УДК 004.021 + 004.623 + 004.421

РАЗРАБОТКА СИСТЕМЫ ДЕТЕКТИРОВАНИЯ ДУБЛИКАТОВ ВИДЕОФАЙЛОВ НА ОСНОВЕ ИХ ЦВЕТОВЫХ КАРТ

Г. А. Нуриева[1], А. А. Ференец[2]

[1,2] Высшая школа информационных технологий и интеллектуальных систем Казанского (Приволжского) федерального университета

[1] nurievag97@gmail.com, [2]ist.kazan@gmail.com

Аннотация

Описана разработка системы детектирования дубликатов видеофайлов на основе методики получения и анализа цветовой карты видеоряда как подхода, не требующего больших вычислительных ресурсов, применимого к широкому спектру типов видеороликов и имеющего малые искажения относительно оригинала.

***Ключевые слова**: видеофайл, цветовая карта, детектирование дубликатов*

ВВЕДЕНИЕ

Согласно модели Web 2.0 [1], содержимое сайтов должно пополняться самими пользователями. Этот подход обеспечивает гораздо более быстрое развитие сайтов, но поддержка сайта всё равно необходима для модерации содержимого, в том числе, исключения дублирования материалов. У крупных веб-порталов невозможно проанализировать каждый ролик вручную. Например, в 2018 году на Youtube каждую минуту загружалось порядка 300 часов видеоматериалов [2]. Поэтому есть необходимость полной или частичной автоматизации этого процесса.

Анализ видеофайлов осложняется тем, что иногда искажается видео- или аудиоряд: используется другой способ кодирования, специально применяются незначительные искажения (изменение разрешения, зеркальное отражение, добавление артефактов, удаление некоторых кадров и прочие изменения). Это исключает точное сравнение последовательностей байт и требует анализа непосредственно видеоданных. Для этого можно анализировать низкоуровневые

признаки: частота кадров, разрешение, глубина цвета. При этом их точные замеры не дают однозначного определения соответствия, так как изменение этих параметров за счёт перекодирования видеофайла сохраняет узнаваемые образы, но изменяет числовые показатели указанных. Соответственно можно анализировать образы, которые передаёт видео. С точки зрения детектирования дублирования видеофайлов это наиболее прогрессивный метод, но требующий использования технологий компьютерного зрения для распознавания образов, значит, обучающих наборов данных для разных тематических категорий видео и определённых вычислительных ресурсов. С другой стороны, возможно выделение некоторых параметров видеоряда, которые связаны не с техническими особенностями представления, а именно с содержимым, и при этом достаточно абстрактными, чтобы их замерять напрямую или без больших вычислительных цепочек. Примером такого параметра является информация о цветах, использованных в видеоряде, или, иначе говоря, о его цветовой карте: непосредственное количество пикселей того или иного цвета или иная информация, следующая из замера количества этих пикселей. Была выдвинута гипотеза, что на основе сравнения цветовых карт видеорядов можно сделать вывод о дублировании одного из них другим даже при незначительных модификациях повторяющегося фрагмента.

1. СОЗДАНИЕ ЦВЕТОВОЙ КАРТЫ

Для получения цветовой карты видеоряда было решено вычислять средний цвет избранных кадров (получать усреднённые показатели цветовых параметров всех пикселей кадров) и далее собирать эту информацию во временные ряды. Кадры для анализа берутся из исходного видеоряда с определённым периодом, который должен быть подобран в зависимости от способа замера цвета.

Были рассмотрены две модели представления цвета: RGB и HSV. В цветовой модели RGB каждый цвет может быть описан как числовое соотношение трех базовых цветов – красного (Red), зеленого (Green) и синего (Blue). Обычно эти параметры варьируются в диапазоне 0–255. HSV (англ. Hue, Saturation, Value) – цветовая модель, в которой координатами цвета являются Hue – тон, Saturation – насыщенность и Value – значение (Brightness – яркость). Тон варьи-

руется от 0 до 360° (но часто приводится к диапазону 0–1), насыщенность и яркость – от 0 до 1. Модель HSV является нелинейным преобразованием модели RGB. Поэтому легко выполнить преобразование RGB в HSV и в обратную сторону. При этом использование того или иного представления может дать более наглядные результаты в зависимости от параметров видеоряда.

