МИНИСТЕРСТВО ОБРАЗОВАНИЯ И НАУКИ РОССИЙСКОЙ ФЕДЕРАЦИИ
федеральное государственное бюджетное образовательное учреждение
высшего образования
«Тольяттинский государственный университет»

Институт энергетики и электротехники
(институт)

Кафедра «Промышленная электроника»

11.04.04 Электроника и наноэлектроника
(код и наименование направления подготовки, специальности)

Электронные приборы и устройства
направленность (профиль)

МАГИСТЕРСКАЯ ДИССЕРТАЦИЯ

на тему **Электронная система фиксации нарушения ПДД на пешеходном переходе**

Студент	А.В. Стариков		
	(И.О. Фамилия)		(личная подпись)
Научный руководитель	Е.С. Глибин		
	(И.О. Фамилия)		(личная подпись)
Руководитель программы	В.В. Ивашин		
	(ученая степень, звание, И.О.Ф.)		(личная подпись)

Допустить к защите

Заведующий кафедрой к.т.н., доцент А.А. Шевцов _____
(ученая степень, звание, И.О. Фамилия) (личная подпись)

«_____»_____20_____г.

Тольятти 2018

Перечень условных обозначений, сокращений и терминов

3D – 3-dimensional – трёхмерное пространство;

ATX – форм-фактор современных персональных компьютеров;

DIMM – двухсторонний модуль памяти;

GSM – Global System for Mobile Communications – глобальный стандарт цифровой мобильной связи;

INRIA – набор изображений, собранных Далалом и Триггсом, а также государственный институт исследований в информатике и автоматике во Франции;

IP-видеокамера – видеокамера с передачей данных по сети Ethernet, используя протокол IP;

HOG – Histogram of Oriented Gradients – гистограмма направленных градиентов – дескриптор особых точек, применяемый для детектирования объектов;

MIT – Массачусетский технологический институт;

OpenCV – библиотека компьютерного зрения с открытым исходным кодом;

SBC – single-board computer – одноплатный компьютер;

SIFT-дескриптор – Scale Invariant Feature Transform – алгоритм детектирования особых точек исследуемой области;

SVM – Support Vector Machine – машина опорных векторов – обучаемый классификатор, алгоритм машинного обучения по прецедентам;

Вейвлет – математическая функция, позволяющая анализировать частотные составляющие данных;

ГИБДД – Государственная инспекция безопасности дорожного движения;

Дескриптор – исследование объекта на соответствие определенным критериям;

ДПС – дорожно-патрульная служба;

ПДД – правила дорожного движения;

ПК – персональный компьютер;

ПО – программное обеспечение;

Трекинг – отслеживание движущихся объектов и определение их координат в пространстве;

ЦАФАП – центр автоматизированной фиксации административных правонарушений.

Содержание

Введение

В настоящее время компьютеры еще не научились «видеть» так, как это может человек. Человеческая система зрения способна опираться на собственные догадки, знания о прошлом и предположения. На текущий момент фотокамеры не способны думать как человек, а должны производить операции по вычислениям и расчетам каждого пикселя, чтобы в целом собрать картину, основываясь на разработанных человеком обучающих базах, чтобы предположить какой объект находится перед ним (пример, фокусировка по лицам). В данный момент разработано множество алгоритмов детектирования, распознавания и отслеживания, большое количество уникальных методов способно обрабатывать изображения таким образом, чтобы определить уникальные свойства присущие именно этому изображению (например, нахождение на снимке человека или любого другого объекта).[1]

Компьютерное (машинное) зрение — это множество технологий, методов и алгоритмов, разработанных с целью детектирования и классификации объектов, а также отслеживания их перемещения в пространстве.[2]

В качестве научной дисциплины, её можно отнести к теории и технологии разработки систем (программных), которые получают информацию из видео или фотоизображений. Как известно видеоданные это последовательно идущие друг за другом ряд изображений (кадров), каждый из которых может использоваться для обработки алгоритмами компьютерного зрения с целями указанными выше.

В качестве технологической дисциплины, машинное зрение стремится применять созданные человеком технологии, методы и алгоритмы с целью создания высокотехнологичных систем. В качестве примера можно назвать такие системы:

- Управления процессами (автопилотируемый транспорт, промышленные установки);[3]
- Видеонаблюдения;
- Организации информации (создания и сохранения баз данных с фото- и видеоизображениями);
- Моделирования объектов или окружающей среды (топография и медицинские визуальные данные)
- Взаимодействия (управление машинами, опираясь на движения человека);
- Дополнительной реальности;
- Цифровой обработки фото- и видеоизображений.[4]

Пока что машинное зрение нельзя представить как замену биологическому зрению, но как дополнение – вполне. Изучив физиологию человека и животных, в процессе познания биологии, люди научились создавать модели зрительного восприятия. Другими словами, люди и животные – прообраз моделей компьютерного зрения. Таким образом, получившийся обмен знаниями между компьютерным и биологическим зрением являлся очень продуктивным для обеих научных дисциплин.[5]

С другой стороны, в большей степени компьютерное зрение является связующим звеном не только в программной, но и аппаратной части. Так для каких же задач оно может использоваться?

В данной магистерской диссертационной работе предлагается рассмотреть решение задачи по увеличению безопасности участников дорожного движения, применив знания по разработке электронных устройств и их программированию на базе технологий компьютерного зрения.

Ежегодно увеличивающееся число камер видеофиксации правонарушения ПДД, устанавливаемое ГИБДД (государственная инспекция безопасности дорожного движения), способствует снижению аварийности на

автотранспортных дорогах. Их увеличение – не такой уж и плохой знак, ведь они делают наши автотранспортные дороги безопаснее.[6]

Применение компьютерного зрения в данной области теоретически может помочь зафиксировать любое нарушение правил дорожного движения.

Непредоставление преимущества пешеходам при переходе по пешеходному переходу отличный пример того, где может помочь компьютерное зрение. Реализация алгоритма поиска и отслеживания пешеходов и автомобилей и их определение в пространстве относительно друг друга – задача, с которой не сможет справиться обычная камера без соответствующей программной технологии.[7]

В рамках данной диссертационной работы предполагается разработка программного и аппаратного обеспечения для детектирования нарушения ПДД на пешеходном переходе. Данная диссертационная работа может быть использована в качестве примера решения подобных задач компьютерного зрения, включающих в себя:

1. Детектирование объектов;
2. Распознавание объектов (автомобиль, пешеход и пр.);
3. Исследование влияния различных отрицательных факторов на качество распознавания.

Объектом исследования является видеоматериал, обрабатываемый для решения задач распознавания и детектирования образов при фиксации нарушения правил дорожного движения (ПДД) на пешеходном переходе.

Цель работы – достижение приемлемого качества детектирования объектов для реализации поставленных задач научно-исследовательской работы, анализ аппаратного и программного обеспечения и их выбор, исследование различных факторов влияющих на точность детектирования (освещение, параметры камеры и пр.).

Задачи работы:

1. Обзор существующих программных решений;

2. Обзор существующих аппаратных решений;

3. Выбор метода детектирования образов и обоснование выбора;

4. Разработка и реализация программы, управляющей логикой работы;

5. Разработка электрической схемы системы фиксации;

6. Обоснования выбора электронных компонентов системы;

7. Исследование влияния негативных факторов.

Программная реализация алгоритма фиксации правонарушения написана на языке программирования C++ в программе Microsoft Visual Studio 2015 с применением технологий компьютерного зрения библиотеки с открытым исходным кодом OpenCV. В результате реализован алгоритм распознавания образов человека и автомобилей. Он был протестирован на нескольких тестовых видеоизображениях, и выявлена взаимосвязь влияния использованных данных в различающихся условиях.

Аппаратная реализация учитывает тенденции в современной электронике к соотношению производительности, компактности и цены разрабатываемого устройства. В результате спроектировано устройство максимально подходящее к данным требованиям. Передача зафиксированных правонарушений производится по технологии Ethernet, что исключает потери данных или их несвоевременное получение. Необходимые электронные устройства наилучшим образом подобраны для качественного детектирования искомых объектов.

Результаты работы опубликованы в виде научных статей в сборнике «Студенческие Дни науки в ТГУ – 2018», а также сборнике «Прикладная математика и информатика: современные исследования в области естественных и технических наук» – Тольятти: ТГУ, 2018.

Магистерская диссертация состоит из 4 глав, в которых решены упомянутые задачи. Общий объем работы составляет 86 страниц, включая 43 рисунка, 3 таблицы и 2 приложения. Список используемой литературы включает 30 наименований, включая 10 источников на английском языке.

1. Обзорная часть

1.1. Формулирование актуальности, цель и задачи работы

Существуют задачи компьютерного зрения, основанные на оценке движения, где последовательно идущий ряд изображений (видеоданные) обрабатываются для определения и оценки скорости каждой точки (пикселя) изображения или 3D (трёхмерной) сцены.[8]

Одним из примеров таких задач является – распознавание, то есть классификация или определения кем или чем является тот или иной объект (например, автомобиль или человек).[9]

Примером данной задачи из жизни являются системы видеофиксаций нарушения ПДД.

Довольно часто можно встретить на улицах камеры, висящие в самых разных местах, зачастую скрытно. Данные камеры записывают правонарушения на дороге в фото или видео формате. Европа, вот откуда пришёл к нам данный метод. Там он полностью производится в автоматическом режиме, а у нас только лишь в начальной стадии и набирает обороты.[10]

В данный момент в России ведется работа по следующим направлениям для развития системы видеофиксации ПДД:

- Проезд по полосе общественного транспорта;
- Непредоставление преимущества пешеходам;
- Проезд по встречной полосе движения;
- Не предоставление преимущества спец. транспорту с включенными сигналами.

Выбор темы диссертационной работы обусловлен несколькими факторами.

Во-первых, в данный момент не существует алгоритма и программного обеспечения для фиксации нарушения по непредоставлению преимущества для движения пешеходов.

Во-вторых, это необходимость развития технологии и алгоритмов компьютерного зрения, которые могут быть применены не только лишь для конкретной одной задачи.

В-третьих, это то, что видеофиксация нарушений ПДД является не только отличным антикоррупционным решением, но и осуществляет применение принципов неотвратимости наказания и может поспособствовать созданию большей безопасности для участников дорожного движения. [11]

Целью данной работы является: реализация программной и аппаратной части системы видеофиксации нарушения ПДД на пешеходном переходе.

Для достижения этой цели поставлены и выполнены следующие задачи:

1. Обзор существующих программных решений;
2. Обзор существующих аппаратных решений;
3. Выбор метода детектирования образов и обоснование выбора;
4. Разработка и реализация программы, управляющей логикой работы;
5. Разработка электрической схемы системы фиксации;
6. Обоснования выбора электронных компонентов системы;
7. Исследование влияния негативных факторов.

Путь решения поставленной задачи по обработке видеоизображения находится в применении методов детектирования, выборе библиотеки компьютерного зрения, написанной на языке программирования C++, а также использование программы Microsoft Visual Studio 2015. Для увеличения вероятности верного работы алгоритмов детектирования необходимо специально подготовить исходные материалы (повысить устойчивость к

помехам, снизить нелинейные искажения и помехи камеры). Важно помнить, что обработке подвергаются лишь изображения с градациями серого (без цветных компонент).

1.2. Обзор существующих решений

1.2.1. Существующие комплексы фиксации нарушений ПДД

Комплексы видеофиксации, применяемые в нашей стране, подразделяются по принципу работы на фото- и видеофиксации, лазерные и радарные. По типу расположения подразделяются на передвижные и стационарные.

Радар-детекторы, используемые большинством автомобилистов, запросто отслеживают работу радарных камер, которые посылают сигнал локатором. В ночное время таким комплексам необходима инфракрасная подсветка, а в её отсутствии они просто не будут работать. Ошибки в работе данного комплекса составляют более 30% проверенных автомобилей, например, из-за недостаточной видимости. Эффект Доплера является тем, на чём основана работа подобных комплексов.[12]

Стоит сказать, что эффект Доплера используется не только лишь для определения скорости передвижения машин, но и для детектирования скоростей прочей передвижной техники и, к тому же, облаков. Также этот эффект применяется в австрономии, где он используется для вычисления радиальной скорости перемещения небесных тел, и даже звезд и галактик.

Компании, разрабатывающие подобные системы, закладывают в них не только функции по детектированию превышения разрешенной скорости, но также и детектирование проезда на запрещающий сигнал светофора, заезд автотранспорта на встречную полосу движения, пересечение сплошной линии. Радарные системы могут исследовать до четырех полос для автотранспортных средств со скоростью их проезда до 250 км/ч.

«Стрелка-СТ» (см. рисунок 1), один из российских комплексов фото- и видеофиксации, которую не улавливает большая часть радар-детекторов. Радар, распознав скорость передвижения автотранспорта, передаёт сигнал фотокамере, которая делает снимок движущегося автомобиля, которая подъехала в зону возможности фотографирования. Так, система строится из двух частей: радара и камеры с широкоугольным объективом. Разработчики говорят о том, что данная система создана на основе разработок военной авиации России. Широкоугольный объектив позволяет «захватывать» до пяти полос движения автотранспорта одновременно. Автотранспорт может двигаться как от комплекса, так и к нему – это не важно. «Стрелка-СТ» применяют в основном на магистральных трассах в связи с высокой стоимостью системы.[13]

Рисунок 1 – Отечественный комплекс фото- и видеофиксации – «Стрелка-СТ»

Принцип работы данной системы схож с лазерными камерами, но они работают на большее расстояние, а максимальная скорость передвигающегося объекта не должна превышать 350 км/ч. А вот в плохую погоду они практически не работают.

Принцип действия комплекса:

- Излучаемые импульсы от видеорадара распространяются по всем полосам дорожного полотна;
- Отраженный от автотранспортных средств сигнал приходит на блок обработки, который находится на расстоянии до 1 км от

основного устройства. Подготовленные данные показывают скорость и расстояние до объекта.

