

Содержание

ВВЕДЕНИЕ	2
ПОСТАНОВКА ЗАДАЧИ НА РАЗРАБОТКУ	4
1.1. Актуальность разрабатываемого приложения	4
1.2. Постановка задачи	4
1.3. Описание предметной области	5
1.4. Обзор и анализ аналогов программного продукта	6
1.4.1. H264, MPEG-4	6
1.4.2. H.265	11
1.4.3. MPEG-2	14
1.4.1 Применение МАИ для анализа представленных аналогов	15
1.5. Назначение разработки	20
1.5.1. Требования к программному изделию	20
1.5.2. Требования к программной документации	21
1.5.3. Порядок контроля и приемки	21
2. ПРОЕКТИРОВАНИЕ СИСТЕМЫ	23
2.1. Обзор и анализ языков программирования	23
2.1.1. Java	23
2.1.2 C++	24
2.1.3. C#	25
2.1.4. Python	25
2.2. Применение МАИ для анализа инструментария	26
2.3 Обоснование выбора инструментария	31
2.4 Сведения о нейронных сетях	32
8.1. Сведения о вычислении оптического потока	39
4. РЕЗУЛЬТАТЫ ТЕСТИРОВАНИЯ	45
4.1. Тестирование программного комплекса	45
ЗАКЛЮЧЕНИЕ	47
СПИСОК ЛИТЕРАТУРЫ	48

					<i>СОДЕРЖАНИЕ</i>			
<i>Изм.</i>	<i>Лист</i>	<i>№ докум.</i>	<i>Подпись</i>	<i>Дата</i>	Разработка программного комплекса оптимизации хранения информации в системах видеонаблюдения с использованием нейронных сетей	<i>Лит.</i>	<i>Лист</i>	<i>Листов</i>
Разраб.		Малахов Д.В.						
Руковод.		Немцев А.О.					5	2
Консульт.		Немцев А.О.				ГФ БГТУ гр. ВТ-41		
Н. Контр.		Полунин А.И.						
Зав. каф.		Поляков В.М.						

ВВЕДЕНИЕ

Постепенно все оборудование для изготовления видео и телевизионных программ в сфере становится цифровым. Почему это происходит? Вовсе не из-за того, что цифровые технологии всех очаровывают и что качество изображения всецело улучшается, первопричины подобного перехода - первоначально экономические, ибо в настоящее время и цена, и стоимость эксплуатации цифрового программного комплекса меньше, чем аналогового с такими же функциями. Сейчас, когда цифровая техника достигла некоторой зрелости, аналоговые установки внезапно оказались менее эффективным, менее надежным и куда менее рентабельным. В настоящее время пересматриваются сами основы видеопроизводства и доставки изображения зрителю, вдобавок скорость развития технологий в нынешнюю - цифровую - эпоху значительно выше, чем в аналоговую. Сейчас, на пример, формат изображения 4: 3 плавно вытесняется форматом 16: 9, чересстрочная развертка - современной (построчной), а изменение стандартов кодирования увеличивает характеристики изображения. Экономика устанавливает и необходимость сжатия видеоинформации. Естественно, все сжатие портит качество визуализации и применяется не от хорошей жизни, а по необходимости. Не стоит думать, что раньше не использовали сжатия видеоинформации и что появилось оно только с перехода на цифровой формат. Нет, сжатие использовали всегда, но только раньше оно было аналоговым, а сегодня надо делать его полностью цифровым, убирая, где это возможно, двойное, то есть и аналоговое, и цифровое, сжатие. Текущие технологии цифрового сжатия, которое подошло на смену аналоговому, особенно в сочетании с компьютерными технологиями, позволяют не только улучшить качество самого видео, но и расширить возможности

					<i>Введение</i>			
<i>Изм.</i>	<i>Лист</i>	<i>№ докум.</i>	<i>Подпись</i>	<i>Дата</i>	<i>Разработка программного комплекса оптимизации хранения информации в системах видеонаблюдения с использованием нейронных сетей</i>	<i>Лит.</i>	<i>Лист</i>	<i>Листов</i>
<i>Разраб.</i>		<i>Малахов Д.В.</i>						
<i>Руковод.</i>		<i>Немцев А.О.</i>					6	2
<i>Консульт.</i>		<i>Немцев А.О.</i>				<i>ГФ БГТУ гр. ВТ-41</i>		
<i>Н. Контр.</i>		<i>Полунин А.И.</i>						
<i>Зав. каф.</i>		<i>Поляков В.М.</i>						

видеопроизводства и оптимизировать просмотр аудиовизуальной продукции Данная дипломная работа посвящена созданию программы для сжатия видеоинформации путем вычисления оптического потока и отсекающей статической части видеофайла. В ходе дипломной работы был проведен анализ существующего программного обеспечения, были поставлены требования к программному средству, выбран метод решения поставленной задачи, проведена программная реализация, тестирования и отладка программного средства.

					Введение	Лист
						7
Изм.	Лист	№ докум.	Подпись			

1. ПОСТАНОВКА ЗАДАЧИ НА РАЗРАБОТКУ

1.1. Актуальность разрабатываемого приложения

Проблема нахождения оптического потока между двумя изображениями (соседними кадрами видео) содержится в построении векторного поля подобного размера, вдобавок будет совпадать с вектором видимого перемещения пикселя с первоначального изображения до второго.

Вычислив такую векторную матрицу между соседними кадрами видео, мы получим цельную картину того, как перемещались те или иные объекты на нем. Иными словами, это задача отслеживания всех пикселей на видео. используется оптический поток безгранично широко — в задачах обнаружения движения, например, подобное векторное поле позволяет сосредоточиться на движениях, происходящих на видео и оставить контекст .

Тут есть пространство определенным неоднозначностям — что именно считать видимым перемещением с точки зрения математики? Обычно, подразумевают что значения пикселей переходят из одного кадра в последующий без изменений, другими словами: оптический поток демонстрирует куда переместился данный пиксель в последующем кадре

1.2. Постановка задачи

Следует разработать программный комплекс для распознавания оптического потока в системах видео наблюдения для последующего отбрасывания контекста

При реализации программного проекта необходимо следует решить ряд задач, таких как:

					<i>Постановка задачи на разработку</i>						
<i>Изм.</i>	<i>Лист</i>	<i>№ докум.</i>	<i>Подпись</i>	<i>Дата</i>							
<i>Разраб.</i>		<i>Малахов Д.В.</i>			<i>Разработка программного комплекса оптимизации хранения информации в системах видеонаблюдения с использованием нейронных сетей</i>	<i>Лит.</i>	<i>Лист</i>	<i>Листов</i>			
<i>Руковод.</i>		<i>Немцев А.О.</i>									
<i>Консульт.</i>		<i>Немцев А.О.</i>					8	19			
<i>Н. Контр.</i>		<i>Полунин А.И.</i>				<i>ГФ БГТУ гр. ВТ-41</i>					
<i>Зав. каф.</i>		<i>Поляков В.М.</i>									

- 1) Проанализировать предметную область;
- 2) Проанализировать аналоги разрабатываемой системы;
- 3) Выбрать инструментальные средства разработки;
- 4) Разработать модульную структуру системы;
- 5) Разработать пользовательский интерфейс;
- 6) Проверить работоспособность программного обеспечения;

1.3. Описание предметной области

В настоящее время для вычисления оптического потока используются следующие подходы:

1. Стандартный подход (метод Лукаса-Канаде).

Главное уравнение оптического потока включает две неизвестных и не может быть однозначно разрешено. метод Лукаса — Канаде минует спорность за счет применения информации о соседних пикселях во всех точках. Алгоритм базируется на гипотезе, что в локальной окрестности любого пикселя вес оптического потока одинаков, таким образом можно записать главное уравнение оптического потока для всех пикселей окрестности и решить получившуюся систему уравнений методом наименьших квадратов.

2. Метод Horn–Schunck

Алгоритм Хорна-Шунка предполагает плавный поток вокруг всего изображения. Поэтому он пытается минимизировать искажения потока и предпочитает решения, которые демонстрируют большую гладкость.

Поток формулируется как глобальная энергетический функционал, который затем стремятся минимизировать. Эта функциональность может быть сведена к минимуму путем решения связанных многомерных уравнений Эйлера — Лагранжа.

3. Применение искусственных нейронных сетей. Данный метод требует либо большого числа образцов задачи распознавания (с верными

					Постановка задачи на разработку	Лист
						9
Изм.	Лист	№ докум.	Подпись			

ответами), либо особенной структуры нейронной сети, учитывающей специфику данной задачи.

На текущий момент все прогрессивные системы видеонаблюдения так или иначе являются цифровыми, то есть в конечном виде данные всегда имеют числовое представление. В связи с этим для оптимального хранения и передачи по сети бесспорно применяется сжатие видео по определённым алгоритмам.

1.4. Обзор и анализ аналогов программного продукта

1.4.1. H264, MPEG-4

MPEG-4 Part 10 или AVC — лицензируемый стандарт сжатия видео, разработанный для достижения высочайшего уровня сжатия видеопотока при сохранении высокого качества. Стандарт H. 264 / AVC / MPEG-4 Part 10 охватывает ряд возможностей, позволяющих существенно увеличить результативность сжатия видео по сравнению с предшественниками (такими, как ASP), обеспечивая ещё большую гибкость использования в всевозможных сетевых средах. Главные из них: Многокадровое предсказание: Применение сжатых прежде кадров в качестве основных значительно более гибко, чем в ранних стандартах. Разрешается применение до 32 ссылок на иные кадры, тогда как в ASP и более ранних количество ссылок ограничено одним или, в случае В-кадров, парой кадров. Это увеличивает результативность кодирования, так как разрешает программисту подбирать для компенсации движения части из большего количества кадров. В большинстве сцен предоставленная функция гарантирует не очень крупную прибавку в качестве и не даёт видимого снижения битрейта. Однако, для отдельных сцен, например, с плотными повторяющимися участками, возвратно-поступательным движением и т. этот подход при сохранении качества позволяет сильно уменьшить расходы битрейта. Свобода порядка воссоздания изображений и порядка основных кадров. В предшествующих стандартах ставилась жёсткая связь между порядком кадров для использования при компенсации перемещения и порядком кадров при воспроизведении. В новом

					<i>Постановка задачи на разработку</i>	<i>Лист</i>
<i>Изм.</i>	<i>Лист</i>	<i>№ докум.</i>	<i>Подпись</i>			<i>10</i>

стандарте данные ограничения в внушительной мере ликвидированы, что позволяет программисту выбрать ход кадров для компенсации перемещения и для воспроизведения с большим уровнем гибкости, которая ограничена исключительно размером памяти, который допускает декодирование. Исключение лимитирования вдобавок позволяет в ряде случаев аннулировать лишнюю задержку, прежде сопряженную с двунаправленным предсказанием. самостоятельность способов обработки кадров и возможности их применения для предсказания перемещений. В предыдущих стандартах изображения, кодированные с применением отдельных технологий (например, двунаправленного предсказания), не могли применяться в качестве основных для предсказания движения прочих кадров видеопоследовательности. Ликвидируя такое ограничение, новый стандарт гарантирует программисту огромную податливость и, во многих случаях, вероятность пользоваться для предсказания перемещений в кадрах, ближе к кодируемому.

