавров и магистров по группам, представленная в виде однородного марковского процесса с дискретным временем.

Марковские процессы принятия решений (МППР) служат ключевым инструментом оптимизации решений в условиях неопределенности. Они широко применяются в автономных системах, анализе больших данных и искусственном интеллекте. Современные методы решения задач МППР включают обучение с подкреплением, обеспечивающее адаптацию моделей к изменяющимся условиям и извлечение знаний из накопленного опыта.

Развитие МППР и связанных с ними подходов значительно ускорилось в последние десятилетия. Классическая теория, разработанная Беллманом (1957), заложила основы динамического программирования и оптимизации. Сегодня исследования в этой области активно интегрируют машинное обучение и обучение с подкреплением для решения сложных прикладных задач [8].

Структурная модель переходов студентов между кластерами представлена на рисунке 2. На основе данной математической модели в дальнейшем планируется разработка алгоритма непараметрической оценки плотности распределения переходов для марковской цепи динамики численности таковых, одновременно учитывая областные и функциональные ограничения.

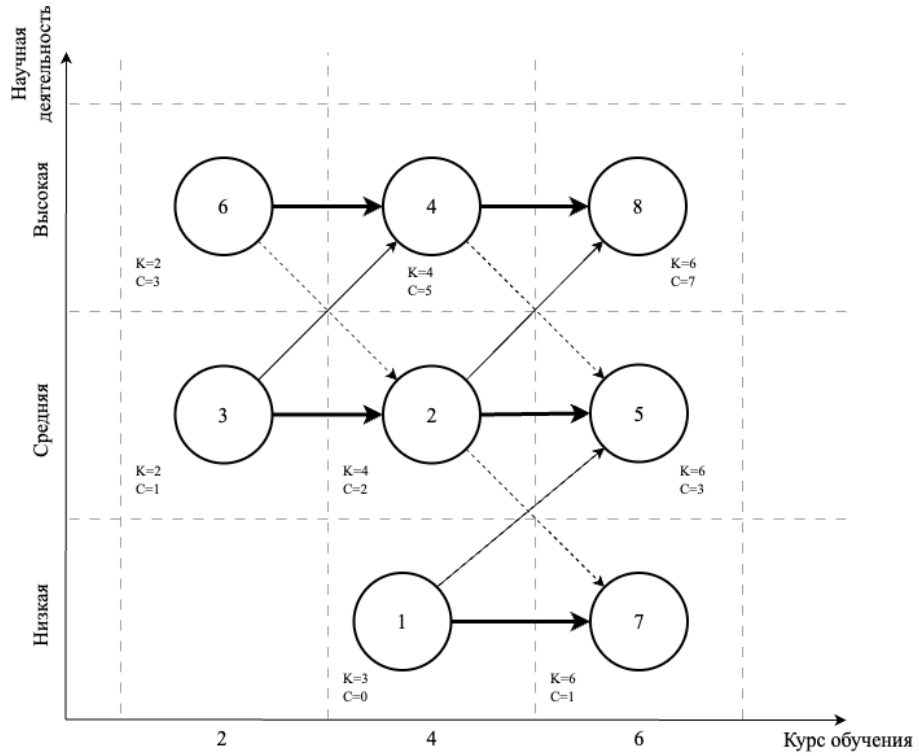


Рисунок 2 – Структурная модель переходов студентов между кластерами (от 1 до 8 – это состояния студентов/кластеры, К – курс обучения, а С – количество публикаций)

Представленная структурная модель переходов предполагает исходное разделение на 8 кластеров, соответствующих различным категориям студентов в зависимости от их научной активности и курса обучения. Модель включает три уровня: верхний уровень "Высокий" (студенты с высокой научной активностью), средний уровень "Средний" (умеренная научная активность) и нижний уровень "Низкий" (минимальная активность). Сплошные стрелки обозначают возможные переходы внутри одного уровня, обычные стрелки - переход на более высокий уровень, а пунктирные стрелки указывают на снижение научной активности с переходом на уровень ниже.

