

# Разработка симулятора для генерации синтетических данных с целью обучения нейронных сетей в условиях неопределенности

В. В. Ковалев<sup>1</sup>, А. А. Корнеева<sup>2</sup>, А. О. Федоркова<sup>3</sup>, Е. А. Масленникова<sup>4</sup>,  
Д. М. Филатов<sup>5</sup>, Д. П. Михайлов<sup>6</sup>

Санкт-Петербургский государственный электротехнический университет  
«ЛЭТИ» им. В.И. Ульянова (Ленина)

<sup>1</sup>vladimir.kovalev.2012@bk.ru, <sup>2</sup>anna.korneeva89@gmail.com, <sup>3</sup>anastasiikanata@gmail.com,  
<sup>4</sup>katerinamaslennikova1494@gmail.com, <sup>5</sup>dmfilatov@etu.ru, <sup>6</sup>dpmikhailov@etu.ru

**Аннотация.** В данной работе рассматривается разработка симулятора для генерации синтетических данных, предназначенных для обучения нейронных сетей в условиях неопределенности. Предлагаемый подход позволяет моделировать сложные сцены с высоким уровнем фотореализма, с целью улучшения обобщающей способности алгоритмов машинного обучения при их применении в реальном мире. Рассматриваются методы генерации искусственных данных, их использование для обучения нейронных сетей, а также способы учета факторов неопределенности. Проведены эксперименты, демонстрирующие эффективность использования синтетических данных для повышения устойчивости моделей к изменяющимся условиям.

**Ключевые слова:** симулятор, генерация синтетических данных, обучение нейронных сетей, фотореализм, компьютерное зрение, машинное обучение, моделирование сложных сцен, условия неопределенности, адаптация алгоритмов, системы поддержки принятия решений

## I. ВВЕДЕНИЕ

С момента появления первых нейронных сетей мир технологий стремительно менялся, открывая новые горизонты в области обработки данных и машинного обучения. Развитие вычислительной мощности компьютеров и улучшение алгоритмов позволили создавать системы, способные решать задачи, ранее доступные только человеку. В последние годы важным шагом вперед стало использование синтетических данных для обучения нейронных сетей, что открыло новые возможности для разработки и тестирования моделей в условиях неопределенности.

Сложные сцены, которые раньше требовали реальных данных, теперь можно эффективно моделировать с помощью симуляторов, создавая фотореалистичные изображения, которые полностью соответствуют реальной жизни. Это позволило значительно ускорить процесс обучения нейронных сетей и повысить их обобщающую способность. Однако при моделировании данных возникает важный вопрос: как учитывать факторы неопределенности, такие как изменения условий освещения, атмосферных явлений или даже непредсказуемые изменения в поведении объектов.

Разработка симулятора для генерации синтетических данных, учитывающего эти неопределенности, представляет собой ключевое направление в современном машинном обучении и компьютерном зрении. В данной работе рассматриваются методы создания таких симуляторов, их применение для обучения нейронных сетей, а также способы повышения устойчивости алгоритмов к изменяющимся условиям. Мы анализируем роль синтетических данных в улучшении работы систем поддержки принятия решений и демонстрируем, как адаптация алгоритмов к неопределенности может повлиять на эффективность решений в реальных ситуациях.

## II. ОБЗОР ГОТОВЫХ РЕШЕНИЙ

Существует два основных направления для визуальных графических симуляторов.

### A. Физическая составляющая

Физическая составляющая симуляторов является одной из ключевых для достижения реалистичности и точности при моделировании различных процессов. В основе физического моделирования лежат математические модели, которые описывают реальные физические явления, такие как движение объектов, взаимодействие материалов, силы, скорость, ускорение, столкновения и другие параметры, влияющие на поведение системы. Эти модели позволяют симулировать реальный мир в виртуальной среде и исследовать его динамику без необходимости вмешательства в реальность, что открывает широкий спектр возможностей для различных отраслей. Основные компоненты физического моделирования:

- Механика движения – определяет движение объектов в пространстве, учитывая такие факторы, как масса, сила, трение, ускорение и гравитация. Механика движения позволяет моделировать поведение тел, их траектории и взаимодействие друг с другом.
- Силы и взаимодействия – важнейшей частью физического моделирования является учет различных сил, таких как силы тяжести, упругости, трения, сопротивления среды и т.д. Эти силы воздействуют на объекты и определяют их поведение в пространстве.