Компенсация перемещений с изменяемым размером блока позволяет очень точно определять зоны перемещений. Векторы перемещений, выступающие за края кадров. В MPEG-2 и предыдущих ему стандартах векторы перемещений могли ориентировать исключительно на пиксели, находящиеся в рамках декодированного основного изображения. технология экстраполяции за края кадра, возникнувшая как функция в Н. 263, интегрирована в новый стандарт. Шеститочечная фильтрация компонента яркости для полупиксельного предсказания с задачей снижения рваности краев и, в окончательном результате, предоставления увеличенной четкости кадра. Достоверность до четверти пикселя при компенсации перемещения гарантирует большую верность описания перемещающихся участков (что исключительно важно для медленного перемещения). Цветность, как правило, держится с разрешением, уменьшенным двукратно по вертикали и горизонтали (прореживание цвета), следовательно компенсация перемещения для компонента цветности применяет точность в одну восьмую пикселя цветности. Взвешенное предсказание, позволяющее применять

					Постановка задачи на разработку	Лист
Изм.	Лист	№ докум.	Подпись			11

масштабирование и сдвиг после компенсации перемещения на значения, указанные программистом. Подобный метод может очень сильно увеличить действенность кодирования для сцен с переменами освещённости, например при эффектах затемнения, градационного появления изображения. Пространственное предсказание от краев соседних частей для I-кадров (в отличие от предсказания исключительно коэффициента модификации в H. 263+ и MPEG-4 Part 2, и дискретно-косинусного коэффициента в MPEG-2 Part 2). Новая технология экстраполяции краёв прежде декодированных частей текущего изображения увеличивает качество сигнала, используемого для предсказания. Сжатие блоков без ущерба: Способ представления блоков без потерь в PCM, при котором видеоданные представлены непосредственно, дающий наверняка характеризовать определенные зоны и позволяющий жесткий лимит на количество закодированной информации для любого блока. Улучшенный метод представления блоков без потерь, позволяющий наверняка характеризовать области, при этом обычно тратят кардинально меньше битов, чем PCM. Гибкие методы чересстрочного сжатия : Приспособленное к кадру кодирование границ (PAFF), позволяющее кодировать отдельное изображение как кадр или как пару блоков (полукадров) — в зависимости от отсутствия\наличия движения. Приспособленное к блокам кодировка границ (MBAFF), позволяющее свободно кодировать любую вертикальную пару блоков (блок 16×32) как современные или чересстрочные. Позволяет использовать блоки 16×16 в режиме разбиения на границы (сравните с 16×8 полублоками в MPEG-2). Почти всегда успешнее PAFF. Новые методы преобразования: Точное целочисленное преобразование пространственных блоков 4×4 (концептуально похожее обширно известному DST, однако упрощенное и способное гарантировать точное декодирование), позволяющее точное расположение разностных сигналов с минимальным шумом, зачастую появляющегося в предыдущих кодеках.

Точное целочисленное преобразование пространственных блоков 8×8 (концептуально подобное широко известному DST, но упрощенное и

					Постановка задачи на разработку	Лист
Изм.	Лист	№ докум.	Подпись			12

способное обеспечить точное декодирование; поддерживается не во всех профилях), обеспечивающее большую эффективность сжатия схожих областей, чем 4×4 .

Адаптивный выбор кодеком между размерами блока 4×4 и 8×8 (поддерживается не во всех профилях).

Дополнительное преобразование Адамара, применяемое к дискретно-косинусным коэффициентам основного пространственного преобразования (к коэффициентов яркости, и, в особом случае, цветности) для достижения большей степени сжатия в однородных областях.

Квантование:

Логарифмическое управление длиной шага для упрощения распределения битрейта кодером и упрощенного вычисления обратной длины квантования.

Частотно-оптимизированные матрицы масштабирования квантования, выбираемые кодером для оптимизации квантования на основе человеческих особенностей восприятия (поддерживается не во всех профилях).

Внутренний фильтр деблокинга в цикле кодирования, устраняющий артефакты блочности, часто возникающие при использовании основанных на DCT техниках сжатия изображений.

Энтропийное кодирование квантованных коэффициентов трансформации:

Context-adaptive binary arithmetic coding (CABAC, контекстнозависимое адаптивное бинарное арифметическое кодирование) — алгоритм сжатия без потерь для синтаксических элементов видеопотока на основе вероятности их появления. Поддерживается только в Main Profile и выше. Обеспечивает более эффективное сжатие, чем CAVLC, но требует значительно больше времени на декодирование.

Context-adaptive variable-length coding (CAVLC, контекстнозависимое адаптивное кодирование с переменной длиной кодового слова) — альтернатива CABAC меньшей сложности. Тем не менее, оно сложнее и

					Постановка задачи на разработку	Лист
Изм.	Лист	№ докум.	Подпись			13

эффективнее, чем алгоритмы, применяемые для тех же целей в более ранних технологиях сжатия видео (как правило это алгоритм Хаффмана).

Часто используемое, простое и высоко структурированное кодирование словами переменной длины многих элементов синтаксиса, не закодированных CABAC или CAVLC, известное как коды Голомба (экспоненциальное кодирование Голомба).

Функции устойчивости к ошибкам:

Определение уровня сетевой абстракции (NAL), позволяющее использовать один и тот же синтаксис видео в различных сетевых окружениях, включая наборы параметров последовательности (sequence parameter sets, SPSs) и наборы параметров изображения (picture parameter sets, PPSs), которые обеспечивают большую надёжность и гибкость, чем предыдущие технологии.

Гибкое упорядочивание макроблоков (FMO), также известное как группы частей (поддерживается не во всех профилях) и произвольное упорядочивание частей (ASO) — методы реструктурирования порядка представления фундаментальных областей (макроблоков) в изображениях. При эффективном использовании гибкое упорядочивание макроблоков может существенно повысить устойчивость к потере данных.

Благодаря ASO, так как каждая часть изображения может быть декодирована независимо от других (при определённых ограничениях кодирования), новый стандарт позволяет посылать и получать их в произвольном порядке друг относительно друга. Это может снизить задержку в приложениях реального времени, особенно при использовании на сетях, имеющих режим работы доставка вне очереди. Эти функции могут также использоваться для множества других целей помимо восстановления ошибок.

Разбиение данных — функция, обеспечивающая разделение данных разной важности (например, векторы движения и другая информация предсказания имеет большую значимость для представления видеоконтента) по разным пакетам данных с разными уровнями защиты от ошибок (поддерживается не во всех профилях).

					Постановка задачи на разработку	Лист
Изм.	Лист	№ докум.	Подпись			14

Избыточные части. Возможность посылки кодером избыточного представления областей изображений, позволяя воспроизвести области изображений (обычно с некоторой потерей качества), данные о которых были потеряны в процессе передачи (поддерживается не во всех профилях).

Нумерация кадров, позволяющая создание «подпоследовательностей» (включая временное масштабирование включением дополнительных кадров между другими) а также обнаружение (и скрывание) потерь целых кадров при сбоях канала или пропаже пакетов.

1.4.2. H.265

H.265 / Высокоэффективное кодирование видео (HEVC) - это кодек-преемник H.264, который, как и H.264, разработан совместно группой экспертов ISO / IEC по движущимся изображениям и группой экспертов по кодированию видео ITU-T (VCEG). Основная цель нового кодека - повышение эффективности сжатия на 50 процентов по сравнению с H.264 и поддержка разрешений до 8192x4320. Стандартная рекомендация была разработана в ответ на растущий спрос на более высокие скорости сжатия движущихся изображений для широкого спектра приложений, таких как Интернет-потокковое вещание, передача данных, видеоконференции, цифровое хранилище и вещательное телевидение.

В качестве предыстории, ITU-T начал разработку преемника H.264 в 2004 году, в то время как ISO / IEC начал работу в 2007 году. В январе 2010 года группы сотрудничали в рамках совместного конкурса предложений, который завершился встречей MPEG и VCEG Совместная группа сотрудничества по кодированию видео (JCT-VC) в апреле 2010 года, на которой для кодека было принято название High Efficiency Video Coding (HEVC).

В октябре 2010 года JCT-VC подготовил первый рабочий проект спецификации, причем проект стандарта, основанный на восьми рабочих проектах спецификаций, был утвержден в июле 2012 года. 25 января 2013 года

					Постановка задачи на разработку	Лист
Изм.	Лист	№ докум.	Подпись			15

МСЭ объявил, что HEVC получил утверждение (согласие) на первом этапе Альтернативного процесса утверждения МСЭ-Т, в то время как MPEG объявила, что HEVC получил статус окончательного проекта международного стандарта (FDIS) в процессе стандартизации MPEG.

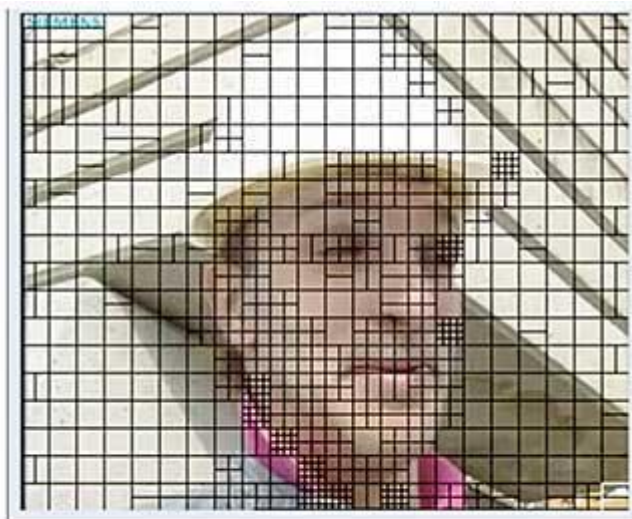
По сути, это означает, что первоначальные версии спецификации были заморожены, чтобы несколько поставщиков могли завершить работу над своими первыми продуктами HEVC.

HEVC будет продолжать развиваться, уже начавшаяся работа над расширениями для 12-битного видео и форматов цветности 4: 2: 2 и 4: 4: 4, а также включение в спецификации масштабируемого кодирования видео и 3D-видео.

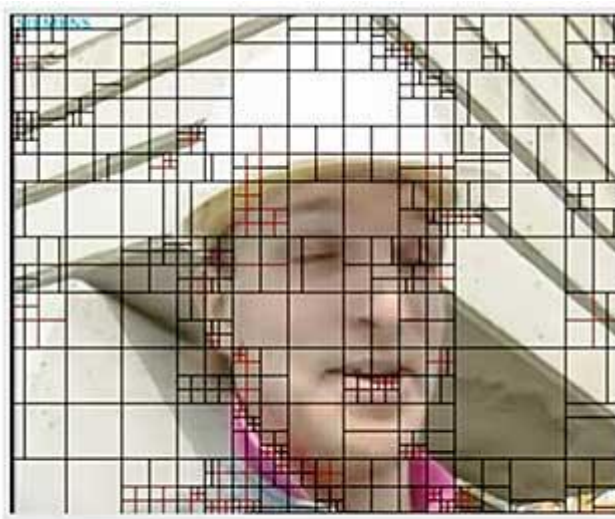
Подобно H.264 и MPEG-2, HEVC использует три типа кадров, I-, B- и P-кадры в группе изображений, включая элементы как межкадрового, так и внутрикадрового сжатия. HEVC включает в себя множество усовершенствований, в том числе:

Блоки дерева кодирования : там, где H.264 использует макроблоки с максимальным размером 16x16, HEVC использует блоки дерева кодирования или CTV с максимальным размером 64x64 пикселей. Блоки большего размера более эффективны при кодировании кадров большего размера, например при разрешении 4K. В результате тестов было показано, что по сравнению с кодированием блоков размером 64x64 пикселей битрейт увеличивается на 2,2 %, когда используются блоки размером 32x32, и увеличивается на 11,0 %, когда используются блоки размером 16x16. В тестах кодирования видео с разрешением 2560x1600 пикселей при использовании блоков размером 32x32 пикселей битрейт увеличивается на 5,7 %, а при использовании блоков размером 16x16 пикселей — на 28,2 % по сравнению с видео, где использованы блоки размером 64x64, при одинаковом пиковом отношении сигнал-шум. Это показано на рисунке 1.1.

					Постановка задачи на разработку	Лист
Изм.	Лист	№ докум.	Подпись			16



H.264

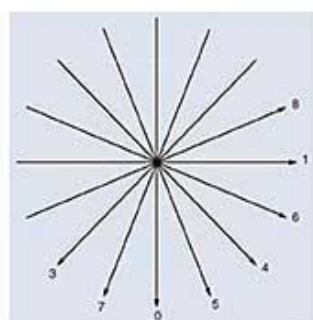


H.265

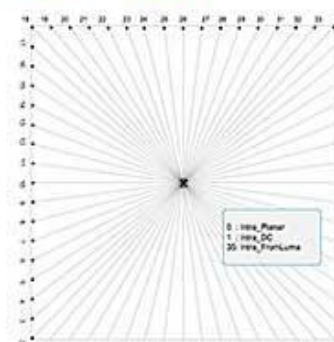
Рисунок 1.1 - Микроблоки алгоритмов H.264 и 265

Больше направлений внутреннего предсказания : там, где H.264 использует 9 направлений внутреннего предсказания, HEVC может использовать более 35, добавляя больше потенциальных блоков опорных пикселей, которые способствуют более эффективному внутрикадровому сжатию см. Рисунок 1.2. Очевидная стоимость - это дополнительное время кодирования, необходимое для поиска в дополнительных направлениях.