- Применяемые технологии компьютерного зрения позволяют цифровой камере отсылать сигнал программе, которая рассчитывает скорость движения автомобиля на основе изменяющихся координат в пространстве при движении;
- Проанализировав данные, программа определяет транспортные средства нарушившие правила дорожного движения при подъезде к комплексу чуть менее 50 м. Данные получаются при совместной работе камеры и радара.

Опишем недостатки и преимущества комплекса «Стрелка-СТ»:

- Расстояние для детектирования правонарушения составляет до 1 км;
- Детектирование перемещения автотранспортных средств с числом кадров 12 в секунду;
- Погрешность в 2 км/ч максимальная для автотранспорта (объекта), находящегося на расстоянии до 50 м от камеры;
- Широкий диапазон распознавания скоростей;
- Автовыделение автотранспортных средств, движущихся с нарушением скоростного режима;
- Автоматическое распознавание гос. номера нарушившего ПДД автомобиля.

Новые системы видеофиксации нарушений ПДД «АвтоУраган» (см. рисунок 2) состоят лишь из широкоугольного объектива и соответственно камеры. В данном комплексе отсутствует радар. И лишь данные с видеокамеры позволяют определять ему скорость передвижения автомобилей с погрешностью до 2 км/ч. Основываясь на времени, которое тратит автомобиль для проезда определенного расстояния, система вычисляет разницу в положении объекта видеопоиска на различных кадрах и

соответственно рассчитывает скорость его передвижения. Геометрия дороги задается в комплексе в процессе его установки.[14]

Рисунок 2 – Комплекс фиксации нарушений «АвтоУраган»

«АвтоУраган-ВСМ» регистрирует до 16 разновидностей нарушений ПДД, находясь на столбе или специальной раме. В систему встроена программа для распознавания образов в видеоизображении. К примеру, система вычисляет контрольные зоны где располагаются автомобильные фары и тем самым может детектировать активность работы фар. Данная система стала основой для другого комплекса под названием «Паркон», детектирующую не верную парковку. Разумеется особенно актуальна данная система для паркующихся автомобилей в Москве.

Подобные системы не имеют радаров, но фиксируют нарушения ПДД благодаря обработке фотографий. Основываясь на времени, за которое автомобиль (его государственный номер) пересекает определенный участок дороги, происходит расчет, совершил ли его владелец правонарушение. Имеется возможность детектировать не только приближающиеся, но и удаляющиеся автотранспортные средства.

Данный комплекс может контролировать железнодорожный переезд или перекресток, следя за сигналами светофора, приходящими лишь благодаря камере, и автомобилист считается нарушившим правила, если выедет на перекресток на запрещающий сигнал.

Подобный комплексы имеют низкую погрешность измерений – до 5%. В отличии от радара, система не присвоит нарушение другому автомобилю, так как основывается на номерах конкретного автомобилиста и «помнит» лишь их. Радар-детектор не позволит отыскать подобные камеры.

Следующее устройство, которое следует рассмотреть, называется «Крис-П» (рисунок 3). Данное устройство также используется для детектирования правонарушений автомобилистами.[15]

Как стационарная, так и передвижная система, «Крис» распознает номера автомобилей и самостоятельно проверяет их по базам ГИБДД с дальнейшей отправкой данных постам ДПС (дорожно-патрульной службы).

Рисунок 3 – Принцип работы радара КРИС-П

Прибор ставится с применением треноги на краю дорожного пути таким способом, чтобы фотообъектив захватывал в одно и то же время все полосы дороги без исключения. Видеокамера системы детектирует правонарушения и отправляет данные о нарушении и водителе в ближайший центр приема сведений о нарушениях ПДД. Предоставление сведений реализуется с помощью GSM (глобальный стандарт цифровой мобильной связи) или сотовой связи, а кроме того с применением обычной телефонной линии.

Информация, составленная системой, отправляется на ПК (персональный компьютер). Кроме того данной системой возможно управлять с ПК, настраивать пороги скорости для детектирования, смотреть

выбранные данные о нарушителе. Отличительной чертой системы считается функция выбора цели, не смотря на направление движения.

На рисунке 4 представлена схема работы автоматических комплексов фото- и видеофиксации нарушений ПДД

Рисунок 4 – Схема работы автоматических комплексов фото- и видеофиксации нарушений ПДД

Как видим из представленной схемы, данные у стационарных комплексов передаются по проводным (Ethernet, UTP) или беспроводным (3G, WiMAX, 802.11) каналам связи в блок приема и конвертации данных, а затем в ЦАФАП (Центр автоматизированной фиксации административных правонарушений).

1.2.2. Ошибки фиксирования нарушения ПДД

Когда в самом начале говорили о фиксировании правонарушений автомобилистами камерами видеофиксации, то думали лишь о положительных последствиях от их работы автоматическом режиме. Все надеялись на соблюдения принципа неотвратимости наказания. О том, что не

будет существовать никаких сговоров с сотрудниками ДПС, и, не смотря на статус того или иного нарушителя, на него будет составлен проток о нарушении ПДД.

Число разновидностей нарушений, которые в данный момент уже умеют регистрировать камеры видеофиксации, неуклонно растет. Сейчас это уже не просто радары, умеющие фиксировать скорость движения автомобиля, но и камеры, регистрирующие выезд на перекресток на запрещающий сигнал светофора, разворот в неположенном месте, пересечение запрещенных дорожных полос, выезд за стоп-линию.

Тем не менее, с увеличением числа регистрируемых нарушений, выросло и число ошибок. И не смотря на то, что штрафы в любом случае должны проверяться и подписываться сотрудниками Госавтоинспекции у них не всегда хватает времени на то, чтобы беспристрастно проверять каждый протокол о правонарушении.[16]

Исследуем типичные ошибки детектирования правонарушения на дороге камерами видефиксации.

Штраф для автомобилиста, тень от автомобиля которого попала на запрещенную полосу (см. рисунок 5) или свет от фар (рисунок 6).

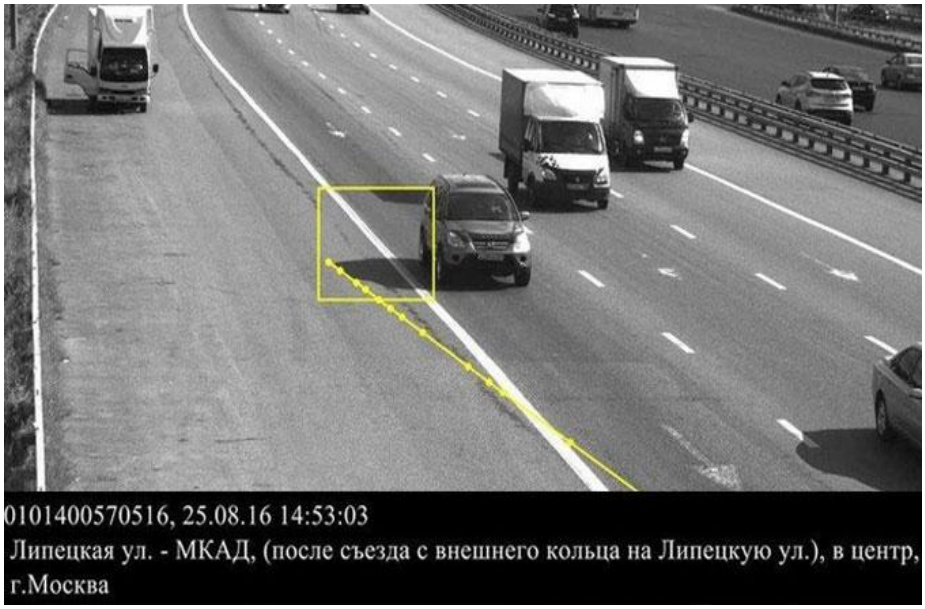

Рисунок 5 – Фотография из пришедшего штрафа за тень на обочине

Рисунок 6 – Фотография из пришедшего штрафа за блик от фар на обочине

Владельцу автотранспортного средства пришёл штраф за неправильную парковку, хотя его остановил сотрудник ГИБДД (см. рисунок 7).

Рисунок 7 – Фотография из штрафа за неправильную парковку

Незаконный штраф о повороте не из того ряда получил следовавший за нарушителем водитель Nissan (см. рисунок 8).

Рисунок 8 – Незаконный штраф о повороте не из того ряда

1.3. Обзор существующих алгоритмов и библиотек распознавания

1.3.1. Основная концепция

Распознавание образов – это теория из раздела информационных технологий и смежных дисциплин, улучшающих основы и методы детектирования и классификации предметов, явлений, процессов, сигналов, ситуаций и т. п. объектов, имеющих конечный набор некоторых признаков, характеристик и свойств. Такие задачи, требующие решения, возникают повсеместно, например, при переходе через дорогу или проезде через перекресток по определенному сигналу светофора. Распознавание цвета светофора даёт важную информацию для водителей и пешеходов о том, можно или нельзя в данный момент проезжать по перекрестку или соответственно переходить через дорогу.[17]

Важность и необходимость такого рода распознавания возникает в различных сферах деятельности – от систем безопасности

жизнедеятельности человека и военного дела до различных производственных нужд.

Особенно в производственной среде проблема распознавания особенно отражает себя. В ситуациях, когда возникает образная, эмоциональная и усталостная перегрузка, человек перестает правильно реагировать на поступающие к нему сигналы и сообщения, в результате чего он совершает ошибку и допускает брак.

В данный момент теория распознавания уже находится в поле зрения множества междисциплинарных исследований и может простираться от информатики и физики до биологии. В настоящий момент ведутся работы по созданию искусственного интеллекта и нейронных сетей, и технологии детектирования и классификации привлекают к себе всё большее внимание.

Выделим два основных направления:

- Изучение способностей животных к распознаванию, анализ их особенностей и построение соответствующих моделей;
- Развитие данной технологии с целью решения прикладных задач и создание программной и аппаратной части электронных устройств.

Рассмотрим классическое определение задачи распознавания объектов: имеем перед собой множество объектов. Необходимо их классифицировать, т.е. соотнести каждый объект в собственный класс. Это множество представлено подмножествами, которые можно назвать классами. Также дано: подробная информация о классах, описание данного целого множества и информация об объекте, по которому отсутствует информация о том, к какому классу он относится. Необходимо по имеющейся информации об описании нашего объекта и классах определить, – к какому классу можно отнести данный объект.[18]

Если говорить об оптическом распознавании, то можно применить методику перебора объекта с различными масштабами, сдвигами, под

различными видами (углами обзора), и пр. Для букв неклассифицированного языка можно перебрать шрифты, их размерности и пр.

Следующий подход — выделить контур объекта и проанализировать его характеристики (наличие скруглений, углов и т. д.)

Ещё один метод — применить искусственные нейронные сети. Данная методика способна по имеющимся базам данных с большим количеством примеров искомого объекта (с правильными и неправильными ответами, которые программа знает) производить его распознавание в изменяющихся условиях окружения.

1.3.2. Методы распознавания образов

Применение алгоритмов компьютерного зрения, начиная с момента их зарождения в 50-х годах прошлого века, во многом упростило жизнедеятельность человека и на данный момент является одним из перспективнейших вопросов для решения. На текущий момент задача распознавания образов не является решенной окончательно, поскольку процент ошибок, возникающий при применении различных методов детектирования объектов, все еще достаточно велик, и всё ещё нельзя говорить о совершенном машинном зрении в качестве прообраза человека.

Существует большое количество методов детектирования объектов и распознавания образов. Рассмотрим такие методы, как: SIFT-дескриптор, контексты формы, HOG-дескриптор (гистограмма направленных градиентов), признаки Хаара.

Чтоб сформировать SIFT-дескриптор (Scale Invariant Feature Transform) для начала определяют показатели величины и угла наклона градиента у всех пикселей, принадлежащих окрестности особой области размером 16x16. Величины градиентов при этом записываются со значениями, сравнимыми с величиной функции плотности нормального

распределения и математическим ожиданием в исследуемой части изображения и стандартным отклонением, равняющимся половине ширины окрестности (веса Гауссова распределения применяются так, чтобы снизить воздействие на результирующий дескриптор, рассчитанных в пикселях, расположенные дальше от особой области).[19]

Во всех квадрантах размером 4x4 пикселя рассчитывается гистограмма ориентированных градиентов с помощью добавления взвешенного значения величины градиента к одному из 8 бинов гистограммы. Для снижения различных "граничные" эффекты, связанных с соотнесением схожих градиентов к разным квадрантам (это может возникать из-за небольшого сдвига расположения особой области) применяется билинейная интерполяция: значение величины всех градиентов добавляется не только в гистограмму, соответствующую квадранту, к которому текущий пиксель принадлежит, но и к другим, которые относятся к близлежащим квадрантам. При этом значение величина прибавляется со значением, пропорциональным расстоянию от пикселя, в котором рассчитан текущий градиент, до центра соответствующего квадранта. Все рассчитанные гистограммы складываются в единый вектор, размером, соответствующим 128=8 (число бинов) 4x4(число квадратов).

Рассчитанный дескриптор изменяется, чтобы уменьшить возможные эффекты от изменения освещенности. Отредактировав контраст изображения (величина интенсивности каждого пикселя умножается на некоторую константу) может также приводить значениях величин градиентов. Поэтому очевидно, что этот эффект может быть снижен путем нормализации дескриптора таким образом, чтобы его длина стала равна единице.

Редактирование яркости изображения (к значению интенсивности каждого пикселя прибавляется некоторая константа) не влияет на значения величин градиентов. Таким образом, SIFT-дескриптор является инвариантным по отношению к аффинным изменениям освещенности. С

другой стороны могут возникать и нелинейные изменения в освещенности вследствие, например, различной ориентации источника света по отношению к поверхностям 3D объекта. Данные эффекты могут вызвать большое изменений в отношении величин некоторых градиентов (при этом оказывается незначительное влияние на угол наклона вектора градиента). Чтобы это избежать, применяют отсечение по некоторому порогу (по результатам экспериментов выяснено, что оптимальное значение 0,2), которое применяют к компонентам нормализованного дескриптора. После применения порогового преобразования дескриптор вновь нормализуется. Таким образом, уменьшая значение слишком больших величин градиентов и увеличивая значение распределения ориентаций исследуемых градиентов в окрестности особой области.