- Столкновения и деформация – когда объекты взаимодействуют между собой (например, при столкновении), важную роль играет моделирование этих столкновений и деформаций. Для этого используют модели жестких и мягких тел, которые позволяют точно симулировать, как объекты могут изменять свою форму или поглощать энергию при столкновении.

В некоторых симуляторах учитываются процессы, связанные с теплообменом, изменением температуры, фазовыми переходами в некоторых областях симуляции, таких как аэродинамика или гидродинамика, важным аспектом является моделирование поведения жидкостей и газов. Это может включать моделирование течений, волн, взаимодействие с твердыми телами и другие явления.

### *В. Визуальная составляющая*

Визуальная составляющая в симуляторах играет ключевую роль в создании реалистичной и погружающей виртуальной среды. От качества и точности визуальных элементов зависит не только восприятие пользователем, но и эффективность симуляции. Современные симуляторы стремятся к максимальному фотореализму, что позволяет использовать их для обучения, тестирования, разработки, а также в развлекательных приложениях, таких как игры и тренажеры.

Визуальная составляющая симуляторов включает в себя несколько ключевых компонентов, каждый из которых отвечает за создание и обработку графических объектов, сцены и окружения:

- Модели объектов – все элементы виртуальной среды, такие как здания, транспортные средства, персонажи или природные объекты, создаются в виде 3D-моделей. Эти модели могут быть как статичными (например, стены, мебель, деревья), так и динамичными (например, движущиеся машины, колеблющиеся объекты, анимация персонажей). Для создания высококачественных моделей используются специализированные 3D-редакторы, такие как Blender, Autodesk Maya, 3ds Max и другие. Чем детальнее и точнее модель, тем более реалистичной будет сцена.
- Текстуры и материалы – текстуры и материалы обеспечивают визуальную реалистичность объектов, добавляя им детали, такие как цвет, шероховатость, отражения, прозрачность, освещенность и другие характеристики поверхности. Текстуры могут имитировать поверхности различных материалов — от бетона и металла до кожи и ткани. В современных симуляторах используется ряд продвинутых методов текстурирования, таких как bump mapping (для имитации рельефа), normal mapping (для детализации поверхности без дополнительной геометрии), specular mapping (для создания блеска на поверхностях), а также PBR (Physically Based Rendering) – система, моделирующая реалистичные взаимодействия света с поверхностями материалов [1].

- Освещение и тени – освещение — один из важнейших элементов визуальной составляющей симуляторов. Свет определяет, как объекты выглядят в виртуальной среде, и является ключевым фактором, влияющим на восприятие глубины, формы и цвета объектов. В симуляторах используются различные типы освещения, такие как точечный свет, направленный свет, освещение окружением и глобальное освещение. Современные симуляторы также используют технологии ray tracing (трассировка лучей) для создания реалистичных теней, отражений и преломлений света. Тени играют важную роль в восприятии сцены, так как они придают объектам объем и помогают формировать реальное восприятие пространства. Современные движки, такие как Unreal Engine и Unity, поддерживают различные алгоритмы для создания динамических и статических теней, которые соответствуют времени суток, погодным условиям и сценарию [2].

- Камеры и ракурсы – визуализация сцены также зависит от того, как расположена виртуальная камера, через которую пользователь наблюдает за окружающим миром. Камера в симуляторе может быть настроена для отображения различных углов обзора и условий, таких как вид от первого или третьего лица, динамическая камера (например, в автомобилях или самолетах) или статические ракурсы для предустановленных сцен.

Также используется постобработка для улучшения визуальных характеристик изображения: цветокоррекция, размытие (например, при движении), глубина резкости, световые блики и другие фильтры, которые помогают создать более привлекательную и реалистичную картину.