- **Luma: 35 prediction directions (33 + Planar + DC)**



H.264/AVC



HEVC

Рисунок 1.2. - Поиск расширен чтоб найти больше опорных пикселей.

Другие достижения включают.

- Адаптивное прогнозирование вектора движения, которое позволяет кодеку находить больше межкадровых избыточностей.
- Улучшенные инструменты распараллеливания, включая параллельную обработку Wavefront, для более эффективного кодирования в многоядерной среде
- Энтропийное кодирование только CABAC, больше не CAVLC
- Улучшения фильтра удаления блочности и создание второго фильтра под названием Sample Adaptive Offset, который дополнительно ограничивает артефакты по краям блока.

1.4.3. MPEG-2

MPEG-2-это название группы стандартов для цифрового кодирования видео и аудиосигналов, организации транспортных потоков видео и аудиоинформации и передачи связанной информации. Стандарты MPEG выдаются группой экспертов по движению на изображениях ISO / IEC (MPEG).

- Стандарт MPEG-2 получил распространение в цифровых видеодисках DVD, системах компрессии видеоизображений, цифровом телевидении DVB.
- В случае использования в цифровом телевидении MPEG-2 активно применяется как стандарт, определяющий структуру транспортных потоков и способы передачи данных.
- Стандарт содержит несколько подразделов (parts). Например, MPEG-2 part 1 определяет тип контейнера, например, может использоваться Transport Stream, который позволяет корректировать ошибки оборудования, принимающего сигнал. Part 2 — структуру компрессированного изображения (элементарный поток MPEG-2).
- Стандарт MPEG-2 намеренно не определяет способы компрессии изображения (звука), он лишь указывает, как должно быть оформлено сжатое изображение (звук). Стандарт не определяет, каким образом

					Постановка задачи на разработку	Лист
Изм.	Лист	№ докум.	Подпись			18

должен быть реализован кодер или декодер MPEG-2, он определяет только структуру данных. Это даёт возможность участникам рынка конкурировать друг с другом за создание более качественных устройств и алгоритмов.

- Использование стандартов MPEG-2 требует уплаты лицензионных отчислений держателям патентов через MPEG Licensing Association. Тексты стандартов MPEG-2 распространяются свободно, но не бесплатно.
- MPEG-2 используется для общего сжатия движущихся изображений и звука и определяет формат видеопотока, который может быть представлен как три типа кадра:
 - независимо сжатые кадры (I-кадры),
 - кадры, сжатые с использованием предсказания на основе предыдущих кадров (P-кадры) и
 - кадры, сжатые с использованием предсказания на основе предыдущих и последующих (B-кадры).
- Соответствующие группы кадров от одного I-кадра до другого образуют GOP (англ. Group Of Pictures) - группу кадров.
- Обычно используются потоки в 30 или 29,97 кадров в секунду.
- MPEG-2 поддерживает видео и в прогрессивной, и в чересстрочной развёртке.

1.4.1 Применение МАИ для анализа представленных аналогов

Для анализа всех представленных проектов воспользуемся методом анализа иерархий. Иерархия представлена на рисунке 1.3. Данные, рассчитываемые в методе, представлены в таблицах 1.1-1.7.

1. Какая программа является лучшей

Критерии:

2. Качество сжатия

3. Скорость кодирования

					Постановка задачи на разработку	Лист
Изм.	Лист	№ докум.	Подпись			19

4. Скорость декодирования
5. Объем требуемой памяти
6. Битрейт

Альтернативы:

7. H.264 Java
8. H.265 C++
9. MPEG-2 C#
10. Разрабатываемая программа Python

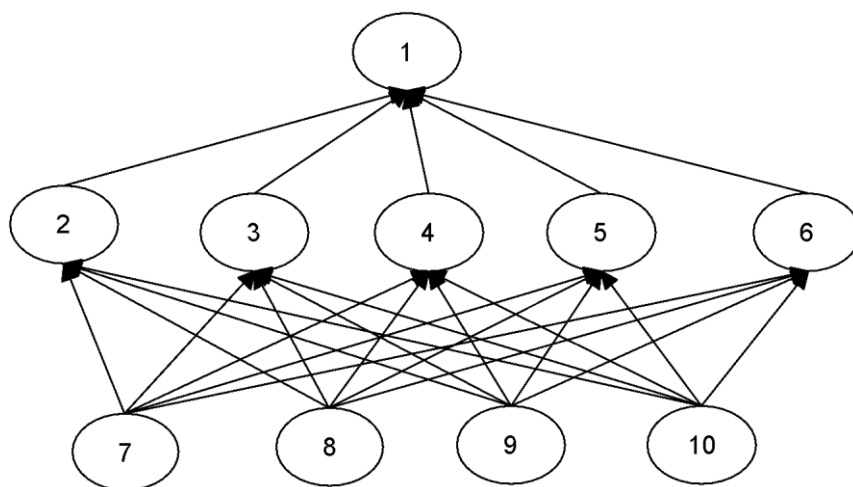


Рисунок 1.3 - Иерархия

Данный этап метода анализа иерархии основан на принципе дискриминации и сравнительных суждений

Матрицы интенсивности предпочтения представлены ниже:

В таблице 1.1 приведена матрица интенсивности предпочтений для сравнения с аналогами.

Таблица 1.1 – Выбор программы для сжатия видео

Программа	2	3	4	5	6	Произведение	Корень	Вектор
2	1,00	3,00	3,00	2,00	3,00	54,00	2,22	0,38
3	0,33	1,00	2,00	3,00	0,33	0,67	0,92	0,16
4	0,33	0,50	1,00	3,00	0,50	0,25	0,76	0,13
5	0,50	0,33	0,33	1,00	0,33	0,02	0,45	0,08

Таблица 1.1. – Продолжение

6	0,33	3,00	2,00	3,00	1,00	6,00	1,43	0,25
Сумма	2,50	7,83	8,33	12,00	5,17	60,94	5,78	1,00
Лямбда макс	0,96	1,25	1,09	0,93	1,28	5,52		
ИС	0,13							
ОС	11,50							

Приоритеты выстроились следующим образом: Качество сжатия> Битрейт> Скорость кодирования> Скорость декодирования> Объем требуемой памяти

Определим приоритеты для каждого из подкритериев 2-го уровня:

В таблице 1.2 представлены приоритеты для критерия «Качество сжатия»

Таблица 1.2 – Качество сжатия

Качество сжатия	7	8	9	10	Произведение	Корень	Вектор
7	1,00	0,11	0,20	0,14	0,00	0,24	0,04
8	9,00	1,00	3,00	0,33	9,00	1,73	0,29
9	5,00	0,33	1,00	0,20	0,33	0,76	0,13
10	7,00	3,00	5,00	1,00	105,00	3,20	0,54
Сумма	22,00	4,44	9,20	1,68	114,34	5,93	1,00
Лямбда макс	0,88	1,30	1,18	0,90	4,26		
ИС	0,09						
ОС	9,71						

Приоритеты критериев второго уровня для качество сжатия: Разрабатываемая программа> H.265> MPEG-2> H.264.

В таблице 1.3 представлены приоритеты для критерия «Скорость кодирования»

Таблица 1.3 – Скорость кодирования

Скорость кодирования	7	8	9	10	Произведение	Корень	Вектор
7	1,00	5,00	4,00	3,00	60,00	2,78	0,55
8	0,20	1,00	3,00	3,00	1,80	1,16	0,23
9	0,25	0,33	1,00	1,00	0,08	0,54	0,11
10	0,33	0,33	1,00	1,00	0,11	0,58	0,11
Сумма	1,78	6,67	9,00	8,00	61,99	5,06	1,00
Лямбда макс	0,98	1,53	0,96	0,91	4,38		
ИС	0,13						
ОС	14,03						

Приоритеты критериев второго уровня по скорости кодирования:
Н.264> Н.265> Разрабатываемая программа> MPEG-2

В таблице 1.4 представлены приоритеты для критерия «Скорость декодирования»

Таблица 1.4 – Скорость декодирования

Скорость декодирования	7	8	9	10	Произведение	Корень	Вектор
7	1,00	3,00	5,00	7,00	105,00	3,20	0,58
8	0,33	1,00	3,00	3,00	3,00	1,32	0,24
9	0,20	0,33	1,00	3,00	0,20	0,67	0,12
10	0,14	0,33	0,33	1,00	0,02	0,35	0,06
Сумма	1,68	4,67	9,33	14,00	108,22	5,54	1,00
Лямбда макс	0,97	1,11	1,13	0,90	4,10		
ИС	0,03						
ОС	3,71						

Приоритеты критериев второго уровня по скорости декодирования:
Н.264> Н.265> MPEG-2> Разрабатываемая программа

В таблице 1.5 представлены приоритеты для критерия «Объем требуемой памяти»

Таблица 1.5 – Объем требуемой памяти

Память	7	8	9	10	Произведение	Корень	Вектор
7	1,00	3,00	5,00	5,00	75,00	2,94	0,53
8	0,33	1,00	5,00	5,00	8,33	1,70	0,30
9	0,33	0,20	1,00	1,00	0,07	0,51	0,09
10	0,20	0,20	1,00	1,00	0,04	0,45	0,08
Сумма	1,87	4,40	12,00	12,00	83,44	5,60	1,00
Лямбда макс	0,98	1,34	1,09	0,96	4,37		
ИС	0,12						
ОС	13,53						

Приоритеты критериев второго уровня по объему требуемой памяти:

Н.264> Н.265> MPEG-2> Разрабатываемая программа

В таблице 1.6 представлены приоритеты для критерия «Битрейт»

Таблица 1.6 – Битрейт

Битрейт	7	8	9	10	Произведение	Корень	Вектор
7	1,00	0,11	0,11	0,20	0,002	0,22	0,04
8	9,00	1,00	0,33	1,00	3,00	1,32	0,25
9	9,00	3,00	1,00	1,00	27,00	2,28	0,43
10	5,00	1,00	1,00	1,00	5,00	1,50	0,28
Сумма	24,00	5,11	2,44	3,20	35,00	5,31	1,00
Лямбда макс	1,01	1,27	1,05	0,90	4,22		
ИС	0,07						
ОС	8,21						

Приоритеты критериев второго уровня по битрейту: MPEG-2> Разрабатываемая программа> Н.265> Н.264

Для принятия окончательного решение по выбору программы для сжатия на основе метода анализа иерархии, необходимо значения векторов приоритета из таблиц 1.1-1.6 по каждому фактору (критерию) рассчитаем глобальный коэффициент

Итоговый глобальный приоритет определяется путем суммирования векторов глобальных приоритетов по критериям.

По данным таблиц 1.1-1.6 определяется Глобальный приоритет (ГП) по каждой альтернативе, который представлен в таблице 1.7

Таблица 1.7 – Глобальный приоритет

Критерии	2	3	4	5	6	ГП
Программа	0,38	0,16	0,13	0,08	0,25	
7	0,04	0,55	0,58	0,53	0,04	0,23
8	0,29	0,23	0,24	0,30	0,25	0,26
9	0,13	0,11	0,12	0,09	0,43	0,20
10	0,54	0,11	0,06	0,08	0,28	0,31

Сравнивая полученные значения, определяют рейтинг программ для сжатия видео с учетом вариантов попарных сравнений, представленных в табл. 1.1-1.6. Высокий рейтинг будет соответствовать наибольшему значению глобального вектора приоритета.

Таким образом наибольший приоритет имеет разрабатываемая программа.

1.5. Назначение разработки

Наименование: Разработка программного комплекса оптимизации хранения информации в системах видеонаблюдения.

Область применения: Сфера контроля безопасности. Позволит эффективнее хранить архивы записей камер видеонаблюдения.