На рисунке 9 схематично показана часть изображения (слева) и (справа) полученный на её основе дескриптор.

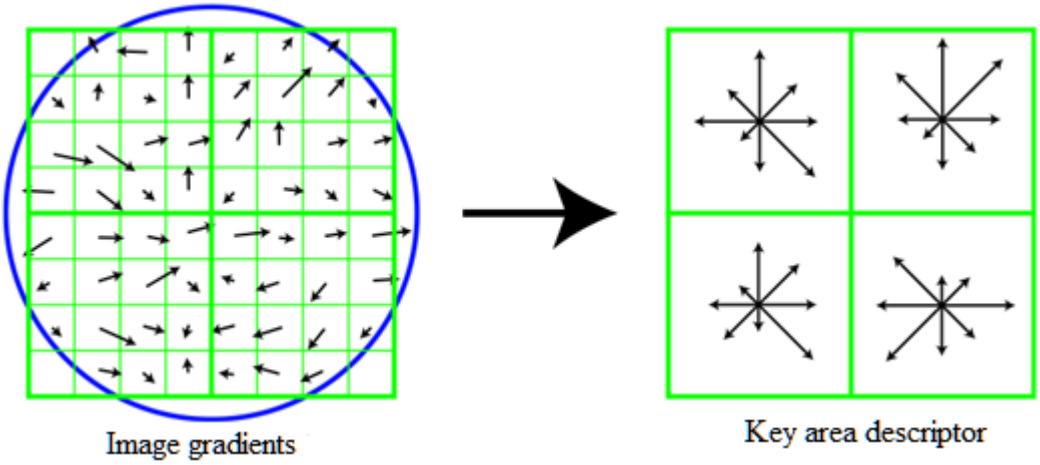

Рисунок 9 – дескриптор SIFT

Следующий метод основан на дескрипторе контекста формы с применением Support Vector Machine (SVM) классификатора.[20]

На рисунке 10 изображена общая схема классификации для рассматриваемого алгоритма.

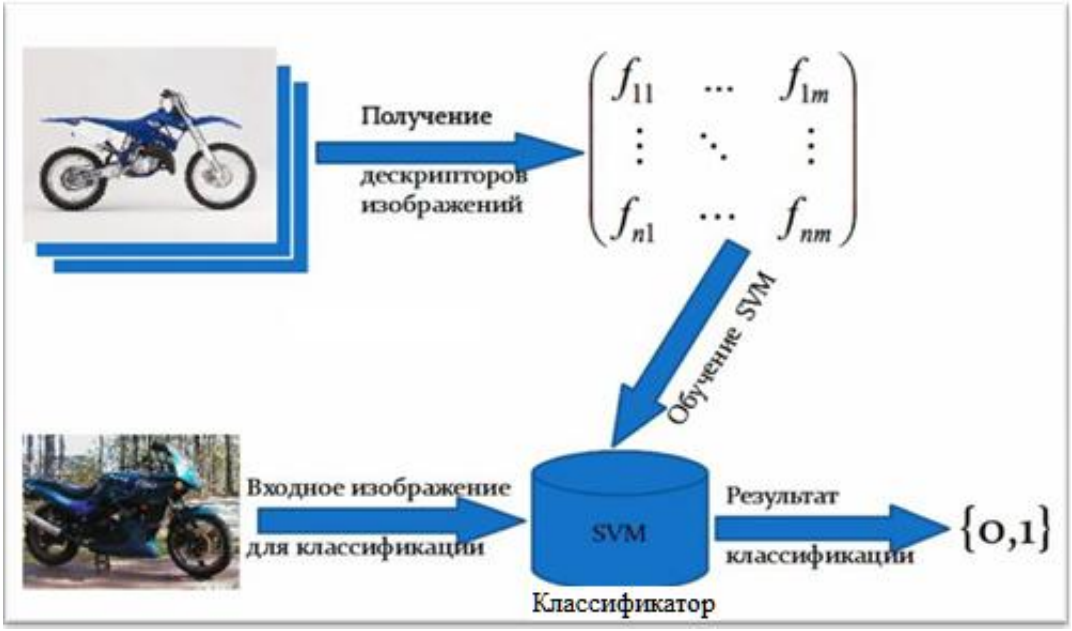

Рисунок 10 – Общая схема распознавания образов и классификации на основе SVM

Рассмотрим методику формирования матрицы дескрипторов. На рисунке 11 схематично представлена данная методика.

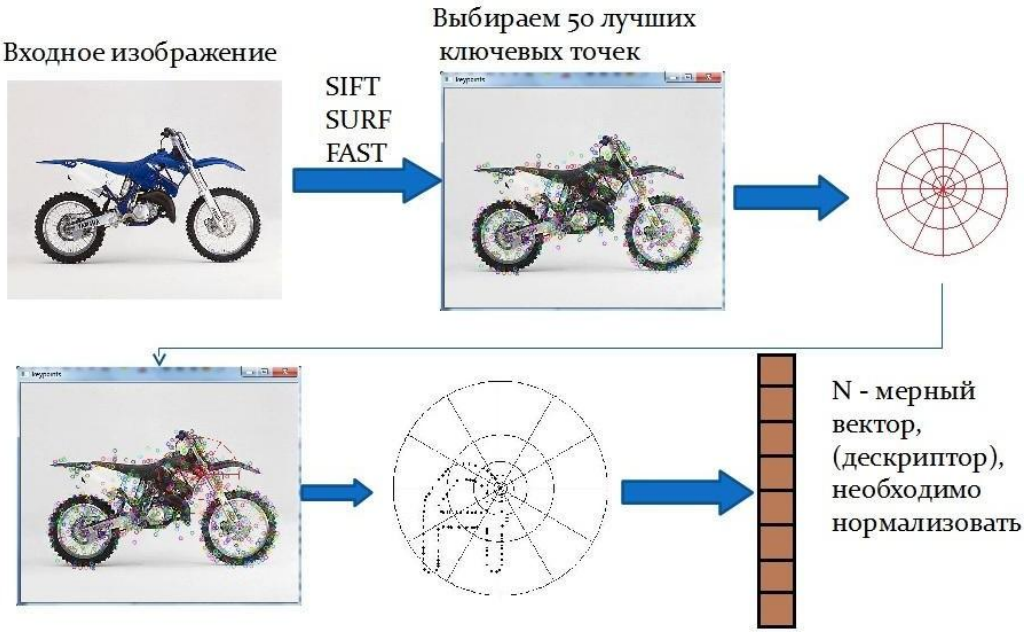

Рисунок 11 – Методика формирования матрицы дескрипторов

Разберём процедуру каждого из этапов формирования дескриптора:

1. Поступающее на вход изображение;

2. Производится построение пирамиды изображений с разными масштабами. В данной случае используется помимо входного изображение два дополнительных с удвоенным размером и уменьшенным в два раза от оригинала;

3. Применяя любой из известных алгоритмов производим выделение ключевых областей на изображениях;

4. Выбирается 50 лучших точек. Основываясь на трёх изображениях, сформируем три облака ключевых точек и, сравнивая их, возьмем самые лучшие, которые наилучшим образом описывают искомый объект;

5. Дескриптор формы распределенный на 48 бинов рассчитывается в каждой ключевой области (точки);

6. Соединяем все дескрипторы, вычисленные для ключевых областей, и получаем один выходной дескриптор, описывающий исходное изображение.

После формирования дескриптора он подается на вход классификатора, и уже классификатор выясняет, имеет ли данный объект отношение к определенному классу объектов или не имеет.

Следующий исследуемый метод распознавания объектов называется гистограмма направленных градиентов или Histogram of Oriented Gradients (HOG).[21]

Из теории по основной концепции алгоритма HOG: объект может быть описан по внешнему виду и его форме в определенной области изображения, основываясь на распределении интенсивности градиентов в данной области или же направлением краев.

Реализовать данный дескриптор можно с помощью разделения изображения на небольшие связанные между собой области, называемые ячейками. Для каждой ячейки рассчитывается гистограмма направлений градиентов или же, для пикселей, направления краев, расположенных внутри

ячейки. Комбинация рассчитанных гистограмм и является решением детектирования (дескриптором).

Рассмотрим метод на изображении, представленном рисунком 12.

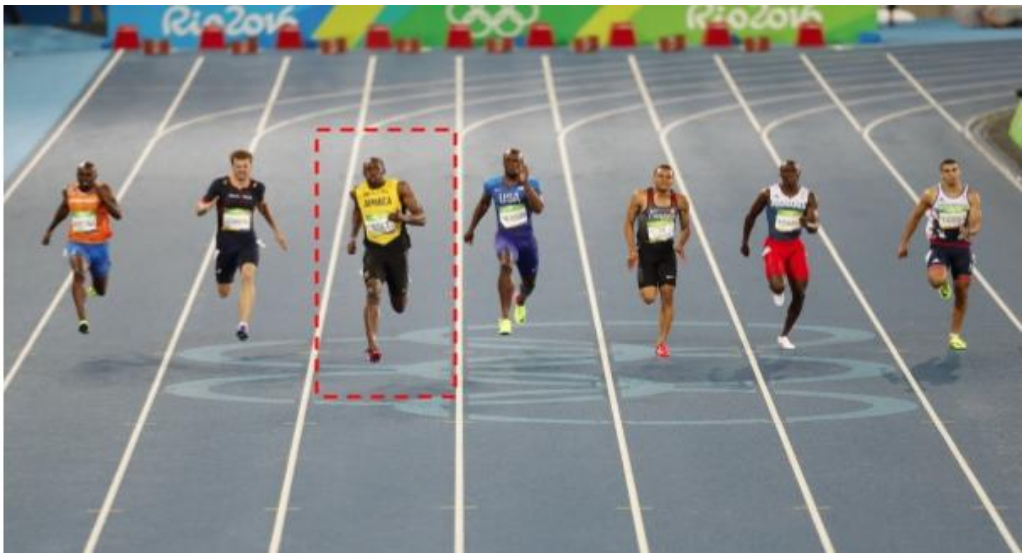

Рисунок 12 – Исследование метода HOG

Для реализации HOG алгоритма на первом шаге рассчитываются значения градиентов с применением одномерной дифференцирующей маски в горизонтальном и/или вертикальном направлении.

Затем находится величина и направление градиента, используя следующие формулы:

$$g = \sqrt{g_x^2 + g_y^2}\,, \qquad (1)$$

$$\theta = \arctan\frac{g_y}{g_x}\,, \qquad (2)$$

где g – величина градиента;

g_x – величина градиента по оси x;

g_y – величина градиента по оси y;

θ – направление градиента, град.

Изображение разбивается на ячейки 8x8 пикселей (рисунок 13), и определяется величина и направление градиентов в каждой из них.

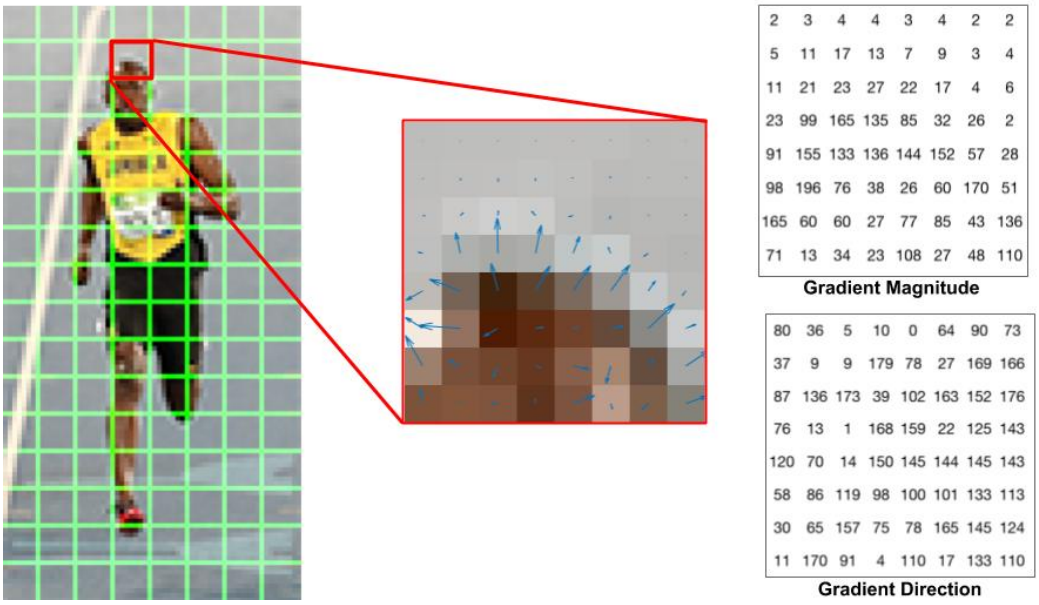

Рисунок 13 – Разбиение скользящего окна на ячейки и нахождения градиентов

Данный алгоритм имеет ряд неоспоримых преимуществ перед другими дескрипторами. HOG-дескриптор работает локально на определенной области, алгоритм поддерживает возможность применения фотопреобразования: геометрических пропорций без изменения ориентации, аффинных или же пороговых. Как обнаружили создатели данного алгоритма Триггс и Далал, жесткое разделение пространства на составляющие, точный расчет направления градиентов и сильная нормализация изображения позволяют игнорировать неточности детектирования, которые появляются в других методах, при определении пешеходов находящихся в движении, при условии, что они поддерживают при движении своё тело вертикально.

Таким образом, HOG-дескриптор – один из лучших алгоритмов детектирования людей на фото- и видеоизображениях, но он также может применяться и для любых других объектов, например, животных и автомобилей.