Для получения среднего цвета кадра было решено проанализировать все пиксели, вычленения значения RGB и их преобразование в HSV при необходимости, а далее проводить вычисление среднего арифметического как общего показателя для массива данных.

2. АЛГОРИТМ ДЕТЕКТИРОВАНИЯ ДУБЛИРОВАНИЯ ВИДЕОРЯДОВ

Для детектирования необходимо произвести поиск похожих последовательностей в списках, содержащих средние цвета извлеченных кадров видеофайлов. Перед выполнением проверки необходимо подготовить цветовые последовательности. Похожесть определяется максимальным количеством кадров со средними цветами, отличающихся более, чем заданное значение. Как было сказано выше, разность цветов определяется в зависимости от цветовой модели.

Таким образом, алгоритм может быть описан кодом на Java, представленным ниже. В нём осуществлено полное поэлементное сравнение двух списков list1 и list2, содержащих значения средних цветов кадров.

```
for (int i = 0; i <= list1.size() - minFragmentDuration; i++) {
  for (int j = 0; j <= list2.size() - minFragmentDuration; j++) {
    distance = getRGBorHSVDistance(list1.get(i), list2.get(j), colorModel);
    if (distance < averageDifference) {
      VideoFragment vf = new VideoFragment(i, j, distance , 1);
      vf.setPossibleDuration(list1.size(), list2.size());
      checkFragmentSimilarity(vf);
     }
   }
}
```

В коде используются переменные и функции:

- minFragmentDuration – количество кадров в последовательности, при которой считается, что найден дублирующийся видеофрагмент;
- getRGBorHSVDistance – метод, возвращающий различие значений средних цветов в зависимости от используемой цветовой модели; для цветовой модели RGB различие значений определяется как евклидово расстояние: $\mathrm{distance} = \mathrm{red}2 + \mathrm{green}2 + \mathrm{blue}2$. Для цветовой модели HSV вычисляется модуль разности для каждого значения Hue, Saturation, Value: $\mathrm{distance} = \Delta\mathrm{Hue}$, $\mathrm{distance} = \Delta\mathrm{Saturation}$, $\mathrm{distance} = \Delta\mathrm{Value}$;
- averageDifference – параметр различия средних цветов кадров; значения параметра различаются для цветовых моделей RGB и HSV; если для кадров distance < averageDifference, то они считаются похожими.

Пусть даны списки средних цветов кадров двух видео. Списки обрабатываются по вложенному циклу – вычисляется разность значений distance и сравнивается с установленным параметром averageDifference.

Если для сравниваемых кадров distance>averageDifference, то алгоритм продолжает поиск по циклу (Рис. 1).

Если встречается пара элементов из двух списков с разностью, меньшей указанного параметра distance<averageDifference, то предполагается, что найден дублирующийся видеофрагмент (Рис. 2). Создается экземпляр класса Fragment, в конструкторе которого указывается информация о найденном фрагменте: индексы первых кадров для обоих видео, параметр сходства, длительность. Каждая следующая пара элементов из двух списков сравнивается между собой, и, если значения их средних цветов близки, обновляются данные о фрагменте: длина, значение сходства.

Рисунок 1 – Схема работы алгоритма при distance<averageDifference

Рисунок 2 – Схема работы алгоритма при distance > averageDifference

Если встречается пара кадров, для которых distance>averageDifference, то попарное сравнение завершается, в ранее созданный объект класса Fragment записываются индексы конца видеофрагмента. Найденный видеофрагмент считается дублирующимся и заносится в список результатов, если его длина duration больше или равна установленному значению minFragmentDuration.

```
if (ed < averageDifference) {
  vf.setDuration(duration + 1);
  vf.setSimilarity(ed);
  checkFragmentSimilarity(vf);
} else {
  if (duration >= minFragmentDuration) {
     vf.setEndFrameOne();
     vf.setEndFrameTwo();
     fragments.add(vf);
  }
}
```

После этого алгоритм возвращается к вложенному циклу, и продолжается поиск.

3. ОПРЕДЕЛЕНИЕ ОПТИМАЛЬНЫХ ПАРАМЕТРОВ РАБОТЫ АЛГОРИТМА

Эффективность работы алгоритма поиска дубликатов видео зависит от выбора цветовой модели и параметров seconds_between_frames, average_difference, min_fragment_duration.

