

МИНИСТЕРСТВО ОБРАЗОВАНИЯ И НАУКИ РОССИЙСКОЙ ФЕДЕРАЦИИ
ФЕДЕРАЛЬНОЕ ГОСУДАРСТВЕННОЕ БЮДЖЕТНОЕ ОБРАЗОВАТЕЛЬНОЕ
УЧРЕЖДЕНИЕ

ВЫСШЕГО ОБРАЗОВАНИЯ

«ТОМЕНСКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ»
ИНСТИТУТ МАТЕМАТИКИ И КОМПЬЮТЕРНЫХ НАУК

Кафедра прикладного обеспечения

ДОПУЩЕНО К ЗАЩИТЕ В ГАК

И ПРОВЕРЕНО НА ОБЪЕМ

ЗАЙМСТВОВАНИЯ

Заведующий кафедрой

д.п.н., профессор

 И.Г. Захарова

26 июля 2016 г.

МАГИСТЕРСКАЯ ДИССЕРТАЦИЯ

АВТОМАТИЗИРОВАННАЯ СИСТЕМА ОПРЕДЕЛЕНИЯ НАРУШЕНИЙ НА ВДИПОМ
ГОСУДАРСТВЕННОМ ЭКЗАМЕНЕ

02.04.03, Математическое обеспечение и администрирование информационных систем

Магистерская программа «Высокопроизводительные вычислительные системы»

Выполнил работу

Студент 2 курса

очной формы обучения

Научный руководитель

к.п.н., доцент

Рецензент

зав.кафедрой математики и информатики

к.т.н., доцент

Цепилов

Андрей

Вадимович

Плющенко

Юрий

Анатольевич

Григорьев

Михаил

Викторович


Томск 2016

ГЛАВА 1. Обзор методов обработки изображений	5
1.1 Актуальность развития технологий компьютерного зрения	5
1.2 Постановка задачи	6
ГЛАВА 2. Описание этапов работы и методов реализации	8
2.1 Считывание видео.....	8
2.2 Предобработка кадра.....	8
2.2.1 Медианный фильтр.....	10
2.2.2 Фильтр Гаусса.....	11
2.3 Выделение объектов.....	13
2.3.1 Метод Оцу	14
2.3.2 Методы нахождения границ.....	15
2.3.3 Оператор Собеля.....	16
2.3.4 Поиск контуров.....	17
2.4 Распознавание объектов.....	18
2.4.1 Сопоставление с образцом.....	18
ГЛАВА 3. Программная реализация. Описание классов.....	19
3.1 Диаграмма классов приложения	19
3.2 Класс FrameReader	19
3.3 Класс Img	20
3.4 Блок-схемы реализованных алгоритмов	22
Заключение.....	27
Список литературы	27

Введение

Видеокамеры в настоящее время активно используются для обеспечения безопасности в общественных местах, изначально они просто записывали данные, а наблюдение производилось операторами. При использовании видеокамер для наблюдения за несколькими объектами достаточно иметь одного оператора, однако это касается лишь легко обнаруживаемых данных, в случае же когда нужно внимательно наблюдать за данными с каждой камеры, качество работы оператора сильно снижается. Вследствие этого появилась технология компьютерного зрения[5], с её помощью начальный анализ можно поручить компьютеру и существенно снизить нагрузку на оператора. Системы, использующие подобный принцип работы устанавливаются в общественных местах, на предприятиях, в больницах, для наблюдения за пациентами[3]. Кроме этих систем существуют полностью автоматические системы, например видеокамеры фиксирующие нарушения правил дорожного движения, такие системы способны работать без оператора[4].

При проведении единого госэкзамена ведется видеонаблюдение с помощью камер, данные с которых записываются, поскольку в каждой аудитории стоит по несколько камер, а экзамен идет в течении нескольких часов, объём данных которые нужно будет обработать, огромен и используется в основном для доказательства нарушений, а поиском занимаются люди-наблюдатели, находящиеся в аудиториях. За нарушение будем считать: появление запрещенного к использованию на экзамене предмета в кадре, уход со своего места, появление новых людей в аудитории, общение с другим экзаменуемым.

Для исследования была выбрана задача поиска запрещенного предмета в кадре, на примере распознавание предметов на рабочем месте экзаменуемого, для решения которой требуется:

1. Загрузить видеопоток.
2. Разделить видео на кадры.
3. Подготовить изображения к поиску элементов(фильтрация).
4. Произвести поиск элементов и их распознавание.

Проблема настоящей работы – анализ и выбор алгоритмов поиска заданных объектов в видеопотоке.