В сравнении с другими дескрипторами, HOG-алгоритм содержим несколько преимуществе. Например, работая по области, он способен

исследовать изображения, подвергнутые различного рода преобразованиям, исключая изменение ориентации. Такие изменения возможны лишь с большими по размеру областями. Создатели данного метода определили, что, разбивая изображение на области, применяя изображения в градациях серого и нормализацию, можно добиться детектирования объектов, даже если он перемещается с небольшой скоростью. В случае с пешеходами, они должны иметь строго вертикальное положение тела. Данный дескриптор идеально подходит для обнаружения людей в кадре, но тем не менее, может использоваться и для любых других объектов, к примеру, автомобилей и даже животных.

В соответствии с углом от 0 до 160 градусов заполняются столбцы будущей гистограммы (рисунок 14).

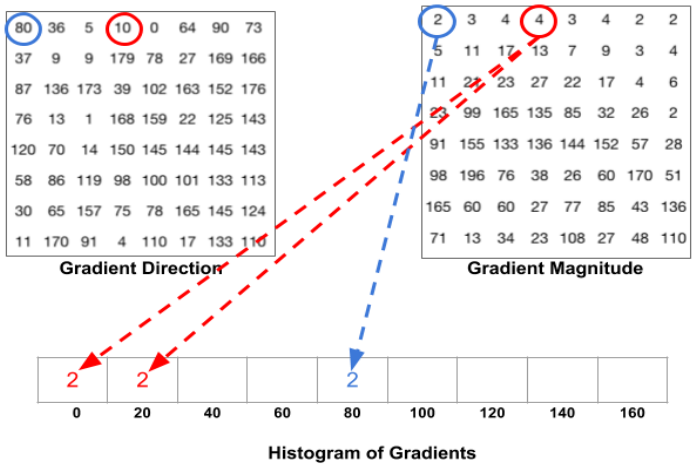

Рисунок 14 – Заполнение столбцов будущей гистограммы

Вклад каждой ячейки с одинаковыми углами складывается и строится гистограмма (рисунок 15).

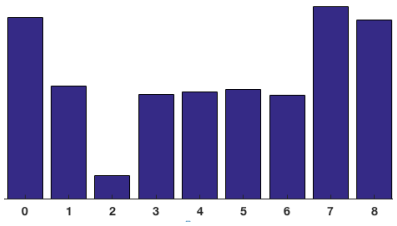

Рисунок 15 – Построение гистограммы на основании величин и направлений векторов в выбранной ячейке

Также производят нормализацию гистограммы по контрасту, чтобы результат её построения не был зависим от освещенности.

Визуализацию направленных градиентов можно представить как на рисунке 16. доминирующее направление гистограммы фиксирует фигуру человека (это отчётливо видно вокруг туловища и ног).

Рисунок 16 – Визуализация построения градиентов

Для создания же системы детектирования образов с помощью данного метода необходимо воспользоваться классификатором SVM.

Изменяющаяся освещенность является фактором снижающим точность детектирования. В связи с этим нормализация контраста помогает немного повысить качество идентификации объектов.

В процессе работы программы осуществляется поиск людей и автомобилей сформированным скользящим окном определенной размерности, внутри которого происходит поиск заданных HOG-признаков. Необходимые HOG-признаки закладываются заранее на этапе обучения классификатора SVM или машины опорных векторов. Обучение происходит множеством фотографий (шаблонов), которые содержат искомый объект или не содержат его вовсе.

В результате исследования изображений-шаблонов получим информацию в виде коэффициентов, которые в процессе работы программы

SVM разделяет на две области. Граница между двумя областями является критерием решения задачи детектирования.

В дальнейшем в зависимости от полученных коэффициентов в скользящем окне программа определяет, имеется ли искомый объект в данной конкретной области или он отсутствует.

Следующим алгоритм детектирования основан на признаках Хаара. По своему названию они схожи с одним из первых простейших математических вейвлетов. Признаки Хаара применялись для создания самого первого детектора лиц.[22]

Так сложилось, что все методы, основанные на распределении интенсивности пикселей в изображении (пример, цветовую модель каждого пикселя) являются сложными для вычисления. Математические вейвлеты Хаара являются основой для создания детектора на основе признаков. Джонс и Виола придумали использовать вейвлеты Хаара и разработали то, что сейчас именуется признаками. Признаки строятся на основе смежных прямоугольных областей. Они идентифицируются на изображении, затем суммируется интенсивность пикселей данных выделенных областей, а затем вычисляется разность между областями. Эта разность и является величиной определенного признака, определенной размерности. Данный признак, можно сказать, спозиционирован особым образом на изображении.

Здесь же, в методе Виолы–Джонса, применяется метод скользящего окна определенной размерности, движущегося по изображению и рассчитывающего признак Хаара. Наличие или отсутствие искомого объекта определенного класса на изображении определяется разницей между признаком и обученным порогом. Признаки Хаара не очень хорошо подходят для обучения и классифицирования, таким образом, в методе Виолы–Джонса признаки организуются в каскадном классификаторе.

Особенностью данного метода является высокая скорость вычисления. Благодаря использованию интегрального представления

изображения время вычисления может быть постоянным, независимо от изображения (приблизительно 60 микропроцессорных инструкций для признака, состоящего из сдвоенной области).

В алгоритме используются признаки различных типов, различных масштабов и положений (рисунок 17). Внутри окна обнаружения таких признаков генерируется более 100,000.

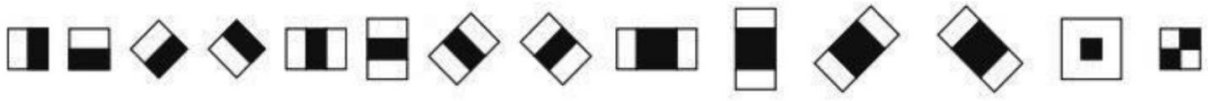

Рисунок 17 – Типы Хаар признаков. Чёрная закраска соответствует положительному весу, белая - отрицательному

1.3.3. Тестирование алгоритмов обнаружения

Даллал и Триггс производили эксперимент по детектированию людей, сравнивая дескрипторы R-HOG, C-HOG, вейвлеты Хаара и контексты формы. В 2001 году обобщённые вейвлеты Хаара применялись Моханом, Папагеоргиу и Поггио в собственных тестах по детектированию объектов.

Существующий метод детектирования SIFT был модифицирован добавлением главных компанент, что проявилось в создании дескриптора PCA-SIFT. Впервые данный дескриптор был исследован в 2004 году в работе Ке и Суктханкара; заявлялось, что PCA-SIFT превосходит по своим характеристикам обычные SIFT-дескрипторы.

Контексты формы, наравне с C-HOG, используют при детектировании круглые бины, но не учитывают ориентацию, а лишь в основном наличие края. Данный алгоритм был создан и исследован в 2001 году в работе Белонги, Малик и Пузича.

Протестировать данные алгоритмы удалось на двух различных наборах данных. MIT (Массачусетский технологический институт) имеет в своем распоряжении базу из 509 изображений, а также тестовую выборку из

200 фотографий. Данная база содержит фотографии людей лишь спереди или сзади, а их позы практически неотличимы. Данная база широко известна в научных кругах и применяется в различных исследованиях. Другая база фотографий была специально разработана Далалом и Триггсом для их исследований, так как на базе фотографий MIT были получены слишком идеальные результаты.

Они назвали свою базу данных INRIA. Она включает 1805 фотографий людей. Их позы не статичны и разные. Также данная база включает фотографии с «трудным» фоном (пример, на фоне толпы людей), таким образом, являясь более сложной для детектирования и распознавания, нежели набор MIT.

В результате экспериментов (таблица 1), дескрипторы C-HOG и R-HOG дают практически идентичные результаты, с учётом, что C-HOG имеет несколько меньшую долю пропущенных изображений при фиксированной доле ошибок первого рода (ложные срабатывания) на обеих базах фотографий.

Таблица 1 – Результаты исследования алгоритмов детектирования

Дескриптор	**Набор изображений**	**Доля пропущенных изображений**	**Доля ошибок первого рода**
HOG	MIT	≈0	10^{-4}
HOG	INRIA	0.1	10^{-4}
Обобщенные вейвлеты Хаара	MIT	0.01	10^{-4}
Обобщенные вейвлеты Хаара	INRIA	0.3	10^{-4}
PCA-SIFT, контексты формы	MIT	0.1	10^{-4}
PCA-SIFT, контексты формы	INRIA	0.5	10^{-4}

На семинаре Pascal Visual Object Classes в 2006 году, экспериментаторы, предъявили результаты использования HOG-дескрипторов при детектировании на фотографиях не только людей, но и автомобилей, автобусов, животных, а также оптимальные характеристики при формировании и нормализации блоков в этих случаях.

Не осталась без внимания Европейская конференция по машинному зрению 2006 года, где Корделия Шмид совместно с Триггсом и Далалом использовали HOG-дескрипторы при детектировании людей на видеоизображении. Способ, который они предложили довольно таки прост так как HOG-дескриптор применялся на каждом кадре, а гистограммы внутреннего движения (англ. Internal Motion Histograms, IMH) на нескольких последовательно идущих кадрах.

1.3.4. Библиотеки компьютерного зрения

Библиотека алгоритмов компьютерного зрения, которая позволяет производить детектирование движения в видеопотоке, и которая будет использоваться при решении поставленных задач диссертационной работы, называется OpenCV (Open Source Computer Vision Library). Логотип показан на рисунке 18 и представляет собой выравненные по вершинам треугольника три заглавные буквы O, C и V.[23]

Рисунок 18 – Логотип OpenCV

Библиотека является бесплатной, как для коммерческого применения, так и для использования в целях обучения. Реализуется на языках

программирования C++, C, Python и Java. Поддерживает современные операционные системы (ОС), такие как Windows, Mac OS, Linux, а также ОС для смартфонов: Android и IOS. Данная библиотека разработана преимущественно для решения задач в режиме реального времени, как, например, распознавание людей на видеопотоке на рисунке 19.

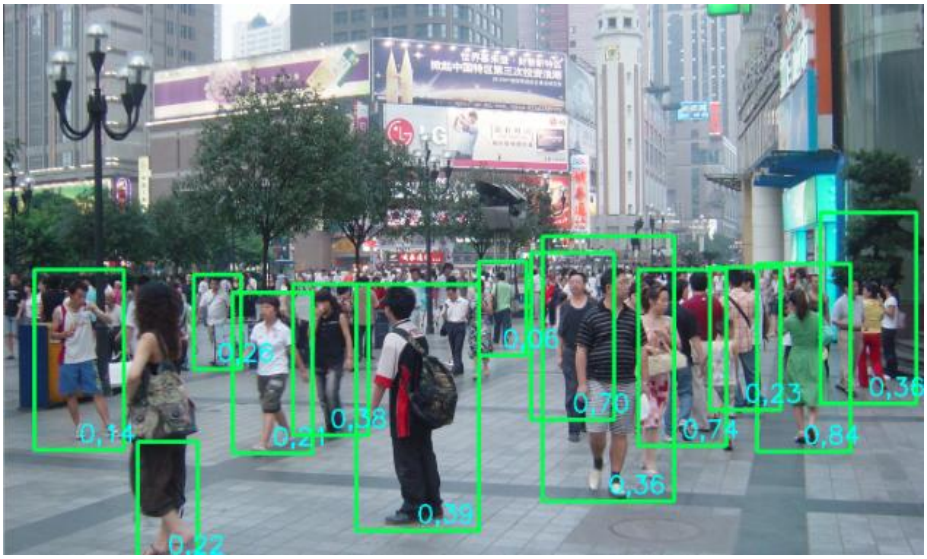

Рисунок 19 – Распознавание людей в видеопотоке с помощью OpenCV

Библиотека позволяет обрабатывать простые изображения и фото, производить геометрические преобразования, фильтрацию (шумов и пр.), производить преобразование цветовых пространств.

При работе с видео в режиме реального времени OpenCV способна анализировать движение, распознавать и отслеживать объекты.

Библиотека имеет более 2500 современных оптимизированных алгоритмов компьютерного распознавания и обучения, которые могут использоваться чтобы:

- Обнаруживать и идентифицировать лица;
- Определять объекты;
- Классифицировать человеческие поступки в видеопотоке;
- Отслеживать перемещение объекта;
- Извлекать 3D модели объектов.

Мировая практика использования данной библиотеки: обнаружение проникновения на охраняемые объекты в Израиле, мониторинг оборудования в Китае, обнаружение несчастных случаев утопления в бассейнах в Европе, проверка взлётно-посадочной полосы на наличие мусора в Турции, проверка продуктовых этикеток на фабриках по всему миру, скоростное обнаружение лиц в Японии.

Следующей библиотекой для рассмотрения является Computer Vision System Toolbox (система, содержащая набор программ разработки и проектирования для компьютерного зрения). Она позволяет с помощью встроенных инструментов и алгоритмов производить разработку моделей и систем по обработке фото- и видеоизображений.[24]

Библиотека включает средства выявления деталей объекта, детектирования и отслеживания движения, стереозрения, анализ и обработку видопотока. Содержит инструменты для импорта/экспорта видео-документов, отображения кадров, компоновки и сборки графики.

Данная библиотека входит в состав математического пакета MATLAB (рисунок 20), представляемая в качестве системных блоков, функций, а также визуальных блоков Simulink. Рассматриваемый набор инструментов вычисление арифметики с фиксированной точкой, и разработку кода на C, что позволит быстро разрабатывать модели и прототипы необходимых систем.

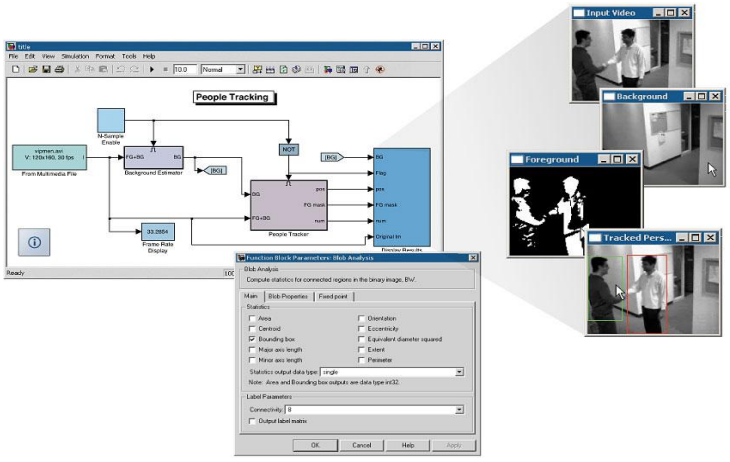

Рисунок 20 – Алгоритмы компьютерного зрения в MATLAB

Также стоит упомянуть библиотеку CCV (A Modern **Computer Vision** Library). Данная библиотека написана на языке C и может рассматриваться как компактная альтернатива библиотеки OpenCV. Из CCV изъяты все редкоиспользуемые функции и возможности, которыми так наполнена OpenCV библиотека. CCV предназначается не для экспериментов с объектами, посредством применения различных алгоритмов, как это делается в других библиотеках, а для практического применения в конкретных задачах и приложениях.

