

ПРАВИТЕЛЬСТВО РОССИЙСКОЙ ФЕДЕРАЦИИ  
ФГАОУ ВО НАЦИОНАЛЬНЫЙ ИССЛЕДОВАТЕЛЬСКИЙ УНИВЕРСИТЕТ  
«ВЫСШАЯ ШКОЛА ЭКОНОМИКИ»

Факультет компьютерных наук  
Образовательная программа «Компьютерные науки и анализ данных»

**Отчет о программном проекте на тему:**  
**Action-based Нейросетевой Видеомонтаж**

**Выполнил студент:**

группы #БКНАД211, 3 курса  
Харитонов Борис Константинович

**Принял руководитель проекта:**

Евсиков Виктор Алексеевич  
Приглашенный преподаватель  
Факультет компьютерных наук НИУ ВШЭ

# Содержание

<b>Аннотация</b>	<b>3</b>
<b>1 Введение</b>	<b>4</b>
1.1 Описание предметной области . . . . .	4
1.2 Задачи проекта . . . . .	4
1.3 Обзор конкурентных продуктов . . . . .	4
1.4 Технические требования . . . . .	5
1.5 Анализ исходного материала . . . . .	6
1.6 Анализ процесса ручного монтажа . . . . .	7
<b>2 Описание высокоуровневой архитектуры предлагаемого решения</b>	<b>7</b>
<b>3 Основная (backbone) модель</b>	<b>9</b>
3.1 Обзор литературы по архитектурам основной модели . . . . .	9
3.2 Описание выбранного датасета . . . . .	9
3.3 Описание архитектурных принципов разработанных моделей . . . . .	10
3.4 Разработанные модули . . . . .	10
<b>4 Алгоритм постобработки</b>	<b>11</b>
<b>5 Усовершенствование кода основной модели для ускорения процесса обработки</b>	<b>11</b>
<b>6 Заключение</b>	<b>13</b>
<b>Список литературы</b>	<b>14</b>
<b>7 Приложения</b>	<b>15</b>
7.1 Приложение А . . . . .	15
7.2 Приложение Б . . . . .	16

## Аннотация

Социальные сети играют ключевую роль в распространении информации и короткие видеоролики становятся самым популярным форматом. Однако процесс создания видеороликов из большого объема исходного видеоматериала, требует значительных временных и трудовых затрат.

Цель работы заключалась в разработке приложения, способного автоматически обрабатывать большие объемы видеоматериала и выделять из них наиболее подходящие фрагменты для создания коротких видеороликов.

Для достижения цели была разработана архитектура состоящая из двух модулей: основная модель для анализа и выборки кадров согласно заданным параметрам пользователя, и алгоритм постобработки трактовки ответов основной модели.

Благодаря разработанному приложению получилось сократить процесс создания коротких видеороликов, что делает предложенный метод актуальным и перспективным для дальнейшего развития в области автоматизированного видеомонтажа.

## Ключевые слова

Машинное обучение, видеомонтаж, глубокое обучение, нейросети, анализ данных, свёрточные слои.

# 1 Введение

## 1.1 Описание предметной области

В современном цифровом мире, где короткие видеоролики, такие как Shorts и Reels, стали самым популярными форматами контента на социальных платформах, существует потребность в автоматизированных инструментах для создания качественных и привлекательных видеороликов, основанных на большом объеме исходного материала, в том числе игровых сессиях. Однако, создание и монтаж видеороликов требует времени и определенных навыков.

В связи с этим, актуальной задачей становится разработка программного проекта, который позволит автоматизировать процесс видеомонтажа и сделать его более доступным для широкого круга пользователей. Результаты данной работы могут быть полезны для блогеров, маркетологов и всех, кто стремится создавать качественный видеоконтент.

## 1.2 Задачи проекта

Основная задача проекта - разработать и протестировать систему, которая автоматически суммаризирует короткие экшн видеоролики, выделяя наиболее значимые и динамичные моменты игры. Это позволит пользователям, геймерам и брендам быстро и эффективно создавать и распространять привлекательный контент, достигая широкой аудитории и повышая охват и количество подписчиков.

Проект представляет собой приложение, которое использует нейросетевые алгоритмы для автоматического монтажа видеороликов на основе заданных действий, тем самым позволит упростить и ускорить процесс создания видеороликов.

