

Применение физических признаков видео к задаче классификации

Р. А. Казанцев, С. В. Звездаков, Д. С. Ватолин

Аннотация—В данной работе рассматривается задача построения минимального репрезентативного набора физических признаков видео, которые могут быть использованы в разных регрессионных и классификационных задачах при работе с видео. Для тестирования полученного набора признаков была рассмотрена задача классификации видео на 4 класса: мультфильм, съемка с дрона, компьютерная игра и спортивная трансляция. В работе описан автоматический подход для создания размеченной по жанрам базы видеопоследовательностей, его аугментации и очистки от аномальных данных. С помощью этого подхода для проведения экспериментальной оценки была создана база из 14271 видео, где минимальное количество видео на класс составляет 2700 видео. Методом градиентного бустинга был обучен ансамбль решающих деревьев, средняя точность и полнота прогнозирования которого на тестовой выборке составили 86.15% и 86.12%. Также были опробованы другие методы машинного обучения такие, как логистическая регрессия, наивный гауссовский классификатор, метод опорных векторов и случайный лес, и они показали более худшие результаты. Наиболее важными признаками для принятия решений в задаче классификации видео оказались две метрики детализации изображения, вычисляемые с помощью оператора Лапласа и на основе эффекта повторного размывания, и пространственная сложность.

Ключевые слова—Классификация видео, признаки видео, сжатие видео, создание набора видеоданных.

I. ВВЕДЕНИЕ

В работе была поставлена задача сконструировать множество (физических) признаков видео, которые бы помогали при решении различных задач, связанных с видео, с помощью методов машинного обучения. Среди таких задач следует выделить сжатие, обработку видео и задачи компьютерного зрения.

Для экспериментальной оценки признаков рассматривается задача классификации видео на четыре жанра: мультфильм, съемка с дрона, компьютерная игра и записи спортивных трансляций. Эти жанры были выбраны по причине наличия у каждого жанра собственных характерных физических свойств (распределение цвета по кадру, характер и сложность движения, резкость, сложность структур и детализация объектов).

Статья получена 18 февраля 2019.

Казанцев Р. А. — МГУ имени М. В. Ломоносова (email: roman.kazantsev@graphics.cs.msu.ru).

Звездаков С. В. — МГУ имени М. В. Ломоносова (email: sergey.zvezdakov@graphics.cs.msu.ru).

Ватолин Д. С. — МГУ имени М. В. Ломоносова (email: dmitriy@graphics.cs.msu.ru).

Задача классификации видео по жанрам хорошо исследована и часто решается классическими методами машинного обучения и с помощью нейронных сетей [1, 2, 3]. В приведенных статьях в качестве практического применения этой задачи рассматривается автоматическая категоризация видео и поиск видео необходимой категории для сервисов видеохостинга. Также мы считаем, что задача полезна при сжатии видео, когда для жанра (категории) существует заранее подготовленная конфигурация кодека для более оптимального сжатия видео. К примеру, в видеокордеке h264 для мультфильмов существует специальная настройка `--tune animation` для оптимального кодирования.

В секции II рассматриваются существующие решения для задачи классификации видео, приведены их достоинства и недостатки, а также рассматриваются используемые признаки видео. Так как в открытом доступе отсутствует необходимая видеобаза для исследования, в секции III описывается подход для создания подобных наборов видеоданных с возможностью их расширения. В секции IV подробно обсуждаются выбранные признаки видео (видеометрики). В секциях V и VI приводятся детали обучения модели автоматической категоризации видео и результаты тестирования.