Студенты младших курсов редко участвуют в научных конференциях, и как следствие – ведут активную научную деятельность, более того, тот же НИРС у многих по учебному плану предусмотрен начиная с 3 курса, поэтому изначально заложенные требования и ожидания от такой группы будет соответствующим [9]. Если предположить, что с учетом продолжения обучения эта выборка студентов будет вести среднюю научную активность, то их переход будет выглядеть так: 3 –> 2 –> 5. Если же таковые студенты с переходом на следующие курсы будут существенно увеличивать свою научную активность, то их переход возможен на высокий уровень, к примеру: 3 –> 4 –> 8 или 3 –> 2 –> 8. Аналогичным образом происходит переход у остальных студентов.

Важно отметить, что эффективность вовлечения студентов в научную деятельность наиболее высока при условии их добровольного участия и подлинной заинтересованности. В случаях, когда исследования выполнялись по внешнему требованию, результаты оказывались слабее. Как показывают данные, ключевым барьером для участия студентов является отсутствие интереса, а не нехватка компетенций [10-11].

На основании приведенной выше структурной модели, а также при наличии ретроспективных данных требуется построение матрицы вероятностей перехода студентов ($P \in R^{6x6}$). Следующим этапом предполагается создание математической модели динамики изменения численности кластеров, на основании чего можно будет прогнозировать количество статей к заданному моменту времени.

Выводы

По доле в общем количестве научных работ (публикаций, патентов, диссертаций, диссертационных советов и др.) Россия значительно уступает таким мировым лидерам, как Китай и США. Это отставание негативно влияет на все аспекты социально-экономического развития страны. Поэтому крайне важно разработать новые механизмы привлечения талантливых студентов к научной деятельности, начиная с первых лет обучения.

Студенческий контингент оказывает влияние на показатели эффективности научной деятельности вуза за счет своих студенческих публикаций, как правило – в рамках НИРС. Данную группу участников научной деятельности вуза необходимо учитывать в итоговых расчетах.

Проведенное исследование подтверждает необходимость включения системной работы со студенческими публикациями в научную политику современных университетов в качестве обязательного компонента. В работе проанализирован вклад студенческих публикаций как индикатора эффективности научной деятельности вузов. Установлена значимая взаимосвязь между объемом таких публикаций и повышением эффективности научной деятельности университетов.

Была построена структурная модель переходов студентов между кластерами, в рамках их публикационной активности, необходимая для дальнейшего создания математической модели динамики изменения численности кластеров, на основании чего можно будет прогнозировать количество статей к заданному моменту времени.

Литература

1. Kahu, E. R. Framing student engagement in higher education / E. R. Kahu. – DOI 10.1080/03075079.2011.598505. – Direct text // Studies in higher education, 2013. – Vol. 38, Issue 5. – P. 758–773.

2. Гринштейн, В. Э. Анализ существующих методик по оценке эффективности работы научных организаций / В. Э. Гринштейн // Правовая защита, экономика и управление интеллектуальной собственностью : материалы Всероссийской научно-практической конференции (г. Екатеринбург, 21 апреля 2015 г.). — Екатеринбург : Изд-во Уральского федерального университета им. первого Президента России Б. Н. Ельцина. — 2015. — С. 40–43.

3. Юань, Ф. Подготовка студентов к научно-исследовательской деятельности в Китае / Ф. Юань. – Текст : непосредственный // Мир науки, культуры, образования. – 2017. – No 3 (64). – С. 98–101.

4. Зятева, О. А. Управление научными показателями вуза: анализ публикационной активности / О. А. Зятева, Е. А. Питухин // ПНиО, 2019. - №4 (40). - URL: https://cyberleninka.ru/article/n/upravlenie-nauchnymi-pokazatelyami-vuza-analiz-publikatsionnoy-aktivnosti (дата обращения: 20.05.2025).

5. Павлова, Л. Л. Анализ вовлеченности студентов в научно-исследовательскую деятельность в высших учебных заведениях / Л. Л. Павлова, Е. Л. Филатова // Известия вузов. Социология. Экономика. Политика, 2024. - №2. - URL: https://cyberleninka.ru/article/n/analiz-vovlechennosti-studentov-v-nauchno-issledovatelskuyu-deyatelnost-v-vysshih-uchebnyh-zavedeniyah (дата обращения: 10.06.2025).