Цель работы – разработать приложение для поиска недопустимых ситуаций на видеозаписи, записанной во время проведения экзамена.

Объект исследования – обработка изображений.

Предмет исследования – поиск элементов изображения определённого типа в видеопотоке.

Задачи исследования:

1. Изучение и анализ литературы по вопросам распознавания образов.
2. Выбор и реализация алгоритма для решения задачи поиска.
3. Разработка приложения для реализации .

Для решения задачи используются методы компьютерной графики и распознавания образов.

Для оптимизации работы использованы методы параллельной обработки данных.

ГЛАВА 1. Обзор методов обработки изображений

1.1 Актуальность развития технологий компьютерного зрения

С момента появления видеокамер небольших размеров, способных работать автономно их применяют, как средство обеспечения безопасности в общественных местах. Изначально всё сводилось к тому, что камеры выдавали картинку на пульт охраны, где за ними наблюдал оператор, такое применение присутствует и сейчас и при небольшом числе камер является весьма эффективным, однако чем больше становится камер тем сложнее за ними уследить одному человеку. Таким образом решения у проблемы два – первое увеличить количество операторов, а второе проводить первичный анализ средствами обработки данных.

Таким образом появилось направление компьютерного зрения - теория и технология создания машин, которые могут производить обнаружение, отслеживание и классификацию объектов. С помощью её методов стало возможным проводить обработку полученного с камеры изображения. Вместе с ней также появилась технология - видеоаналитика, которая занимается обработкой, полученных с помощью компьютерного зрения данных. Благодаря этим технологиям возможно в разы снизить нагрузку на операторов, что позволяет одному человеку контролировать большую территорию с помощью камер вместе с технологиями видеоаналитики и компьютерного зрения. Также существуют полностью автоматические системы видеонаблюдения, например в нашей стране с 1 июля 2008 года действует система фиксации нарушений ПДД камерами наблюдения, система полностью автоматическая – она самостоятельно определяет нарушения правил дорожного движения, распознает номер машины и отправляет почтой уведомление о штрафе, работа обслуживающего персонала состоит лишь в проверке работы системы и поддержании её в рабочем состоянии.

В большинстве стран мира с развитой инфраструктурой повсеместно используются системы наблюдения, обеспечивающие общественную безопасность во всех местах с массовыми скоплениями людей. Одним из лидеров в использовании таких технологий является Израиль, так в центральном аэропорту Тель-Авива установлено более 700 камер, передающих данные системе для поиска возможных террористов. Повсеместно разворачиваются системы наблюдения на таких крупных мероприятиях, как олимпийские игры или саммиты глав государств.

В начале 2014 года было начато создание системы видеонаблюдения для московского метрополитена, в разработку было вложено около 400 миллионов рублей, а созданием занималась фирма Винтегра Секьюрители.

Кроме общественных мест подобные системы вводятся также на предприятиях, с 2011 года существует продукт Microsoft's OneVision Video Recognizer, который позволяет обнаруживать, идентифицировать и отслеживать лица на видео. Такая система подходит для огромных предприятий, где требуется контроль за перемещением сотрудников или ограничение доступа в определенные зоны.

Уже несколько лет в нашей стране, во время проведения единого государственного экзамена на пунктах проведения устанавливаются камеры, в 2016 году была организована онлайн трансляция, где любой мог проследить за ходом его проведения, можно предположить, что одним из следующих шагов может стать создание системы видеоаналитики для автоматического наблюдения за экзаменом.

1.2 Постановка задачи

Производится съемка рабочего места участника экзамена. Под нарушением будет подразумеваться появление постороннего объекта, на примере письменных принадлежностей экзаменуемого.

Требуется программно определить факт появления постороннего предмета на столе в кадре, передать кадр на обработку оператору для ручной проверки.

Входными данными является видео в формате avi, демонстрирующее рабочее место.

Выходными данными являются кадры с обнаруженными на них посторонними предметами.

Исходя из задачи требуется произвести обзор алгоритмов, использующихся в подобных задачах, выбрать подходящие для решения данной задачи и разработать приложение.

Требования к приложению :

1. Возможность настройки параметров методов.
2. Возможность сохранить кадры с нарушением.

Исходя из задания, требуется разработать программу, реализующую следующие функции:

1. Считывание видео из входного файла
2. Разделение видео на кадры
3. Удаление шумов с изображения, с помощью усредняющих фильтров
4. Выделение границ оператором Собеля
5. Определение контуров
6. Распознавание объектов, через сравнение с образцом
7. Вывод отчета о работе

ГЛАВА 2. Описание этапов работы и методов реализации.

