Поиск видео по изображению - технология "Video Color"

А.П. Алексеев[1], Т.В. Алексеева[1]

[1] ИТ-компания "AAP Software"

Аннотация. Перед всеми нами ежедневно встаёт задача поиска информации. Требуется найти текст, изображения, аудио или видео информацию. Чаще всего для поискового запроса используется текст. Реже - изображения. Есть сервисы вроде "Shazam", которые ищут музыку используя запись звука. Мы сосредоточились на создании поискового сервиса, который осуществляет поиск видео. В качестве параметров для запроса мы используем изображения.

Ключевые слова: Video Color, поиск видео, поиск видео по изображению, поиск видео по скриншоту, технология видео поиска

Video search by image - technology "Video Color"

A.P. Alekseev[1], T.V. Alekseeva[1]

[1] "AAP Software" IT company

Abstract. We all face the challenge of finding information every day. It is required to find text, images, audio or video information. Most often, text is used for the search query. Less often - images. There are services like "Shazam" that search for music using sound recording. We focused on creating a search service that searches for videos. We use images as parameters for the request.

Keywords: Video Color, Video Search, Video Search by Image, Video Search by Screenshot, Video Search Technology

1. Постановка задачи поиска видео по изображению

Имеется набор исходных видео файлов. Имеется также набор изображений из этих видео файлов, взятых случайным образом. Необходимо для этих изображений определить из какого именно видео они взяты и указать точную позицию кадра (времени).

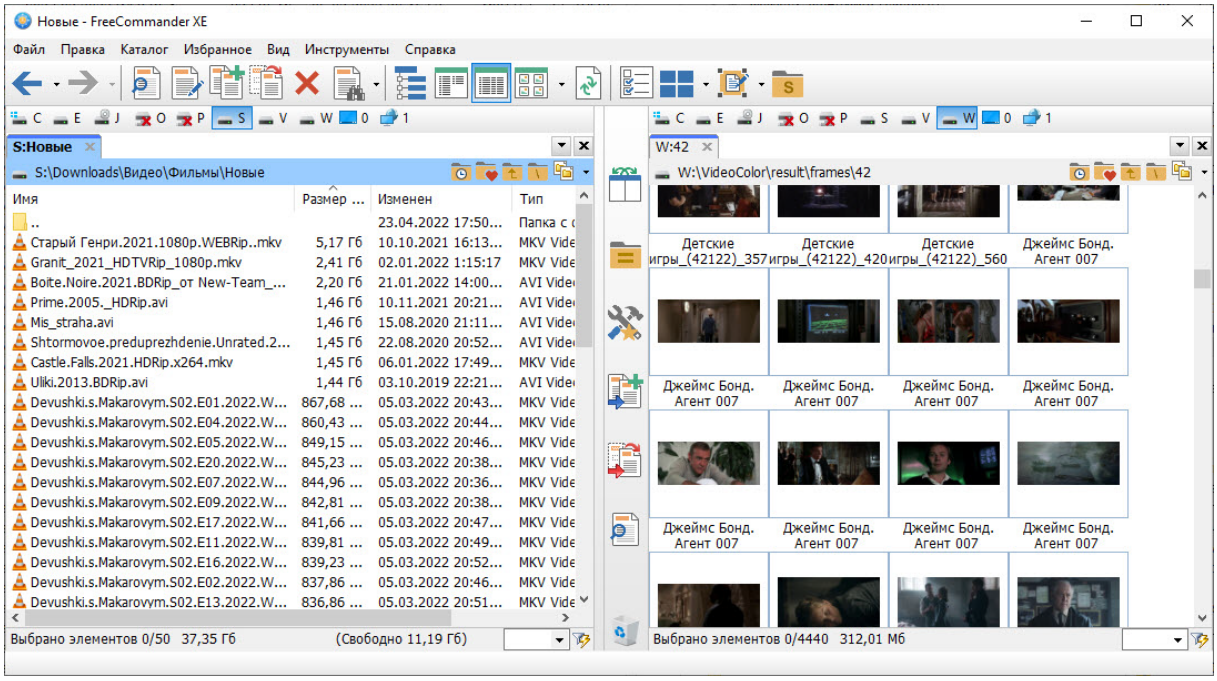

Рис. 1. На левой панели каталог с видео, на правой - каталог с изображениями

2. Область применения

- Идентификация старых и неизвестных видео фильмов.
- Нахождение и отсечение рекламных блоков.
- Проверка частей видео на предмет заимствования их из других фильмов (плагиат).
- Определение точной даты публикации и названия шоу (передачи) если в репосте отсутствует данная информация. Определение более-менее точной позиции проигрываемого потокового видео, если идёт вещание ранее проиндексированного видео.

3. Простейший алгоритм поиска кадра по скриншоту

Простейший алгоритм поиска заключается в том, что каждый кадр в видео рассматривается как отдельное изображение по которому может вестись поиск. Каждое индексируемое, а затем и каждое искомое изображения, делятся на табличные области и в каждой из её ячеек

находятся усреднённые значения компонент красного, зелёного и синего цветов. По ним, в дальнейшем, можно производить сравнение для нахождения искомого кадра методом перебора всех имеющихся записей.

Рис. 2. Разбивка кадра на табличные ячейки

4. Проблема поиска видео когда проиндексированный образец и скриншот из видео имеют разные соотношения сторон

Рис. 3. Различная ширина кадров

Стандартным соотношением сторон для телефильмов является соотношение **4:3**. В то же время для кинотеатров соотношения были более широкие, причём их было несколько. В последние годы стандартом де факто стало соотношение сторон **16:9**. Это соотношение используется в современных телевизорах, мониторах, видеокамерах и фотоаппаратах. Тем не менее обозначилась тенденция использования более широкоэкранных телевизоров и мониторов, смартфонов и планшетов, а также съёмочной аппаратуры. С другой стороны, в то же время, некоторые популярные сервисы используют очень узкие соотношения сторон **1:1** и даже используют альбомную ориентацию, подстраиваясь под пользователей смартфонов.