6. Хачикян, Е. И. Грантовая активность ученых в контексте оценки эффективности деятельности научных организаций и вузов / Е. И. Хачикян, Т. И. Филиппова, И. И. Пацакула // Ежегодник российского образовательного законодательства. – 2018. – Т. 13, № 18. – С. 225-237. – EDN TFBVEN.

7. Леонова, Т.Н. Эффективность грантового финансирования научно-исследовательских работ: мировой опыт и российские перспективы // ЭНСР, 2014. - № 4 (67).

8. Беллман, Р. Динамическое программирование и уравнения в частных производных [Текст] / Р. Беллман, Э. Энджел ; Пер. с англ. С. П. Чеботарева ; Под ред. А. М. Летова. - Москва : Мир, 1974.

9. Лапин, П. М. Способы вовлечения студентов в научно-исследовательскую работу в вузе / П. М. Лапин. – Текст : непосредственный // Социальные и гуманитарные науки : теория и практика. – 2020. – No 1 (4). – С. 319–325.

10. Russell, S. H. Undergraduate research opportunities : Facilitating and encouraging the transition from student to scientist / S. H. Russell. – Direct text // Creating effective undergraduate research programs in science. – 2008. – P. 53–80.

11. Резник, С. Д. Развитие интереса студенческой молодежи к научному поиску : опыт и проблемы регионального университета / С. Д. Резник, М. В. Черниковская. – Текст : непосредственный // Вестник Кемеровского государственного университета. Серия: Политические, социологические и экономические науки. – 2020. – Т. 5, No 2. – С. 186–194. - DOI 10.21603/2500-3372-2020-5-2-186-194.

Ульяненко А.Э. Анализ публикационной активности студентов и построение структурной модели движения бакалавров и магистров по группам. *Рассмотрен вклад студенческих публикаций, как один из показателей эффективности научной деятельности вузов. Выявлена корреляция между количеством студенческих публикаций и роста позиций университетов в рейтинге, обнаружена большая дифференциация, как в разрезе образовательных институтов, так и в разрезе ученых степеней. Была построена структурная модель переходов студентов между кластерами, в рамках их публикационной активности, необходимая для дальнейшего создания математической модели прогнозирования.*

Ключевые слова: *студенческие публикации, наукометрические показатели, индекс Хирша, университетские рейтинги, научная политика, марковские процессы.*

Ulianenko A.E. Publication activity of students. the model of movement of bachelors and masters in groups. *The contribution of student publications as one of the indicators of the effectiveness of scientific activity of universities is considered. The correlation between the number of student publications and the growth of the universities' positions in the ranking is revealed, and a large differentiation is found both in terms of educational institutions and in terms of academic degrees. A structural model of student transitions between clusters was constructed within the framework of their publication activity, which is necessary for further creation of a mathematical forecasting model.*

Keywords: *student publications, scientometric indicators, Hirsch index, university ratings, scientific policy, markov processes.*

Статья поступила в редакцию 12.06.2025
Рекомендована к публикации доцентом Карабчевским В. В.

Об авторах

Алымов Даниил Андреевич - магистрант кафедры программной инженерии им. Л. П. Фельдмана факультета интеллектуальных систем и программирования ФГБОУ ВО «Донецкий национальный технический университет».

Боднар Алина Валериевна - кандидат технических наук, доцент, доцент кафедры программной инженерии им. Л. П. Фельдмана факультета интеллектуальных систем и программирования ФГБОУ ВО «Донецкий национальный технический университет».

Гурко Андрей Владимирович - кандидат технических наук, доцент, доцент кафедры информационных систем и вычислительной техники, Санкт-Петербургский Горный Университет императрицы Екатерины II.

Зори Сергей Анатольевич - доктор технических наук, доцент, заведующий кафедрой программной инженерии им. Л. П. Фельдмана факультета интеллектуальных систем и программирования ФГБОУ ВО «Донецкий национальный технический университет».

Койбаш Александр Андреевич – старший преподаватель кафедры компьютерной инженерии факультета интеллектуальных систем и программирования ФГБОУ ВО «Донецкий национальный технический университет».