В библиотеке CCV помогает использовать следующие алгоритмы:

1. Скоростной метод распознования недвижимых объектов BBF (Brightness Binary Feature);
2. Детектор объектов сложной формы DPM (Deformable Parts Model);
3. Оператор распознавания текста на изображении SWT (Stroke Width Transform);
4. SIFT алгоритм обнаружения наборов областей и точек при детектировании.

1.4. Обзор существующих аппаратных решений

1.4.1 Одноплатный компьютер или процессорный модуль

Одноплатный компьютер (SBC) представляет собой компьютер, построенный на единой плате, с микропроцессором, памятью, портами входа/выхода (I/O) и обладающий возможностями необходимыми для функционирования полноценного компьютера. Одноплатные компьютеры были созданы как системы демонстрации или разработки, для образовательных систем или для использования в качестве встроенных контроллеров. Они объединяют многие типы домашних или переносных компьютеров на одной печатной плате.[25]

Процессорные модули применяются, если необходимо реализовать собственную несущую плату, а одноплатный компьютер, как готовое решение, позволяет не тратить время на разработку и производство. Это позволяет сосредоточиться главным образом на программной части устройства. Стоимость одноплатных компьютеров может оказаться ниже, чем разработка своей платы, и также их использование позволяет как можно скорее выводить продукцию на рынок. Разумеется, одноплатные компьютеры выпускаются в таких форматах, чтобы удовлетворять требования любых заказчиков. Это сказывается на том, что всегда будет иметься избыточное количество портов и интерфейсов, а также, возможно, и количество и объем памяти. Минусы применения подобных компьютеров в том, что при появлении на рынки новых, более совершенных модификаций, если понадобиться обновить имеющееся оборудование, необходимо будет смотреть на новое расположение портов, а также на необходимость переустановки всего применяемого программное обеспечение (ПО). Самые часто используемые форм-факторы: 3,5″ (146×102 мм), 2,5″(100×72 мм, альтернативное название Pico ITX) и PC/104 (96×90 мм).

На рисунке 21 можно увидеть процессорные модули. Данные компоненты неплохо избавляют общую систему от избыточности и устанавливаются на печатную плату. При разработке устройства на процессорном модуле удобно следовать всем условиям технического задания, оглядываясь на размер платы для разрабатываемого устройства. Разработчик устанавливает лишь те интерфейсы, порты и периферию, и то их количество, которое заложено в задании. Также, благодаря тому, что большинство модулей стандартизировано, система может быть модифицирована простой заменой модуля на другой модуль того же размера на плате. Изменяя модуль можно изменить целую архитектуру процессора и системы, а также, если потребуется, сделать её более производительной. Появляется возможность применять всё более новые выпускаемые

процессоры, что также увеличивает срок жизни устройства, снижает устаревание и позволяет не зависеть от какого-то конкретного производителя.

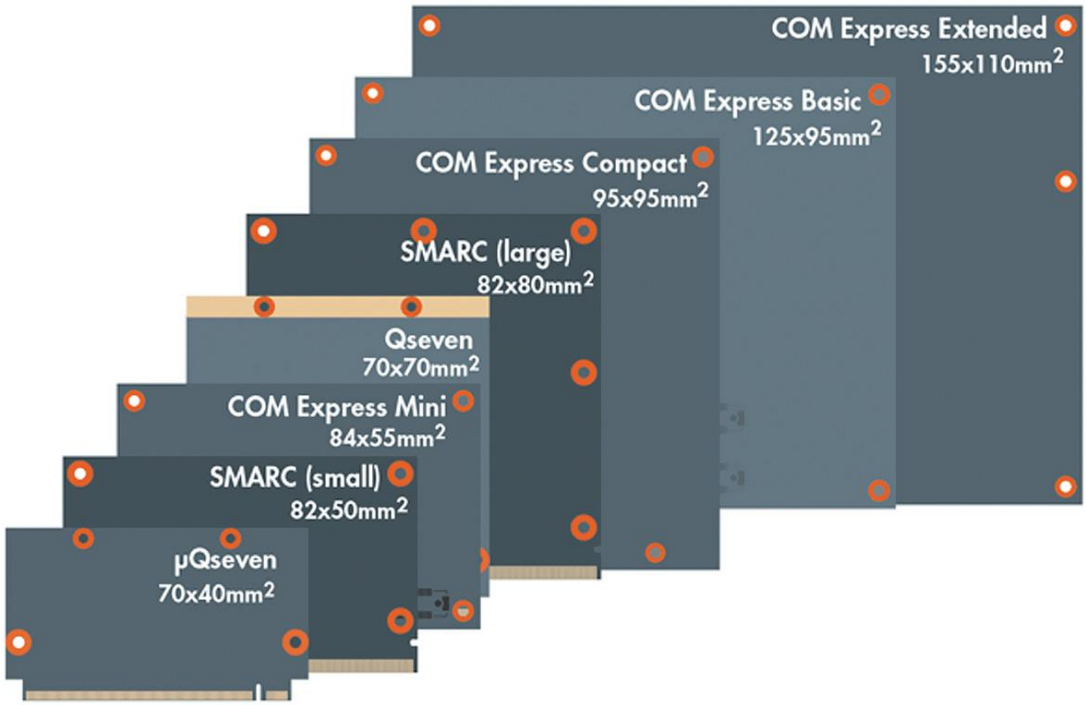

Рисунок 21 – Виды процессорных модулей

При сравнении одноплатных компьютеров и персональных, первые чаще не полагаются на слоты для периферийных устройств или расширений. Одноплатные компьютеры разрабатываются с применением большого множества микропроцессоров. Упрощенные в сборке, например, разработанные любителями в данной области, чаще применяющими статическое оперативную память (ОЗУ) и дешевые 8 или 16-разрядные процессоры.

Первый настоящий одноплатный компьютер под названием «dyna-micro» был основан на Intel C8080A, а также использовался первый EPROM от Intel - C1702A. Dyna-micro была переименована в E&L Instruments of Derby, CT в 1976 году как «MMD-1» (Mini-Micro Designer 1), представленная

на рисунке 22, и была известна как пример микрокомпьютера в очень популярной серии 8080 «BugBook». SBC также задумывались в ранней истории домашних компьютеров, например, в Acorn Electron и BBC Micro. Другие типичные ранние одноплатные компьютеры, такие как KIM-1, часто отправлялись без приложений, которые должны были быть добавлены владельцем.

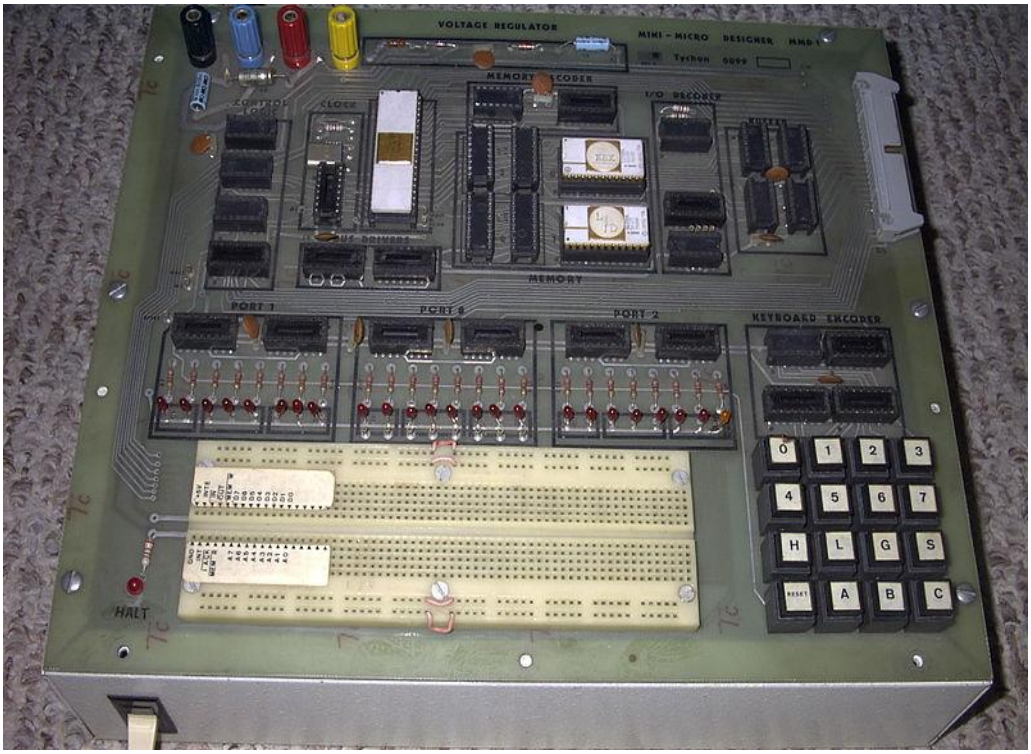

Рисунок 22 – Прототип MMD-1, 1976

По мере того, как рынок ПК стал более распространенным, в компьютерах использовалось меньше SBC. Основные компоненты были собраны на материнской плате, а периферийные компоненты, такие как последовательные порты, контроллеры дисков и графические процессоры, были расположены на дочерних платах. Доступность современных чипов, обеспечивающих большинство функций ввода-вывода встроенных компонентов, позволяет производителям материнских плат предлагать материнские платы с I/O, традиционно предоставляемыми дочерними платами.

Большинство материнских плат теперь предлагают встроенную поддержку дисковых накопителей, включая IDE и SATA с RAID, графикой, Ethernet и традиционным вводом-выводом, такими как последовательный и параллельный порты, USB и клавиатура/мышь. Встраиваемые карты теперь чаще представляют собой высокопроизводительные графические карты (на самом деле графические сопроцессоры), высокопроизводительные RAID-контроллеры и специализированные карты ввода-вывода, такие как сбор данных и платы DSP (Digital Signal Processor).

Одноплатные компьютеры стали возможными благодаря увеличению плотности интегральных схем. Одноплатная конфигурация снижает общую стоимость системы за счет сокращения количества требуемых печатных плат и устранения соединительных схем и шины, которые в противном случае были бы применены. Поместив все функции на одной плате, можно получить меньшую общую систему, например, как в ноутбуках. Коннекторы являются частым источником проблем с надежностью, поэтому одноплатная система устраняет эти недочёты.

Одноплатные компьютеры чаще всего используются в промышленности, где они применяются в формате стойки для управления технологическими процессами или встроены в другие устройства для обеспечения управления и взаимодействия. Они были на глубинах океана (глубинные морские зонды ALICE), в космическом пространстве (ARRIANE, Pegasus Rocket и Shuttle) и на всех континентах планеты. Из-за очень высоких уровней интеграции, уменьшения количества компонентов и сокращения количества подключений, SBC часто меньше, легче, более энергоэффективны и более надежны, чем сопоставимые многоплатные компьютеры.

Основным преимуществом материнской платы ATX по сравнению с SBC является стоимость. Материнские платы производятся миллионами для потребительских и офисных рынков, что позволяет добиться огромной

экономии за счет масштаба. Одноплатные компьютеры являются рыночной нишей и производятся реже и по более высокой цене. Материнские платы и SBC теперь предлагают аналогичные уровни интеграции функций, что означает, что отказ материнской платы в любом стандарте потребует эквивалентной замены.

В общем разнообразии одноплатных компьютеров используются стандартные компьютерные форм-факторы, предназначенные для использования в корпусе объединяющей платы. Некоторые типы одноплатных компьютеров – CompactPCI, PXI, VMEbus, VXI и PICMG. SBC построены вокруг различных внутренних структур обработки, включая архитектуру Intel, многопроцессорные архитектуры и более низкомощные системы, такие как RISC и SPARC. В мире ПК Intel интеллектуальная и интерфейсная/управляющая схема размещены на однородной плате, которая затем вставляется в пассивную (или активную) объединительную плату. Конечный результат аналогичен системе, построенной с материнской платой, за исключением того, что объединительная плата определяет конфигурацию слота. Объединительные платы поставляются с набором слотов (ISA, PCI, PCIX, PCI-Express и т. Д.), обычно на 20 или менее, то есть он будет входить в 19-дюймовый корпус в стойку (рейку шириной 17 дюймов).

На некоторых одноплатных компьютерах есть разъемы, которые позволяют собирать стопку печатных плат, каждая из которых содержит расширительное оборудование, без традиционной объединительной платы. Примеры укладки форм-факторов SBC включает в себя PC / 104, PC / 104-Plus, PCI-104, EPIC и EBX; эти системы обычно доступны для использования во встроенных системах управления.

Разные типы укладок SBC часто имеют память, поставляемую на модулях, таких как SIMM и DIMM. Платы жестких дисков также не учитываются для определения того, является ли компьютер SBC или нет по двум причинам, во-первых, поскольку жесткий диск рассматривается как

единый блок хранения данных, а во-вторых, поскольку SBC может вообще не требовать жесткого диска, поскольку большинство могут быть загружены из сетевого подключения.

Одноплатные компьютеры применяются при жестких сроках выполнения проекта, когда, времени разработки для несущей платы уже совсем мало, а проект необходимо реализовать в ближайшие пол года. Данные компьютеры практически выполненная система, в которой требуется лишь подключить электропитание, подключить необходимые слоты расширения под определенные интерфейсы, установить операционную систему и отдалить ПО.

