МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ РОССИЙСКОЙ ФЕДЕРАЦИИ

ФЕДЕРАЛЬНОЕ ГОСУДАРСТВЕННОЕ АВТОНОМНОЕ ОБРАЗОВАТЕЛЬНОЕ УЧРЕЖДЕНИЕ ВЫСШЕГО ОБРАЗОВАНИЯ

«НОВОСИБИРСКИЙ НАЦИОНАЛЬНЫЙ ИССЛЕДОВАТЕЛЬСКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ»

(НОВОСИБИРСКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ, НГУ)

Факультет Механико-математический

Кафедра Программирования

1. Направление подготовки Математика и компьютерные науки

**ВЫПУСКНАЯ КВАЛИФИКАЦИОННАЯ РАБОТА БАКАЛАВРА**

Сухомлинов Дмитрий Игоревич

(Фамилия, Имя, Отчество автора)

Тема работы: Разработка библиотеки поиска по фотографии в базе данных видеоматериалов

**«К защите допущена» Научный руководитель**

Заведующий кафедрой, доцент каф. АФТИ ФФ НГУ,

д.ф.-м.н., профессор лаб. 13 ИАИЭ СОРАН

Марчук А.Г /\_\_\_\_\_\_\_\_ Таранцев И. Г. /\_\_\_\_\_\_\_\_\_\_

(фамилия , И., О.) (подпись, МП) (фамилия , И., О.) (подпись, МП)

«…»………………20…г. «…»………………20…г.

Дата защиты: «…» ……………20…г.

Новосибирск, 2019

**Введение**

В настоящее время видеокамеры стали неотъемлемой частью нашей жизни. Каждый день записывается и сохраняется огромное количество видеозаписей – от любительских, снятых с помощью мобильных телефонов и домашних видеокамер, до профессиональных видеорепортажей, созданных с помощью специализированного видеооборудования.

Подобные видеозаписи содержат большое количество детальной информации – номера автомобилей, различные происшествия и, в первую очередь, лица людей. По наличию человека на том или ином видеоролике можно судить о его местоположении и действиях в тот или иной момент. Благодаря этому появляется возможность быстрого нахождения пропавших людей, обнаружения опасных преступников и т.д.

Однако, вручную обработать огромные массивы данных человеку не под силу – для того, чтобы проверить базы данных на наличие того или иного человека может уйти огромное количество времени. К тому же, зачастую бывает крайне непросто на глаз определить того или иного человека, здесь играют важную роль большое количество факторов – освещение, поворот головы, выражение лица и т.д.

Для решения этого вопроса предлагается создать автоматическую систему обнаружения и сравнения лиц, которая, отработав в автоматическом режиме, выдаст необходимый результат. При этом, система должна быть легко настраиваемой, понятной и дружелюбной по отношению к конечному пользователю.**Постановка задачи**

Целью работы является разработка механизмов создания базы данных по набору входящих видеороликов и поиска в созданной базе данных вхождений лиц с определенной фотографии. Предполагается создать кроссплатформенную библиотеку, легко встраиваемую в любой пользовательский продукт. Данная библиотека должна работать на стандартном ноутбуке (т.е. не должно требовать специализированного программного обеспечения) и должна осуществлять следующие операции:

* Чтение видеоролика с жесткого диска.
* Разбиение видеоролика на последовательность кадров.
* Обнаружение лиц людей, находящихся на каждом кадре.
* Объединение лиц, принадлежащих одному и тому же человеку, в отдельную группу с указанием того, на каких кадрах видеоролика данное лицо встречается, а также «описание лица» в некотором виде.
* Формирование набора вышеописанных объединений и сохранение их в базе данных.
* Поиск в наборе объединений лиц.

Стоит учесть тот факт, что лица мы храним в виде «описаний», сделанных по некоторому шаблону. Качество данного шаблона напрямую зависит от качества изображения. Поскольку, качество изображения может быть разным – от «размытого», снятого на камеру скрытого видеонаблюдения, до очень четкого, сделанного на профессиональную аппаратуру – необходимо сделать возможность настройки порогов сравнения «описаний лиц», хранящихся в нашей базе. Необходимо учитывать, что чем выше порог сравнения, тем больше времени потребуется для необходимого анализа. Пороги сравнения могут подбираться индивидуально для каждого набора входных видеозаписей, зависят от качества видеоролика и могут быть подобраны экспериментально.

Реализация библиотеки распознавания лиц является крайне трудоемкой задачей, поэтому данный вопрос в рамках данной работы не рассматривается. Вместо этого рассматривается задача создания интерфейса для встраивания любой готовой реализации распознавания лиц с последующим его применением на практике.

Для четкого понимания происходящего в ходе работ, библиотека должна иметь подробную систему логирования и интуитивно понятный интерфейс, не требующий долгой предварительной настройки.

Разрабатываемая библиотека должна обеспечивать:

* Минимальную погрешность при составлении базы «описаний лиц» и поиску по базе (ошибки допустимы только при неправильно подобранных порогах сравнения и прочих не консистентных настройках).
* Поиск лица человека по базе данных за минимально возможное время.
* Вывод результирующих данных в наиболее удобном формате.
* Простоту настройки.