Если всё исходное видео обрабатывать по схеме **M*N** областей, то можно столкнуться с ситуацией когда видео проиндексировано (данный фильм присутствует в базе данных), а поиск по кадрам из этого видео не приносит положительных результатов. Это может случиться если соотношение сторон проиндексированного видео и кадра, по которому осуществляется поиск, различно.

5. Проблема поиска видео, связанная с логотипами каналов

Следующая проблема связана с тем, что одни и те же фильмы и сериалы могут транслироваться на разных ТВ-каналах и содержать их логотипы. А могут и не содержать и это следует учитывать.

Рис. 4. Примеры кадров видео, содержащих логотипы ТВ-каналов

6. Проблема поиска видео, связанная с большим размером данных

Согласно базе данных по кинематографу IMDb (на 2018 год), всего было снято около 2,6 миллионов кинолент, включая отдельные эпизоды сериалов, мультфильмы и короткометражки. Из этого колоссального числа можно выделить около 350 тысяч полнометражных кинокартин, около 390 тысяч короткометражных лент, около 97 тысяч сериалов (более 1 миллиона эпизодов) и около 67 тысяч документальных кинолент. Принято считать, что крупнейший в своем роде ресурс IMDb предоставляет наиболее корректную по данному вопросу информацию.

Прежде чем перейти к основной теме имеет смысл взглянуть на проблему со стороны.

- Сколько кадров содержит среднестатистический видео фильм?
- Сколько фильмов должно быть в базе данных, чтобы пользователи начали пользоваться данным сервисом?
- Сколько времени пользователь готов ждать ответа от поисковой системы?

Попробуем ответить на эти вопросы.

- 150 000 кадров содержит среднестатистический фильм.

- 1 000 000 видео — столько должна содержать современная база данных, чтобы быть востребованной.
- Большинство пользователей готовы ждать ответа от поисковой системы не более секунды.

Получается, что в нашей базе данных должно быть порядка 150 миллиардов кадров. Это очень много! Как осуществить поиск за доли секунды по такой огромной таблице?

7. Обзор текущей ситуации с поиском видео с использованием изображений

На данный момент есть возможность поиска по картинкам. Для поиска необходимо загрузить изображение и получим в ответ набор похожих изображений с ссылками. Вот список основных игроков:

- MicroSoft (bing.com)
- Yandex (yandex.ru)
- Google (google.com)
- TinEye

MicroSoft и Yandex лучше других осуществляют поиск по картинкам. Но результат, к сожалению, не всегда положительный. Если известные фильмы проиндексированы довольно неплохо и есть высокая вероятность по сделанному скриншоту найти описание фильма, то с менее популярными фильмами ситуация заметно хуже.

Стоит отметить, что если в кадре есть лицо известного человека крупным планом, то его почти наверняка идентифицируют и покажут множество похожих изображений именно этого человека.

В случае, если на экране смазанное изображение или редкие неопределённые предметы, то вероятность успешного поиска близка к нулю.