Также, одноплатные компьютеры целесообразно разрабатывать при небольших партиях продукции (к примеру, 200 шт.), тогда как затраты средств, которые понадобятся, если вы захотите использовать процессорный модуль.

При длительных сроках эксплуатации системы, от 10-15 лет, целесообразнее использовать систему с собственным процессорным модулем. Системы на процессорных модулях разрабатываются с определенным запасом по мощности, тогда как одноплатные компьютеры обладают меньшим сроком жизни и зависит от составных компонентов установленных на его плате. Одноплатный компьютер уже невозможно модифицировать, и в будущем будет необходима его замена, если понадобится более мощная вычислительная система. Таким образом, важнейшим компонентом одноплатного компьютера является его процессор и чипсет.

Практически каждая новая серия процессоров компании Intel производится в течение 7 лет, а увеличенный срок жизни системы на собственном разработанном процессорном модуле определяется тем, что имеется возможность произвести простую замену модуля на плате на более современный.

Также сложным выбором является выбор между x86 процессорами и ARM. Очень важную роль играет энергопотребление и разрабатываемое ПО. Если используется Windows, то выбор идет в пользу x86 процессоров, а если используется система на базе Linux, где необходимо сверхнизкое энгергопотребление – в пользу ARM.

В итоге, можно судить о том, что у разработчика всегда должно быть несколько вариантов, какую систему лучше использовать и на чём она должна основываться.

1.4.2 Сравнение современной линейки одноплатных компьютеров

Когда дело доходит до одноплатных компьютеров, Raspberry Pi является бесспорным лидером в своём классе. Конечно, это не самый мощный или самый компактный или даже самый дешевый одноплатный компьютер, но его преданными являются множество пользователей.[26]

Текущая версия – модель Raspberry Pi Model 3 B+ (рисунок 23), выпущенная для «International Pi Day» 14 марта 2018 года. Спецификации Topline – это четырехъядерный процессор с тактовой частотой 1,4 ГГц, двухдиапазонная беспроводная локальная сеть, Bluetooth 4.2/BLE, более быстрый Ethernet, и поддержка Power-over-Ethernet (с отдельным PoE HAT).

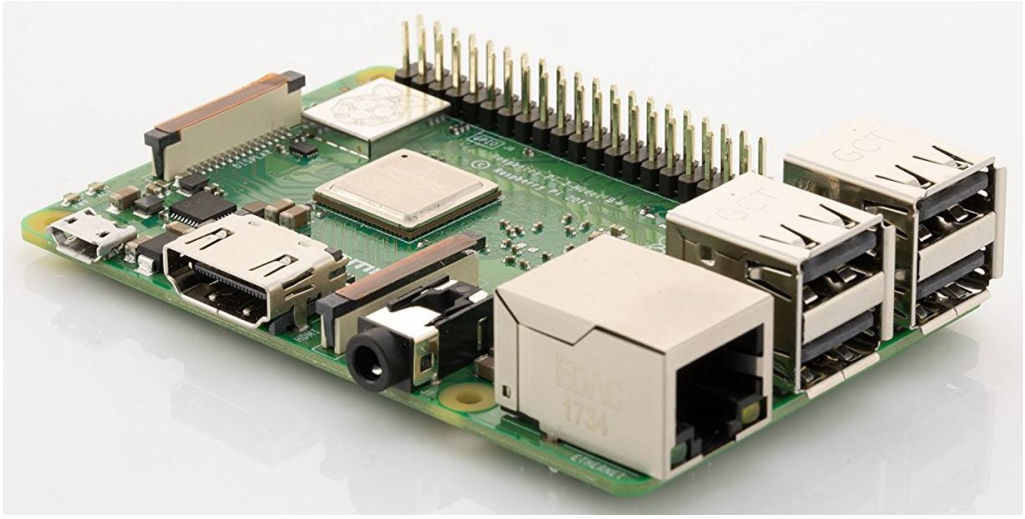

Рисунок 23 – Изображение Raspberry Pi Model 3 B+

В розничной торговле он стоит 35 долларов США, и правда состоит в том, что Raspberry Pi пробила тропу для целого ряда одноплатных компьютеров. Таким образом, вслед за ней последовало множество подражателей, и есть большая группа людей, которые нуждаются в чем-то более мощном, более компактном или даже дешевом.

Рассмотрим некоторые лучшие альтернативы Raspberry Pi, которые можно купить в 2018 году, и выберем лучший одноплатный компьютер подходящий под наши задачи.

Модель Cubieboard (рисунок 24) имеет не только разъемы для питания, GPIO, SATA, HDMI, USB, VGA, Ethernet, но и стандартный «мини-джек» разъем аудиовыхода. Данные одноплатные компьютеры создаются в Китае. Они разработаны в 2012 году, однако первая модель была не очень успешной. Начиная с третьей версии они стали более востребованными. В отличие Raspberry, плата имеет несколько большее количество портов, установлены инфракрасный и «блютуз» порты, а также интерфейс для беспроводной связи. Плата работает с 1/2 Гб оперативной памяти (зависит от версии) на процессоре ARM Cortex-A7.

Одноплатные компьютеры этого разработчика разработаны специально для работы на Linux. Разработчик установил особенную версию, собранную самостоятельно. В целом устройство похожее на описанный ранее Raspberry P. В некоторых моментах Cubieboard повторяет характеристики многих современных одноплатных компьютеров, а иногда их превосходит. Цена чуть выше, чем для Raspberry: средняя стоимость $85.

Рисунок 24 – Модель одноплатного компьютера Cubieboard1

Одноплатный компьютер от BeagleBone Black появился в 2013 году (рисунок 25). Он имеет разъем питания, а также прочие стандартные порты. В своей линейке BeagleBone считают одним из мощнейших, если не самым мощным и лучшим. Разработчики отнеслись с высокой грамотностью к каждому разработанному порту. К компьютеру можно докупить множество интересных приспособлений (дополнений). Плата работает на процессоре Cortex-A8, частота функционирования ядер 1 ГГц. Но оперативная память составляет лишь 512 Мб.

Как и прочие вышеописанные модели, система у BeagleBone работает в основном на Linux. Производитель установил в качестве долговременной ёмкости 2-4 Гб eMMC памяти. В других модификациях установлен дистрибутив Debian. Плата может быть подключена абсолютно к любому устройству и наоборот за счёт наличия большого количества портов. Таким образом если использовать периферийные устройства, производительность данного одноплатного компьютера может быть увеличена в разы. Цена на данный компьютер $45.

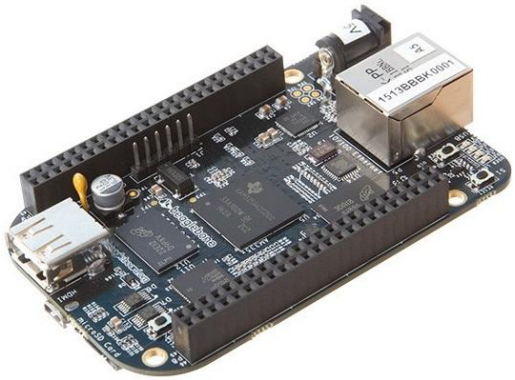

Рисунок 25 – Модель одноплатного компьютера BeagleBone Black

2. Схемотехническая часть

При разработке схемотехнической части нашего электронного устройства первоначально составим структурную схему, которая поможет понять из чего должна состоять электрическая часть системы. На рисунке 26 представлена такая структурная схема.

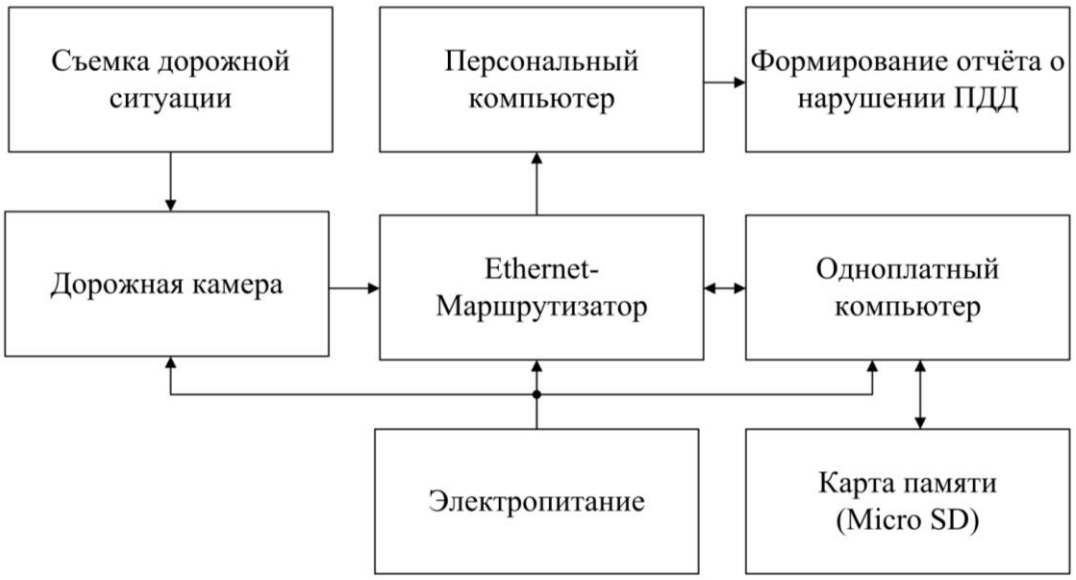

Рисунок 26 – Структурная схема разрабатываемого устройства

Из всевозможных вариантов, выбор для решения поставленных задач одноплатного компьютера Raspberry Pi 3 B+ – это самый оптимальный вариант не только из-за соотношения цена/качество, но и высокой распространенности и производительности модели, простоты установки операционной системы и драйверов, а также доступность дополнительных устройств под данную модель, например, модуля камеры. В данном компьютере имеются все необходимые порты и слоты расширения для решения поставленных задач диссертационной работы.

Raspberry Pi 3 B+ будет использоваться для работы с разработанным программным обеспечением. Одноплатный компьютер в процессе выполнения заложенной программы сможет совершать необходимые расчёты, производить съемку дорожной ситуации с помощью видеокамеры,

детектировать и классифицировать движущиеся объекты и передавать полученные видео и фотоизображения на персональный компьютер, где сотрудник Госавтоинспекции сможет оценивать и подписывать сформированные отчёты о правонарушении на пешеходном переходе.

Разумеется, обыкновенный модуль камеры, и разъем под него на одноплатном компьютере не подойдет, так как такая камера имеет очень слабый объектив не способный производить качественную съемку на большое расстояние с широким углом (рисунок 27).

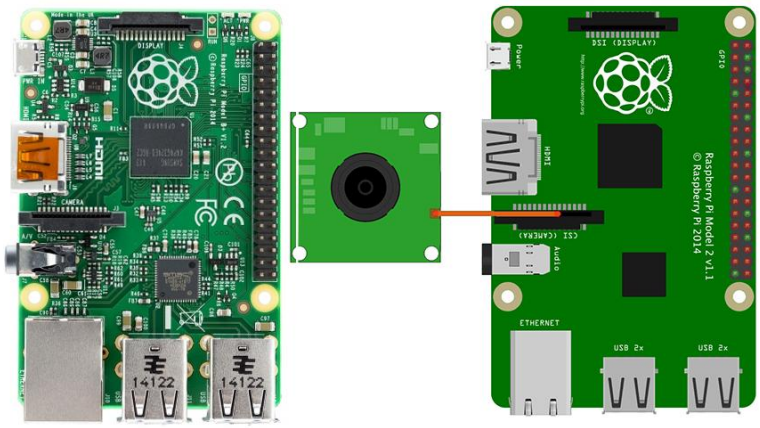

Рисунок 27 – Raspberry Pi вид сверху и схема подключения модуля камеры

Для съемки дорожной ситуации необходима качественная IP-видеокамера с хорошим широкоугольным объективом для захватывания всего пешеходного перехода. Произведем подбор необходимой камеры в одном ценовом сегменте.

Камера от Dahua IPC-HFW2531TP-ZS (рисунок 28) обладает не самыми яркими характеристиками, но, тем не менее, не слишком высокой ценой.

Рисунок 28 – Камера IPC-HFW2531TP-ZS от Dahua

Технические характеристики данной камеры:

- Матрица: 1/2.7” CMOS с прогрессивным сканированием;
- Чувствительность: 0.02 Лк/F1.6 (цвет, 1/3 с, 30IRE), 0.2 Лк/1.6 (цвет, 1/30 с, 30IRE), 0 Лк при ИК вкл;
- Электронный затвор Авто/ручной 1/3 с до 1/10000 с;
- Объектив: 2.7 – 13.5мм@F1.4, вариобъектив;
- Угол обзора объектива: 95° - 27°;
- Цена: 17741 руб.

Уличная IP-видеокамера RVi-IPC42M4 V.2 от RVI также является одним из интересных решений при выборе устройства для видеофиксации (рисунок 29).

Рисунок 29 – Камера RVi-IPC42M4 V.2 от RVI

Технические характеристики камеры от RVI:

- Матрица: Sony IMX290;
- Чувствительность: 0.01 лк F1.4 (Цвет), 0 лк F1.4 (ИК вкл.);
- Объектив: 2.7 – 12 мм@F1.4, вариобъектив;
- Угол обзора объектива: 99° - 34°;
- Цена: 22000 руб.

Обязательно необходима к рассмотрению камера DS-2CD4224F-IS от HikVision (рисунок 30). Данная компания является лидером по разработке комплексов видеонаблюдения, и данную камеру можно применить в составе разрабатываемого комплекса видеофиксации нарушения ПДД.

Рисунок 30 – Видеокамера DS-2CD4224F-IS от HikVision

Характеристики камеры представлены в нижеследующем списке:

- Матрица: 1/2.8’’ Progressive Scan CMOS;
- Чувствительность: 0.01лк@(F1.2,AGC вкл.), 0.014лк@(F1.4,AGC вкл.), 0лк с ИК;
- Скорость электронного затвора: 1с ~ 1/100000с, поддержка медленного затвора;
- Объектив: 2.8 - 12мм@F1.4, вариобъектив;
- Крепление объектива: Ф14;
- Угол обзора объектива: 113° - 33.8°;
- Цена: 25780 руб.