II. ИССЛЕДОВАНИЯ В ДАННОЙ ОБЛАСТИ

Нейронные сети хорошо себя зарекомендовали в решении задач компьютерного зрения (детекция и классификация объектов, сегментация). Одно из достоинств нейросетевых методов по сравнению с классическими методами машинного обучения — не требуется конструирование признаков вручную для описания изображения и видео. Однако плата за это преимущество, как правило, — это большой размер модели и ее плохая алгоритмическая интерпретируемость. Во многих работах модель обучают на основе готовой нейросети, обученной на большом наборе данных. Такой подход называется переносом знаний (англ. *transfer learning*). К примеру, в работе [1] разработана нейросетевая модель для прогнозирования жанра видео со средней точностью 85%, где для извлечения признаков (описания) кадров видео используется сеть VGG-16 [4]. При этом полученный вектор-признак кадра имеет высокую размерность, равную 4096, и не имеет очевидной интерпретации, а количество параметров модели превышает 8 миллионов. Несмотря на то, что нейросети, используемые для извлечения признаков, могут работать с изображениями различных разрешений, они прежде всего ориентированы на распознавание образов, чем на

извлечение общей информации на пиксельном уровне [5]. Однако наличие пиксельной информации является критичной для сжатия видео, так как все современные кодеки оперируют пиксельными значениями.

Поэтому также был рассмотрен ряд работ [2, 3, 5, 11], где используются вручную сконструированные признаки видео и с помощью машинного обучения успешно решаются задачи по сжатию видео и классификации. Из этих работ были отобраны признаки видео для решения задачи классификации видео на четыре жанра: мультфильм, съемка с дрона, компьютерная игра и спорт. Описание и формулы для вычисления признаков приводятся в секции IV.

Некоторые работы [2, 3] приводят неубедительные результаты по классификации видео, поскольку для обучения и тестирования модели используется набор данных совсем небольшого размера (всего 20-50 объектов на класс). Проблема небольшой обучающей выборки в задаче классификации видео в работе [15] решена с помощью простой идеи, но не описана в соответствующей статье [1]. Эта идея приведена в секции III данной статьи и дополнена подходом для увеличения количества данных путем разбиения видеопоследовательности на сцены.

III. СОЗДАНИЕ БАЗЫ ВИДЕОДАНЫХ

Поскольку среди открытых баз видео не удалось найти подходящей для задачи классификации видео на четыре жанра (мультфильм, съемка с дрона, компьютерная игра и спорт), была создана собственная база.

На видеохостинге YouTube существует большое количество плейлистов видео для каждого рассматриваемого класса. Группировка видео по плейлистам обеспечивает автоматическую разметку. Используя программный модуль youtube-dl [17], было скачано 3509 видео, среди которых 947 анимаций, 738 видеосъемки с дрона, 920 видео из компьютерных игр и 904 записи спортивных трансляций. Скачанные видео были дополнительно верифицированы на принадлежность к заявленному классу, о чем будет рассказано дальше.

Для того чтобы проводить более точную классификацию и увеличить количество рассматриваемых примеров, каждое видео было разбито на сцены. Определение смены сцен в видеопоследовательности производилось с помощью алгоритма поблочного сравнения гистограмм распределения яркости [12].

В результате в набор данных вошла 14271 сцена: 2747 мультфильмов, 2747 съемок с дрона, 4546 видео из компьютерных игр, 4231 запись спортивных трансляций. Примеры полученных сцен изображены на рисунке 1.

Недостатком описанного способа создания базы видеоданных для задачи классификации может быть содержание значительного числа выбросов (аномальных данных). К примеру, в плейлист могут попасть видео, не относящиеся к заявленной категории, или среди сцен, полученных разбиением видеопоследовательности, могут оказаться вступительные заставки с текстом и

черным фоном. Из-за большого количества выбросов процесс обучения может быть осложнен и качество итоговой модели может пострадать. Поэтому были предприняты меры по их устранению. Во-первых, исключены первая и последняя сцены каждого видеоряда из базы, так как среди них наиболее вероятно встретить заставку и черный фон. Во-вторых, после процесса обучения был проведен ручной отсмотр некорректно спрогнозированных сцен с целью обнаружения выбросов, обнаруженные выбросы были исключены из набора данных.