В тестовой выборке видеофайлов использованы дубликаты с небольшими искажениями, экранные снимки, видео с низким качеством.

average_difference определяет величину расстояния между значениями средних цветов двух кадров. Необходимо задать такое значение параметра, при котором алгоритм будет считать дубликатами извлеченные кадры видео и экранные снимки, сделанные в один и тот же короткий промежуток времени.

Два кадра, извлеченные из видео с промежутком в 1–3 секунды, можно считать похожими. Из видеофайлов извлечены 10 таких пар кадров и определено различие средних цветов для каждой пары в разных цветовых моделях. Результаты представлены в Таблице 1.

На одном и том же моменте видеоролика извлекается кадр и производится экранный снимок. Это моделирует ситуацию распознавания «экранок» видео. Сравнение среднего цвета производилось на 10 парах таких изображений. Результаты представлены в Таблице 2.

Таблица 1 – Различие средних цветов похожих кадров

Пары кадров	Параметр сравнения кадров			
	RGB	HSV (Hue), 10-4	HSV (Saturation), 10-4	HSV (Value), 10-4
1	2.1	3.7	11	3.28
2	0.13	4.4	8.9	3.72
3	0.08	3.1	9.7	2.27
4	0.12	4.1	10.7	4.05
5	0.19	5.5	11.9	3.36
6	0.18	2.3	6.3	3.48
7	0.16	3.8	7.25	3.26
8	0.09	2.5	3.46	2.97
9	2.08	1.7	7.85	2.53
10	1.13	2.9	9.2	3.78

Таблица 2 – Разница средних цветов кадра и экранного снимка

Пары кадров	Параметр сравнения кадров			
	RGB	HSV (Hue), 10-2	HSV (Saturation), 10-2	HSV (Value), 10-2
1	23	3	0.54	2.64
2	45	10	0.77	1.78
3	39	8	1.22	1.55
4	17	22	0.93	2.93
5	28	15	0.57	3.15
6	53	19	0.71	1.69
7	48	9	1.24	1.05
8	25	30	0.73	1.89
9	34	7	1.51	2.48
10	39	14	0.86	1.15

Также оценено различие среднего цвета изображений, не являющихся дубликатами. Для пространств HSV и RGB полученные результаты представлены в Таблице 3.

Таблица 3 – Различие средних цветов различных изображений

Параметр сравнения кадров	Среднее значение разницы цветов	Минимальное значение разницы средних цветов
RGB	38.35	26.48
HSV (Hue)	0.034	0.0165
HSV (Saturation)	0.0413	0.0328
HSV (Value)	0.0529	0.0356

Для цветовой модели RGB и параметра Hue пространства HSV вычисленная разность для экранных снимков и изображений, не являющихся дубликатами, очень близка. Из этого следует, что поиск по данным параметрам будет давать ошибочные результаты для экранных съемок. Сравнение изображений по параметру Saturation является оптимальным при определении экранных снимков. С учетом погрешностей измерений установлены следующие значения average_difference при различных параметрах сравнения: Hue – 0.02 (не применимо для поиска экранных съемок), Saturation – 0.03, Value – 0.03.

Параметр seconds_between_frames устанавливает число секунд между извлечением кадров.

Значение параметра должно определяться с учетом следующих условий:

1. частота извлечения должна быть достаточной для обнаружения дубликатов кадров;

2. при увеличении значения seconds_between_frames уменьшается число извлеченных кадров, что способствует увеличению производительности.

На Рис. 3 представлен график зависимости разницы средних цветов от интервала извлечения seconds_between_frames.