2.1 Считывание видео

Основные понятия:

Видеозапись – это запись сигналов изображения, то есть по сути представляет из себя последовательный набор кадров.[6]

Для использования компьютерного зрения требуется получить из видеозаписи отдельные кадры и дальнейшую работу проводить уже с ними. Существует множество библиотек для работы с видеозаписями в рамках данной работы я выбрал для использования библиотеку AForge.NET, с её помощью будет осуществляться открытие видеозаписей и получение из них кадров, также следует отметить, что большинство камер снимают видео с количеством кадров от 15 до 60 в секунду. Отличие соседних кадров друг от друга минимально, поэтому стоит предусмотреть возможность пропуска кадров.

2.2 Предобработка кадра.

Основные понятия:

Цифровой шум — дефект изображения, вносимый фотосенсорами и электроникой устройств, которые их используют (цифровой фотоаппарат, теле-/видеокамеры и т. п.) вследствие несовершенства технологий.[1]

Свертка – это операция вычисления нового значения выбранного пикселя, учитывающая значения окружающих его пикселей[1]

Матрица свертки – это матрица коэффициентов, которая «умножается» на значение пикселей изображения для получения требуемого результата[1]

Одной из задач предварительной обработки кадра является минимизация шума на изображении. Существуют различные виды борьбы с шумом, делящиеся на группы в зависимости от того на каком уровне их применяют, поскольку на вход подается уже готовое видео, бороться с шумом можно лишь на уровне постобработки.

Одним из способов убрать шумы с изображения являются усредняющие фильтры, которые делают пиксель более схожим с его окружением. В основе всех усредняющих списков лежит операция свертки, позволяющая изменить значения пикселя в зависимости от его соседей.

Усредняющие фильтры обладают некоторым количеством общих свойств:

1. Нулевой сдвиг. Фильтр не вносит фазовый сдвиг ни в какую из периодических компонент изображения. Для этого передаточная функция должна быть действительной, а маска фильтра – симметричной.[2]
2. Сохранение среднего значения. Сумма всех коэффициентов маски должна быть равна единице.[2]
3. Изотропия. Сглаживание должно быть одинаковым во всех направлениях. В дискретном пространстве, это условие может выполняться лишь приблизительно.[2]

Самым простым усредняющим фильтром, является так называемый фильтр ящик. Принцип его работы – все пиксели внутри маски складываются и делятся на количество элементов. С его помощью можно преобразовать резкую границу в линейное изменение.

Пример работы фильтра ящика с матрицей [1,1,1]

Начальное состояние

0	0	1	1
0	0	1	1
0	0	1	1
0	0	1	1
0	0	1	1

После применения фильтра

0	1/3	2/3	1
0	1/3	2/3	1
0	1/3	2/3	1
0	1/3	2/3	1
0	1/3	2/3	1

В данном случае фильтр сработал правильно, однако это работает только в случае удачных границ, в другой ситуации он может полностью удалить структуру, поэтому применение его весьма ограничено.

Фильтр ящик является плохим фильтром для низких частот и сильно отклоняется на контурах, свойство изотропности не выполняется. В общем выходит, что применять его к любому изображению не стоит, а поскольку требуется в дальнейшем решать задачу нахождения границ, отклонения на контурах нежелательны.

2.2.1 Медианный фильтр

Медианный фильтр имеет выгодное преимущество по сравнению с ящик фильтром, он практически не значительно слабее влияет на контуры, что будет полезно в дальнейшем. При ошибке передачи, уровень яркости

пикселя изменяется, в случае с линейными фильтрами, он будет размыт на соседние пиксели, но не удален. Медианный же фильтр удаляет данные шумы.

Принцип работы медианного фильтра прост, берется некая окрестность пикселя, производится её сортировка и значение пикселя заменяется на среднее.

Пример работы фильтра:

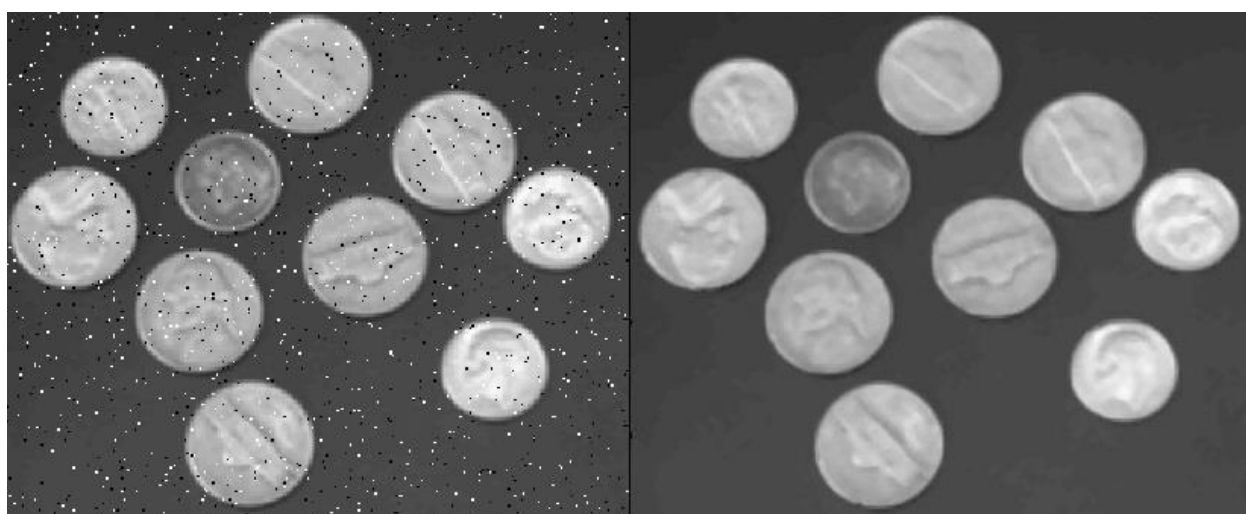


Рис. 1 - Пример работы медианного фильтра

Из недостатков фильтра можно отметить, то что на небольшом изображении он может подавить лишние значения, которые могут быть не шумом, а частью другой области.

2.2.2 Фильтр Гаусса.

Фильтр Гаусса – это матричный фильтр, применяющий матрицу Гаусса для свертки изображения.

Представим исходное изображение в виде функции $I(x,y)$, где x,y – координаты пикселя, а значение функции – значение пикселя.

Матрица свертки является квадратной матрицей с размерностью $2*r+1$, где r -радиус гауссова фильтра. Коэффициенты вычисляются через функцию Гаусса[1]. Для расчета матрицы Гаусса используется формула:

$$g(x) = \frac{1}{\sqrt{2\pi}\sigma} e^{-\frac{x^2+y^2}{2\sigma^2}}$$

Где сигма – отклонение, такая матрица называется Гауссианом. Таким образом значение пикселя будет считаться по формуле:

$$I'(x, y) = \sum_{m=-r}^r \sum_{n=-r}^r g(r+m, r+n) * I(x+m, y+n)$$

При применении свертки появляется проблема единая для большинства фильтров, что делать с граничными элементами? Существуют 2 решения: первое – игнорирование краев, однако при большом размере матрицы, большая область останется необработанной и зашумленной, второе – добавление элементов по краям, на основе известных соседних элементов. Второй вариант является более предпочтительным. После свертки изображение уменьшается до начального размера.