1.5.1. Требования к программному изделию

Программный комплекс для оптимизации хранения информации в системах видеонаблюдения будет обладать следующими функциональными возможностями:

- Вывод в реальном времени записи с камеры видеонаблюдения
- Запись и воспроизведение архивных записей камер

					Постановка задачи на разработку	Лист
Изм.	Лист	№ докум.	Подпись			24

Требования к надежности:

Для устойчивого и без коллизионного функционирования программы рекомендуется использование аппаратной платформы защищенной от повреждений электросети и нарушения параметров питающего напряжения, для чего рекомендуется использование оптимизированных блоков питания, а также источников резервного питания, емкость питающих элементов которых и функции могут адекватно отреагировать на возникшую проблему.

Условия эксплуатации программы совпадают с условиями эксплуатации ПК, на которых будет использоваться программа. Для обслуживания данного программного продукта требуется один человек, квалификация которого должна быть не ниже оператора ПК.

Требования к составу и параметрам технических средств:

- ПК, выполняющий роль сервера, на котором будет использоваться программа, должен иметь:
- Видеоадаптер, поддерживающий DirectX 11.0
- Оперативная память не менее 2 Гб
- Наличие сетевой платы с разъемом Ethernet

Требования к информационной и программной совместимости:
операционная система UNIX системы

1.5.2. Требования к программной документации

Программа должна включать справочную информацию. В состав сопровождающей документации должны входить пояснительная записка, содержащая описание процесса разработки, руководство пользователя, пример работы

1.5.3. Порядок контроля и приемки

1. Испытания программы должны проводиться на объекте Заказчика.
2. Контроль и приемка разработки осуществляются на предоставляемой Заказчиком технике.
3. Испытания программы должны проводиться согласно разработанной

					Постановка задачи на разработку	Лист
Изм.	Лист	№ докум.	Подпись			25

Исполнителем и согласованной Заказчиком Программы и методик испытаний.

4. При проведении испытаний проверяется выполнение всех функций программы.

5. Ход проведения испытаний Заказчик и Исполнитель документируют в Протоколе проведения испытаний.

6. На основании Протокола проведения испытаний Исполнитель совместно с Заказчиком подписывают Акт приемки-сдачи программы в эксплуатацию.

Выводы

В процессе выполнения этого этапа были определены цели реализации данного дипломного проекта, определены 3 аналога разрабатываемой программы. В результате анализа аналогов была разработана техническая задача разработки программного обеспечения для оптимизации хранения информации в системах видеонаблюдения. Оценивая проект, который будет завершен выполнен по техническому заданию, с аналогами, методом анализа иерархий, было установлено, что разрабатываемый проект будет достойным конкурентом для его аналогов и даже превзойдет их с точки зрения степени сжатия видеопотока. Был определен список задач, которые необходимо решить в ходе дипломного проекта.

					<i>Постановка задачи на разработку</i>	Лист
						26
Изм.	Лист	№ докум.	Подпись			

2. ПРОЕКТИРОВАНИЕ СИСТЕМЫ

2.1. Обзор и анализ языков программирования

2.1.1. Java

Java - это язык программирования высокого уровня, первоначально разработанный Sun Microsystems и выпущенный в 1995 году. Java работает на различных платформах, таких как Windows, Mac OS и различные версии UNIX.

- **Объектно-ориентированный** - в Java все является объектом. Java можно легко расширить, поскольку она основана на объектной модели.
- **Независимость платформы** от - в отличие от многих других языков программирования, включая C и C ++, при компиляции Java она не компилируется в платформенно-зависимую машину, а в независимый от платформы байт-код. Этот байт-код распространяется по сети и интерпретируется виртуальной машиной (JVM) на какой бы платформе он ни работал.
- **Просто** - Java разработана таким образом, чтобы ее было легко изучить. Если вы понимаете основную концепцию ООП Java, освоить ее будет несложно.
- **Безопасность** - с помощью функции безопасности Java она позволяет разрабатывать системы, свободные от вирусов и несанкционированного доступа. Методы аутентификации основаны на шифровании с открытым ключом.
- **Не зависящий от архитектуры** - компилятор Java генерирует не зависящий от архитектуры формат объектного файла, что делает скомпилированный код исполняемым на многих процессорах при наличии системы времени выполнения Java.

					Проектирование системы			
Изм.	Лист	№ докум.	Подпись	Дата	Разработка программного комплекса оптимизации хранения информации в системах видеонаблюдения с использованием нейронных сетей	Лит.	Лист	Листов
Разраб.		Малахов Д.В.						
Руковод.		Немцев А.О.					27	19
Консульт.		Немцев А.О.				ГФ БГТУ гр. ВТ-41		
Н. Контр.		Полунин А.И.						
Зав. каф.		Поляков В.М.						

- **Переносимость** - независимость от архитектуры и отсутствие зависимых от реализации аспектов спецификации делает Java переносимой.

Компилятор на Java написан на ANSI C с чистой границей переносимости, которая является подмножеством POSIX.

- **Надежность** - Java прилагает усилия для устранения ситуаций, подверженных ошибкам, уделяя особое внимание проверке ошибок во время компиляции и проверке во время выполнения.

2.1.2 C++

Язык программирования C ++ - это язык программирования базового назначения, который можно обозначать по-разному, в том числе - статически типизированный, произвольный, мультипарадигмальный и скомпилированный. Бьярн Страуструп создал основу для C ++ в 1979 году, когда он работал в Bell Labs. Он имеет объектно-ориентированные функции, такие как классы, но также включает другие улучшения по сравнению с оригиналом. C ++ был улучшением по сравнению с исходным названием «C с классами», которое не имеет такого приятного звучания. C ++ (произносится как «си плюс плюс»), а также группа языков среднего уровня, потому что он имеет смесь функций, имеющихся как в языках высокого, так и в языках низкого уровня. Новое название было дано в 1983 году и представляет собой игру слов. Буквально это означает «Увеличить C на 1», имея в виду тот факт, что он лучше, чем C.

C ++ - один из наиболее широко распространенных языков программирования, нашедший свое применение во многих аппаратных средствах и платформах ОС. Известно, что он может эффективно компилировать собственный код и используется для разработки операционных систем, программирования встроенных систем, драйверов устройств, встроенного программного обеспечения, разработка настольных и

					Постановка задачи на разработку	Лист
Изм.	Лист	№ докум.	Подпись			28

кроссплатформенных приложений, разработка игр. Ряд групп предлагают как бесплатные, так и платные компиляторы языка программирования C ++, включая GNU Project, Microsoft, Intel и Embarcadero Technologies. Язык программирования C ++ оказал значительное влияние на другие популярные языки программирования, в частности C # и Java. Другие широко используемые и популярные языки, такие как Objective-C, используют заметно иной синтаксис и подход к добавлению классов в язык C ++

2.1.3. C#

C # (произносится C Sharp) - это объектно-ориентированный язык программирования. Он был разработан в 1998-2001 годах. Группа инженеров Microsoft под руководством Андерса Хейлсберга и Скотта Уилтаумота в качестве языка разработки приложений для платформы Microsoft .NET Framework. Затем он был стандартизирован как ECMA-334 и ISO / IEC 23270.

C # принадлежит к семейству языков с C-подобным синтаксисом, из которого его синтаксис ближе всего к C ++ и Java. Язык имеет статические типы данных, поддерживает полиморфизм, перегрузку оператора (включая явное и неявную типизацию), делегаты, атрибуты, события, переменные, свойства, общие типы и методы, итераторы, анонимные функции с поддержкой глобальных и локальных переменных, ссылка, исключения, комментарии в формате XML .

2.1.4. Python

Python - это язык программирования с несколькими парадигмами, который поддерживает императивное, процедурное, структурированное, объектно-ориентированное программирование, метапрограммирование и функциональное программирование. Общие проблемы программирования решаются с помощью динамического определения типов данных.

Ориентированное на аспект Программирование частично поддерживается декораторами, более полная поддержка обеспечивается дополнительными библиотеками. Такие методы, как контрактное программирование и

					Постановка задачи на разработку	Лист
Изм.	Лист	№ докум.	Подпись			29

логическое программирование, могут применяться с использованием библиотек или расширений. Основными архитектурными особенностями являются динамическим определением типов данных, автоматическое управление памятью, полная интроспекция, механизм обработки исключений, поддержка многопоточных вычислений с глобальной блокировкой интерпретатора (GIL), структуры данных высокого уровня. Поддерживается разделение программ на модули, которые, в свою очередь, могут быть объединены в пакеты

2.2. Применение МАИ для анализа инструментария

Для анализа вариантов применяемого инструментария воспользуемся методом анализа иерархий. Данные, рассчитываемые в методе, представлены в таблицах 2.1-2.7.

1. Какой язык программирования является лучшим

Критерии:

2. Скорость работы конечного продукта.

3. Скорость разработки программы.

4. Кроссплатформенность.

5. Скорость внесения изменений, скорость тестирования

6. Объем требуемой памяти.

Альтернативы:

7. Java

8. C++

9. C#

10. Python

Данный этап метода анализа иерархии основан на принципе дискриминации и сравнительных суждений

Матрицы интенсивности предпочтения представлены ниже:

В таблице 2.1 приведена матрица интенсивности предпочтений для выбора языка программирования.

					Постановка задачи на разработку	Лист
						30
Изм.	Лист	№ докум.	Подпись			

Таблица 2.1 – Выбор языка программирования

Выбор языка программирования	2	3	4	5	6	Произведение	Корень	Вектор
2	1,00	3,00	3,00	2,00	3,00	54,00	2,22	0,38
3	0,33	1,00	2,00	3,00	0,33	0,67	0,92	0,16
4	0,33	0,50	1,00	3,00	0,50	0,25	0,76	0,13
5	0,50	0,33	0,33	1,00	0,33	0,02	0,45	0,08
6	0,33	3,00	2,00	3,00	1,00	6,00	1,43	0,25
Сумма	2,50	7,83	8,33	12,00	5,17	60,94	5,78	1,00
Лямбда макс	0,96	1,25	1,09	0,93	1,28	5,52		
ИС	0,13							
ОС	11,50							

Приоритеты выстроились следующим образом: Скорость работы конечного продукта> Объем требуемой памяти> Скорость разработка> Объем занимаемой оперативной памяти> Кроссплатформенность

Определим приоритеты для каждого из подкритериев 2-го уровня:

В таблице 2.2 представлены приоритеты для критерия «Качество сжатия»

Таблица 2.2 – Скорость работы конечного продукта

Скорость работы конечного продукта	7	8	9	10	Произведение	Корень	Вектор
7	1,00	0,11	0,20	0,14	0,00	0,24	0,04
8	9,00	1,00	3,00	0,33	9,00	1,73	0,29
9	5,00	0,33	1,00	0,20	0,33	0,76	0,13

Таблица 2.2 – Продолжение

10	7,00	3,00	5,00	1,00	105,00	3,20	0,54
Сумма	22,0 0	4,44	9,20	1,68	114,34	5,93	1,00
Лямбда макс	0,88	1,30	1,18	0,90	4,26		
ИС	0,09						
ОС	9,71						

Приоритеты критериев второго уровня для Скорость работы конечного продукта: Python> C++> C#> Java.