В рамках работы будет проведен анализ предметной области, изучены существующие методы искусственного интеллекта в области видеомонтажа, а также разработан и протестирован программный продукт, демонстрирующий эффективность и практическую применимость предложенного подхода.

## 1.3 Обзор конкурентных продуктов

В последние годы наблюдается значительный рост интереса к применению нейросетевых технологий в сферах кинематографии и мультимедиа. Несмотря на это, большинство существующих проектов имеют определенные ограничения и не полностью соответствуют задачам, которые ставятся в рамках данной курсовой работы.

В процессе разработки было выявлено несколько инициатив, они характеризуются высокой стоимостью, закрытым исходным кодом и необходимостью обработки данных на удаленных серверах, что существенно ограничивает их доступность и удобство использования. Хотя качество этих решений остается предметом предположений, можно предположить, что они могут быть эффективны в решении общих задач видеомонтажа.

Одним из примеров является продукт компании Typeframes, специализирующейся на создании текстовых видео на основе сценария. Хотя этот продукт представляет определенный интерес, он не полностью соответствует задаче обработки видео на основе существующих файлов.

Кроме того, появились генеративные модели, такие как Gen-1 и Gen-2 от Runway, Sora от OpenAI и Veo от Google DeepMind, которые могут создавать новый контент с нуля. Однако эти модели не предназначены для выполнения задач по монтажу и адаптации уже имеющихся видеоматериалов, что делает их неприменимыми в контексте данного исследования.

В дополнение к упомянутым продуктам стоит также отметить разработки в области автоматизированного видеомонтажа, использующие нейросети, такие как Magisto, Animoto, Pictory и Lumen5. Эти инструменты предлагают автоматизированные решения для создания видео с использованием предварительно настроенных шаблонов. Они удобны для быстрого создания презентаций и рекламных видео, но ограничены в возможностях кастомизации и не предназначены для работы с сложными видеоматериалами, требующими детальной обработки и монтажа.

Перспективным проектом является AutoPod, специализирующийся на монтаже подкастов. Развитие предлагаемого проекта направлено на создание доступного и открытого (open source) решения, которое будет способствовать прогрессу в этой области. Расширение функционала проекта позволит конкурировать с AutoPod в данной нише за счет предложения более гибких и мощных инструментов для видеомонтажа.

## 1.4 Технические требования

В рамках данного проекта были установлены технические требования, направленные на обеспечение широкой доступности и удобства использования разрабатываемого приложения на различных устройствах.

Приложение должно быть совместимо с современным пользовательским оборудованием и функционировать без необходимости подключения к удаленному серверу. Такой подход

исключает необходимость создания и поддержания серверной инфраструктуры, что снижает сложности, связанные с ее эксплуатацией. Кроме того, данное решение способствует повышению уровня безопасности личных компьютеров и данных пользователей, минимизируя риски, связанные с кибератаками на серверы. Таким образом, обеспечивается более высокая защита информации и удобство использования приложения для широкого круга пользователей.

Кроме того, весь проект написан на Python и предоставляется с открытым исходным кодом, что обеспечивает прозрачность разработки и предоставляет пользователям возможность самостоятельно аудировать и модифицировать код в соответствии с их потребностями. Такой подход способствует созданию безопасного, надежного и гибкого продукта, доступного для использования широким кругом пользователей.

## 1.5 Анализ исходного материала

Для разработки и тестирования, в рамках проекта, основным материалом являются видеозаписи игровых сессий по игре Minecraft. Minecraft - это трехмерная видеоигра для персональных компьютеров, телефонов и игровых консолей, которая предлагает игрокам возможность играть как от первого, так и от третьего лица. Для анализа и разработки метода суммаризации видеороликов будут использоваться записи игровых сессий в одном из командных соревновательных режимов игры.

В этом режиме игроки выполняют следующие задачи:

- Уничтожение баз противника: команда игроков должна атаковать и разрушить базы команды противника.
- Защита базы союзников: команда игроков должна защитить свою базу от атак противника.
- Сбор ресурсов с различных частей карты: команда игроков должна собирать ресурсы с разных участков карты для использования в игре.