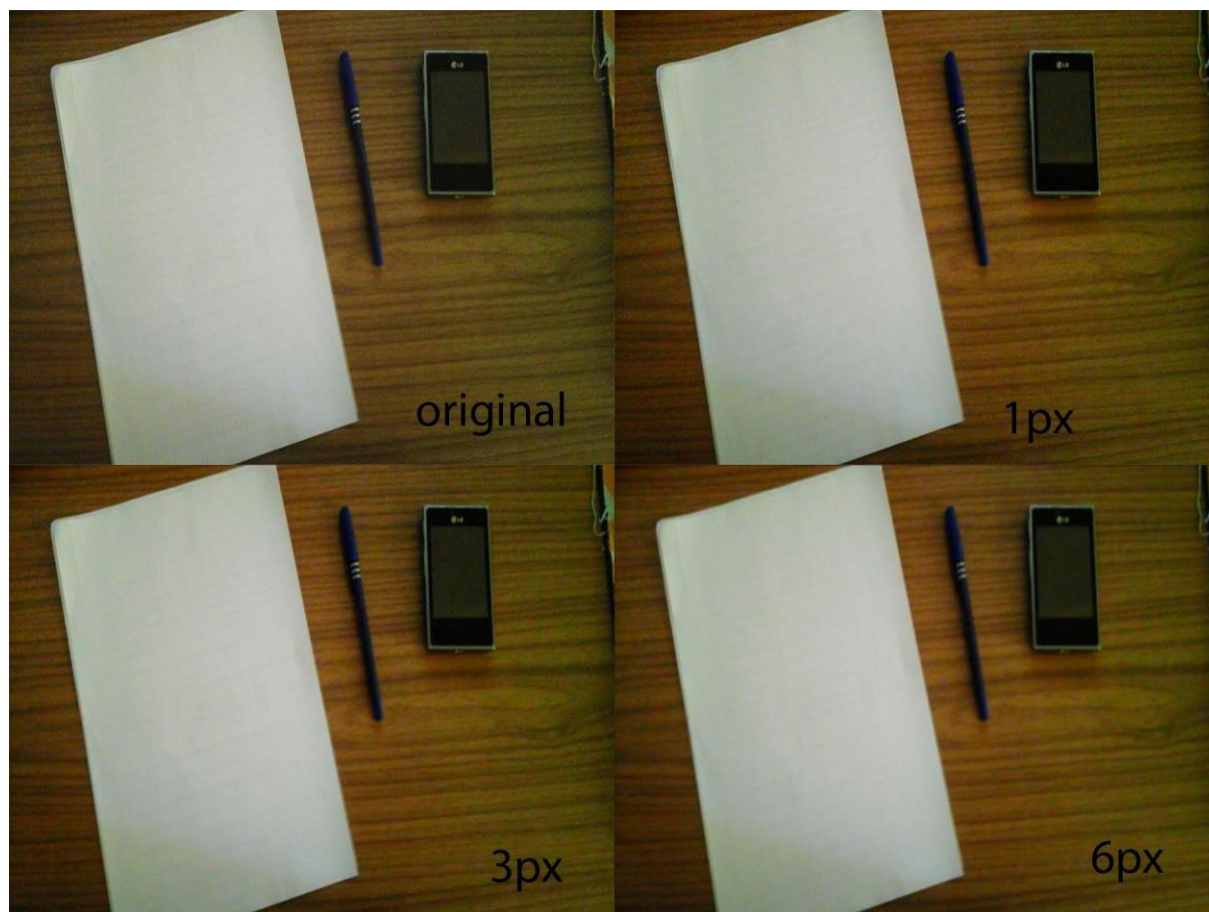


Рис. 2 - Результат работы фильтра Гаусса

2.3 Выделение объектов.

После удаления шума с изображения можно перейти к поиску на ней требуемых объектов, на данном этапе стоит перейти от работы с полноцветным изображением, к его версии заданной через уровень яркости, чтобы перейти от модели RGB к GrayScale(оттенки серого), используется следующая формула (с округлением результата до целого)[9]:

$$Gray = R * 0.2125 + G * 0.7154 + B * 0.0721$$

Переход к модели с оттенками серого позволяет сократить время на вычисления, и может использоваться всегда, кроме случаев, когда объекты выделяются на основе их цвета.

2.3.1 Метод Оцу

Метод Оцу – это алгоритм вычисления порога бинаризации для полутонового изображения. С его помощью можно отделить фоновые пиксели, путем расчета порога при котором внутриклассовая дисперсия будет минимальной.

Основные понятия:

Гистограмма — это график статистического распределения элементов цифрового изображения с различной яркостью, в котором по горизонтальной оси представлена яркость, а по вертикали — относительное число пикселей с конкретным значением яркости.

Для определения порога используется гистограмма изображения, с уровнями яркости от 0 до 255.

Метод сводится к поиску минимума внутриклассовой дисперсии, которая задаётся как взвешенная сумма дисперсий двух классов, взвешенная сумма дисперсий:

$$\sigma_{\omega 2} = \omega_1 \sigma_{12} + \omega_2 \sigma_{22}$$

Где ω_1 и ω_2 – вероятности первого и второго классов. Оцу было доказано, что минимизация внутриклассовой дисперсии эквивалентна максимизации межклассовой дисперсии:

$$\sigma_{b2} = \omega_1 \omega_2 (a_1 - a_2)^2$$

Где a_1 и a_2 – среднее арифметическое значене для каждого класса.[8]

Отсюда получаем алгоритм нахождения порога:

1. Рассчитываем значения гистограммы.

2. Проходим гистограмму, пересчитывая на каждом шаге дисперсию, если она оказывается больше максимума, то обновляем дисперсию и порог.
3. Выводим последний получившийся порог.

Из плюсов метода Оцу следует отметить простоту реализации, быстрое время выполнения ($O(N)$ операций), а также отсутствие дополнительных входных параметров. Из недостатков следует отметить, что при большом числе объектов с разной яркостью метод Оцу будет малополезен, однако эту проблему можно решить применяя его к частям изображения вместо целого.