Колесников Алексей Евгеньевич - студент кафедры автоматизированных систем управления факультета информационных систем и технологий ФГБОУ ВО «Донецкий национальный технический университет».

Левкина Анастасия Викторовна - ассистент кафедры прикладной математики и искусственного интеллекта факультета интеллектуальных систем и программирования ФГБОУ ВО «Донецкий национальный технический университет».

Лукащук Михаил Олегович - магистрант кафедры программной инженерии им. Л. П. Фельдмана факультета интеллектуальных систем и программирования ФГБОУ ВО «Донецкий национальный технический университет».

Мальчева Раиса Викторовна – кандидат технических наук, доцент, доцент кафедры компьютерной инженерии факультета интеллектуальных систем и программирования ФГБОУ ВО «Донецкий национальный технический университет».

Мартыненко Татьяна Владимировна - кандидат технических наук, доцент, доцент кафедры автоматизированных систем управления факультета информационных систем и технологий ФГБОУ ВО «Донецкий национальный технический университет».

Нестеренко Александра Романовна - студент кафедры программной инженерии им. Л. П. Фельдмана факультета интеллектуальных систем и программирования ФГБОУ ВО «Донецкий национальный технический университет».

Радевич Екатерина Владимировна - ассистент кафедры прикладной математики и искусственного интеллекта факультета интеллектуальных систем и программирования ФГБОУ ВО «Донецкий национальный технический университет».

Савицкая Ирина Валериевна – ассистент кафедры прикладной математики и искусственного интеллекта факультета интеллектуальных систем и программирования ФГБОУ ВО «Донецкий национальный технический университет».

Семёнова Анастасия Павловна - старший преподаватель кафедры прикладной математики и искусственного интеллекта факультета интеллектуальных систем и программирования ФГБОУ ВО «Донецкий национальный технический университет».

Сорокин Артём Александрович – студент кафедры информационных систем и вычислительной техники, Санкт-Петербургский Горный Университет императрицы Екатерины II.

Сухорукова Елена Олеговна- ассистент кафедры прикладной математики и искусственного интеллекта факультета интеллектуальных систем и программирования ФГБОУ ВО «Донецкий национальный технический университет».

Ульяненко Антон Эдуардович - аспирант кафедры компьютерного моделирования и дизайна факультета информационных систем и технологий ФГБОУ ВО «Донецкий национальный технический университет».

Шуватова Екатерина Александровна - ассистент кафедры автоматизированных систем управления факультета информационных систем и технологий ФГБОУ ВО «Донецкий национальный технический университет».

Ястребов Александр Романович - магистрант кафедры программной инженерии им. Л. П. Фельдмана факультета интеллектуальных систем и программирования ФГБОУ ВО «Донецкий национальный технический университет».

**Требования к статьям,
направляемым в редакцию научного журнала
«Информатика и кибернетика»**

Редколлегией принимаются к рассмотрению статьи, в которых рассматриваются важные вопросы в области информатики и кибернетики. Научный журнал издаётся с 2015 года, периодичность издания – 4 раза в год.

В журнале предусмотрены следующие рубрики:

- информатика и вычислительная техника;
- компьютерные и информационные науки;
- инженерное образование.

В соответствии с номенклатурой специальностей научных работников МОН ДНР первые две рубрики соответствуют следующим укрупненным группам специальностей научных работников:

05.01 – «Инженерная геометрия и компьютерная графика»,

05.13 – «Информатика, вычислительная техника и управление».

С 01.02.2019 Научный журнал включён в Перечень рецензируемых научных изданий, в которых должны быть опубликованы основные научные результаты диссертаций на соискание учёной степени кандидата наук, на соискание учёной степени доктора наук (приказ МОН ДНР № 135) по группам специальностей 05.01.00 и 05.13.00.

Рубрика «Инженерное образование» предназначена опубликования сотрудниками научно-методических статей.

Журнал также включён в базу данных РИНЦ (Российский индекс научного цитирования) (лицензионный договор № 425-07/2016 от 14.07.2016).