Также стоит избегать утечек данных, когда сцены одного и того же видео используются для обучения и тестирования. Автоматический способ разбиения, предложенный в библиотеках машинного обучения, не подходит, так как данные разбиваются случайным образом и сцены одного видео могут попасть в разные части. Поэтому с помощью класса *PredefinedSplit* из библиотеки *scikit-learn* [13] был реализован собственный механизм разбиения данных для кросс-валидации, обучения и тестирования модели, предотвращающий утечку данных. Реализованный механизм обеспечивает, чтобы все сцены одного видео попали либо в обучающую выборку, либо в тестовую.

IV. ФИЗИЧЕСКИЕ ПРИЗНАКИ ВИДЕО И ИХ ВЫЧИСЛЕНИЕ

В задаче рассматриваются 4 категории видео: мультфильм, съемка с дрона, компьютерная игра и спорт. У каждой категории видео можно найти свои физические особенности. К примеру, мультфильмы имеют характерное разреженное распределение цвета в кадре, присутствуют простота в формах, контрастные переходы и ярко выраженные границы объектов. Видеоигры также характеризуются разреженным распределением цвета, но в свою очередь могут быть более реалистичными, иметь более сложные формы объектов и плавные переходы на границах объектов. Для съемки с дрона характерно плавное движение камеры в определенном направлении. Основной отличительной особенностью спортивного видеоконтента является большое количество движений объектов разного размера. Исходя из особенностей каждой категории мы выбрали 9 признаков видео для решения задачи классификации.

SI (Spatial perceptual Information) [9] оценивает пространственную сложность видеокadra, как среднее квадратичное отклонение длины градиента изменения яркости пикселя внутри кадра. Вычисляется по формуле: $SI = std_{space}[Sobel(x(i, j))]$.

TI (Temporal perceptual Information) [9] оценивает временную сложность кадра, как среднее квадратичное отклонение изменения яркости пикселей соседних кадров. Вычисляется по формуле: $TI = std_{space}[F_n(i, j) - F_{n-1}(i, j)]$, где $F_n(i, j)$ — яркость пикселя (i, j) на n -ом кадре.

TI_ME (Temporal perceptual Information with Motion Estimation) [10] оценивает временную сложность кадра с учетом компенсации движения. Вычисляется по формуле: $TI_ME = std_{space}[\min(F_n(i, j) - F_{n-1}(k, l))]$, где $F_n(i, j)$ — яркость пикселя (i, j) на n -ом кадре.



мультфильм



съемка с дрона



компьютерная игра



спорт

Рис. 1. Примеры полученных сцен для каждого класса видео.

Признак характеризует различия в изменении яркости пикселей соседних кадров после компенсации движения. Для оценки и компенсации движения использовался блочный алгоритм [6], со следующими параметрами: $min_block_size=8$, $max_block_size=8$, $precision="QuarterPixel"$, $loss_metric="SAD"$.

Blur — метрика четкости с использованием эффекта второго размытия [7]. Эффект заключается в том, что после повторного размытия изображения разница между яркостными значениями пикселей изменяется мало. Чем выше метрика, тем четче изображение.

Blur_Lap — метрика четкости с использованием оператора Лапласа [8]. Метрика вычисляется как среднеквадратичное отклонение значений отклика оператора Лапласа, примененной к каждому пикселю кадра, по следующей формуле:

$$LAP_{var} = \frac{1}{width \cdot height} \sum_m \sum_n [L(m, n) - \bar{L}]^2,$$

где $\bar{L} = \frac{1}{width \cdot height} \sum_m \sum_n |L(m, n)|$ — среднее значение абсолютного значения отклика оператора Лапласа.