Каждая сессия имеет разный состав команд, но задачи игроков и способы их достижения остаются неизменными от сессии к сессии, как следствие в исходном материале содержится большое количество повторов и незначительных событий, которые могут быть неинтересными для зрителя, особенно для тех, кто уже знаком с командным соревновательным режимом игры.

## 1.6 Анализ процесса ручного монтажа

Для выполнения общих задач монтажа видео необходимы определённые навыки и знания. В процессе работы требуется умение пользоваться специализированным программным обеспечением, а также следует учитывать, что даже монтаж коротких видеороликов может занимать значительное время. Анализ процесса ручной суммаризации видеозаписей игровых сессий в игре Minecraft позволяет выделить два основных этапа: отбор наиболее интересных фрагментов видео и их последующая компоновка в финальный видеоролик.

Алгоритм отбора следующий: Игроки часто имеют определенный план действий в игре, например, сосредоточение атак на одну из команд противника, если она проявляет высокую активность. Этот план реализуется через краткосрочные стратегии — действия, которые игрок предпринимает сразу после воскрешения. Основная задача суммаризации заключается в выборе наиболее захватывающих моментов из этих стратегий. В рамках краткосрочных стратегий выделяются два основных типа поведения игрока: "подготовка" и "атака". Для создания коротких видеороликов обычно отбираются сцены "атаки", так как моменты "подготовки" часто повторяются и не несут уникальной информации.

Следующим шагом является сборка отобранных фрагментов в один видеоролик с удалением резких переходов для обеспечения плавности восприятия. Так же как и процесс отбора это времязатратный процесс, так как переходы должны быть достаточно плавными, чтобы это не вызвало головокружение у зрителя. Иногда к видеороликам добавляется музыкальное сопровождение, что делает их более привлекательными и интересными для зрителей, хотя это требует дополнительного времени и профессиональных навыков.

## 2 Описание высокоуровневой архитектуры предлагаемого решения

Аналогично процессу ручного монтажа, предлагаемое решение состоит из двух модулей. Модуль основной модели размечает фрагменты исходных данных на заданные классы. Модуль алгоритма постобработки, который интерпретирует ответы основной модели.

Для наглядности работа моделей представлена на графиках ниже. Задача основной модели - классифицировать действия на заданных временных отрезках, каждый из которых состоит из 11 кадров, что эквивалентно примерно одной трети секунды. Для иллюстрации, рассмотрим поставленную в главах 1.5-1.6 задачу идентификации двух поведений игрока: "подготовки" и "атаки", которые на Рисунке 2.1 обозначены соответственно синим и желтым

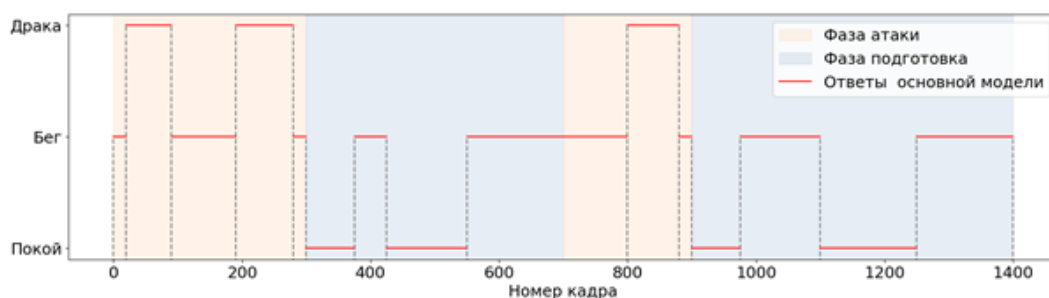


Рис. 2.1: Интерпретация результата работы основной модели на отрезке видеофайла сессии игры, синим и жёлтым обозначены фазы игры, красным - ответы основной модели.