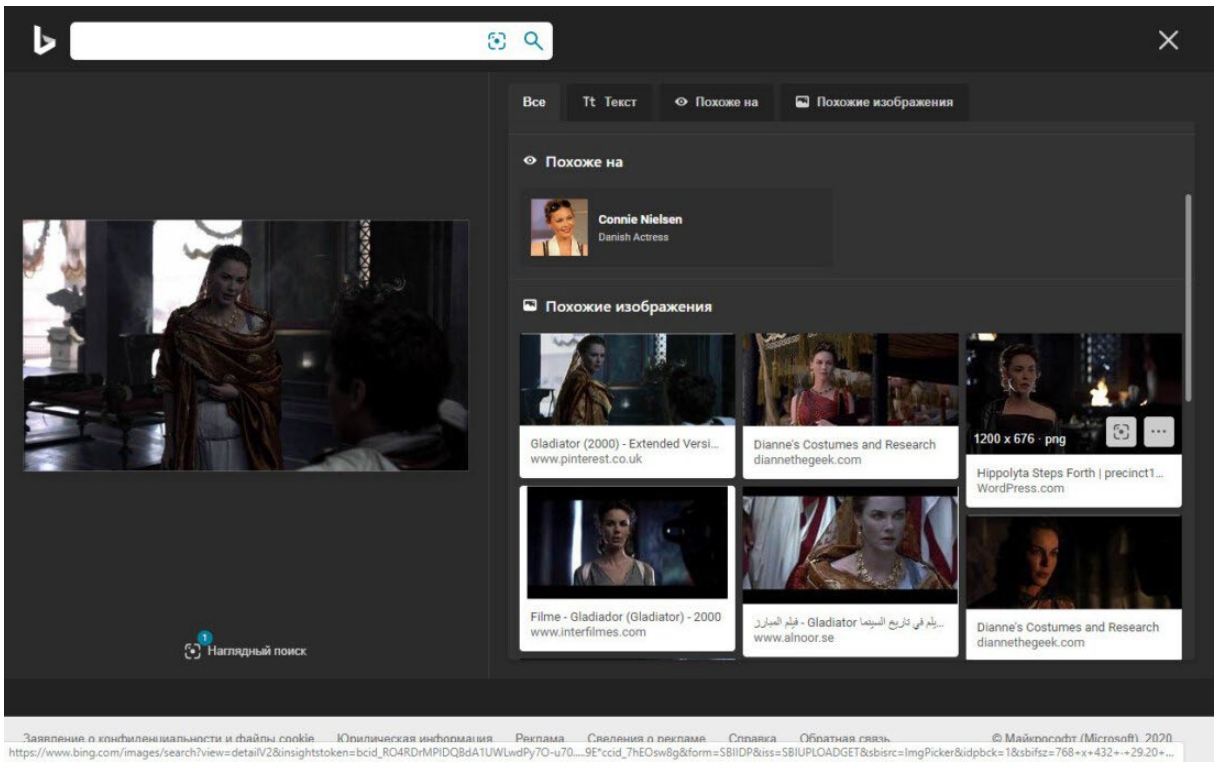

Рис. 5. Успешный результат поиска на сайте Bing.com

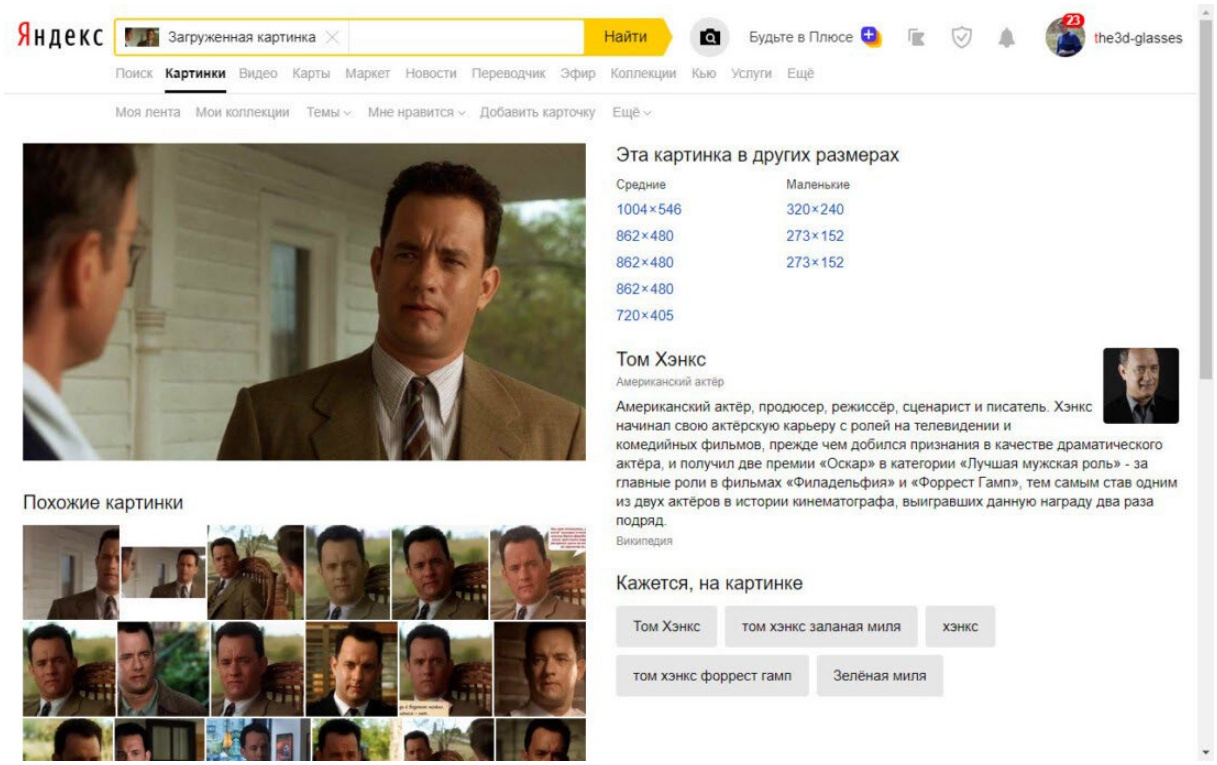

Рис. 6. Успешный результат поиска на сайте Yandex.ru

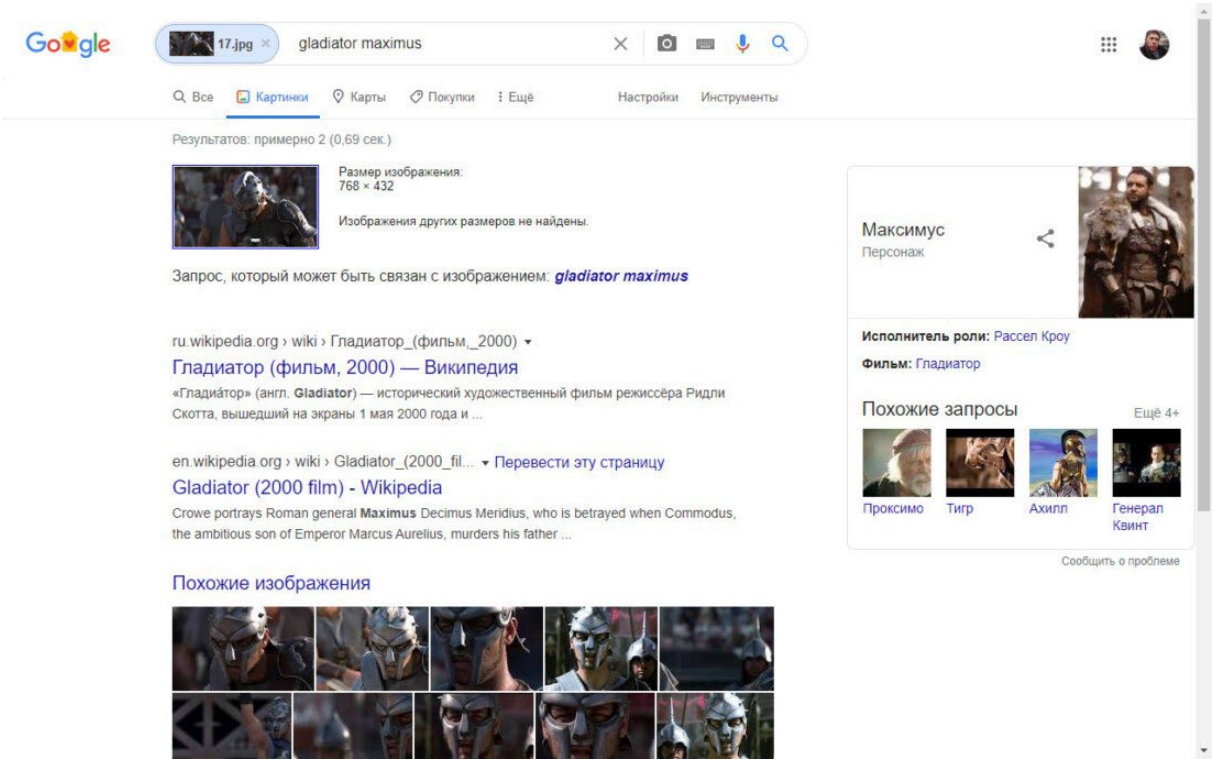

Рис. 7. Успешный результат поиска на сайте Google.com

8. Технология поиска "Video Color"

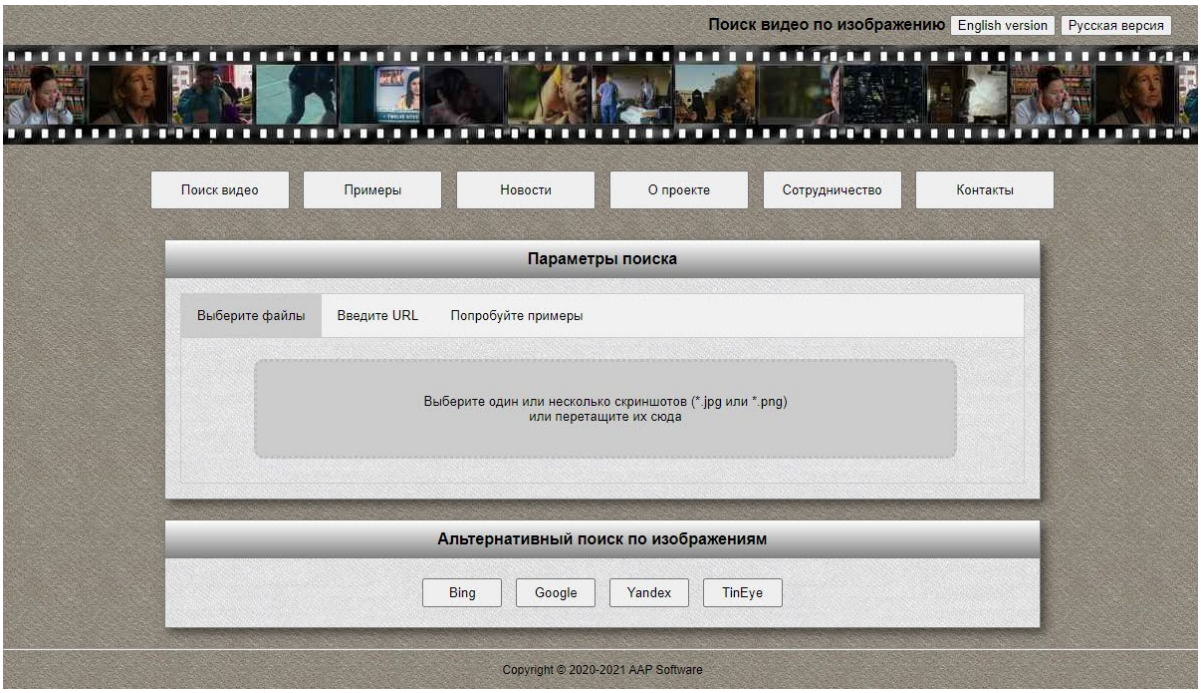

Рис. 8. Поисковая форма на сайте https://www.videocolor.aapsoftware.ru

Технология поиска «Video Color» заключается в том, что решает целый ряд перечисленных выше проблем, а также ряд других, а именно:

- Проблема поиска видео когда проиндексированный образец и скриншот из видео имеют разные соотношения сторон.
- Проблема поиска видео, связанная с логотипами каналов.
- Проблема поиска видео, связанная с большим размером данных.
- Проблема неоднозначности при использовании индексных хэшей.
- Проблема неоднозначности при использовании перцептивных хэшей.

9. Результаты поиска

В случае успешного поиска пользователь получает следующую информацию:

- Название фильма
- Режиссёр
- Страна
- Год выпуска.
- Жанр
- Список актёров
- Описание
- Продолжительность видео
- Позиция кадра по которому осуществляется поиск в видео
- Скринлист видео.

Подробности на представленном ниже скриншоте.