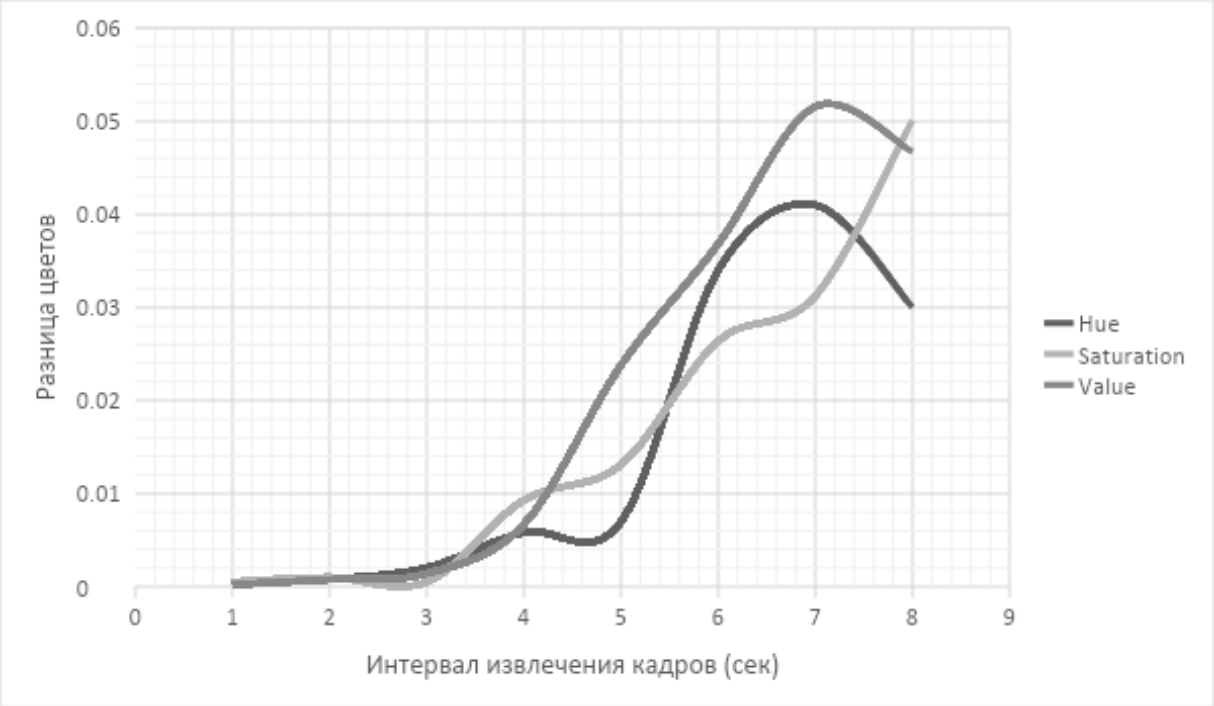


Рисунок 3 – Зависимость разницы среднего цвета кадров от интервала извлечения из видео