цветами. Модель обучена распознавать три категории действий: покой, бег и бой. Результаты классификации модели представлены красным цветом. Кадры, где игрок находится в состоянии покоя, относятся к поведению при “подготовке”, в то время как кадры, где игрок участвует в бою, относятся к поведению при “атаке”. Действие бега может встречаться в любом из поведений игрока, что создает сложности в определении, к какой из этих фаз следует отнести эти кадры. Модуль алгоритма постобработки интерпретирует результаты,

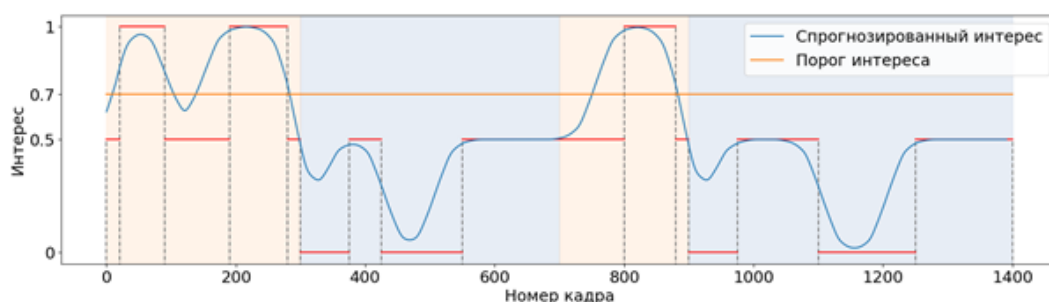


Рис. 2.2: Результат работы алгоритма постобработки, для размеченных основной моделью данных.

полученные от основной модели, и оценивает степень интереса зрителей к различным фрагментам видео. Для этого могут применяться разнообразные методы, включая усреднение покадровых оценок модели или применение фильтра скользящего среднего для сглаживания данных. Основная задача модуля, исходя из приведенного выше примера, заключается в повышении оценки интереса к действию "бег" в состоянии “атаки” и снижении этой оценки в состоянии “подготовки”. Примерный результат работы модуля постобработки представлен на Рисунке 2.2. Для получения конечного видео достаточно установить порог интереса. Все кадры, у которых оценка интереса превышает этот порог, будут интерпретированы как поведение при “атаке”. Использование более низкого порога позволит сохранить больше контекста в видео, облегчая понимание того, кто и на кого нападает. В то же время, более высокий порог сделает конечное видео более динамичным и резким.



## 3 Основная (backbone) модель

### 3.1 Обзор литературы по архитектурам основной модели

Задача, которую решает основная модель, относится к области распознавания действий (Action Recognition). Эта задача направлена на классификацию или сегментацию действий, выполняемых на видео. Например, один из датасетов YouCook2 [1] для данной задачи предлагает классифицировать различные моменты готовки.

При рассмотрении различных моделей, используемых для решения данной задачи, можно выделить две основные группы, хотя они не охватывают все существующие подходы, но включают в себя большинство из них.

Первая группа моделей в области Action Recognition характеризуется использованием более старых методов искусственного интеллекта. Такие модели обучаются на небольших датасетах, и количество классифицируемых действий в них ограничено. В этом контексте часто используются устоявшиеся архитектуры, такие как сверточные нейронные сети, анализ оптического потока или рекуррентные сети. Преимуществом таких моделей является их относительная простота и возможность обучения на небольших вычислительных мощностях, а также необходимость в небольшом объеме данных для обучения. Однако недостатком является то, что эти модели обычно не могут быть эффективно применены к решению задач на нескольких различных датасетах. Примеры таких моделей: Channel-Separated Convolutional Networks [2], SlowFast Networks [3], и просто Convolutional Neural Networks [4]

Вторая группа моделей использует более продвинутые методы решения задач в области Action Recognition, такие как различные версии слоев внимания и архитектуры типа трансформеров. Наиболее известными моделями трансформеров являются CLIP и GPT. Преимущества таких методов заключаются в их способности быть дообученными для решения новых задач. Однако они требуют значительных вычислительных ресурсов для обучения из-за больших объемов видеопамати и расчётов, необходимых для работы с ними. Примеры таких моделей: Vision-Language Model [5], Multiview Transformer [6]

### 3.2 Описание выбранного датасета

Датасет применяется для обучения основной модели, тогда как алгоритм постобработки подбирается эмпирическим путём.

Оригинальные видеоданные имеют разрешение 1920x1080 пикселей и частоту 30 кадров в секунду. Для анализа на вход модели подаются сжатые линейно до 480x270 пикселей

изображения. Такое сжатие не меняет основных признаков, к тому же не даст моделям переобучаться на дефектах видеоданных. С другой стороны это позволит любому рядовому пользователю не использовать модели, но и иметь возможность обучать их самому, без дополнительных трат на оборудование.