Mean_Y, **Mean_U** и **Mean_V** — среднее значение каждой из трех цветовых компонент пикселя в n -ом кадре в пространстве YUV [11]. Вычисляются по следующей формуле:

$$Mean_Y(U, V)_n = \frac{1}{width \cdot height} \sum_{i=1}^{width} \sum_{j=1}^{height} F_n(i, j).$$

Var_Y, **Var_U** и **Var_V** — среднеквадратичное отклонение значений каждой из трех цветовых компонент пикселя в n -ом кадре в пространстве YUV [11]:

$$Var_Y(U, V)_n = \sqrt{\frac{1}{width \cdot height} \sum_{i=1}^{width} \sum_{j=1}^{height} (F_n(i, j) - Mean_Y(U, V)_n)^2}.$$

Color3D_Hist трехмерная цветовая гистограмма видеокadra оценивает распределение значений пикселя кадра в цветовом пространстве RGB [3]. Каждый канал цвета равномерно разбивается на 4 части. Таким образом размер гистограммы составляет $64=4 \times 4 \times 4$.

Motion_Hist двумерная гистограмма движения в кадре оценивает распределение длины и направления (угла) вектора движения. Размер гистограммы равен $25=5 \times 5$. Причем используется логарифмическая сетка по основанию 2 для разбиения диапазона значений длины вектора движения, так чтобы диапазон значений для векторов меньшей длины разбивался чаще. Для нахождения векторов движения каждого пикселя использовался алгоритм блочной оценки движения [6], со следующими параметрами: $min_block_size=8$, $max_block_size=8$, $precision="QuarterPixel"$, $loss_metric="SAD"$.

Для каждого кадра из более 5 млн. кадров (14271 сцены) были вычислены 9 составных признаков (или 100 простых признаков). Вычисление заняло примерно две недели на двух компьютерах Intel(R) Core(TM) i7-4770R CPU @ 3.20GHz, 16GB RAM. Так как количество кадров у разных сцен может различаться, в качестве вектора-признака для сцены берутся значения признаков центрального кадра данной сцены. Таким образом, по входному вектору-признаку видео (сцены) прогнозируется его категория.

V. ПОСТРОЕНИЕ МОДЕЛИ КЛАССИФИКАЦИИ

В секции III было отмечено, что недостатком предложенного подхода для создания базы видеоданных могут быть аномальные сцены (выбросы). В созданном

наборе данных из четырех классов наблюдалось достаточное количество аномальных сцен, от которых пришлось избавиться для получения более точной модели. Было проведено 2 итерации обучения модели решающих деревьев с помощью градиентного бустинга, после каждой проводилась ручная диагностика некорректно классифицированных сцен, среди которых были выявлены аномальные сцены. Из аномалий для мультфильмов были выделены следующие случаи: сцены про то, как создавался мультфильм и интервью от создателей. Среди съемок с дронов встретились следующие аномальные сцены: реклама дронов, заставки в начале ролика, ускоренное воспроизведение (time-lapse), съемка с камер мобильных устройств. Категория видеоигр пострадала от заставок, сцен с видеоблогерами, рекламы и новостей. Меньше всего аномальных видео было зарегистрировано у спортивных видео, так как соответствующие плейлисты оказались более качественными. Однако, в этой категории удалось обнаружить следующие выбросы: сцены с развевающимися флагами, сцены награждения спортсменом, кадры таблиц с результатами. Таким образом, была выявлена 391 аномальная сцена: 93 в категории «мультфильм», 224 — «съемка с дрона», 48 — «компьютерная игра» и 26 в категории «спортивные видео».

Итоговые (очищенные) данные были разделены следующим образом: 11115 видео для обучения (и кросс-валидации) и 2765 для тестирования. Для кросс-валидации данные разбивались на три части. Нужно заметить, что в обоих случаях был реализован собственный механизм разбиения данных с помощью класса *PredefinedSplit* из библиотеки *scikit-learn* [13], который предотвращает утечку данных, то есть попадание сцен одного и того же видео в разные части. Автоматический механизм разбиения данных, реализованный в библиотеках машинного обучения не подходит из-за высокой вероятности утечки данных.

Для сравнения разными методами были обучены несколько классификаторов:

- Логистическая регрессия с точностью прогнозирования 64%;
- Ансамбль решающих деревьев, полученный методом случайного леса, с точностью прогнозирования 83%;
- Ансамбль решающих деревьев, полученный с помощью градиентного бустинга, с точностью прогнозирования 86.1%.