С помощью видеокамеры и встроенного Ethernet-интерфейса можно организовать потоковую передачу видео, например, с помощью программы mjpg-streamer, с определенным разрешением и числом кадров (к примеру, 640x480, 15 fps), используя обыкновенный маршрутиризатор для создания локальной сети между одноплатным компьютером и камерой. Также можно легко организовать передачу фотоизображений с нарушением передвижения автомобиля через пешеходный переход, с целью дальнейшего формирования отчёта о правонарушении на персональном компьютере.

Рассмотрим варианты выбора маршрутиризатора для связи IP-камеры и Raspberry Pi и дальнейшей передачи информации на ПК Госавтоинспекции, которые предлагают различные компании.

Маршрутизатор D-link DIR-120 представлен на рисунке 31. Отличается наличием 4-х LAN-портов и одним WAN-портом для выхода в интернет на скорости до 100 Мбит/с. Цена на него составляет около 1750 руб.

Рисунок 31 – Изображение маршрутизатора D-link DIR-120

Маршрутизатор TP-LINK TL-R480T+ показан на рисунке 32. Имеет те же порты, что и D-link роутер, представленный выше, но имеет возможность установки в стойку. Как пишут пользователи, отличается более высоким качеством работы, а его стоимость составляет 1800 руб. Но в данном варианте также отсутствует PoE (Power over Ethernet) питание, которое необходимо для питания IP видеокамеры.

Рисунок 32 – Изображение маршрутизатора TP-LINK TL-R480T+

Рассмотрим такой маршрутиризатор как MikroTik PowerBox RB750P-PBR2 на рисунке 33. Он имеет 1 WAN порт и 4 LAN порта с поддержкой технологии PoE, которая необходима для питания IP камеры. Цена на такое устройство 4020 руб, и данный вариант подходит больше всего к выбору.

Рисунок 33 – Изображение маршрутизатора MikroTik PowerBox RB750P-PBR2

Современная карта памяти microSDXC может применяться для промежуточного хранения полученных данных, а выбирать её объем можно до 256 Гб и с технологией высокой скорости записи до 20-25 Мб/с. Подобной картой памяти может выступать Samsung MB-MC256GA, которую и выберем для разрабатываемого комплекса.

Воспользовавшись знаниями по начертанию электронных схем соединения, отобразим на рисунке 34 схему электрических соединений разрабатываемого комплекса фиксации нарушения ПДД.[27]

В качестве конструкции корпуса для Raspberry Pi и маршрутиризатора, которая может находиться в нижней или срединной части столба с камерой, необходимо разработать или воспользоваться существующим корпусом (монтажным бокс) со степенью защищенности на уровне IP 67, чтобы обеспечить достаточную пыле- и влагозащищенность. К данному боксу будет подводиться питание переменным напряжением 220 В от ближайшей подстанции или здания, а также выделенная линия интернет.

Подобный корпус мог бы выглядеть как на рисунке 23, но также необходимо предусмотреть дополнительную защиту от влаги для портов, которые необходимо закрыть.

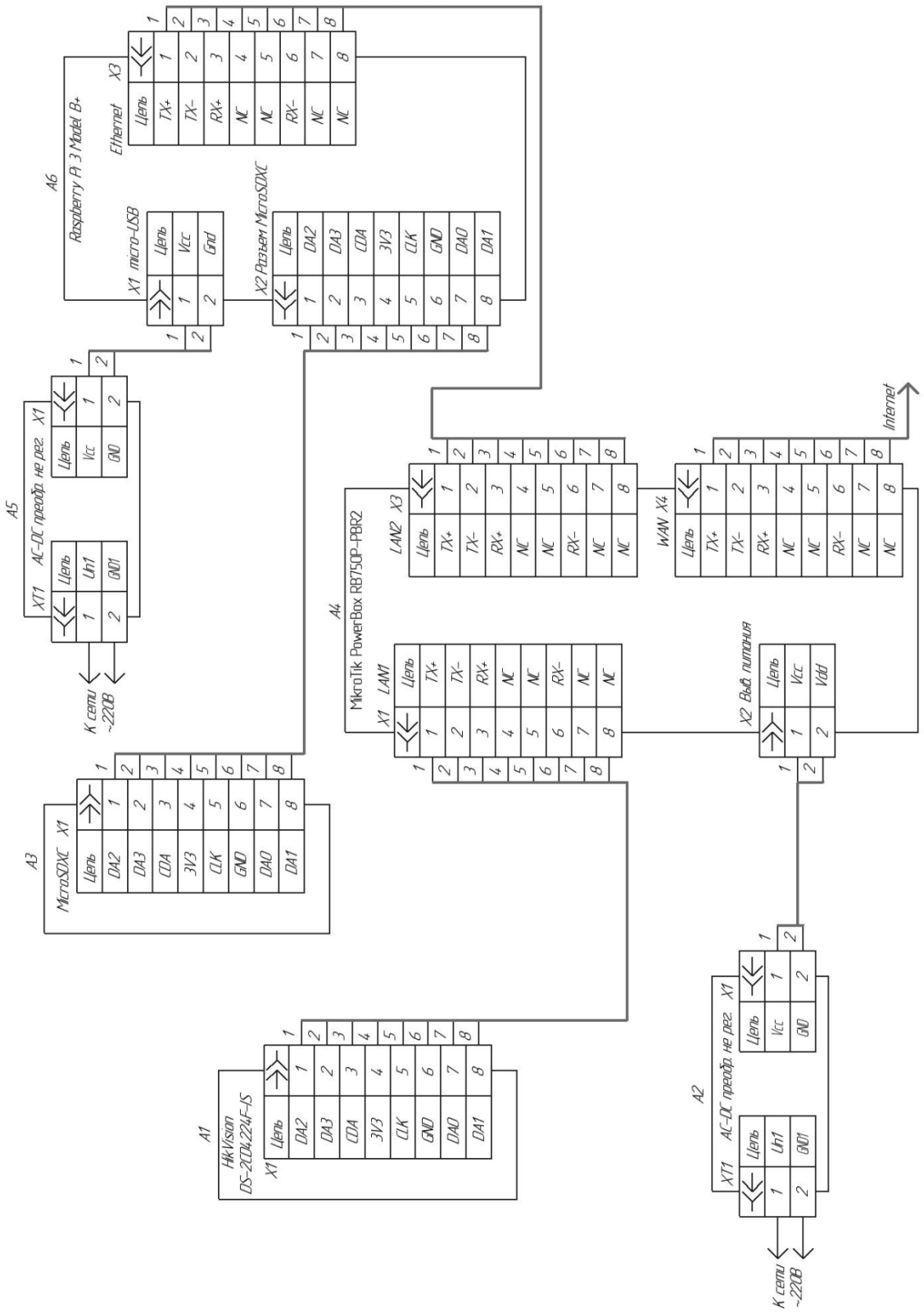

Рисунок 34 – Схема электрических соединений разрабатываемого устройства

3. Программная часть

3.1. Описание предлагаемой программы детектирования

Для реализации программного управления системы фиксации нарушения ПДД на пешеходном переходе необходимо создать программную часть, которая будет исполняться на подобранном устройстве в схемотехнической части диссертации.

Воспользовавшись знаниями о существующих языках программирования и библиотеках компьютерного зрения, выберем библиотеку OpenCV v.2.4.2 в качестве основополагающей для написания программы.

Интегрированная среда разработки программных продуктов Microsoft Visual Studio 2015 подойдет для написания консольного приложения на языке программирования C++. При написании программы необходимо будет подключить выбранную библиотеку компьютерного зрения и воспользоваться её функциями для детектирования и классификации движущихся объектов на видеоизображении.

На основе рассмотренных методов компьютерного зрения для детектирования объектов применим имплементацию HOG алгоритма и SVM для нахождения людей и автомобилей. Данный алгоритм является самым эффективным по качеству распознавания объектов (как людей, так и автомобилей и пр.) по сравнению с другими рассмотренными методами детектирования. С другой стороны происходит плата за это низкой скоростью распознавания и сложностью в обучении классификатора.

Гистограмма направленных градиентов (HOG-дескриптор) является алгоритмов трансформирования фото или видеокадров в многомерный вектор, так, что происходит сопоставление изображений с близкими по значению векторами. Такая трансформация является важной для успешного применения классификатора (SVM-классификатор, нейросеть). Для

классификации HOG-дескрипторов чаще всего используется машина опорных векторов (SVM). На практике в машинном зрении часто появляется задача детектирования на произвольном изображении необходимых нам объектов. В данном случае применяется метод скользящего окна с некоторым количеством масштабов, перемещающегося по изображению в поиске схожих значений векторов, в действующем окне и заложенных в базе классификатора, как на рисунке 35.

Суть метода состоит в том, что исходное изображение изменяется в масштабе в несколько раз в меньшую, а затем в большую сторону. Так получается несколько изображений разного масштаба (размера). Этот процесс позволяет искать объекты разного размера при одинаковом размере скользящего окна. Затем выполняется проход окна определенной размерности (под которое обучен классификатор). HOG-дескриптор – это очень ресурсозатратный метод, так как он вычисляется на каждом шаге скользящего окна, после чего дескриптор отправляется классификатору с целью определения наличия или отсутствия искомого объекта на изображении.

Для формальной записи разрабатываемой программы необходимо воспользоваться составлением БСА (блок-схемы алгоритма) её работы (рисунок 36, а-в). Запишем то, как по нашему мнению должна работать программа, и это поможет в будущем производить корректировки для её улучшения.