Необходимо протестировать приложение для того, чтобы выяснить границу его применимости: наилучшие значения порогов сравнения, при которых минимизируются как вероятность ошибки сравнения, так и время работы; наилучшие изображения с точки зрения распознавания лиц на них.

**Теория (Название)**

Задача поиска лиц на фотографии является одним из основных элементов разрабатываемого приложения. Данная задача наиболее затратна в плане требуемых ресурсов компьютера, однако, от ее реализации зависит успех и качество выполнения второй части, связанной с поиском конкретного лица.

Вышеупомянутая задача тесно связана с задачей преобразования входных данных в необходимый нам формат. Например, видеоролики необходимо обрабатывать покадрово, то есть стоит задача разбиения входного видеоролика на отдельные кадры. Каждый кадр должен отдельно обрабатываться на предмет наличия на нем лиц.

Для реализации описанной задачи предлагается создать два интерфейса – один для считывания входных данных, другой для реализации задачи поиска и распознавания лиц. Каждый из этих интерфейсов включает в себя набор чисто виртуальных функций, требующих реализации в классе-наследнике. В случае интерфейса для считывания входных данных под «наследником» подразумевается реализация для различных форматов видеозаписей. В случае интерфейса для задачи поиска и распознавания «наследником» является реализация интерфейса с использованием конкретной библиотеки распознавания лиц. Данная концепция позволяет абстрагироваться от конкретного формата входных данных и от определенной библиотеки, предназначенной для распознавания лиц.

Из множества имплементированных интерфейсов предполагается создать встраиваемую библиотеку, ориентированную на конечного пользователя. Поскольку реализация библиотеки является задачей программиста, а вопрос использования этой реализации интересует только конечного пользователя, то предполагается поставлять последнему реализацию в виде, во-первых, динамической (статической? Какой угодно?) библиотеки, закрытой для просмотра извне, и, во-вторых, двух интерфейсов, описывающих функции данной библиотеки, как, когда и в каком порядке они должны быть вызваны. Очевидно, что первым должен быть вызван конструктор той или иной имплементации интерфейса. Поскольку, как уже упоминалось, сама имплементация закрыта для конечного пользователя, а конструктор интерфейса не позволит нам сконструировать объект наследника, мы инкапсулируем конструктор интерфейса, сделав его недоступным, а для создания объекта будем использовать статическую функцию, принадлежащую интерфейсу и предназначающуюся для создания экземпляра класса наследника.

В рамках данной работы реализованы интерфейсы для формата видеороликов “.avi” и для библиотеки распознавания лиц Luxand FaceSDK.

**Пример использования библиотеки**

Для начала работы нам необходимо создать два объекта, один – загрузчик входного видеофайла, другой – «Искатель лиц», занимающийся поиском лиц на отдельных изображения и составлением базы:

ILoader\* loader = ILoader::createLoader();

IFaceFinder\* faceFinder = IFaceFinder::createFaceFinder();

Далее работа протекает с этими объектами. Каждый из них должен быть проинициализирован. Инициализация в данном случае – некая предварительная обработка, которая не относится к непосредственно созданию объекта. Она реализуется индивидуально, различные реализации загрузчиков, как и различные обработчики входных файлов могут иметь разные требования:

loader->init();

faceFinder->init();

Далее, загрузчику должен получить сам входной файл. Для этого вызывается соответствующая функция, в которую передается файл в виде пути к нему на жестком диске:

loader->loadFile("path/to/file");

Дальнейшая работа протекает в цикле. Загрузчик файла должен разбить видеозапись на последовательность кадров, причем, он должен иметь информацию об общем количестве кадров и о том, когда они подойдут к концу. Следовательно, нам необходимо выполнять цикл до тех пор, пока кадры не закончатся. Внутри цикла загрузчик предоставляет всю возможную информацию о кадре из видеозаписи – номер кадра; указатель на массив данных, представляющих из себя последовательность пикселей, считанных слева-направо снизу-вверх; длину и ширину картинки, а также ее глубину цвета. Все эти данные передаются в соответствующую функцию «Искателя лиц», где он уже делает свою работу – определяет лица на фотографии, создает по ним «описания лиц» и добавляет их в базу данных не так – сначала создает первичные описания лиц, а потом по выходу из цикла (писать в другом месте) собирает другие описания и сохраняет их в базу:

while (loader->hasFrameToRead()) {

const int curentDataFrameNumber = loader->getLastReadFrameNumber();

unsigned char\* data = loader->readNextFrame();

const int width = loader->getPictureWidth();

const int height = loader->getPictureHeight();

const int scanLine = loader->getSkanLine();

faceFinder->addImage(curentDataFrameNumber, data, width, height, scanLine, IFaceFinder::ColorDepth::Bit24);

delete[] data;

}

Внутри цикла мы обладаем возможностями внести дополнительные параметры обработки. Например, мы можем обрабатывать только каждый 10-й кадр:

if (curentDataFrameNumber % 10 != 0) {

continue;

}

После окончания обработки видеозаписи необходимо завершить работу как загрузчика, так и обработчика лиц. Во время завершения загрузки происходят действия постобработки, окончательно (здесь текст про сохранение – что именно сохраняем) формируется база данных, закрывается входной файл видеозаписи и т.д.