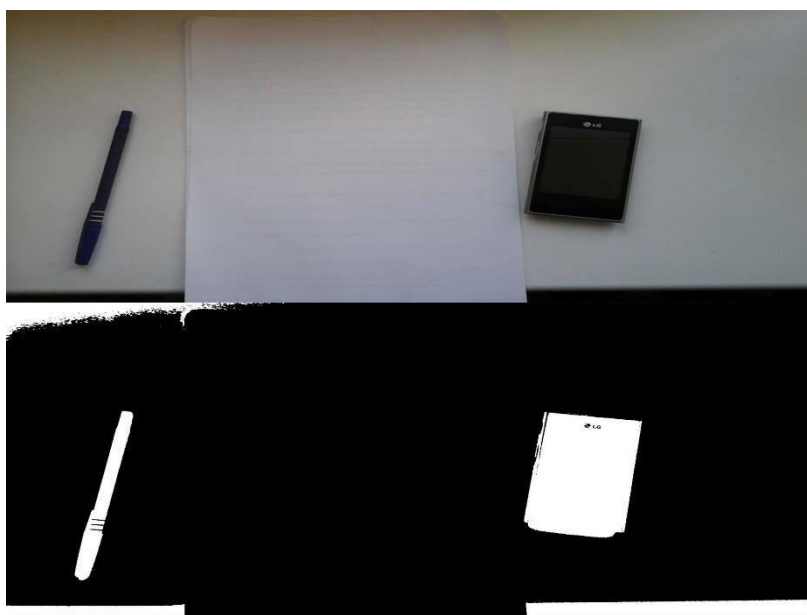


Рис. 3 - Применение метода Оцу

2.3.2 Методы нахождения границ

С помощью метода Оцу можно выделить объекты, однако кроме этого подхода существует другой, с поиском границ, где после применения одного из алгоритмов мы получаем границы объектов в виде линий. Для данного подхода существует множество алгоритмов, самые известные из них – детектор границ Кенни и оператор Собеля, также существуют алгоритмы Робертса, Прюитта, Кирша, Робинсона, Марр-Хилдрета и другие.

2.3.3 Оператор Собеля

Оператор Собеля был введен Ирвином Собелем в 1970м году. С его помощью можно вычислить приблизительное значение градиента яркости изображения в каждой точке.

Оператор использует ядра 3 на 3,

Матрица Собеля для y

$$\begin{matrix} -1 & -2 & -1 \\ 0 & 0 & 0 \\ 1 & 2 & 1 \end{matrix}$$

Матрица Собеля для x

$$\begin{matrix} -1 & 0 & 1 \\ -2 & 0 & 2 \\ -1 & 0 & 1 \end{matrix}$$

Каждое ядро применяет операцию свертки к исходному изображению, после чего градиент можно вычислить используя полученные приближенные значения производных[10]

$$G = \sqrt{G_x^2 + G_y^2}$$

Оператор Собеля позволяет получить только приближенное значение градиента, но он достаточен для применения в большом числе практических задач. Из недостатков стоит отметить его чувствительность к шуму, а также возможное появление шумов на полученном с его помощью изображении.

На рисунке 4, представлено выделение границ оператором Собеля, с промежуточными шагами, отображающими изменения после свертки по x и по y.