Статьи, представляемые в данный сборник, должны отвечать следующим требованиям. ***Содержание статьи*** должно быть посвящено актуальным научным проблемам и включать следующие необходимые элементы:

- постановку проблемы в общем виде, её связь с важными научными и практическими задачами;
- анализ последних исследований и публикаций, в которых решается данная задача и на которые опирается автор, выделение нерешенных ранее частей общей проблемы, которым посвящается статья;
- формулировка цели статьи и постановка задач, решаемых в ней;
- изложение основного материала с полным обоснованием полученных научных результатов;
- выводы и перспективы последующих исследований в данном направлении.

Каждый элемент должен быть выделен соответствующим названием раздела, например, «введение», «постановка задачи», «цель и задачи работы», «цель статьи», «цель исследования», «цель разработки», «анализ … », «сравнительная оценка … », «разработка … », «проектирование … », «программная реализация», «тестирование … », «полученные результаты», «выводы», «литература». Разделы «введение», «выводы», «литература» являются обязательными. Включать в названия разделов нумерацию не разрешается.

В основном тексте статьи формулируются и обосновываются полученные авторами утверждения и результаты. Выводы должны полностью соответствовать содержанию основного текста. Языки публикаций: русский, английский.

Объём статьи, формат страницы

Для оформления статьи следует использовать листы формата А4 (210х297 мм) с полями по 2,5 см со всех сторон. Нумерацию страниц выполнять не нужно.

Рекомендуемый объём статьи – 6-12 страниц. Рукописи меньшего объёма могут быть рекомендованы к публикации в качестве коротких сообщений.

Последняя страница текста статьи должна быть заполнена не менее чем на две трети, но содержать не менее трёх пустых строк в конце.

Форматирование текста

Подготовка статьи осуществляется в текстовом редакторе Microsoft Office Word.

Весь текст статьи оформляется шрифтом Times New Roman 10 пт с одинарным междустрочным интервалом, если ниже в требованиях не сказано иного. Абзацный интервал «перед» – 0 пт, «после» – 0 пт.

На первой строке с выравниванием по левому краю располагается УДК.

Заголовок (название) статьи оформляется шрифтом Times New Roman 14 пт, полужирное начертание, с выравниванием по центру (без абзацных отступов). Заголовок статьи следует печатать с прописной буквы без точки в конце, переносы слов не допускаются. Абзацный интервал «перед» – 12 пт, «после» – 12 пт.

После названия статьи следует информация об авторах, которая выравнивается по центру (без абзацных отступов). На одной строке указываются инициалы и фамилии всех авторов через запятую. Между двумя инициалами ставится пробел. С новой строки указывается название вуза (организации) и город (для каждого автора, если не совпадают). На следующей строке указываются адреса электронной почты (один адрес либо каждого автора – по желанию). Адрес электронной почты оформляется в виде гиперссылки.

К тексту аннотации применяется курсивное начертание, с выравниванием по ширине, отступы слева и справа по 1 см. Заголовок «Аннотация» выделяется полужирным начертанием. Объём аннотации – 450-550 символов (без пробелов). Абзацный интервал «перед» – 12 пт, «после» – 12 пт.

Основной текст статьи разбивается на две колонки шириной по 7,5 см (промежуток между столбцами – 0,99 см), выравнивается по ширине. Абзацный отступ первой строки – 1 см. Автоматический перенос слов не применяется.

Заголовки разделов выполняются шрифтом Arial 10 пт, полужирное курсивное начертание. Абзацный отступ отсутствует, интервал перед абзацем – 12 пт, после абзаца – 6 пт. Для заголовка «Введение» установить интервал «перед» – 0 пт, «после» – 6 пт.