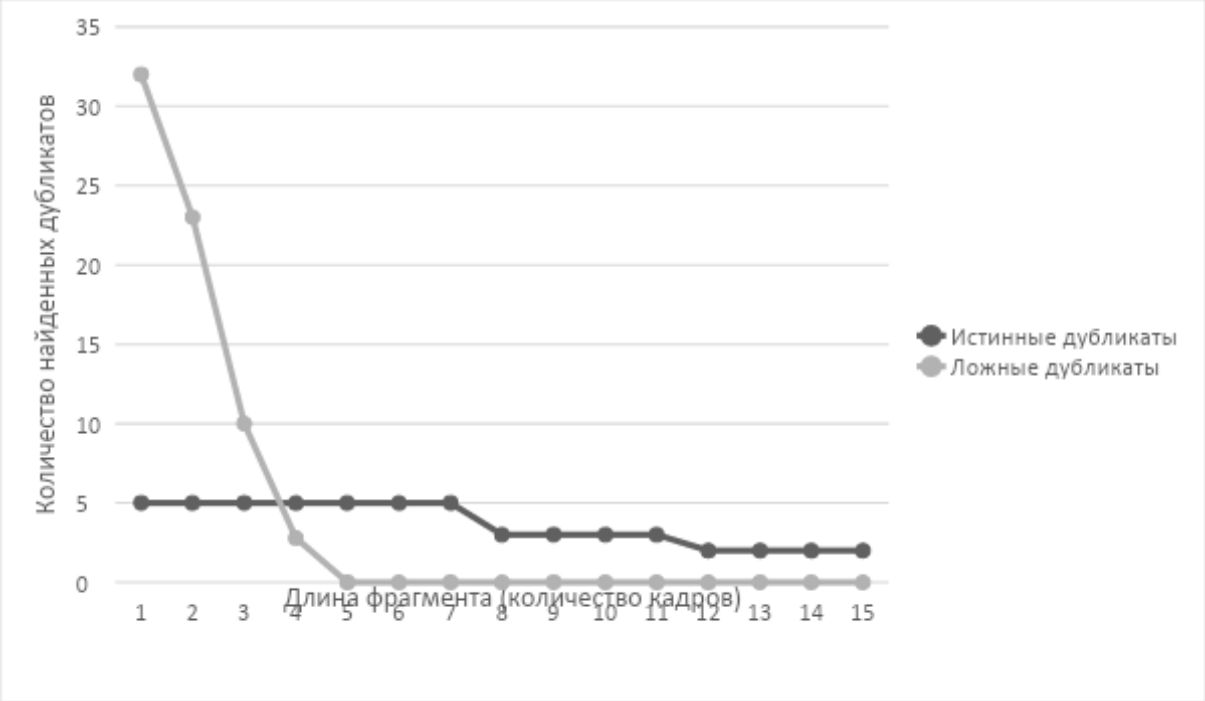


Рисунок 4 – Зависимость разницы эффективности поиска дубликатов от длины искомого фрагмента

На Рис. 4 представлен график зависимости эффективности поиска дубликатов от длины искомого фрагмента min_fragment_duration.

Согласно полученным результатам, оптимальное значение для параметров seconds_between_frames и min_fragment_duration соответственно равно 4 и 5.

4. ИСКЛЮЧИТЕЛЬНЫЕ СИТУАЦИИ В РАБОТЕ АЛГОРИТМА

Существуют различные методы обхода алгоритмов поиска дубликатов видео. Основной способ состоит в искажении исходного видео. При этом следует учитывать, что искаженное видео является нечетким дубликатом только в том случае, если оно сохраняет визуальное содержимое оригинального видео. Поэтому видео, подвергшиеся сильным модификациям и потерявшие целостность содержания, не представляют информационной ценности как дубликаты и не попадают под действие алгоритма поиска нечетких дубликатов.

Возможны следующие варианты искажений:

1. добавление рамки в видео;
2. поворот и растягивание видео в кадре;
3. наложение фильтров на видео.

Для обнаружения рамок, цветовых вставок, яркостных, геометрических искажений или их комбинаций следует добавить предобработку кадров после их извлечения из видеоряда. Это позволит удалять либо игнорировать такого рода искажения.

Существует метод предварительной обработки изображений в рамках подхода, основанного на вычислении хэш-значений в скользящем окне [3]. Предварительная обработка заключается в препарировании исходного изображения при помощи одного из преобразований: снижение яркостного диапазона, вычисление градиента, разложение поля яркости по ортонормированному базису, адаптивное линейное контрастирование, локальный бинарный шаблон. Исследования, проведенные в работе [4], показали устойчивость большинства преобразований к яркостному сдвигу.

5. ОПРЕДЕЛЕНИЕ ПАРАМЕТРОВ ДУБЛИРОВАНИЯ ВИДЕОФАЙЛОВ

Результат работы алгоритма представляет собой информацию о наличии дублирующихся последовательностей для каждого видео: длина (количество кадров), индексы начала и конца фрагмента, значения сходства тона, насыщенности и яркости. На основе этих данных составляется описание параметров дублирования видеофайлов и формируется отчет.

По длине фрагмента определяется процент дублирования от общей длительности видео. Информация о величине параметра seconds_between_frames позволяет найти длину фрагмента в секундах и временный интервал дублирования по индексам начала и конца фрагмента в списке кадров.

По полученным параметрам производится оценка характера дублирования:

1. полный дубликат – более 90% дублирования;

2. частичный дубликат – найден один или более фрагментов дублирования, процент общего дублирования менее 90%;

3. экранная съемка – определяется по среднему значению сходства видео.

В результате анализа параметров дублирования видеофайлов генерируется отчет о проверке, имеющий следующую структуру:

1. длина дублирования,

2. процент дублирования от общей длительности,

3. интервал(ы) дублирования,

4. характер дублирования (полный дубликат, частичный дубликат, экранная съемка).

6. ТЕСТИРОВАНИЕ

Для тестирования системы использовался набор из 50 видео, взятых с видеосервиса YouTube, содержащий дубликаты видео, созданные с помощью обрезки, искажения, экранной съемки оригинальных видео. Скорость подготовки цветовых последовательностей в зависимости от длительности видео приведена в Таблице 4.