Модель принимает последовательность из нескольких подряд идущих кадров, представленных в виде тензора размером  $11 \times 480 \times 270 \times 3$ , где 11 это количество последовательных кадров,  $480 \times 270$  это разрешение изображения, и 3 канала RGB. Количество подаваемых на вход модели кадров - это гиперпараметр, который позитивно влияет на качество классификации действий, но негативно влияет на время выполнения модели. В качестве целевых меток используются данные о действиях, происходящих в видео, закодированные в формате one-hot, где каждая целевая переменная представлена вектором из нулей, и единицы для позиции соответствующей предсказываемому классу.

Действия классифицируются на 5 категорий: просмотр инвентаря, подготовка к игре, бездействие, бег и драка - они, хоть и не являются исчерпывающими, описывают большинство важных действий для монтажа экшн видеороликов.

### 3.3 Описание архитектурных принципов разработанных моделей

В рамках проекта использована модель со свёрткой по времени. Она принимает входные данные размером  $480 \times 270 \times 33$ , где 11 кадров объединяются в одно изображение с 33 каналами. Такой подход помогает снизить количество слоёв свёртки, что уменьшает время обучения и требования к видеопамяти.

Для модели был написан прототип. Он включает 11 слоев свертки, что позволяет сжать разрешение изображения до  $4 \times 7$ , создавая при этом большое количество каналов с различной информацией. После этого следуют полносвязный слой и еще один, формирующий 5 выходных нейронов для получения ответа.

### 3.4 Разработанные модули

В процессе разработки приложения было решено реализовать различные архитектуры моделей как отдельные модули на языке Python, чтобы их замена была максимально проста для пользователя. Их было решено поделить на несколько версий, так как некоторые реализации основных моделей потенциально не могут быть использованы с некоторыми реализациями алгоритмов постобработки.

Предложенная выше архитектура модели была разработана и реализована в формате

двух различных модулей, соответствующих двум версиям модели. Сами модели идентичны по своей архитектуре и способу обучения, однако отличаются тем, как они возвращают информации о классах на видео.

## 4 Алгоритм постобработки

Цель работы алгоритма постобработки состоит в том, чтобы оставлять меньше слишком длительных сегментов видео, чтобы поддерживать динамику, но и сохранять при этом стабильность и плавность в переходах между сегментами, и другую потенциально интересную информацию для зрителя, которую основная модель могла определить в другой класс. Для работы алгоритма постобработки требуется предоставить примерные значения интереса для каждого из классов выбранные эмпирическим путём.

Основные модели первой и второй версий возвращают разные форматы ответов о классах на видео, поэтому под каждую из версий требуются свои реализации алгоритмов постобработки.

Первая версия основной модели возвращает класс каждого из кадров на основе пяти кадров идущих ранее и пяти кадров идущих позже. Для неё написано два алгоритма постобработки - на основе BSpline и на основе плавающего окна усреднения.

Вторая версия основной модели возвращает классы для окон взятых через определённый шаг, для неё написан всего один алгоритм постобработки, который усредняет интерес классов, которые модель вернула для каждого кадра. Так как шаг может быть любым числом от одного до размера окна, модель может возвращать несколько классов для некоторых кадров. Качество получаемых видеороликов, согласно собранным отзывам получается не хуже, чем для первой версии модели.

## 5 Усовершенствование кода основной модели для ускорения процесса обработки

Большая часть времени разработки была потрачена на оптимизацию модели для ускорения ее работы, так как изначально была поставлена задача локального исполнения. Для оценки скорости работы модели было введено обозначение в процентах от времени исходного видео, где 50% означает, что на обработку 60-минутного видеофайла модель потратит 30 минут. В сравнении с человеческим показателем, который находится в диапазоне 150-300%.

В самом начале разработки, модель выдавала показатель в 80%. Анализ нагрузки на вычислительные мощности показал, что решение было не оптимизировано, что приводило к дополнительным затратам времени на передачу данных и выполнение операций. Основной упор был сделан на скорость передачи данных с процессора на видеокарту.