Таблицы в тексте статьи

Название следует помещать над таблицей с абзацного отступа (1 см) в формате: слово «Таблица», пробел, номер таблицы, пробел, тире, пробел, название таблицы. Название таблицы записывают с прописной буквы без точки в конце строки и выравнивают по ширине. В ячейках таблицы устанавливается выравнивание текста по центру по вертикали. По горизонтали текст выравнивается по центру либо по левому краю. Границы ячеек таблицы должны быть только чёрного цвета, толщина линии – 1 пт. На все таблицы должны быть приведены ссылки в тексте статьи, при ссылке следует писать слово «табл.» с указанием её номера, например, « … данные приведены в табл. 5». Таблицы нумеруются в пределах статьи. Таблица располагается сразу после ссылки на неё, если это возможно (например, после окончания абзаца). Если же таблица не помещается на текущей странице, то она должна быть расположена в начале следующей страницы (или колонки). При необходимости допускается включение в статью таблицы, ширина которой превышает ширину колонки. В этом случае таблица и её название размещаются по центру страницы. Таблица не должна выступать за границы полей страницы. Таблица и её название отделяются от основного текста статьи одной пустой строкой до и после.

Рисунки в статье

Ссылки на иллюстрации по тексту статьи обязательны и оформляются в виде « ... на рис. 2» и т. п. Рисунок и его подпись выравниваются по центру колонки (без абзацных отступов), положение рисунка – «в тексте». Размещается рисунок после его первого упоминания в тексте, если это возможно (например, после окончания абзаца). Если же иллюстрация не помещается на текущей странице, то она должна быть расположена в начале следующей страницы (или колонки). При необходимости допускается включение в статью рисунка, ширина которого превышает ширину колонки. В этом случае рисунок и его подпись выравниваются по центру страницы. Иллюстрация не должна выступать за границы полей страницы. Подпись рисунка оформляется в формате: слово «Рисунок», пробел, номер иллюстрации, пробел, тире, пробел,

название рисунка. Название рисунка записывают с прописной буквы без точки в конце строки. Для подписи иллюстрации применяют курсивное начертание. Иллюстрация и её подпись отделяются от основного текста статьи одной пустой строкой до и после. Не допускается выполнять рисунки с помощью встроенного графического редактора Microsoft Office Word. Если на иллюстрации имеется текст, размер шрифта должен быть не менее чем аналогичный текст, набранный шрифтом Times New Roman 10-го размера. Иллюстрация не должна содержать много незаполненного пространства.

Формулы

Формулы и уравнения рекомендуется набирать с использованием MathType (предпочтительно) или MS Equation. Формулы и математические символы не должны существенно отличаться по размеру от основного текста. Обязательной является нумерация формул, на которые имеется ссылка в тексте статьи. Ссылки в тексте на порядковые номера формул дают в скобках, например, « ... согласно формуле (2)». Формулы размещаются по центру колонки, а их номера – по правому краю. Как для строки с формулой, так и для первой строки пояснений (при наличии), абзацный отступ убирается. Первая строка пояснения начинается со слова «где», после которого следует поставить табуляцию на 1 см, затем само пояснение в формате: символ, подлежащий объяснению, пробел, тире, пробел, поясняющий текст, запятая, обозначение единицы измерения физической величины. Пояснения перечисляются через точку с запятой, выравниваются по ширине. Вторая и последующие строки пояснений начинаются с абзацного отступа (1 см). Весь блок текста, связанный с формулой (только формула, несколько формул подряд или формула с пояснениями), отделяется от основного текста одной пустой строкой до и после. Переносить формулы на следующую строку допускается только на знаках выполняемых операций, причем знак в начале следующей строки повторяют. При переносе формулы на знаке умножения применяют знак «×». Формулы и математические уравнений могут быть записаны в тексте документа, если их высота не превышает высоту строки. При этом следует учитывать, что знаки математических операций отделяются от чисел или символов пробелами с обеих сторон. Например, «Если учесть, что $y < 0$ и $2x + y = 1$, то из формулы (3) можно выразить x…». К символам, которые приведены в формуле, при дальнейшем их употреблении (в том числе в пояснениях к формуле) должно применяться курсивное начертание. При этом к любым числам (верхние и нижние индексы, содержащие цифры и т. п.), а также к математическим знакам курсивное начертание не применяется. Не допускается вставлять формулы, выполненные в виде рисунков.