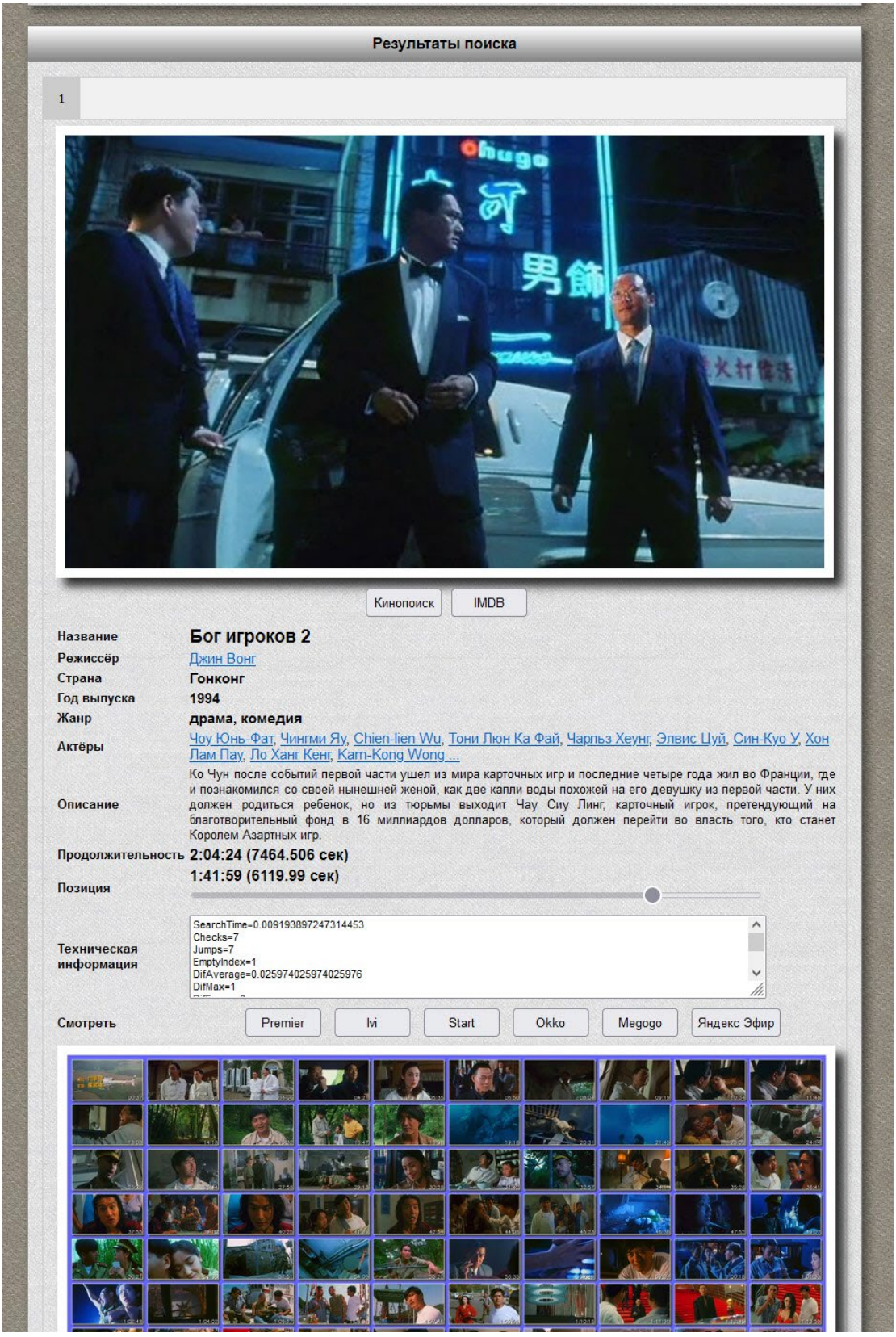

Рис. 9. Положительные результаты поискового запроса

9. Использование индексных хэшей для ускорения поиска

Для решения указанной проблемы есть решение и оно заключается в использовании хэшей. Хэш — это некоторое число, которое получается путём применения хэш-функции к исходным данным. Использование хэшей даст нам возможность ускорить поиск нашего кадра в огромном массиве информации.

В качестве исходных данных можно использовать усреднённые значения RGB определённых областей заданного изображения. А сам хэш можно сделать 32-битовым беззнаковым целым числом.

10. Проблема неоднозначности при использовании индексных хэшей и её решение

При вычислении хэшей нам необходимо привести усреднённое значение к целому числу, а вот здесь возможны неприятности.

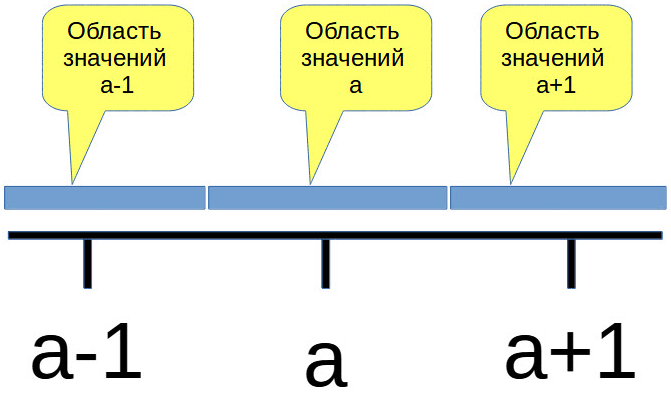

Рис. 10. Вычисляемые средние значения

Поскольку при перекодировании видео под разное разрешение или битрейт получаемые при просмотре кадры немного отличаются от оригинала, то может случиться следующее. Хэш от исходного изображения, возможно, будет отличаться от хэша перекодированного изображения. Это связано с тем, что усреднённые значения в пограничных областях могут изменяться в большую или меньшую стороны (пусть даже на небольшую величину) и мы можем получить вместо **a** значение **a+1**.

Кроме того, при захвате или экспорте кадра из видео в формат JPEG или другой формат (с потерями) происходит дополнительное изменение исходного кадра.