Первым улучшением стала замена чисел с плавающей точкой. Каждый кадр представлялся как набор чисел с плавающей точкой, что приводило к общему размеру одного пакета данных в районе 32 ГБ для 1024 наборов кадров. Это очень большой объем данных, который занимает большое время на передачу по шине PCI-e. Глубина цвета в большинстве случаев составляет 1 байт. Поэтому числа с плавающей точкой были заменены на 1-байтные числа, что позволило сократить затраты памяти до 4 ГБ для 1024 наборов кадров и время работы до 50%. Показатели уже были достаточны, чтобы сказать, что программа могла запускаться на пользовательском оборудовании.

Второе улучшение было основано на понимании того что центральные процессоры совершают действия над матрицами медленнее, чем графические. Поэтому было решено перенести часть предобработки, а именно шаги транспонирования наборов кадров на видеокарту - это не уменьшило затраты по памяти, однако позволило более эффективно использовать оборудование. Финальное время работы модели стало 28%.

Дальнейшие исследования показали, что без смены подхода к интерференсу модели ускорить её время работы не получится. Поэтому была разработана вторая версия модели, которая возвращает информацию о классах окон, взятых через указанный шаг. Вторая версия модели работает быстрее первой, однако ускорение будет зависеть от размера шага. Для шага в 1 кадр мы не увидим разницы с первой версией модели, а для шага в 5 кадров, время работы модели составляет 7%.

Все метрики были собраны в процессе выполнения модели на видеокарте. Тестирование последней версии модели с окном в 11 кадров и шагом в 5 на процессоре показало, что скорость работы составляет 14% от времени исходного видеофайла. Это означает, что минимальные требования к аппаратному обеспечению для данного приложения не включают наличие видеокарты. Однако, следует отметить, что данные показатели могут варьироваться в зависимости от конкретных характеристик компонентов компьютера. Список компонентов ПК, на котором проводилось тестирование, приведен в Приложениях.

После этого видео были размещены в социальную сеть Tiktok для анализа отклика аудитории. Видео набрали количество просмотров и реакций больше, чем созданные вручную. Отдельно собранная фокус-группа, отметила, что сгенерированное видео в целом имеет сходство с видеоматериалами, смонтированными человеком.

## 6 Заключение

Цель данной курсовой работы состояла в разработке приложения, способного автоматически обрабатывать большие объемы видеоматериала и выделять из них наиболее подходящие фрагменты для создания коротких видеороликов.

Для достижения поставленной цели была разработана архитектура, произведена разметка первоначального датасета, обучены модели, разработаны несколько прототипов алгоритма постобработки, проведены тесты, написано приложение, проведена оптимизация, проведено тестирование работы на основе игровых сессий, проведен контроль качества полученных видеороликов, оценен уровень интереса со стороны зрителей.

Разработанное в рамках данной курсовой работы приложение значительно сокращает процесс отбора и видеомонтажа материала. За один час оно способно обработать более 14 часов видеозаписей игровых сессий Minecraft. Это даёт предложенному методу потенциал для дальнейшего развития в области автоматизированного видеомонтажа не только для игровых видео, но и для разнообразных других типов контента.

## Список литературы

- [1] Luowei Zhou, Chenliang Xu и Jason J. Corso. *Towards Automatic Learning of Procedures from Web Instructional Videos*. 2017. arXiv: [1703.09788 \[cs.CV\]](#).
- [2] Du Tran, Heng Wang, Lorenzo Torresani и Matt Feiszli. *Video Classification with Channel-Separated Convolutional Networks*. 2019. arXiv: [1904.02811 \[cs.CV\]](#).
- [3] Christoph Feichtenhofer, Haoqi Fan, Jitendra Malik и Kaiming He. *SlowFast Networks for Video Recognition*. 2019. arXiv: [1812.03982 \[cs.CV\]](#).
- [4] Andrej Karpathy, George Toderici, Sanketh Shetty, Thomas Leung, Rahul Sukthankar и Li Fei-Fei. “Large-scale Video Classification with Convolutional Neural Networks”. В: *CVPR*. 2014.
- [5] Wenhao Wu, Zhun Sun и Wanli Ouyang. *Revisiting Classifier: Transferring Vision-Language Models for Video Recognition*. 2023. arXiv: [2207.01297 \[cs.CV\]](#).
- [6] Shen Yan, Xuehan Xiong, Anurag Arnab, Zhichao Lu, Mi Zhang, Chen Sun и Cordelia Schmid. *Multiview Transformers for Video Recognition*. 2022. arXiv: [2201.04288 \[cs.CV\]](#).