В таблице 2.3 представлены приоритеты для критерия «Скорость разработки программы»

Таблица 2.3 – Скорость разработки программы

Скорость разработки программы	7	8	9	10	Произведение	Корень	Вектор
7	1,00	5,00	4,00	3,00	60,00	2,78	0,55
8	0,20	1,00	3,00	3,00	1,80	1,16	0,23
9	0,25	0,33	1,00	1,00	0,08	0,54	0,11
10	0,33	0,33	1,00	1,00	0,11	0,58	0,11
Сумма	1,78	6,67	9,00	8,00	61,99	5,06	1,00
Лямбда макс	0,98	1,53	0,96	0,91	4,38		
ИС	0,13						
ОС	14,03						

Приоритеты критериев второго уровня по Скорость разработки программы: Java> C++> Python> C#

В таблице 2.4 представлены приоритеты для критерия «Кроссплатформенность»

Таблица 2.4 – Кроссплатформенность

Кроссплатформенность	7	8	9	10	Произведение	Корень	Вектор
7	1,00	3,00	5,00	7,00	105,00	3,20	0,58
8	0,33	1,00	3,00	3,00	3,00	1,32	0,24
9	0,20	0,33	1,00	3,00	0,20	0,67	0,12
10	0,14	0,33	0,33	1,00	0,02	0,35	0,06
Сумма	1,68	4,67	9,33	14,00	108,22	5,54	1,00
Лямбда макс	0,97	1,11	1,13	0,90	4,10		
ИС	0,03						
ОС	3,71						

Приоритеты критериев второго уровня по Кроссплатформенности:

Java> C++> C#> Python

В таблице 2.5 представлены приоритеты для критерия «Скорость внесения изменений, скорость тестирования»

Таблица 2.5 – Скорость внесения изменений, скорость тестирования

Память	7	8	9	10	Произведение	Корень	Вектор
7	1,00	3,00	5,00	5,00	75,00	2,94	0,53
8	0,33	1,00	5,00	5,00	8,33	1,70	0,30
9	0,33	0,20	1,00	1,00	0,07	0,51	0,09
10	0,20	0,20	1,00	1,00	0,04	0,45	0,08
Сумма	1,87	4,40	12,00	12,00	83,44	5,60	1,00
Лямбда макс	0,98	1,34	1,09	0,96	4,37		
ИС	0,12						
ОС	13,53						

Приоритеты критериев второго уровня по объему требуемой памяти:

Python> C#> C++> Java

В таблице 2.6 представлены приоритеты для критерия «Объем требуемой памяти»

Таблица 2.6 – Объем требуемой памяти

Объем требуемой памяти	7	8	9	10	Произведение	Корень	Вектор
7	1,00	0,11	0,11	0,20	0,002	0,22	0,04
8	9,00	1,00	0,33	1,00	3,00	1,32	0,25
9	9,00	3,00	1,00	1,00	27,00	2,28	0,43
10	5,00	1,00	1,00	1,00	5,00	1,50	0,28
Сумма	24,00	5,11	2,44	3,20	35,00	5,31	1,00
Лямбда макс	1,01	1,27	1,05	0,90	4,22		
ИС	0,07						
ОС	8,21						

Приоритеты критериев второго уровня по Объему требуемой памяти.:

C#> Python> C++> Java

Для принятия окончательного решение по выбору языка программирования на основе метода анализа иерархии, необходимо значения векторов приоритета из таблиц 2.1-2.6 по каждому фактору (критерию) рассчитаем глобальный коэффициент

Итоговый глобальный приоритет определяется путем суммирования векторов глобальных приоритетов по критериям.

По данным таблиц 2.1-.6 определяется Глобальный приоритет (ГП) по каждой альтернативе, который представлен в таблице 1.7

Таблица 1.7 – Глобальный приоритет

Критерии	2	3	4	5	6	ГП
Языки программирова ния	0,38	0,16	0,13	0,08	0,25	

Таблица 1.7 – Продолжение

7	0,04	0,55	0,58	0,53	0,04	0,23
8	0,29	0,23	0,24	0,30	0,25	0,26
9	0,13	0,11	0,12	0,09	0,43	0,20
10	0,54	0,11	0,06	0,08	0,28	0,31

Сравнивая полученные значения, определяют рейтинг программ для сжатия видео с учетом вариантов попарных сравнений, представленных в табл. 2.1-2.6. Высокий рейтинг будет соответствовать наибольшему значению глобального вектора приоритета. Таким образом наибольший приоритет имеет Python.

2.3 Обоснование выбора инструментария

Прикладной уровень реализуется с помощью Python.

Достоинства Python:

Python-это объектно-ориентированный язык общего назначения, разработанный с целью повышения производительности разработчиков.

Низкий порог входа. Синтаксис этого языка легко выучить.

- *Логичный, лаконичный и понятный.* Если произвести сравнение с другими языками, то у Python свободно читаемый синтаксис. Данное свойство очень сильно помогает при работе в команде разработчиков, когда необходимо читать, либо поправлять чужой код.
- *Кроссплатформенный:* Подходит для различных платформ, также существуют реализации интерпретаторов для мобильных устройств и не популярных систем.
- *Широкое применение.* Данный язык используется для создания веб-приложений, машинного обучения, различных математических вычислений. Существует реализация под названием Micro Python, оптимизированная для работы на микроконтроллерах (вы можете писать инструкции, логику взаимодействия устройств, организовывать связь, внедрять умный дом).

- *Сильная поддержка гигантов ИТ-индустрии.* Такие компании, как *Google, Facebook, Dropbox, Spotify, Quore и Netflix*, использовали *Python* на определенных этапах разработки.
- *Высокая востребованность на рынке труда.* Библиотеки: Множество различных библиотек, модулем, которые существенно облегчают разработку приложения, то есть необходимо писать все подряд, а можно просто воспользоваться уже готовым решением для задачи.
- Для большинства задач: для веб-разработки, для скриптов, прототипирования, машинного обучения и работы с большими данными, — один из лучших языков.

Было принято решение работать на языке Python 3.8 в среде разработки PyCharm. Данная среда была выбрана по следующим причинам:

- 3.Среда легка для самостоятельного изучения;
- 4.Статический анализ кода, подсветка синтаксиса и ошибок;
- 5.Навигация по проекту и исходному коду: отображение файловой структуры проекта, быстрый переход между файлами, классами, методами и использованиями методов;
- 6.Наличие библиотек для глубокого обучения, таких как Tensor flow и Keras
- 7.Встроен отладчик для Python.
- 8.При написании какого-либо символа, сразу предлагается выбрать полное написание предполагаемой конструкции

2.4Сведения о нейронных сетях

Нейронная сеть - математическая модель, а также её программное или аппаратное воплощение, построенная по принципу организации и функционирования биологических нейронных сетей — сетей нервных клеток живого организма. Пример простейшей нейронной сети изображен на рисунке 2.1.

Сверточные нейронные сети (CNN) специально разработаны для обработки данных, состоящих из нескольких массивов / матриц, таких как

					Постановка задачи на разработку	Лист
						36
Изм.	Лист	№ докум.	Подпись			

изображение, состоящее из трех матриц в каналах RGB. Ключевой идеей CNN является операция свертки, которая заключается в использовании нескольких небольших ядер / фильтров для извлечения локальных функций путем перемещения по одному и тому же входу. Каждое ядро может выводить карту характеристик, и все карты характеристик объединяются вместе, это также известно как сверточный слой и является основным компонентом CNN. Обратите внимание, что эти составные карты могут быть дополнительно обработаны следующим слоем. Чтобы снизить вычислительные затраты, к этим картам функций обычно применяется операция объединения, такая как максимальное объединение. Типичная CNN обычно структурирована как серия слоев, включая несколько сверточных слоев и несколько полностью связанных слоев. Например, знаменитый LeNet состоит из двух сверточных слоев и трех полностью связанных слоев, и операция объединения используется после каждого сверточного слоя.

Искусственная нейронная сеть состоит из трех компонентов:

- Входной слой;
- Скрытые (вычислительные) слои;
- Выходной слой.

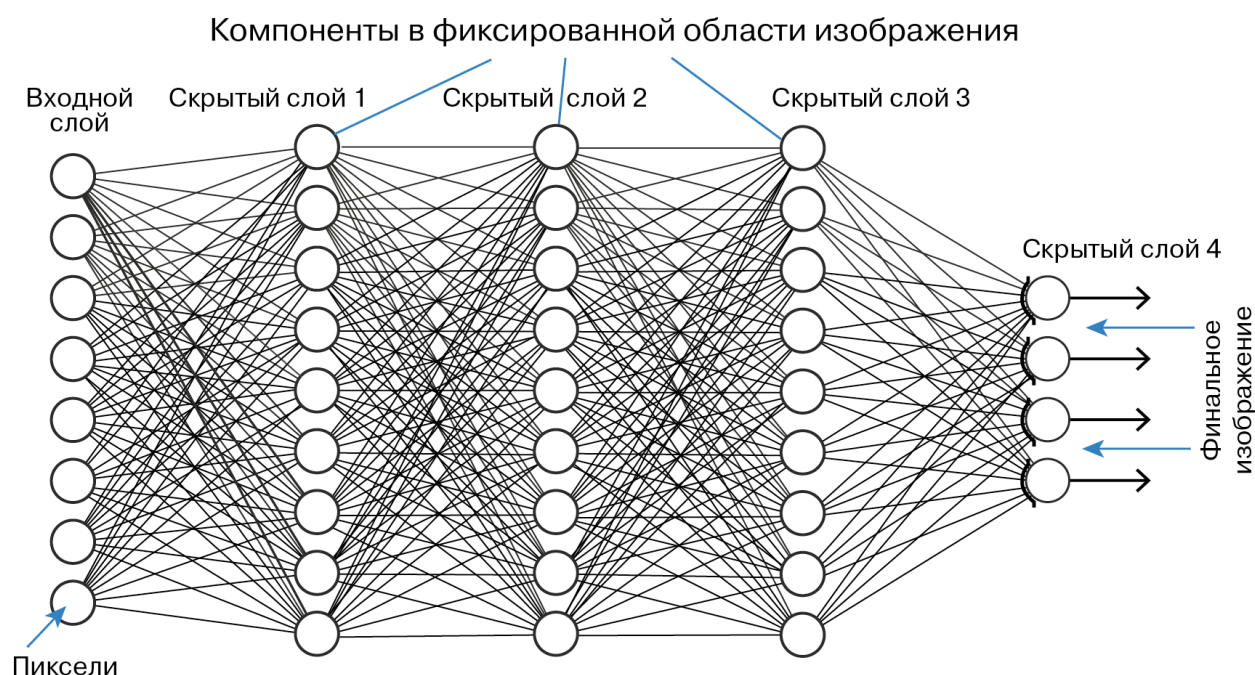


Рисунок 2.1 - Искусственная нейронная сеть

Обучение нейронных сетей проходит в несколько этапов:

- Прямое распространение ошибки;
- Обратное распространение ошибки.

Во время прямого распространения ошибки осуществляется прогнозирование ответов. Обратное распространение сводит к минимуму ошибку между фактическим ответом и прогнозируемым ответом.

При обучении нейронных сетей используется несколько важных параметров:

Функция активации является одним из самых мощных инструментов, влияющих на силу, приписываемую нейронным сетям. Другими словами, он частично определяет, какие нейроны будут активированы, и какая информация будет передана следующим слоям.

Без функций активации глубокие сети теряют большую часть своих возможностей обучения. Нелинейность этих функций отвечает за увеличение степени свободы, что позволяет обобщать проблемы больших размерностей в более низких измерениях. Некоторые функции активации изображены на рисунке 2.2.

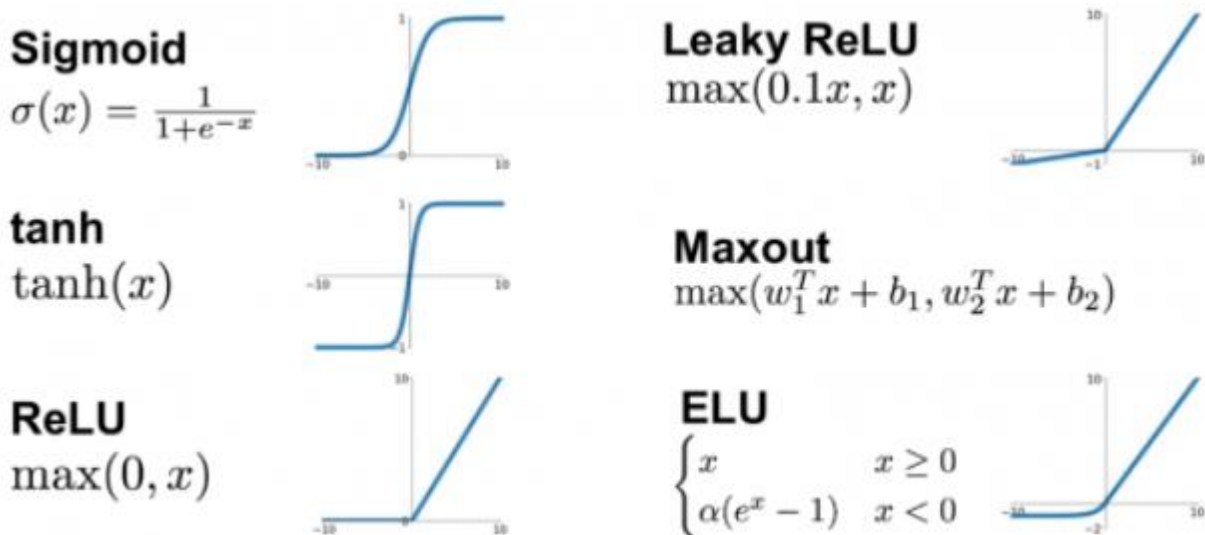


Рисунок 2.2 - Некоторые функции активации

Функция потерь находится в центре нейронной сети. Он используется для расчета ошибки между реальными и полученными ответами. Наша

глобальная цель-минимизировать эту ошибку. Таким образом, функция потерь эффективно приближает обучение нейронных сетей к этой цели.