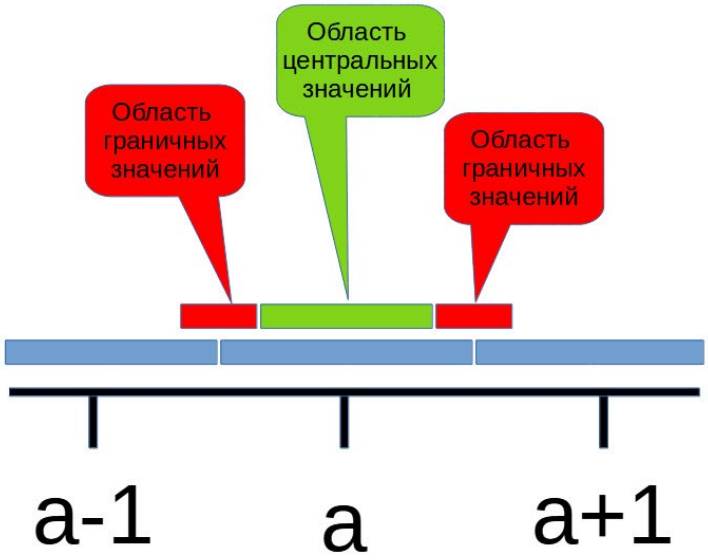

Рис. 11. Области реально вычисляемых значений

В результате мы получаем неоднозначность хэшей от различных версий кадров одного и того же видео. Это реальная проблема.

11. Использование перцептивных хэшей для ускорения поиска

Перцептивное хеширование - это использование алгоритма, который создает фрагмент или отпечаток пальца различных форм мультимедиа.

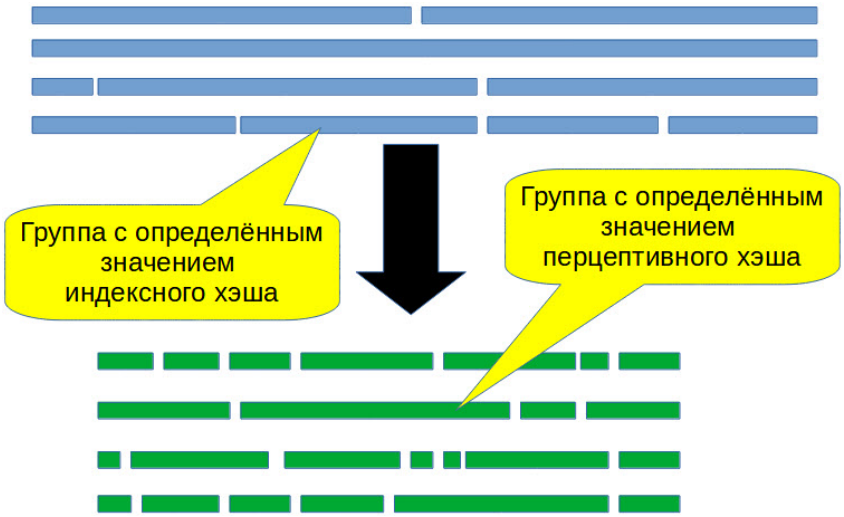

Рис. 12. Расположение записей в базе данных, сгруппированных по индексным хэшам (синие группы) и внутри них по перцептивных хэшам (зелёные группы)

Для чего можно использовать перцептивные хэши? Для ускорения поиска записей, как обычные индексы. Этим можно многократно увеличить скорость поиска. Ну что ж, этим и займёмся.

На картинке видно, что все записи упорядоченны по значению индексного хэша (синие группы). Внутри каждой группы есть подгруппы (выделены зелёным цветом), где записи имеют одинаковое значение перцептивного хэша.

На приведённом ниже изображении можно видеть, как из таблицы **11x7** выбирается 32 центральных элемента (выделены серым цветом). Их можно использовать для построения перцептивного хэша размером 4 байта. В некоторых случаях этого вполне достаточно. Разумеется, можно использовать и любые другие схемы.

0	1	2	3	4	5	6	7	8	9	10
11	12	13	14	15	16	17	18	19	20	21
22	23	24	25	26	27	28	29	30	31	32
33	34	35	36	37	38	39	40	41	42	43
44	45	46	47	48	49	50	51	52	53	54
55	56	57	58	59	60	61	62	63	64	65
66	67	68	69	70	71	72	73	74	75	76

Рис. 13. Кадр, разбитый на табличные ячейки. Серым цветом выделены те ячейки, что используются для вычисления перцептивного хэша

Итак, мы находим средние значения сумм R, G, B в указанных областях, подсчитываем среднее значение для всех выбранных областей S и если оно больше S, то считаем текущий бит равным 1, в противном случае равным 0.

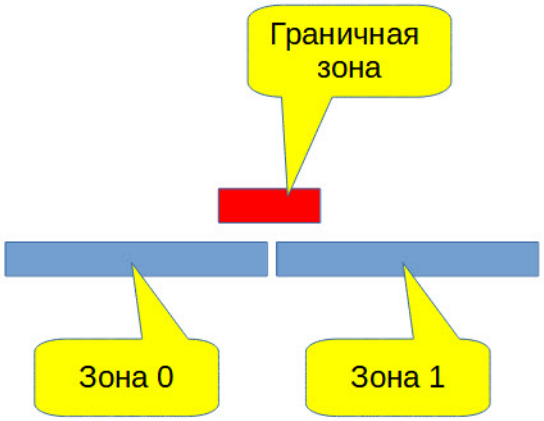

Рис. 14. В граничной зоне значения для одного и того же кадра могут различаться

При вычислении хэша нам необходимо привести усреднённое значение либо к 0 либо 1, и здесь возможны неприятности.

Поскольку при перекодировании видео под разное разрешение или битрейт получаемые при просмотре кадры немного отличаются от оригинала, то может случиться следующее. Значение в той или иной точке будут различны для двух схожих изображений. Это приведёт к тому, что значения перцептивных хэшей будут разные. А это, в свою очередь, приведёт к тому, что мы гарантированно не найдём искомый кадр.