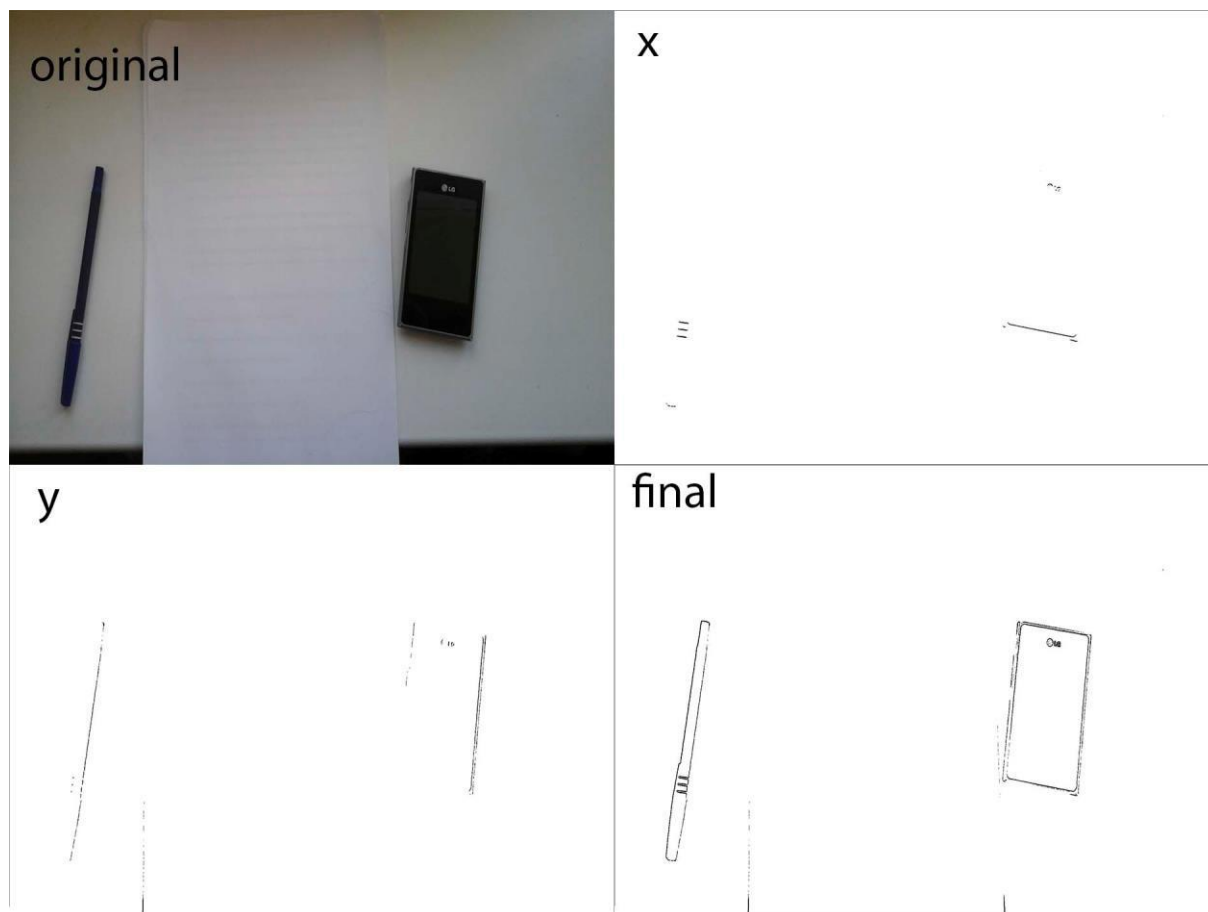


Рис. 4 - Выделение границ оператором Собеля

2.3.4 Поиск контуров

Для распознавания объекта необходимо сначала отделить его от остального изображения, для этого будет использоваться наивный алгоритм, который будет искать пиксели у которых есть соседи.

Описание алгоритма:

1. Главный цикл, которым обходятся все пиксели изображения
2. Поиск первого черного пикселя
3. Проверка его на соседей в восьми направлениях, и сохранение всех этих пикселей в лист, в главном цикле все эти пиксели в дальнейшем игнорируются.
4. Поиск границ массива, выделение их прямоугольной областью

5. Проверка на превышение минимального размера
6. Пока не обойдена вся картинка, возврат к шагу 1
7. Возвращает список с найденными контурами.

2.4 Распознавание объектов

Для распознавания объектов сперва требуется привести все изображения к одному размеру. В дальнейшем к ним можно применять методы сравнения.

2.4.1 Сопоставление с образцом

Для распознавания объекта используется метод сопоставления с образцом, приведенные к одному размеру сравниваются изображениями из галереи образцов, при превышении порога совпадения, который можно задать в программе определяется идентичны ли объекты.

ГЛАВА 3. Программная реализация. Описание классов.

Для разработки была выбрана Microsoft Visual Studio 2013, язык разработки C#.

3.1 Диаграмма классов приложения



Рис. 5 – Диаграмма классов

3.2 Класс FrameReader

Класс FrameReader отвечает за считывание видеофайла в формате «avi», получение из него кадров. Класс включает в себя методы.

Таблица 3

Методы класса FrameReader

Название метода	Назначение
OpenVideo(string filename)	Открывает видеофайл для работы
CloseVideo()	Освобождает видеофайл
GetNextFrame(int skip)	Получает следующий кадр, пропуская заданное число кадров

SetPosition(int p)	Переводит курсор в заданный кадр
--------------------	----------------------------------

3.3 Класс Img

Класс `Img` является основным классом, выполняющим обработку изображения. Он включает в себя фильтры, выделение границ, поиск контуров, изменение размера изображений, конвертацию изображения в массив и обратно.