Перечисления: оформление списков

Основной текст статьи может содержать перечисления, оформленные в виде маркированного списка. В качестве маркера элемента списка разрешается использовать только короткое тире «–». Каждый элемент перечисления записывается с новой строки с абзацного отступа, равного 1 см. После символа короткого тире текст располагается с отступом в 1,5 см от левой границы строки, выравнивается по ширине, при переносе на новые строки располагается без отступов. Нумерованные и многоуровневые списки включать в статью не разрешается.

Литература

В тексте статьи обязательны ссылки на все литературные источники, номер источника указывается в квадратных скобках. Ссылки на неопубликованные работы не допускаются. Рекомендуемое количество источников, на которые ссылается автор, не менее 10. Перечень источников приводится в порядке их упоминания в статье. Библиографическое описание каждого литературного источника оформляется в соответствии с ГОСТ Р 7.0.100–2018. Перечень литературных источников оформляется в виде нумерованного списка. В качестве маркеров элементов списка используют порядковые арабские цифры с точкой. Каждый источник представляет собой отдельный элемент перечисления, записывается с новой строки с абзацного отступа, равного 1 см. После порядкового номера с точкой текст располагается с отступом в 1,5 см от левой границы строки, выравнивается по ширине, при переносе на новые строки располагается без отступов.

В конце статьи обязательно приводятся аннотации на русском и английском языках, каждая заканчивается перечнем 5-6 ключевых слов.

К тексту аннотации применяется курсивное начертание, с выравниванием по ширине, отступы слева и справа по 1 см. Слово «Аннотация» опускается. Текст аннотации начинается с ФИО авторов и названия статьи, выделяемых полужирным начертанием. Аннотация на русском языке совпадает с аннотацией, приведенной в начале статьи. В тексте аннотации на английском языке после фамилии автора указывается только первая буква имени с точкой. Абзацный интервал «перед» – 12 пт, «после» – 12 пт. Ключевые слова оформляются с новой строки аналогично тексту аннотации. Заголовок «Ключевые слова:» (англ. «Keywords:») выделяется полужирным начертанием. Ключевые слова перечисляются через запятую.

Порядок представления статьи и сопроводительные документы

В редакцию необходимо представить:

– файл с текстом статьи;

– файл, содержащий фамилию, имя и отчество авторов полностью; ученую степень, ученое звание; место работы с полным указанием должности, подразделения и наименования организации, города (страны); номера телефонов и e-mail для связи;

– экспертное заключение о возможности публикации статьи, подписанное руководителем и заверенное печатью организации, в которой работает автор статьи;

– выписка из заседания кафедры или письмо организации с просьбой об опубликовании и указанием, что изложенные в статье результаты ранее не публиковались.

Статьи и сопроводительные документы следует высылать на электронный адрес infcyb.donntu@yandex.ru.

К сведению авторов

Если статья оформлена с нарушением указанных выше требований и правил, редакция после предварительного рассмотрения может отклонить статью.

На рецензирование статьи направляются членам редакционной коллегии журнала. Все статьи публикуются при наличии положительной рецензии.

В статью могут быть внесены изменения редакционного характера без согласования с автором. Ответственность за содержание статьи и качество перевода аннотаций несут авторы.

Публикация статей в научном журнале «Информатика и кибернетика» осуществляется на некоммерческой основе.

Все номера Научного журнала размещаются в электронной библиотечной системе ФГБОУ ВО «ДонНТУ» и на сайте http://infcyb.donntu.ru/.

.

CONTENT

Informatics and computer engineering

Computer science

Электронное периодическое издание

Научный журнал

ИНФОРМАТИКА И КИБЕРНЕТИКА

(на русском, английском языках)

№ 2 (40) - 2025

Ответственный за выпуск Р. В. Мальчева

Технический редактор Р. В. Мальчева

Компьютерная верстка Р. В. Мальчева

Подписано к выпуску 18.06.2025. Усл. печ. лист. 9,0. Уч.-изд. лист. 5,9.
Адрес редакции: ДНР, 283001, г. Донецк, ул. Артема, 58, ФГБОУ ВО «ДонНТУ»,
4-й учебный корпус, к. 36.
Тел.: +7 (856) 301-07-35, +7 (949) 334-89-11
E-mail: infcyb.donntu@yandex.ru, URL: http://infcyb.donntu.ru