## 7 Приложения

### 7.1 Приложение А

Описание характеристик систем, на которых были проведены тестирования основной модели.

Таблица 7.1: Характеристики систем, на которых были проведены тесты скорости работы основной модели. V1 и V2 - обозначают две версии основной модели первую и вторую соответственно

	OS	CPU	GPU	Скорость V1	Скорость V2
Основная система	Ubuntu 20.04	Ryzen 5500x	Radeon VII	28%	7%
ПК	Windows 10	Ryzen 5600x	RTX 3080 12 GB	17%	7%
ПК	Windows 10	Ryzen 5600x	—	50%	14%

## 7.2 Приложение Б

Документация по использованию приложения

### Подготовка датасета для обучения модели

Подготовка датасета включает в себя создание набора видеодорожек, каждая из которых представляет определенный класс. Для этого необходимо вручную аннотировать исходные видеофайлы. В процессе просмотра видеоматериала выбираются и извлекаются фрагменты, соответствующие заданным классификационным категориям. Извлеченные фрагмен-

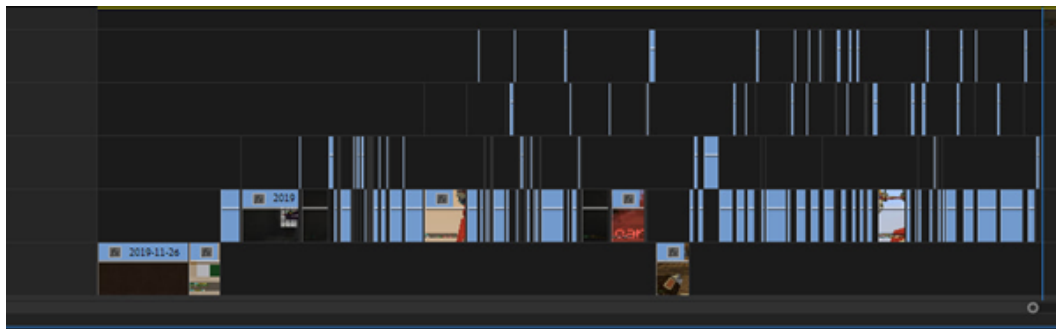


Рис. 7.1: Интерфейс видеоредактора, в котором подготовлен датасет для модели.

ты размещаются на отдельных видеодорожках в соответствии с их классификацией. При этом важно соблюдать последовательность фрагментов и размещать их строго вверх на дорожке, соответствующей классу, к которому они относятся, без каких-либо сдвигов по временной оси (узкие синие вертикальные полосы на Рисунке 7.1). Пространство на дорожках между извлеченными фрагментами остается незаполненным и не требует дополнительной обработки. Такой подход обеспечивает простую для пользователя организацию видеоматериала.

Все видеодорожки должны быть экспортированы в формате MP4. При экспорте каждого видео следует добавить к его названию порядковый номер, чтобы обеспечить корректную идентификацию и последующую обработку файлов. Нумерация должна быть последовательной и начинаться с нуля. Количество видеофайлов в датасете должно соответствовать количеству классов. После выполнения этих шагов датасет будет готов к использованию для обучения моделей.

### Подготовка приложения к работе

Для начала работы необходимо выбрать и установить модуль основной модели и подходящий к нему алгоритм постобработки. Модели и алгоритмы предоставляются на платформе GitHub в виде отдельных модулей на языке Python, которые можно устанавливать и использовать внутри приложения. В приложении можно выбрать используемые модули, если их установлено более одного.



При запуске программы, она в автоматическом режиме подготовит файлы датасета из представленных видео, обучит модель, сохранит её, для дальнейшего переиспользования и обработает видео. Если какой-либо шаг будет выполнен, приложение пропустит его и перейдёт к следующему, так например, если модель обучена, то приложение перейдёт к обработке видеофайла.