Также были опробованы метод опорных векторов и гауссовский наивный байес (Gaussian Naive Bayes), но не было получено приемлемого результата.

Таким образом, оптимальным методом построения классификатора видео по жанрам оказался метод градиентного бустинга над решающими деревьями. Для обучения и тестирования модели использовалась библиотека XGBoost [14]. Выбор параметров модели проводился с помощью кросс-валидации. Были найдены следующие оптимальные параметры: *booster=gbtree*, *gamma=1*, *reg_alpha=1*, *learning_rate=0.001*, *max_depth=4*, *subsample=0.6*. Процесс обучения и

тестирования модели представлен на рисунке 2.

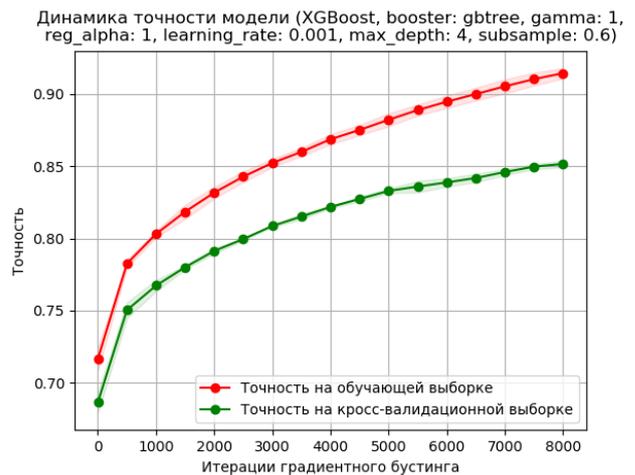


Рис. 2. Процесс обучения классификатора методом градиентного бустинга.

Наиболее значимыми признаками оптимальной модели оказались метрики детализации изображения *Blur_Lap* и *Blur*. На рисунке 3 приводится диаграмма важности физических признаков видео для решения данной задачи классификации.

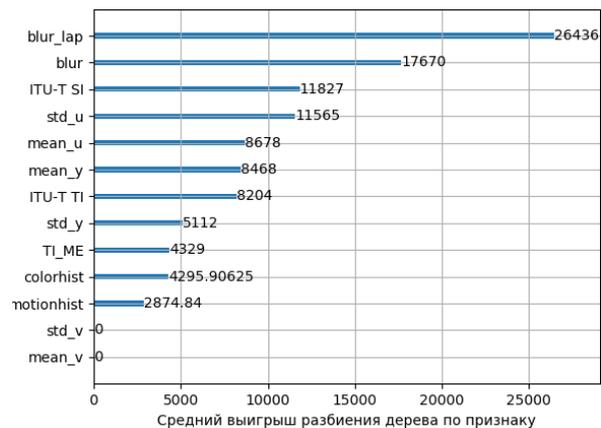


Рис. 3. Важность физических признаков видео для классификации.

VI. РЕЗУЛЬТАТЫ

Средняя точность прогнозирования на тестовой выборке составила 86.1%, истинность которой подтверждается результатами кросс-валидации из рисунка 2. Хуже всего прогнозируются мультфильмы с точностью 80.6%, а лучше всего — спортивный жанр с точностью 91.6%. Матрица ошибок на тестовой выборке, точность и полнота модели представлены в таблице 1.

Для каждого жанра была проведена диагностика и выявлены проблемные места модели. Предложенный метод плохо распознает съемку с дрона в случаях плохой освещенности, многочисленных близко движущихся около камеры объектов и регулировки зума, что приводит к оценке движению нехарактерной для съемки с дрона. Также есть проблемы с распознаванием игр в случаях: низкое качество видео из-за сильного сжатия; движение камеры в игре подобно

Таблица 1. Матрица ошибок на тестовой выборке, точность и полнота модели

		Спрогнозированный жанр				Точность модели (%)
		Мультфильм	Съемка с дрона	Компьютерная игра	Спорт	
Настоящий жанр	Мультфильм	428	13	60	30	80.6
	Съемка с дрона	19	427	34	12	86.7
	Компьютерная игра	64	33	765	31	85.6
	Спорт	25	14	32	778	91.6
	Полнота модели (%)	79.8	87.6	85.8	91.4	

съемке с дрона; спортивные игры с реалистичной графикой.