12. Тестирование полученных результатов

Тестирование поиска осуществляется следующим образом. При индексации видео случайным образом сохраняются 10 кадров. Записывается ID видео и позиция в видео. После завершения очередного этапа индексации массива видео, строится файл поисковых запросов на основании обработки каждого сохранённого кадра. Выполняются запросы к пиковому веб-серверу и полученные результаты сравниваются с исходными данными по каждому кадру. На момент последнего тестирования проиндексировано было около 30 000 видео. Тестовых изображений - 295458. В результате выполнения поисковых запросов 291382 кадров (около 98.62%) находятся успешно. Оставшуюся часть 4076 (неуспешные запросы) можно разделить на две части:

- Совпадающие кадры. Это может быть абсолютно чёрный кадр (ночь, эффект перехода, начало фильма и т.п.). В этой ситуации выбрать правильный кадр из множества одинаковых физически невозможно.
- Кадры отличаются, но, индексный хэш и перцептивный хэш одинаковые. Такие ситуации крайне редки, но всё же случаются и обусловлены тем, что усреднение разных кадров может дать одинаковый результат. По сути они дают ошибочный результат, но их на порядки меньше, чем в первой труппе.

13. Отдельные приложения для поиска видео по скриншоту

Помимо поиска на сайте можно использовать приложения для операционных систем Android и Windows. Эти приложения можно загрузить по ссылкам с сайта https://www.videocolor.aapsoftware.ru .