Потребовалось также создать один дополнительный класс для хранения объектов.

```
public class OB
{
    public byte[,]
mas;
    public int
x1; public int x2; public
int y1; public int y2;
}
```

Таблица 4

Методы класса `Img`

Название метода	Назначение
<code>Img()</code>	Конструктор класса
<code>median(byte[,], mas, int k)</code>	Выполняет медианный фильтр с заданной размерностью маски, на выход дает новый массив
<code>imgtomas(Bitmap bmp)</code>	Переводит картинку в массив байтов, со значениями компонентов в формате <code>rgb</code>

med(byte[] mas)	Возвращает медиану массива
bytetobmp(byte[, ,] mas)	Создает из массива с rgb значениями, картинку для вывода на экран либо сохранения
printing(byte[, ,] masnew, string type)	Сохраняет массив rgb в файл, с названием type
GaussianMatrix(int k)	Строит гауссову матрицу заданного размера
GaussianBlur(double[][][] matrix, byte[, ,] mas, int k)	Выполняет гауссов фильтр
otsu(byte[, ,] mas)	Находит порог для разделения объектов методом Оцу
binar(byte[, ,] mas, int threshold)	Производит бинаризацию картинки, используя заданный порог
morf(byte[, ,] mas, int size)	Производит операции эрозии и коррозии с заданным размером матрицы
rgbtogray(byte[, ,] mas)	Переводит массив из значений rgb в оттенки серого
sobelx(byte[, ,] mas)	Производит свертку оператором Собеля по координатам x
sobely(byte[, ,] mas)	Производит свертку оператором Собеля по координатам y
countgrad(int[, ,] masx, int[, ,] masy)	Считает градиент и определяет границы
printsob(int[, ,] mas)	Переводит серое изображение в rgb формат для распечатки или вывода на экран
ScaleImage(Image source, int width, int height)	Масштабирует изображение
GetObjects(byte[, ,] mas)	Возвращает список с найденными объектами

3.4 Блок-схемы реализованных алгоритмов

Блок-схема реализации медианного фильтра



Рис. 5 – Блок-схема медианного фильтра

Реализация фильтра Гаусса

Для фильтра Гаусса сначала требуется заполнить матрицу Гаусса.

Блок схема создания матрицы Гаусса



Рис. 6 – Блок-схема создания матрицы Гаусса

Блок-схема реализации фильтра Гаусса



Рис. 7 – Блок-схема фильтра Гаусса

Блок-схема реализации метода Оцу



Рис. 8 – Блок-схема метода Оцу

Блок-схема реализации оператора Собеля



Рис. 8 – Блок-схема поиска границ с помощью оператора Собеля

Заключение.

В результате выполнения выпускной квалификационной работы были изучены различные существующие методы обработки изображений. Были рассмотрены различные алгоритмы и методы для реализации различных этапов обработки изображений, такие как:

1. Медианный фильтр
2. Фильтр Гаусса(подавление цифрового шума)
3. Метод Оцу(поиск порога бинаризации)
4. Оператор Собеля(выделение границ)

Среди них были выбраны наиболее подходящие для решения поставленной задачи.

Разработано приложение на языке C#, которое позволяет произвести поиск запрещенных предметов на видео и вывести кадры оператору для ручного анализа. Созданное приложение предоставляет возможность настройки параметров и отвечает поставленным в постановке задачи требованиям.

Список литературы

1. Гонсалес Р., Вудс Р. Цифровая обработка изображений: Пер. с англ./Под ред //Чочиа ПАМ: Техносфера. – 2005. – Т. 1072.
2. Яне Б. Цифровая обработка изображений //М.: Техносфера. – 2007. – Т. 584. – С. 77-30569.
3. Осин В. А., Анищенко С. И., Шапошников Д. Г. Мониторинг движений головы пациента в процессе томографии по видеоизображениям.

4. Иванов Р. Е. Разработка и исследование автоматических систем регистрации нарушений ПДД с использованием RFID-технологии и широкополосных беспроводных средств.
5. Шапиро Л., Стокман Д. Компьютерное зрение //М.: Бином. Лаборатория знаний. – 2006. – Т. 752. – С. 8.
6. Лукьяница А. А., Шишкин А. Г. Цифровая обработка видеоизображений //М.: Ай-Эс-Эс Пресс. – 2009. – Т. 518.
7. Otsu N. A threshold selection method from gray-level histograms //Automatica. – 1975. – Т. 11. – №. 285-296. – С. 23-27.
8. Burger W., Burge M. J. Digital image processing: an algorithmic introduction using Java. – Springer Science & Business Media, 2009.
9. Gao W. et al. An improved Sobel edge detection //Computer Science and Information Technology (ICCSIT), 2010 3rd IEEE International Conference on. – IEEE, 2010. – Т. 5. – С. 67-71.