Таблица 4 – Зависимость времени обработки от длительности видео

Длительность видео (min)	0.5	1	2	5	10	20	60
Время извлечения кадров (s)	2	5	12	31	84	218	718
Время вычисления среднего цвета (ms)	8	19	27	74	168	385	1384

Для оценки времени работы алгоритма поиска дубликатов был проведён поиск дубликатов в базе видео методом полного перебора, то есть производилось сравнение каждого фильма с каждым (меньшее по длине видео искалось в большем) за исключением сравнения с самим собой. Таким образом, для базы из N фильмов выполнялось N(N–1)/2 сравнений. Время обработки видео не учитывалось.

Для оценки качества работы алгоритма использовались величины Полнота (Recall) и Точность (Precision): Recall=M/R, Precision=M/F, где M – количество правильно найденных дубликатов, R – действительное количество дубликатов, F – количество дубликатов, найденное алгоритмом.

При помощи изменения порога сравнения меры схожести сцен average_difference для HSV(Saturation) был получен график Recall/Precision, представленный на Рис. 5. Максимальная величина F-меры равна 0,839.

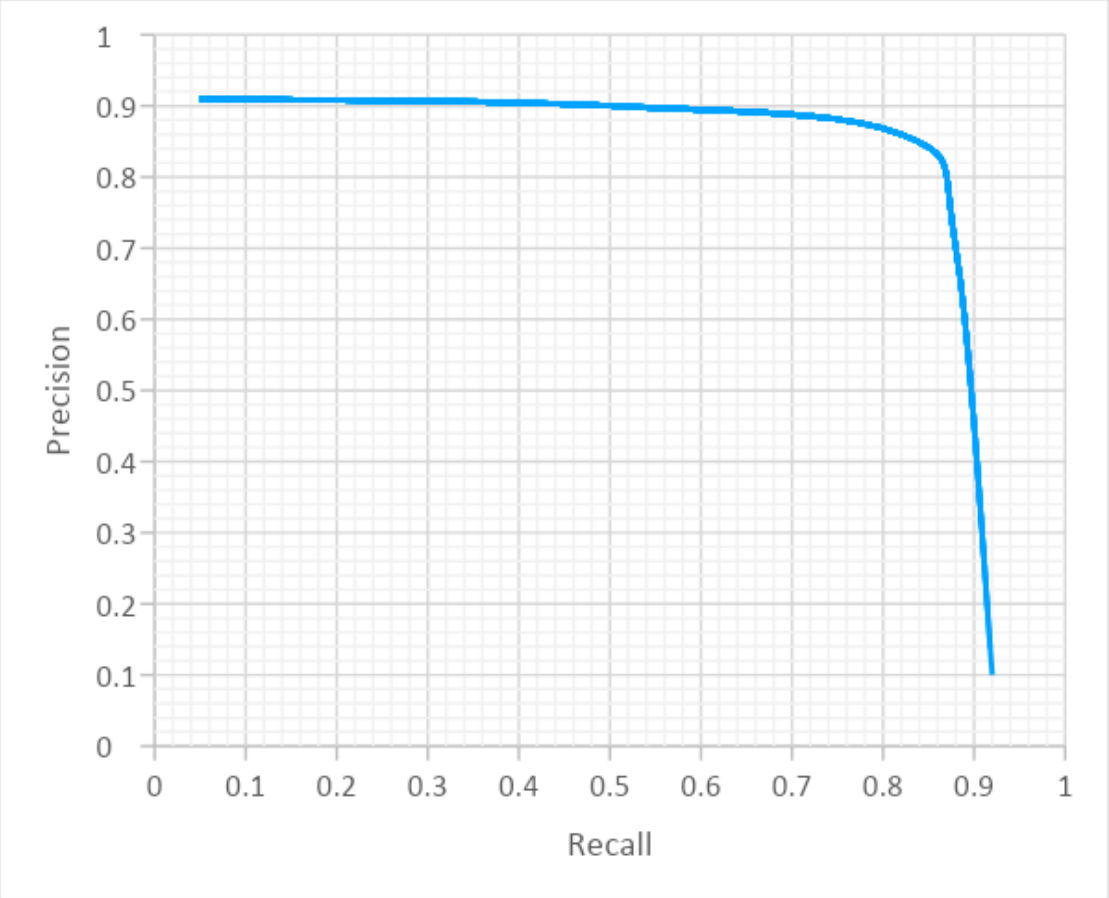


Рисунок 5 – Отношение Recall/Precision

ЗАКЛЮЧЕНИЕ

Программа обнаружения дублирования видеофайлов способна распознавать экранную съемку, частичное дублирование, зеркальные копии и устойчива к различным искажениям цифрового видео, таким, как изменение качества изображения, размеров, ориентации видео (повороты), добавление субтитров и небольших логотипов и т. д.