Функция потери измеряет “насколько хороша ” нейронная сеть для данного набора данных и ожидаемых ответов. Это также может зависеть от переменных, таких как вес и смещение.

Функция потерь является одномерной и не векторной, поскольку она оценивает, насколько хорошо работает нейронная сеть в целом.

Некоторые известные функции потерь:

- Квадратичная (среднеквадратичное отклонение);
- Кросс-энтропия;
- Экспоненциальная (AdaBoost);
- Расстояние Кульбака — Лейблера или прирост информации.

Функция потерь в нейронной сети должна удовлетворять двум условиям:

- Функция потерь должна быть записана как среднее;
- Функция потерь не должна зависеть от каких-либо активационных значений нейронной сети, кроме значений, выдаваемых на выходе.

Также важную роль играют слои, из которых сделана архитектура искусственной нейронной сети. Давайте посмотрим на некоторые из них.

Свёрточные нейронные сети получили свое название от математической операции, называемой сверткой. Это специализированный вид линейной операции, и CNN используют эту математическую операцию вместо умножения матриц по крайней мере на одном из слоев. Это то, что отличает ConvNets от других нейронных сетей в глубоком обучении.

Слой сверточной нейронной сети должен иметь следующие элементы:

- Сверточные ядра (фильтры), определяемые шириной и высотой. Эти ядра просматривают изображение, по одной единице или пикселю за раз, чтобы получить значение пикселя, чтобы его можно было добавить в матрицу.

- Входные и выходные каналы, входные и выходные объемы. В этих входных и выходных каналах карты функций входят и выходят из разных слоев. Это помогает передать информацию, которую узнала нейронная сеть, на следующий уровень, где можно выполнить дополнительную матрицу и анализ.
- Глубина фильтра свертки должна быть равна количеству каналов входной карты признаков. Это означает, что сколько бы раз фильтр ни применялся к изображению, именно такой глубиной должна быть карта конечных характеристик. Это связано с тем, что каждый раз, когда фильтр проходит над изображением или картой функций, добавляется новый слой.

На рисунке 2.3. изображена операция свертки в нейронной сети.

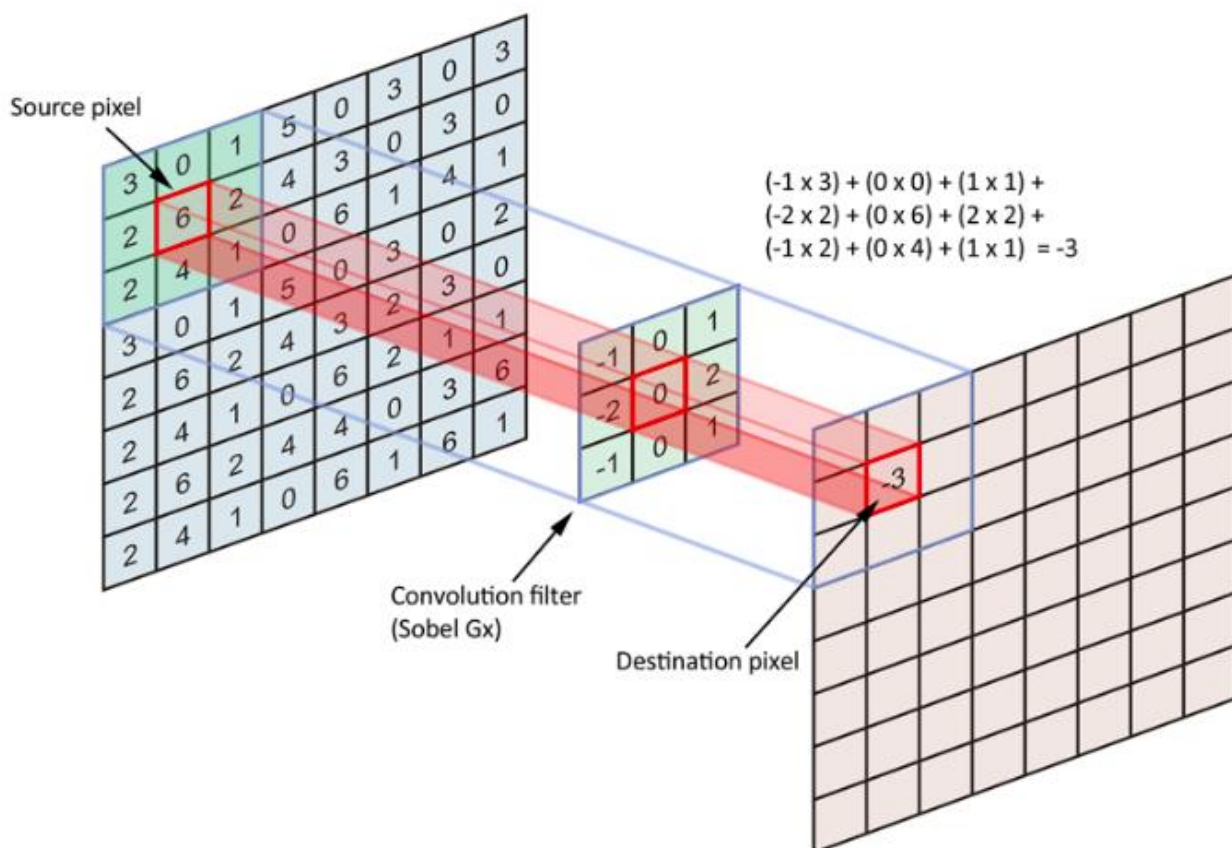


Рисунок 2.3 - Пример свертки

Слой активации - скалярный результат каждой свертки переходит в функцию активации, которая является типом нелинейной функции. Слой

активации обычно логически сочетается со сверточным слоем (считается, что функция активации встроена в сверточный слой). Функция нелинейности может быть любым выбором исследователя.

Слой объединения используется для уменьшения пространственных размеров, но не глубины, в модели сверточной нейронной сети, в основном это то, что вы получаете:

1. Имея меньше пространственной информации, вы увеличиваете производительность вычислений
2. Меньше пространственной информации также означает меньше параметров, поэтому меньше шансов на подгонку
3. Вы получаете некоторую неизменность перевода

Некоторые проекты не используют объединение, особенно когда они хотят «изучить» положение определенного объекта.

На рисунке 2.5 показано наиболее распространенный тип объединения слоя max-pooling, который сдвигает окно, как обычная свертка, и получает на выходе наибольшее значение в окне.

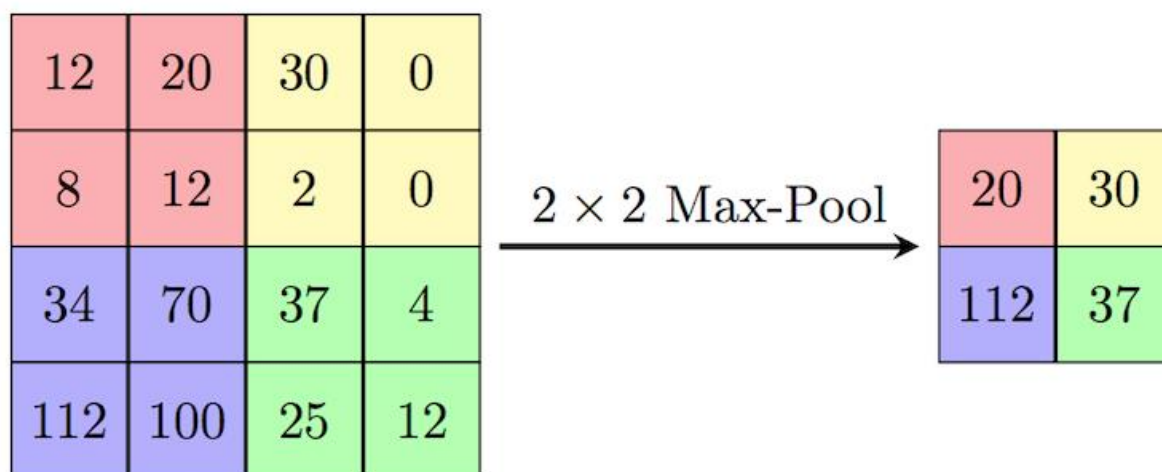
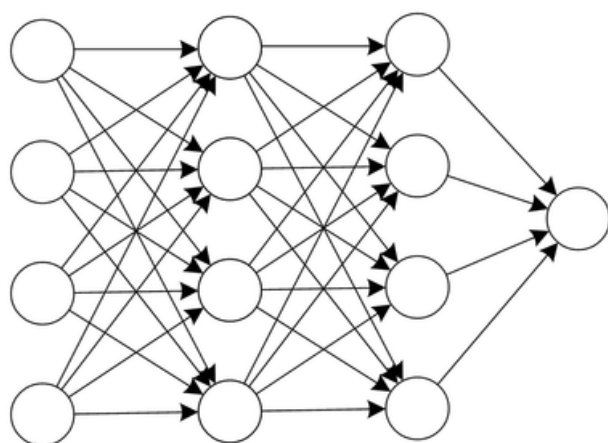
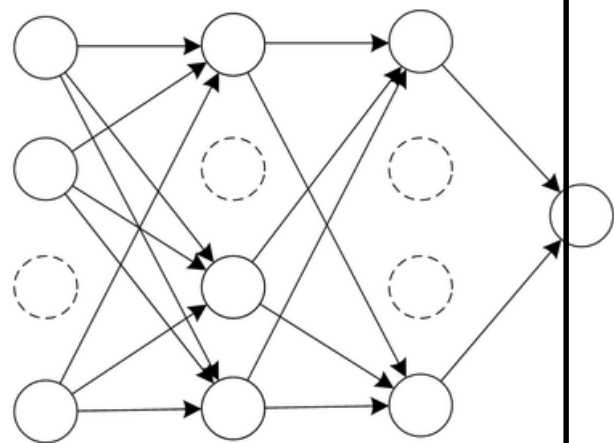


Рисунок 2.4 - Пример слоя пулинга по максимуму

Dropout – слой, который исключает часть нейронов. Исключенные нейроны не способствуют процессу обучения ни на одном из этапов алгоритма обратного распространения.



(a) Standard Neural Network



(b) Network after Dropout

Рисунок 2.5 - Графическое представление Dropout

Пакетная нормализация позволяет нам использовать гораздо более высокие скорости обучения и менее осторожно подходить к инициализации, а в некоторых случаях устраняет необходимость в отсеве. Применительно к современной модели классификации изображений, пакетная нормализация обеспечивает такую же точность за счет уменьшения количества шагов обучения в 14 раз и значительно превосходит исходную модель.

Нормализация входного слоя нейронной сети обычно выполняется путем масштабирования данных, подаваемых в функции активации. Например, когда есть признаки со значениями от 0 до 1 и некоторые признаки со значениями от 1 до 1000, то их необходимо нормализовать, чтобы ускорить обучение. Нормализацию данных можно выполнить и в скрытых слоях нейронных сетей, что и делает метод пакетной нормализации.