VII. ЗАКЛЮЧЕНИЕ И ДАЛЬНЕЙШЕЕ ИССЛЕДОВАНИЕ

В работе было рассмотрено 9 признаков для описания пространственной и временной сложности видео, четкости границ, распределения цвета и характера движения объектов в кадре. С использованием этих признаков была успешно решена задача распознавания 4 категорий видео (мультфильм, съемка с дрона, компьютерная игра и спорт) с точностью 86.1%.

В качестве дальнейшего исследования мы планируем применить эти видеометрики для нахождения оптимальной конфигурации кодера для сжатия видео и сравнить с результатами работы [5], в которой отсутствуют признаки для оценки временной составляющей видеосигнала после компенсации движения, как **TI_ME**.

БИБЛИОГРАФИЯ

- [1] K. Sivaraman, G. Somappa, "Moviescope: Movie trailer classification using deep neural networks," University of Virginia, 2016.
- [2] R. Zumer, S. Ratté, "Color-independent classification of animation video," *IJMIR*, pp. 187–196, 2018.
- [3] Narra Dhana Lakshmi, Y. Madhavee Latha, A. Damodaram, K. Lakshmi Prasanna, "Implementation of Content Based Video Classification using Hidden Markov Model," *IEEE 7th International Advance Computing Conference*, 2017.
- [4] K. Simonyan, A. Zisserman, "Very deep convolutional networks for large-scale image recognition," *International Conference on Learning Representations*, 2015.
- [5] O. Murashko, "Using machine learning to select and optimise multiple objectives in media compression," PhD thesis, University of St. Andrews, 2018.
- [6] K. Simonyan, S. Grishin, D. Vatolin, D. Popov, "Fast video super-resolution via classification," *15th IEEE International Conference on Image Processing*, pp. 349–352, 2008.
- [7] F. Crete, T. Dolmiere, P. Ladret, M. Nicolas, "The Blur Effect: Perception and Estimation with a New No-Reference Perceptual Blur Metric," *Human Vision and Electronic Imaging XII*, pp. 1–11, 2007.
- [8] R. Bansal, G. Raj, T. Choudhury, "Blur image detection using Laplacian operator and Open-CV," *IEEE System Modeling & Advancement in Research Trends*, pp. 63–67, 2016.
- [9] TU-T Recommendation P.910: Subjective video quality assessment methods for multimedia applications, 1999. – 37 p.
- [10] Браиловский И., Соломещ Н. Моделирование качества для видеокодирования // Информационные технологии. – 2012. — №1 – С. 42–48.
- [11] Хабибуллина Н.А., "Разработка новых методов анализа качества видеокодексов и оптимизация систем сжатия видеoinформации, Диссертация на соискание ученой степени кандидата технических наук," МФТИ, 2014.
- [12] Паршин А. Е., Глазистов И. В. Алгоритм поиска дубликатов в базе видеопоследовательностей на основе сопоставления иерархии смен сцен // Новые информационные технологии в автоматизированных системах. – 2009. – № 12. – С. 51–61.
- [13] Pedregosa et al., "Scikit-learn: Machine Learning in Python," *JMLR* 12, pp. . 2825–2830, 2011.
- [14] T. Chen, C. Guestrin, "Xgboost: A scalable tree boosting system," *arXiv:1603.02754*, 2016
- [15] K. Sivaraman, G. Somappa, MovieScope, GitHub repository, <https://github.com/maximus009/MovieScope>
- [16] x264, <https://www.videolan.org/developers/x264.html>
- [17] youtube-dl, <https://rg3.github.io/youtube-dl/>