### **III. РАЗРАБОТКА СИМУЛЯТОРА ДЛЯ ГЕНЕРАЦИИ ДАННЫХ**

Для того чтобы разработать качественный, хороший симулятор, такой чтобы удовлетворял наши потребности в генерации данных, требуется прежде определить его архитектуру, уточнить вопросы в каком формате будут поступать данные и какие вообще процессы необходимо отобразить.

#### *А. Подготовка моделей и формирование архитектуры симулятора*

Прежде всего, отметим, что мы рассматриваем задачу генерации данных номерных знаков автомобилей, значит в нашей системе уже будут необходимы трёхмерные модели автомобилей их номерные знаки и самое главное требуется сделать так, чтобы сами номерные знаки могли изменяться. Продемонстрируем на изображении ниже архитектуру симулятора:

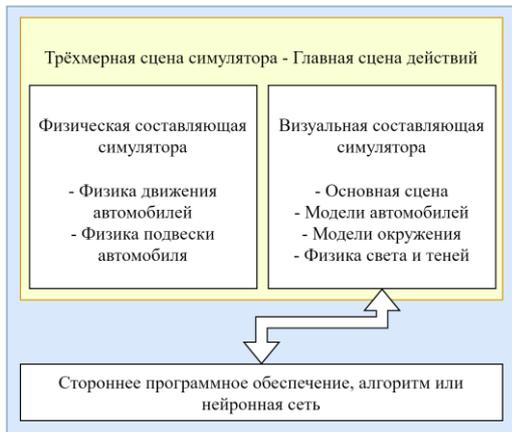


Рис. 1. Схема архитектуры симулятора для отработки алгоритмов компьютерного зрения и работы нейронных сетей

Из рис. 1 можно увидеть, что для реализации симулятора под отработку алгоритмов компьютерного зрения или работу нейронных сетей требуется создать физическую симуляцию автомобилей их простейшее поведение и визуализировать данные процессы в трёхмерной сцене, также обязательно требуется реализовать вывод изображений со сцены в компьютерную сеть, чтобы у стороннего программного обеспечения были актуальные изображения сцены симулятора.

### В. Реализация симулятора в Unity

Разработка трёхмерных моделей автомобилей и их номерных знаков производилась с помощью программы моделирования Blender, однако трёхмерная модель сама по себе нас не интересует, необходимо вписать её в окружение, поэтому реализуем сцену парковки автомобилей в Unity, результат представлен на изображении ниже:



Рис. 2. Парковочная сцена в Unity

На рис. 2 четко видно все модели и номерные знаки автомобилей, и хотя качество графики оставляет желать лучшего, для работы с образом автомобиля и алгоритмом распознавания текста данный кадр вполне подходит, хотя бы для реализации комбинаций номерных знаков автомобилей, отсутствие или меньшая выборка определенных символов для распознавания текста может стать критичной для алгоритма распознавания.

Кроме реализации статичной сцены парковки автомобилей также в симуляторе на Unity реализовано несколько классов автомобилей, с целью симуляции их движения, а картинка со сцены сразу же попадает на две камеры, одна камера отвечает за наблюдателя внутри

сцены как правило именуется MainCamera и позволяет игроку наблюдать сцену. Для реализации передачи изображения из симулятора куда-либо в сеть компьютера необходимо создать второй источник захвата изображения и реализовать в нём функцию передачи изображения по IP-протоколу. Для простоты понимания продемонстрируем реализацию на изображении ниже:

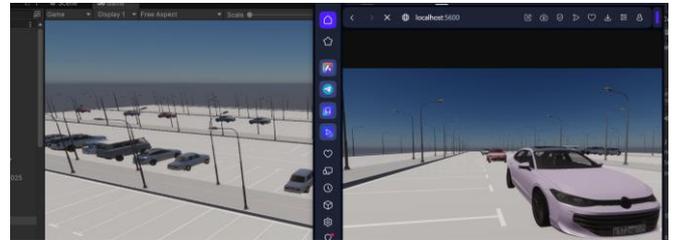


Рис. 3. Слева сцена внутри движка Unity, справа окно браузера с открытым локальным соединением по порту 5600

На рис. 3 можно наблюдать результат работы симулятора, который генерирует трёхмерные модели автомобилей, даёт им случайный цвет и номерной знак, также присутствует смещение автомобилей симулируя их движение. Все указанные детали отображаются на камере для передачи видеопотока в локальной сети компьютера на порт 5600. В правой части изображения видно окно браузера с открытым соединением с портом 5600.