Можно говорить о временной базе данных – сначала собираем все превичные описания в локальную временную базу данных, потом (в finish()) выполняем сравнение, объекдинение и прочее, сокращаем временную базу и переносим из нее в постоянную базу данных только минимум информации.

loader->finish();

faceFinder->finish();

**Сравнение лиц**

Одним из наиболее важных моментов является сравнение двух лиц и определение степени их схожести. Назовем эту задачу верификацией [1]. «Описание лица» представляет из себя некую внутреннюю структуру, определенную в конкретной библиотеке распознавания лиц. Верификация двух лиц происходит посредством сравнения этих двух структур. Результатом сравнения двух лиц является вероятностная мера [2] – численное значение µ(), принадлежащее отрезку [0; 1], где событие обозначает принадлежность этих двух лиц одному человеку. В зависимости от значения µ() принимается одно из двух бинарных решений: да, лица принадлежат одному и тому же человеку, или нет, лица принадлежат разным людям. Итого мы имеем 2 варианта решения и 2 варианта того, как оно есть на самом деле. Мы получаем 4 возможных положения вещей:



Рис. 1 Цветом фона обозначено реальное положение вещей (голубой – принять, оранжевый – отвергнуть), а цветом рамки – решение, принятое алгоритмом (голубой – принять, оранжевый – отвергнуть). Сделай цвета констрастными в черно-белом исполнении

В соответствии с возможными исходами имеется 2 корректных решения и 2 ошибочных:



Рис. 2 Левый столбец представляет из себя описание реального положения вещей, а верхняя стока – выбор алгоритма. Возможны 2 корректных результата, ошибка 1 рода («ложная тревога») и ошибка 2 рода («пропуск цели»).

Лица с двух фотографий принимаются одинаковыми при µ() больше, либо равному некоторому заданному пороговому значению µ1. Соответственно, при уменьшении значения µ1 увеличивается вероятность ошибки 1 рода, а вероятность ошибки 2 рода уменьшается, и наоборот, при увеличении значения µ1 вероятность ошибки 1 рода уменьшается, а вероятность ошибки 2 рода увеличивается. В соответствии различными задачами есть смысл либо искусственно занизить значение, чтобы получить гораздо большую выборку лиц, пусть и с высокой вероятностью получения ошибочных данных, либо завысить значение, благодаря чему выборка будет меньше объемом, однако точность совпадения заметно возрастет. Помимо этих двух вариантов существует так же вопрос нахождения некоего среднего оптимального значения – такого, при котором вероятности ошибок 1 и 2 рода окажутся примерно равны. Выбор необходимого значения зависит от конкретной задачи и подбирается экспериментально для каждой библиотеки распрзнавания лиц.

Помимо вышеописанного, важным фактором является процедура «двойной верификации». При верификации лица №1 и лица №2 степень их схожести может оказаться ниже заявленного порога, определяющего принадлежность двух лиц одному и тому же человеку. Это может произойти даже если лица принадлежат одному человеку, но повернуты в разную сторону, находятся в неудачном ракурсе, освещение затемняет определенные части и т.п. Однако, существует лицо №3, степень схожести которого как с лицом №1, так и с лицом №2 является достаточно высокой для положительной верификации, из чего следует, что лицо №1 и лицо №2 тоже принадлежат одному и тому же человеку, однако, так как на момент сравнения лица №1 и лица №2 друг с другом они были определены как принадлежащие разным людям, они находятся в разных элементах базы данных. Для того, чтобы избежать подобного разделения лиц, принадлежащих одному человеку, процедура верификации проводится дважды. В первый раз с каким порогом? – каждый кадр видеоролика проверяется на наличие лиц на нем и проводится процедура верификации всех найденные лиц со всеми элементами, уже сохраненными в базу. Если порог сравнения оказывается достаточно высоким, то эти два (какие два?) лица считаются принадлежащими одному человеку. В этом случае (каком этом?), происходит сравнение качества «описаний лиц» - того, что уже находится в базе и нового. Если находящееся в базе «описание лица» хуже, чем новое, то новое заменяет собой находящееся в базе. Если же находящееся в базе лучше, то сохраняется только порядковый номер кадра, на котором наше «худшее» описание лица находится. Поскольку во время первой процедуры верификации есть вышеописанная возможность разделения двух лиц, принадлежащих одному и тому же человеку на два элемента базы, то после окончания первой процедуры верификации сразу же запускается вторая процедура, в ходе которой каждый элемент базы данных сравнивается с каждым, с целью найти дополнительные совпадения. И что сделать дальше?

Здесь нужно немного ранее ввести описание «качества описания лица», которое правильнее сделать в обзоре выше (см. письмо)

**Литература**

1. [Электронный ресурс] Оценка качества алгоритмов распознавания лиц <https://habr.com/ru/company/ntechlab/blog/329412/>
2. Чернова Н. И. Теория вероятностей: Учеб. пособие / Новосиб. гос. ун-т. Новосибирск, 2007