Application of physical video features in classification problem

R. Kazantsev, S. Zvezdakov, D. Vatolin

Abstract— In this paper, we propose hand-crafted physical features for video that can be used in a wide range of different regression and classification problems in video processing. For evaluation of resulted set of features we consider video genre classification problem among four classes: animation, drone video, computer game and sports. In this work we describe an automatic approach for video dataset creation, its augmentation and anomaly detection. In order to arrange the experiment we create dataset from 14271 samples, having a minimal number of samples per class is equal to 2700. Using gradient boosting we trained decision tree ensemble model with average precision and recall equal to 86.15% and 86.12% on test dataset. Also, other machine learning methods such as linear regression, naïve Gaussian classifier, support vector machine and random forest demonstrated worse results. The most relevant physical video features are two blur metrics using Laplacian operator and based on re-blur effect.

Keywords—Hand-crafted video features, video classification, video compression, video dataset creation.

REFERENCES

- [1] K. Sivaraman, G. Somappa, "Moviescope: Movie trailer classification using deep neural networks," University of Virginia, 2016.
- [2] R. Zumer, S. Ratté, "Color-independent classification of animation video," *IJMIR*, pp. 187–196, 2018.
- [3] Narra Dhana Lakshmi, Y. Madhavee Latha, A. Damodaram, K. Lakshmi Prasanna, "Implementation of Content Based Video Classification using Hidden Markov Model," *IEEE 7th International Advance Computing Conference*, 2017.
- [4] K. Simonyan, A. Zisserman, "Very deep convolutional networks for large-scale image recognition," *International Conference on Learning Representations*, 2015.
- [5] O. Murashko, "Using machine learning to select and optimise multiple objectives in media compression," PhD thesis, University of St. Andrews, 2018.
- [6] K. Simonyan, S. Grishin, D. Vatolin, D. Popov, "Fast video super-resolution via classification," *15th IEEE International Conference on Image Processing*, pp. 349–352, 2008.
- [7] F. Crete, T. Dolmiere, P. Ladret, M. Nicolas, "The Blur Effect: Perception and Estimation with a New No-Reference Perceptual Blur Metric," *Human Vision and Electronic Imaging XII*, pp. 1–11, 2007.
- [8] R. Bansal, G. Raj, T. Choudhury, "Blur image detection using Laplacian operator and Open-CV," *IEEE System Modeling & Advancement in Research Trends*, pp. 63–67, 2016.
- [9] TU-T Recommendation P.910: Subjective video quality assessment methods for multimedia applications, 1999. – 37 p.
- [10] Brailovskij I., Solomeshh N. Modelirovanie kachestva dlja videokodirovaniya // *Informacionnye tehnologii*. – 2012. — #1 – S. 42–48.
- [11] Habibullina N.A., "Razrabotka novyh metodov analiza kachestva videokodekov i optimizacija sistem szhatija videoinformacii, Dissertacija na soiskanie uchenoj stepeni kandidata tehniceskikh nauk," MFTI, 2014.
- [12] Parshin A. E., Glazistov I. V. Algoritm poiska dublikatov v baze videoposledovatel'nostej na osnove sopostavlenija ierarhii smen scen // *Novye informacionnye tehnologii v avtomatizirovannyh sistemah*. – 2009. – # 12. – S. 51–61.
- [13] Pedregosa et al., "Scikit-learn: Machine Learning in Python," *JMLR* 12, pp. . 2825–2830, 2011.
- [14] T. Chen, C. Guestrin, "Xgboost: A scalable tree boosting system," *arXiv:1603.02754*, 2016
- [15] K. Sivaraman, G. Somappa, *MovieScope*, GitHub repository, <https://github.com/maximus009/MovieScope>
- [16] x264, <https://www.videolan.org/developers/x264.html>
- [17] youtube-dl, <https://rg3.github.io/youtube-dl/>