Модуль определения параметров дублирования обрабатывает данные, полученные в результате обнаружения дубликатов, и формирует отчет, содержащий информацию о параметрах и характере дублирования.

Таким образом, система позволяет не только детектировать наличие дублирования видеофайлов, но и измерить количественные и качественные показателей дублирования, за счет чего можно построить индивидуальную политику в отношении дубликатов видео.

СПИСОК ЛИТЕРАТУРЫ

1. What Is Web 2.0 [Электронный ресурс] URL: https://www.oreilly.com/pub/a/web2/archive/what-is-web-20.html
2. Youtube [Электронный ресурс] URL: https://www.youtube.com/
3. Мясников В.В., Глумов Н.И. Поиск дубликатов на цифровых изображениях // Компьютерная оптика, 2013, Т. 37, № 3, С. 360–367.
4. Кузнецов А.В., Мясников В.В. Исследование методов предварительной обработки изображений в задаче обнаружения дубликатов на изображении// Сборник трудов III международной конференции и молодежной школы «Информационные технологии и нанотехнологии», Самара. 2017. С. 904–911.

DEVELOPMENT OF A SYSTEM FOR DETECTING VIDEO DUPLICATES BASED ON THEIR COLOR MAPS

G. A. Nurieva[1], A. A. Ferenetz[2]

[1-2] Higher School for Information Technologies and Intelligent Systems of Kazan (Volga Region) Federal University

[1]nurievag97@gmail.com, [2]ist.kazan@gmail.com

Abstract

This article describes the development of a system for detecting duplicate video files based on the method of obtaining and analyzing the color map of a video sequence, as an approach that does not require large computational resources, is applicable to a wide range of types of video clips and has small distortions relatively to the original.

Keywords*: video file, color map, duplicate detection*

REFERENCES

1. What Is Web 2.0 https://www.oreilly.com/pub/a/web2/archive/ what-is-web-20.html
2. Youtube https://www.youtube.com/
3. Myasnikov V.V., Glumov N.I. Poisk dublikatov na cifrovyh izobrazhe-niyah // Komp'yuternaya optika, 2013, T. 37, № 3, S. 360–367.

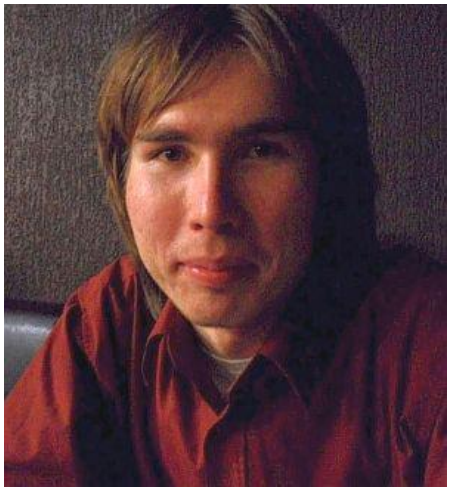
4. Kuznecov A.V., Myasnikov V.V. Issledovanie metodov predvaritel'-noj obrabotki izobrazhenij v zadache obnaruzheniya dublikatov na izobrazhenii// Sbornik trudov III mezhdunarodnoj konferencii i molodezhnoj shkoly «Infor-macionnye tekhnologii i nanotekhnologii», Samara. 2017. S. 904–911.

СВЕДЕНИЯ ОБ АВТОРАХ

НУРИЕВА Гульшат Аталасовна – выпускник бакалавриата Высшей школы информационных технологий и интеллектуальных систем Казанского (Приволжского) федерального университета.

Gulshat Atlasovna NURIEVA – graduate bachelor of Higher School of ITIS KFU.

email: nurievag97@gmail.com



ФЕРЕНЕЦ Александр Андреевич – ассистент, преподаватель кафедры программной инженерии Высшей школы информационных технологий и интеллектуальных систем Казанского (Приволжского) федерального университета.

***Alexande*r Andreevich FERENETS** – assistant at Department of Software Engineering of Higher School of ITIS KFU

email: ist.kazan@gmail.com

Материал поступил в редакцию 15 августа 2019 года