Полносвязная нейронная сеть - это сеть, в которой каждый нейрон связан со всеми остальными нейронами, находящимися в соседних слоях. Последовательная сеть прямого распространения - FFNN (Feed-Forwarded Neural Net), нейронная сеть, в которой все связи направлены только от входа сети к ее выходу (без операций распараллеливания или циклов). Полносвязный слой изображен на рисунке 2.6

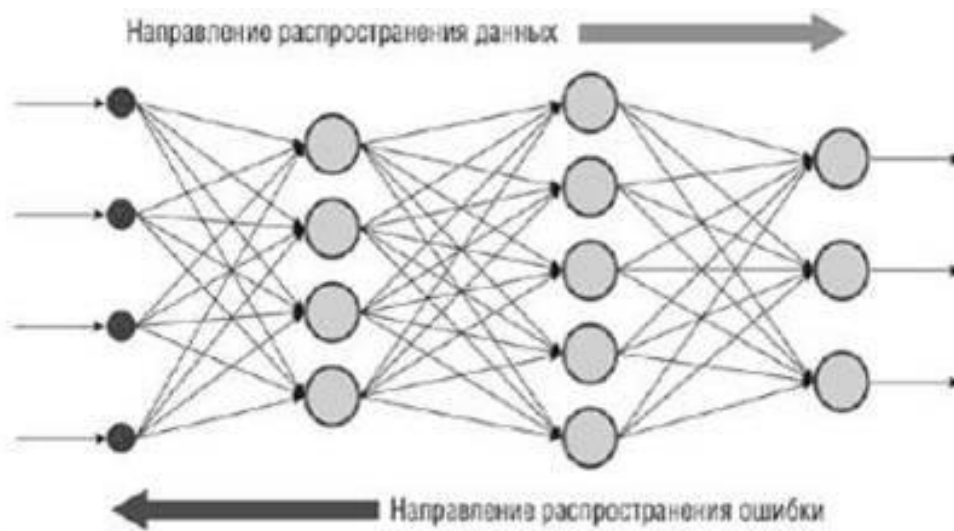


Рисунок 2.6 - Графическое представление полносвязного слоя

8.1.Сведения о вычислении оптического потока

Оптический поток определяется как изменение структурированного света в изображении, например, на сетчатке или датчике камеры, из-за относительного движения между глазным яблоком или камерой и сценой. Дальнейшие определения из литературы подчеркивают различные свойства оптического потока.

- Гельмгольца: «Я тоже считаю, что одноглазые люди могут формировать правильные представления о материальных формах своего окружения, главным образом, за счет вариаций изображения на сетчатке глаза из-за движений тела».
- Гибсон: «Аналитически, это полное преобразование массива, кажется, означает, что элементы этой текстуры смещены, причем элементы рассматриваются как пятна. С интроспективной точки зрения поле повсюду изобилует движением, когда движется наблюдатель ».
- Хорна: «Явное движение яркостных паттернов, наблюдаемое при движении камеры относительно объектов, на которых изображено изображение, называется оптическим потоком».

Гельмгольца в основном заботит восприятие глубины, и он описывает оптический поток как «вариации изображения сетчатки», которые возникают из-за движения тела и зависят от структуры, а именно расстояния, а также

жесткости окружающей среды. Гибсон описывает смещение структуры в оптической матрице как преобразование, которое «оживает, когда движется наблюдатель». Более новое определение оптического потока дано Хорном, который приписывает «движение яркостных паттернов» в изображении относительным движением между наблюдателем и объектами в окружающей среде.

С технической точки зрения и в контексте компьютерного зрения изменения окружающей среды на изображении представлены серией кадров изображения. Оптический поток фиксирует изменение этих изображений через векторное поле. В исследованиях делается акцент на точной оценке оптического потока по пикселям, что является сложной вычислительной задачей. В настоящее время оптический поток можно оценить практически в реальном времени для разумного разрешения изображения.

Оптический поток постоянно обрабатывается в нашей визуальной системе и может использоваться для решения различных задач. Такие задачи включают оценку самодвижения (Bruss & Horn, 1983), сегментацию сцены на независимо движущиеся объекты и жесткие части или передний план и фон (Weiss, 1997; Cremers & Schnörr, 2003). Кроме того, оптический поток содержит информацию о времени контакта с местами в окружающей среде, изображенными на изображении (Lee, 1976; Horn et al., 2007; Alenya et al., 2009). Оптический поток также позволяет оценить относительную глубину всех видимых и твердых объектов. Информация, извлеченная из оптического потока, может использоваться в системах помощи водителю для обнаружения других автомобилей и их движения, пешеходов и движения автомобиля (Wedel et al., 2011). Другое применение - обнаружение других самолетов для предотвращения столкновений в воздушном пространстве (Zufferey & Floreano, 2006; Socol et al., 2007). Оптический поток используется в видеокодеках для интерполяции изображения между ключевыми кадрами (Chahine & Konrad, 1995). Быстрые дисплеи с высоким разрешением используют оптический поток для синтеза дополнительных

					Постановка задачи на разработку	Лист
Изм.	Лист	№ докум.	Подпись			44

кадров изображения в качестве интерполяции между существующими (Csillag & Boroczky, 1999). Часто более высокая частота кадров невозможна из-за ограниченной полосы пропускания подключения к дисплею, и поэтому используется этот метод интерполяции. На рисунке 2.7 визуализировано обнаружение движения.

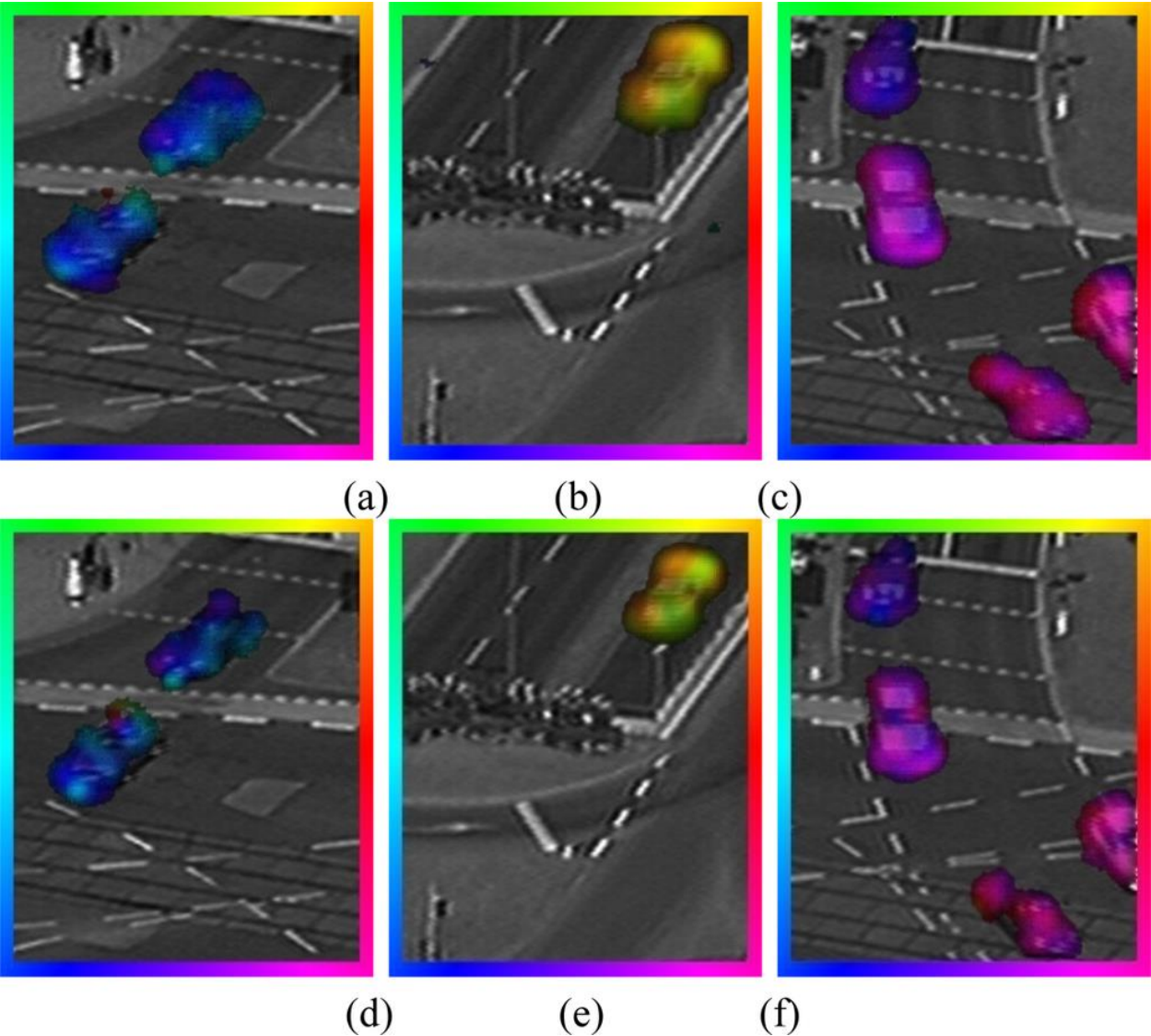


Рисунок 2.7 – Пример визуализации оптического потока

Выводы

В процессе выполнения этого этапа были определены программные средства необходимые для реализации данного дипломного проекта, введены основные определения нужные для понимания принципов работы разрабатываемой программы.

3. РАЗРАБОТКА СИСТЕМЫ

3.1. Проектирование нейронной сети

В качестве основной модели, была взята базовая модель сверточной нейронной сети и обучающая выборка MPI Sintel.

Набор данных MPI Sintel устраняет ограничения существующих тестов оптического потока. Он предоставляет натуралистические видеопоследовательности, которые сложно использовать современными методами. Он разработан для поощрения исследований движения на большие расстояния, размытия при движении, многокадрового анализа и нежесткого движения.

Набор данных содержит поля потока, границы движения, несовпадающие области и последовательности изображений. Последовательности изображений визуализируются с разным уровнем сложности.

Sintel - это короткометражный анимационный фильм с открытым исходным кодом, созданный Тоном Розендалом и Blender Foundation.

Сама структура сети состоит из базовых блоков, который представлен на рисунке 3.3:

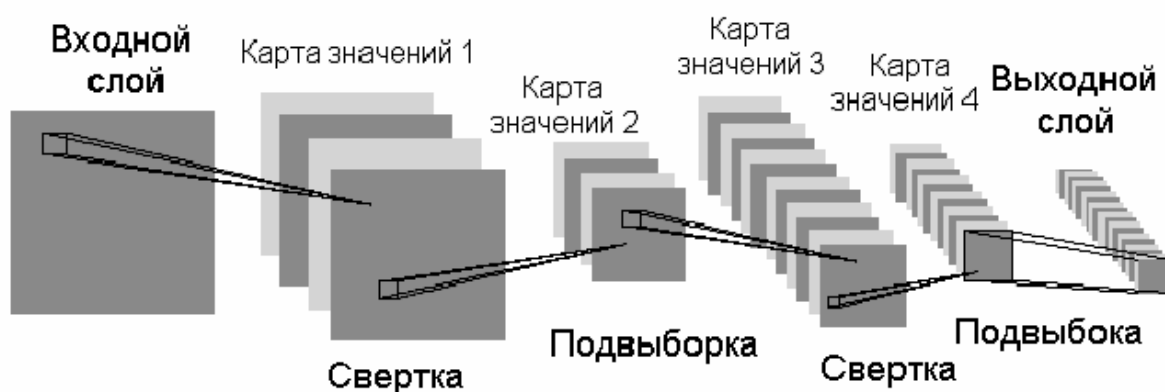


Рисунок 3.3 – Базовый блок

					<i>Разработка системы</i>			
Изм.	Лист	№ докум.	Подпись	Дата				
Разраб.		Малахов Д.В.			Разработка программного комплекса оптимизации хранения информации в системах видеонаблюдения с использованием нейронных сетей	Лит.	Лист	Листов
Руковод.		Немцев А.О.					45	2
Консульт.		Немцев А.О.				ГФ БГТУ гр. ВТ-41		
Н. Контр.		Полунин А.И.						
Зав. каф.		Поляков В.М.						

Интересующая нас сверточная часть сети состоит из одного обычного сверточного слоя с сверткой 3x3 в начале и тринадцати блоков, показанных на рис.3.3, с постепенно растущим числом фильтров и все меньшим пространственным размером тензора.