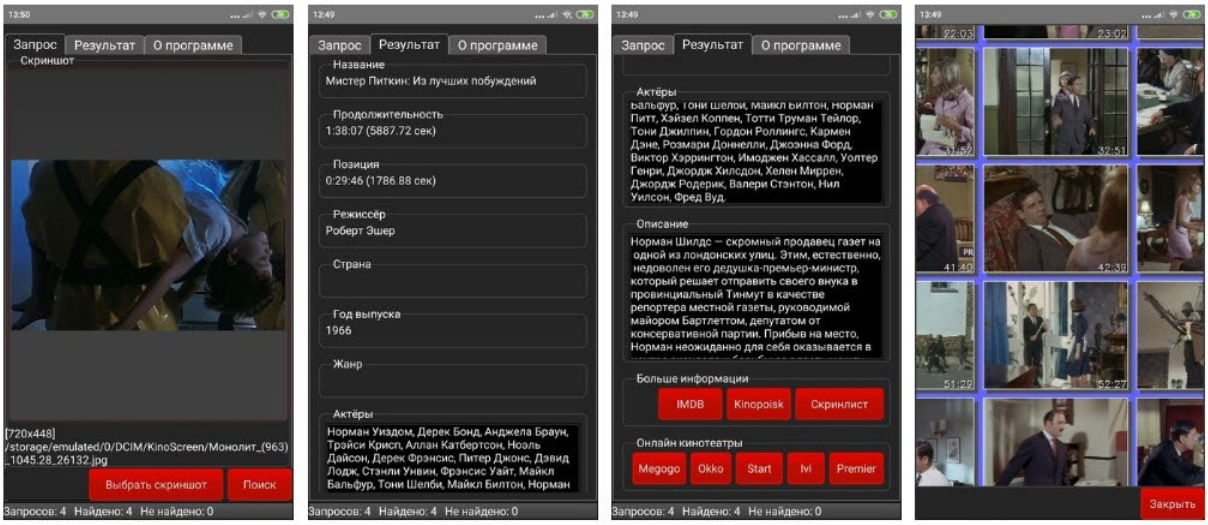

Рис. 15. Приложение для ОС Android для поиска видео по скриншоту

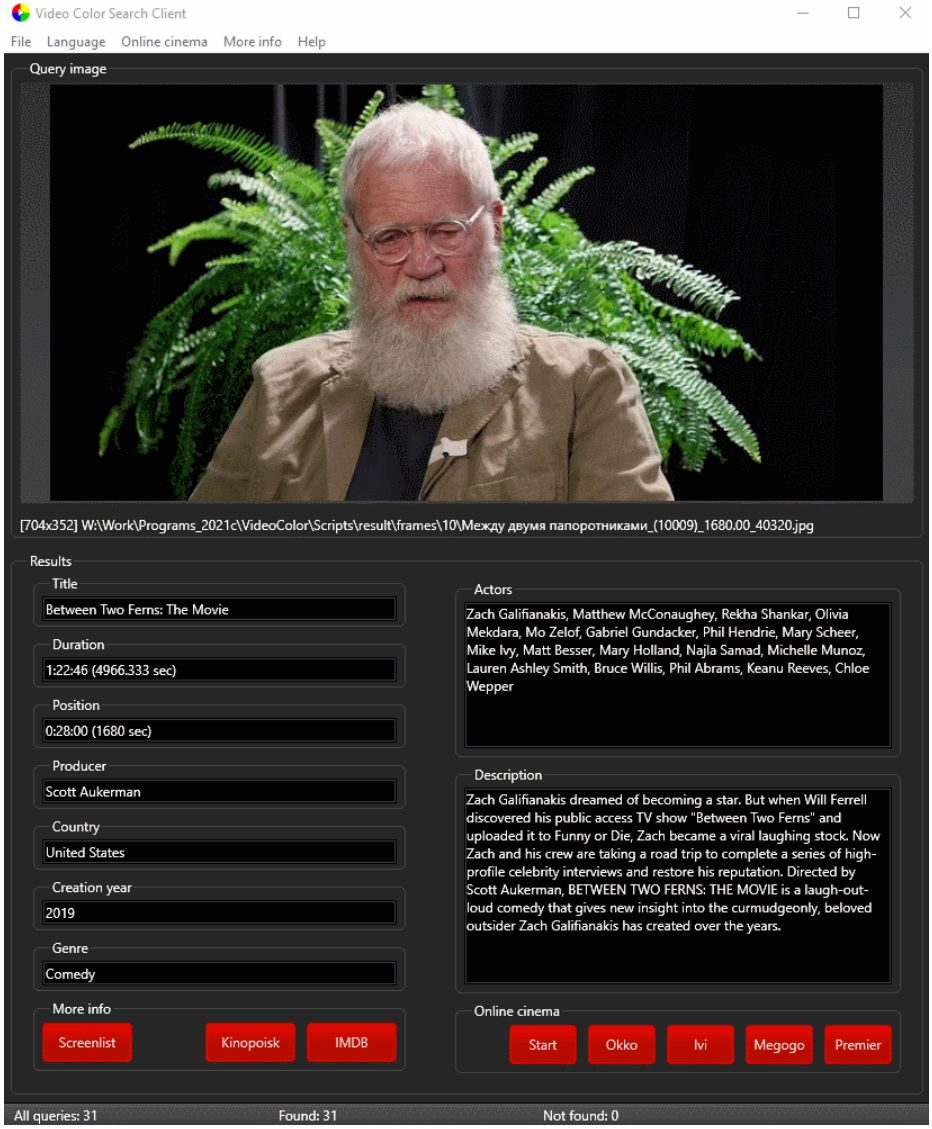

Рис. 16. Приложение для ОС Windows для поиска видео по скриншоту

14. Бесплатная библиотека на PHP для поиска видео по скриншоту

Для поиска видео по скриншоту для сторонних веб-сайтов на PHP была написана библиотека ***Video Color Search Client***.

Есть два варианта получения данной библиотеки:

- GitHub
- Packagist.org

Для клонирования проекта с ***GitHub*** откройте окошко терминала, перейдите в рабочий каталог и выполните следующую команду:

git clone https://github.com/alekseev23/VideoColorSearchClientPHP.git

Для установки пакета с помощью ***Composer*** введите в консоли следующую команду:

php composer.phar require aap_software/video_color_search_client

После её выполнения в каталоге vendor появится подкаталог aap_software. Библиотека содержит описание и пример использования. С её помощью любой желающий может на своём сайте предоставить поиск по скриншоту своим пользователям. Однако количество бесплатных поисковых запросов в сутки имеет ограничение.

15. Что достигнуто на текущий момент

- Есть работающий сайт, содержащий поисковую форму для изображения. Производительность составляет около 100 поисковых запросов в секунду. Адрес сайта: https://www.videocolor.aapsoftware.ru
- Есть база данных проиндексированных видео, содержащая информацию о более чем 30 000 фильмов.

- Помимо поиска на сайте с помощью веб-браузера можно использовать приложение для поиска видео по скриншоту "***Video Color Search Client***" (Windows, Android).
- Написана библиотека "***Video Color Search Client***" на PHP, которая позволяет использовать поисковый функционал на любых других сайтах, правда с ограничениями по количеству запросов. Для решения этой проблемы нужно связаться с владельцем ресурса.

16. Статистика запросов

На текущий момент сервис работает в тестовом режиме. Всё же приведём некоторую статистику по поисковым запросам.

Параметр	Значение
Всего запросов	1 977
Успешные запросы (фильм найден в базе данных)	590
Неудачные запросы (кадр не идентифицирован)	1 387
Уникальных IP-адресов	213
Среднее количество запросов в сутки	10

Большое количество неудачных запросов объясняется относительно небольшой базой проиндексированных видео фильмов.

17. Планы на будущее

- Довести количество проиндексированного видео до 1 000 000 фильмов и эпизодов сериалов.
- Реализовать поиск по короткому видео фрагменту (есть некоторые наработки).
- Попробовать реализовать поисковый функционал по короткому аудио фрагменту, подобно тому, что сделано в приложении **Shazam**.

Литература

1. А.П. Алексеев. Технология видео поиска Video Color // https://medium.com/@grifer163/технология-видео-поиска-video-color-8960214cc911
2. А.П. Алексеев. Технология видео поиска «Video Color» (следующая статья) // https://habr.com/ru/post/517048/
3. А.П. Алексеев. Проблемы поиска кадров в базе данных, связанные с соотношением сторон и их решение // https://habr.com/ru/post/588899/
4. А.П. Алексеев. Использование индексных хэшей для ускорения поиска кадров в базе данных // https://habr.com/ru/post/589013/
5. А.П. Алексеев. Использование перцептивных хэшей для ускорения поиска кадров в базе данных «VideoColor» // https://habr.com/ru/post/589383/
6. А.П. Алексеев. Поисковая система «Video Color» для любителей фильмов // https://habr.com/ru/post/596857/
7. А.П. Алексеев. Библиотека PHP для поиска видео по скриншоту // https://habr.com/ru/post/653371/

References

1. A.P. Alekseev. Video Search Technology "Video Color" // https://medium.com/@grifer163/технология-видео-поиска-video-color-8960214cc911
2. A.P. Alekseev. Video Search Technology «Video Color» (new article) // https://habr.com/ru/post/517048/
3. A.P. Alekseev. Aspect ratio problems in database search and solutions // https://habr.com/ru/post/588899/
4. A.P. Alekseev. Using index hashes to speed up the search for frames in the database // https://habr.com/ru/post/589013/
5. A.P. Alekseev. Using perceptual hashes to speed up the search for frames in the "VideoColor" database // https://habr.com/ru/post/589383/
6. A.P. Alekseev. Search engine "Video Color" for movie lovers // https://habr.com/ru/post/596857/
7. A.P. Alekseev. PHP library for searching video by screenshot // https://habr.com/ru/post/653371/