### С. Реализация симулятора в Unreal Engine

Попробуем также реализовать похожую сцену в Unreal Engine, требуется использовать такие же модели для воссоздания аналогичного окружения. Результат представлен ниже на изображении.



Рис. 4. Пример изображения, созданного с помощью генерации данных

Из рис. 4 видно, та же самая сцена воссоздана и в Unreal Engine 5, но на данном этапе существуют сложности с передачей изображения из трёхмерной сцены в сеть компьютера при работе с Unreal Engine 5.

Нельзя не заметить качественную разницу в изображениях аналогичных друг к другу сцен, невооруженным глазом видно, что начальная картинка с добавлением простых трёхмерных моделей в Unity и Unreal Engine отличается, так в последнем присутствует элемент фотореалистичности за счёт более тонкой настройки теней и отражений, в то время как в Unity картинка требует дополнительной настройки художников.

#### D. Генерация данных и результаты

Если же говорить про использование сгенерированных данных, то проще всего взять стандартную нейронную сеть типа YOLOv5 и провести обучение и тесты на распознавание автомобиля, впоследствии данные тесты можно уточнить для распознавания номерных знаков и задачи по распознаванию текста, в данной работе упор совершен на генерацию данных.

При обучении сети можно рассмотреть три пути, обучение на настоящем датасете, на искусственном наборе данных и на смешанном наборе данных, сравним все три подхода к обучению сети YOLOv5 при условии что мы проводим обучение до 500 эпох всего на наборе данных до 5000 изображений, это может показаться достаточно малым числом для реализации бесконечной генерации изображений, но это точно первый показательный результат нашего подхода. Если учитывать показатель mAP, количество изображений, разность точности определения показатели представлены в табл. 1:

ТАБЛИЦА 1.

Режим обучения	Выборка	mAP_0.5:0.95	1Т	2Т	3Т
Реальная выборка	2200	0.49255	0.82	0.92	0.88
Искусственная выборка	2500	0.97803	0.97	0.85	0.61
Смешанная выборка	4700	0.83255	0.88	0.95	0.95

Анализ результатов, представленных в таблице, показывает, что внедрение синтетических данных в процессе смешанного обучения приводит к значительному улучшению показателей. Рассмотрено формирование наборов данных от 2000 изображений согласно [3].

#### IV. ЗАКЛЮЧЕНИЕ

В результате продемонстрированы возможности создания визуальных графических симуляторов для отработки алгоритмов и работы нейронных сетей с целью генерации новых данных для повышения точности обучения нейронных сетей и отслеживания работы алгоритмов, из полученных данных о возможностях программ Unity и Unreal Engine, а также из результатов обучения нейронной сети на настоящих, искусственных и смешанных форматах данных можно сделать выводы о пригодности новых методов для обучения нейронных сетей.

#### СПИСОК ЛИТЕРАТУРЫ

- [1] Creating Physically Based Materials [Электронный ресурс]. – Режим доступа: <https://learn.unity.com/tutorial/creating-physically-based-materials> (дата обращения: 14.03.2025).
- [2] Lumen Technical Details [Электронный ресурс]. – Режим доступа: <https://dev.epicgames.com/documentation/en-us/unreal-engine/lumen-technical-details-in-unreal-engine> (дата обращения: 14.03.2025).
- [3] Wenkao Yang, Wei Zhang. Real-time Traffic Signs Detection Based on YOLO Network Model // 2020 International Conference on Cyber-Enabled Distributed Computing and Knowledge Discovery (CyberC). С 357.