Особенностью этой архитектуры является отсутствие слоев максимальной подвыборки. Вместо этого для уменьшения пространственного измерения используется свертка с параметром шага, равным 2.

Обучение проходило в 20 эпох с выборкой, состоящей из 1041 кадров анимационного фильма.

Результаты обучения на последней эпохе:

loss: 0.0359

accuracy: 0.9921

val_loss: 0.0943

val_accuracy: 1.0000

Результаты обучения представлены на рисунке ниже:

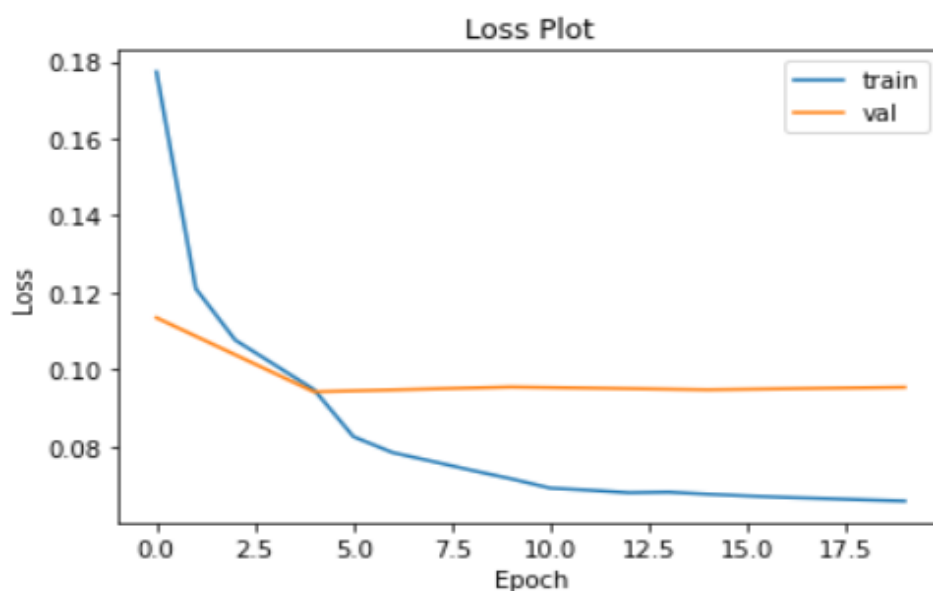


Рисунок 3.4 – Результаты тренировки модели

Как видно из графика ошибка на тренировочных и на тестовых данных стремиться к нулю, а точности на тех же данных стремиться к единице. Можно сделать вывод, что нейронная сеть обучена хорошо.

3.2. Описание применяемых алгоритмов обработки данных

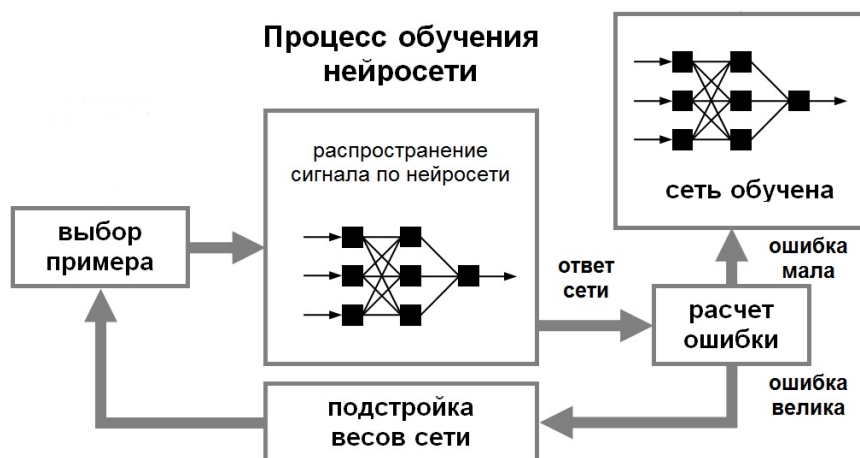


Рисунок 3.5 – Алгоритм тренировки сети

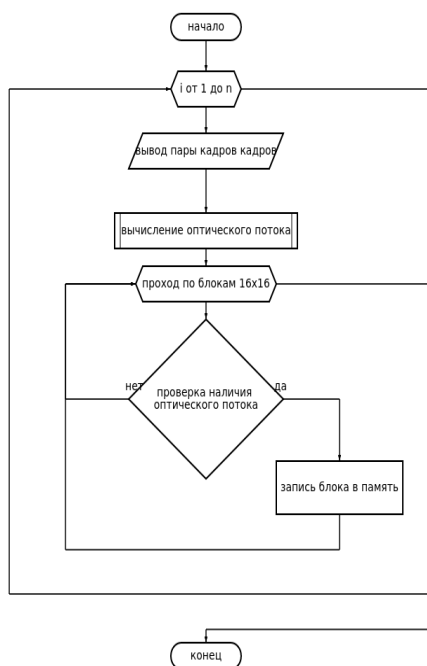


Рисунок 3.6 – Алгоритм сжатия видеофайла

Выводы

В процессе выполнения этого этапа были реализованы цели реализации данного дипломного проекта, спроектирована нейронная сеть и алгоритм сжатия информации.

4. РЕЗУЛЬТАТЫ ТЕСТИРОВАНИЯ

4.1. Тестирование программного комплекса

Программный комплекс был протестирован на съемках с камер видео наблюдения и выполняет поставленные перед ним задачи оптимизации хранения информации Рисунки 4.1 - 4.2



Рисунок 4.1 – Кадр исходного видео

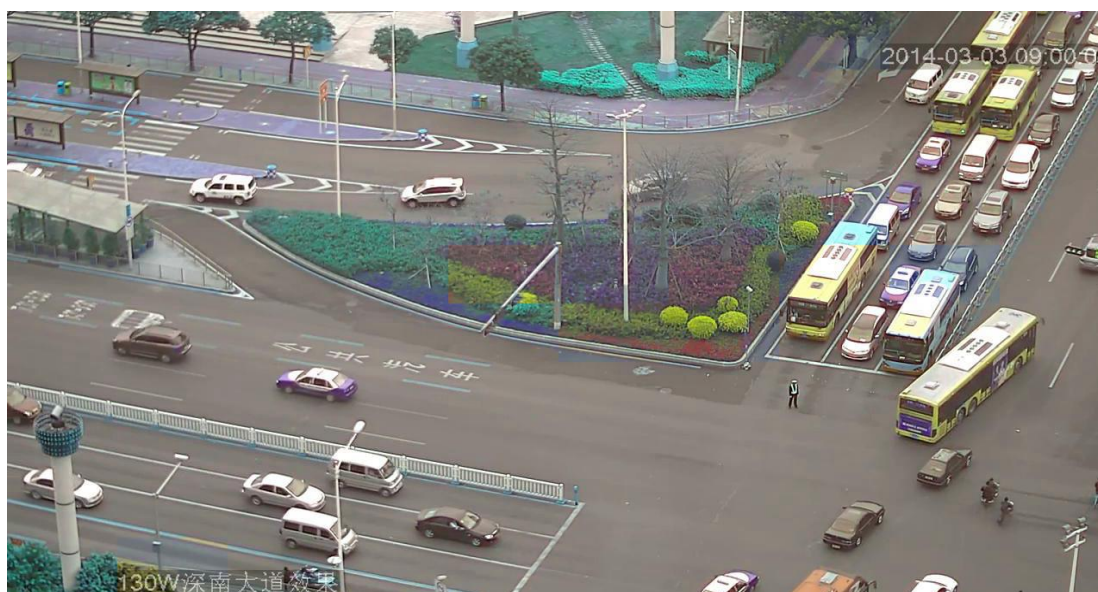


Рисунок 4.2 – Кадр сжатого видео

					РЕЗУЛЬТАТЫ ТЕСТИРОВАНИЯ			
Изм.	Лист	№ докум.	Подпись	Дата				
Разраб.		Малахов Д.В.			Разработка программного комплекса оптимизации хранения информации в системах видеонаблюдения с использованием нейронных сетей	Лит.	Лист	Листов
Руковод.		Немцев А.О.					48	2
Консульт.		Немцев А.О.				ГФ БГТУ гр. ВТ-41		
Н. Контр.		Полунин А.И.						
Зав. каф.		Поляков В.М.						

при тестировании алгоритма были получены следующие характеристики
алгоритма :

Битрейт: 0,22 Мб/с

исходный размер видео файла: 4,1 Мб

Длительность видео файла: 12с

размер сжатого видео файла 2,646 Мб

					Разработка системы	Лист
						49
Изм.	Лист	№ докум.	Подпись			

ЗАКЛЮЧЕНИЕ

Был разработан программный комплекс для оптимизации хранения информации в системах видеонаблюдения с использованием нейронных сетей. При разработке использованы актуальные на данный момент инструментальные средства.

В дипломном проекте были успешно решены следующие задачи:

- анализ предметной области;
- анализ аналогов разрабатываемой системы
- Выбраны инструментальные средства разработки;
- Разработана структура системы
- Проверена работоспособность программного обеспечения

Программный продукт позволяет оптимизировать хранения информации в системах видеонаблюдения вследствие получить увеличение сроков хранения архивных записей и ускорение передачи информации по сети.

Тестирование программного продукта показало, что все модули работают корректно.

Программный комплекс решает поставленные задачи и способен оптимизировать процесс хранения информации.

					Заключение			
Изм.	Лист	№ докум.	Подпись	Дата	Разработка программного комплекса оптимизации хранения информации в системах видеонаблюдения с использованием нейронных сетей	Лит.	Лист	Листов
Разраб.		Малахов Д.В.						
Руковод.		Немцев А.О.					50	1
Консульт.		Немцев А.О.				ГФ БГТУ гр. ВТ-41		
Н. Контр.		Полунин А.И.						
Зав. каф.		Поляков В.М.						

СПИСОК ЛИТЕРАТУРЫ

1. Васильев, А. Н. Python на примерах: практический курс по программированию / А. Н. Васильев. — 2-е изд. — Санкт-Петербург: Наука и Техника, 2017. — 432 с. — ISBN 978-5-94387-741-4. — Текст: электронный // Электронно-библиотечная система IPR BOOKS: [сайт]. — URL: <https://www.iprbookshop.ru/73043.html> (дата обращения: 02.04.2021). — Режим доступа: для авторизир. пользователей
2. Рик, Гаско Простой Python просто с нуля / Гаско Рик. — Москва: СОЛОН-Пресс, 2019. — 256 с. — ISBN 978-5-91359-334-4. — Текст: электронный // Электронно-библиотечная система IPR BOOKS: [сайт]. — URL: <https://www.iprbookshop.ru/94940.html> (дата обращения: 23.03.2021). — Режим доступа: для авторизир. пользователей
3. Соловьев Н.А. Цифровая обработка информации в задачах и примерах: учебное пособие для СПО / Соловьев Н.А., Тишина Н.А., Юркевская Л.А. — Саратов: Профобразование, 2020. — 122 с. — ISBN 978-5-4488-0596-7. — Текст: электронный // Электронно-библиотечная система IPR BOOKS: [сайт]. — URL: <http://www.iprbookshop.ru/92201.html> (дата обращения: 04.05.2021). — Режим доступа: для авторизир. Пользователей
4. ГОСТ 19_201-78 ЕСПД. Техническое задание. Требования к содержанию и оформлению.
5. Васильев А.Н. Python на примерах : практический курс по программированию / Васильев А.Н.. — Санкт-Петербург : Наука и Техника, 2017. — 432 с
6. Горожанина Е.И. Нейронные сети : учебное пособие / Горожанина Е.И.. — Самара: Поволжский государственный университет телекоммуникаций и информатики, 2017. — 84 с.
7. Review: MobileNetV2 — Light Weight Model (Image Classification) [Электронный ресурс]: URL <https://towardsdatascience.com/review-mobilenetv2-light-weight-model-image-classification-8febb490e61c> Яз.англ. — (дата обращения: 04.05.2021).

					<i>Список литературы</i>	Лист
						6
Изм.	Лист	№ докум.	Подпись			