Рисунок 35 – Метод скользящего окна с масштабированием

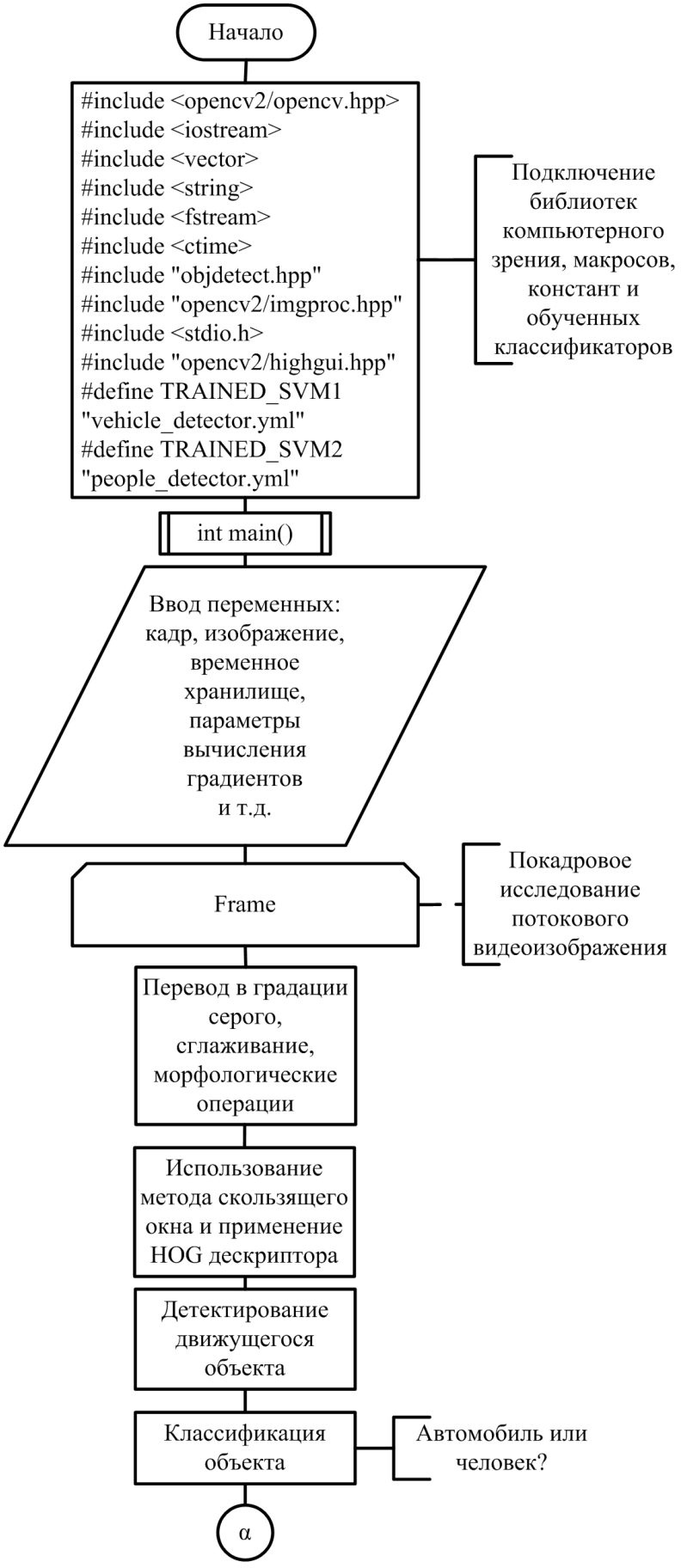

Рисунок 36а – БСА предлагаемой программы

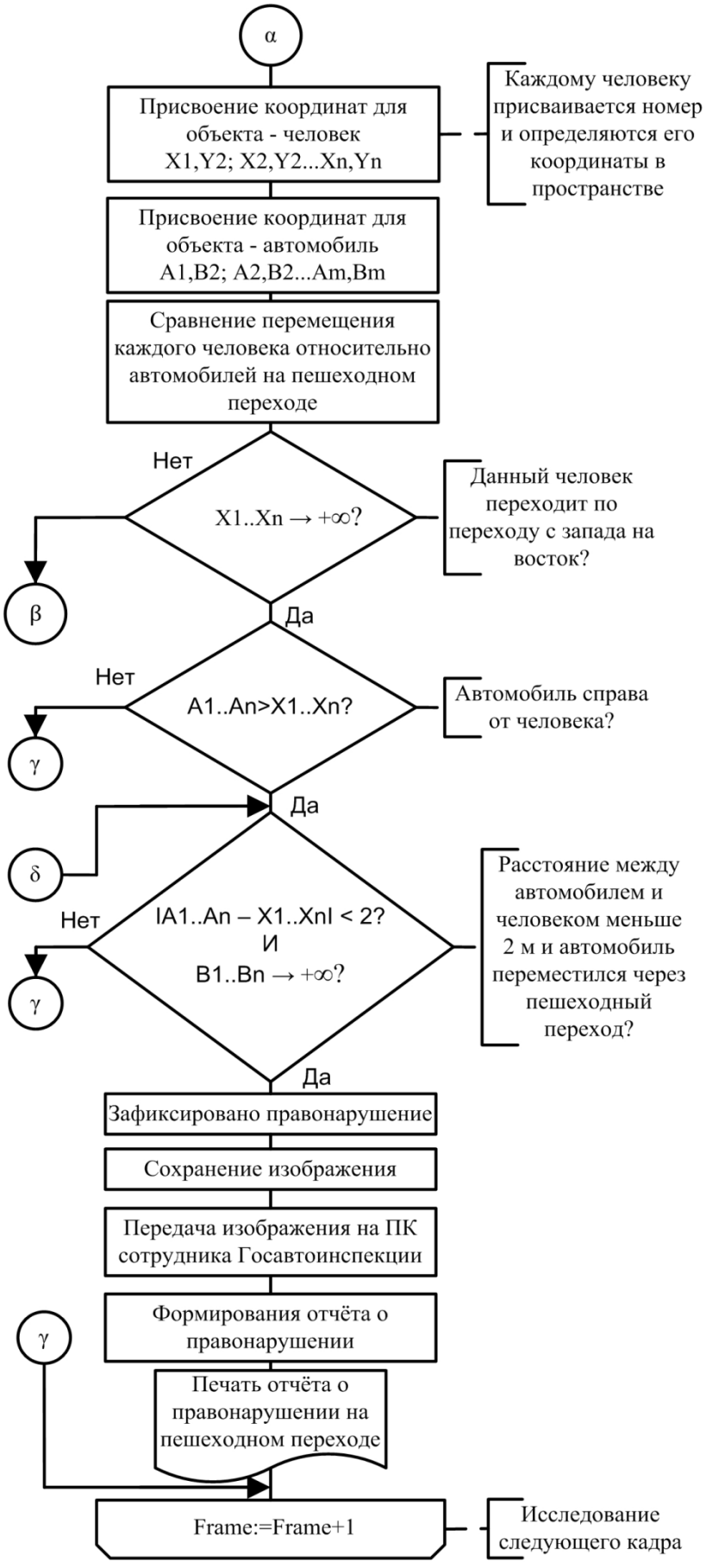

Рисунок 36б – БСА предлагаемой программы

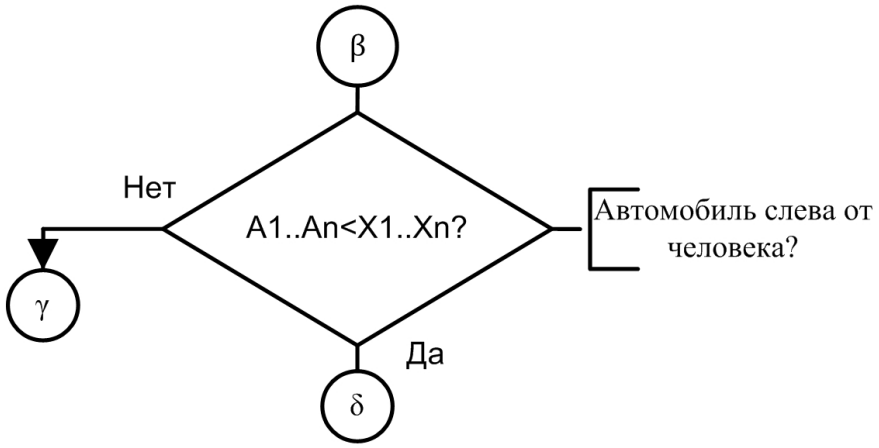

Рисунок 36в – БСА предлагаемой программы

Таким образом, программе необходимо вычислять до нескольких тысяч HOG-дескрипторов, что выливается в использовании более совершенных процессоров с высокой тактовой частотой.

БСА программы позволяет выделить ключевые функции необходимые для её реализации.

Первым делом подключаются библиотеки переменных, макросов, и библиотека компьютерного зрения OpenCV. Вводятся переменные для реализации детектирования, классификации и отслеживания, подключаются обученные классификатора формата «.yml».

Так как разрабатываемое устройство находится в режиме постоянной съемки над пешеходным переходом, необходимо захватывать видео с помощью функции «cvCreateCameraCapture()» и исследовать каждый кадр, для чего нужно использовать цикл и после каждого обработанного изображения производить переход к следующему кадру (Frame:=Frame+1).

К каждому кадру должно применяться преобразование изображения в градации серого, как писалось ранее – это необходимо для обнаружения с помощью HOG-дескриптора. Также могут применяться различные морфологические преобразования для выделения контуров объектов, а также сглаживание изображения.

Простейший способ использовать HOG-дескриптор – вызвать функцию «defaultHog.setSVMDetector(HOGDescriptor::getDefaultPeopleDetector())» в OpenCV, которая позволит использовать SVM-классификатор со стандартным набором данных для определения людей, который обучен на базе изображений людей MIT.

Каждый HOG-дескриптор (для людей и автомобилей) выделяет в каждом кадре наличие движущихся объектов и производит их классификацию. В процессе идентификации в кадре HOG-дескриптор, используя метод скользящего окна, выделяет найденный объект в фигуру прямоугольник.

Каждому объекту, будь то человек или автомобиль, должен присваиваться свой номер в кадре и производиться отслеживание их координат в пространстве (рисунок 37). Положение каждого человека, как самостоятельной единицы, должно сравниваться с каждым автомобилем в отдельности для обнаружения правонарушения.

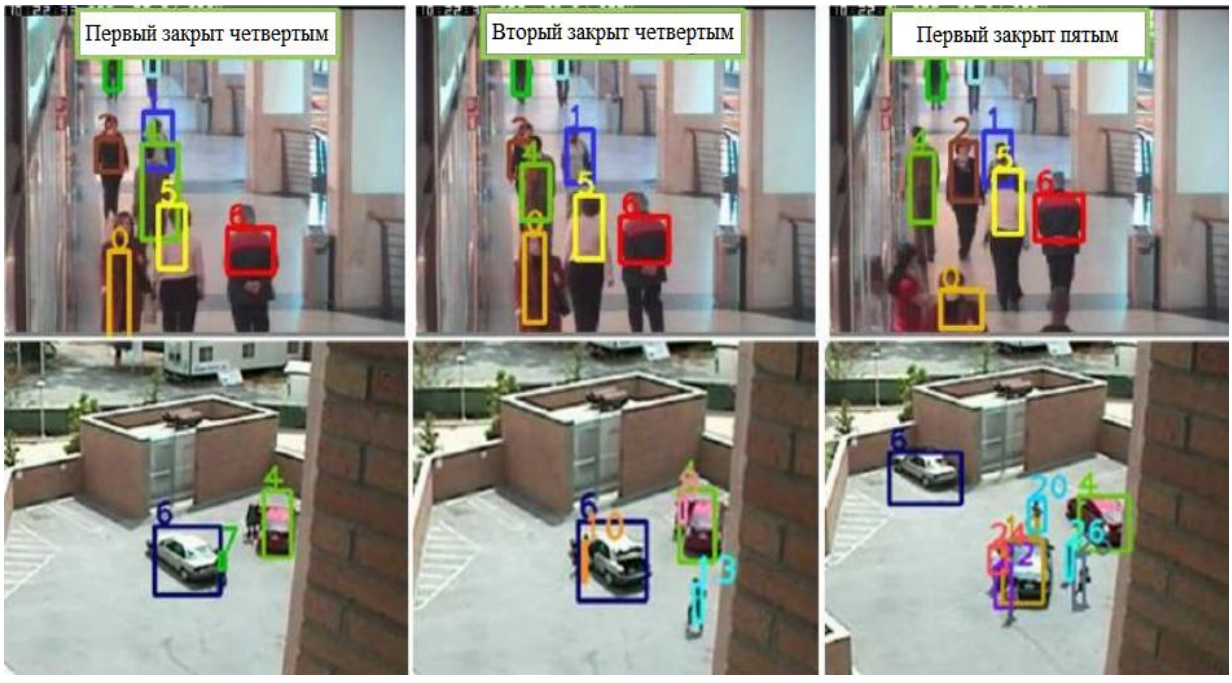

Рисунок 37 – Демонстрация работы системы трекинга

Здесь возникает одна из первых проблем появляющаяся при создании подобного рода программы. HOG-дескриптор не совершенен, и на данный момент не существует совершенных алгоритмов детектирования. Он склонен совершать ошибку в детектировании, например, возможно, что алгоритм не определит со 100% вероятностью искомый объект на всех кадрах, или же ошибется и примет за искомый объект предмет совершенно не относящийся к решаемой задаче.

Одним из возможных решений подобной проблемы может быть применение методов трекинга или отслеживания объекта, не основанные на алгоритмах детектирования, а применяемые после них. Другими словами, можно допустить, что, если HOG-дескриптор в определенной области на 3-5 (это число можно подобрать для каждой конкретной ситуации) кадрах подряд сможет идентифицировать объект, например, как человека, то далее к этому объекту необходимо применять метод трекинга. Далее данный движущийся объект стопроцентно считается человеком и к нему не применяется повторное детектирование, а лишь отслеживание его перемещения, т.е. координат. Аналогично с автомобилями.

Одним из простейших и очень перспективным методом трекинга во многих задачах, но не самым аккуратным и точным, является отслеживание на видеоизображении определенного цвета. Но здесь же возникает проблема отделения объектов друг от друга, искажения и пр. Движущиеся объекты, например толпа людей, могут перекрывать друг друга, находясь на разных планах кадра (переднем и заднем).

Другим популярным типом трекинговой системы является отслеживание объектов на основе фильтра Калмана, предсказывающего будущее состояние динамической системы по имеющейся информации о состоянии системы в прошлом и настоящем. Также здесь применяется метод вычитания фона.[28]

Детектирования и трекинг должны быть множественными, т.е. для всех объектов находящихся в кадре.

При удачной реализации подобных алгоритмов, можно переходить к взаимодействию людей и автомобилей в кадре.

Трекинговая система в первую очередь необходима для понимания того, с какой стороны от проезжей части пешеход начал переходить дорогу по пешеходному переходу. Программе не нужно случайным образом детектировать правонарушение у автомобилистов, которые, пропустив пешехода, проезжают за его спиной.

Следующий важный момент заключается в том, что по состоянию на 2018 год в п.14.1 ПДД содержится следующая формулировка: водитель транспортного средства, приближающегося к нерегулируемому пешеходному переходу, обязан уступить дорогу пешеходам, переходящим дорогу или вступившим на проезжую часть (трамвайные пути) для осуществления перехода. Если ранее 2014 года все автомобилисты были обязаны останавливаться перед пешеходным переходом как только пешеход ступил на него, то сейчас ситуация изменилась.[29]

Термин «уступить дорогу» означает – не создать помех, т.е. автомобиль не должен препятствовать пешеходу в пересечении дороги по пешеходному переходу. Одним из признаков того, что автомобиль воспрепятствовал – это остановка пешехода во время перехода, либо инстинктивное уменьшение скорости передвижения из-за проезжающего перед ним автомобиля.

В связи с тем, что нужно точно формулировать для программы, что именно будет являться правонарушением, предлагается фиксировать нарушение автомобилистом п.14.1 ПДД, если при пересечении пешеходом пешеходного перехода автомобиль проезжает перед ним на расстоянии от 2-х до 3-х метров (это число можно редактировать).

Скорость пересечения пешеходом пешеходного перехода также можно отслеживать, но для формализации для программы нарушения п.14.1 ПДД это будет являться более расплывчатым понятием, хоть и также имеющим право на реализацию в качестве теста.

При фиксации правонарушения на заключительном этапе происходит отправка изображения с нарушением вышеуказанного пункта правил на рабочий ПК сотрудника Госавтоинспекции с целью формирования отчёта. На рабочем компьютере автоматически может работать система распознавания автомобильных государственных регистрационных номеров. Далее сформированный отчёт о правонарушении печатается и отправляется нарушителю – владельцу транспортного средства.

3.2. Разработка программ детектирования людей и автомобилей

В результате выполнения научно-исследовательской работы были созданы две программы детектирования на основе алгоритма гистограммы направленных градиентов на базе библиотеки компьютерного зрения OpenCV.

Используя язык программирования C++ и программу Microsoft Visual studio 2015, напишем программы для детектирования объектов.[30]

Первая программа использует классификатор для детектирования людей, разработанный Далалом и Триггсом, на основе базы данных INRIA. Листинг программы представлен в приложении А.

Вторая программа использует обученный классификатор детектирования передней части различных автомобилей. Листинг второй программы представлен в приложении Б.

Программы имеют довольно схожие листинги, в основе которых лежит детектирование движущегося объекта в видеопотоке с применением HOG-дескриптора и классификатора SVM, но в программе из приложения Б

встроен код для обучения SVM классификатора. Обученный классификатор содержится в файле формата «.yml». Если при запуске программы файл отсутствует, она запрашивает датасет фотографий содержащих объект, и не содержащих его вовсе, с целью обучения нового классификатора.

Объединение представленных двух программ, дальнейшее добавление трекерной системы и условий детектирования правонарушения позволят продолжить разработку программной части устройства.

4. Исследовательская часть

Предлагается провести несколько экспериментов и тестов для разработанных программ по детектированию людей и автомобилей с целью определения: качества детектирования; необходимой производительности процессора разрабатываемого устройства; влияния на качество детектирования таких факторов как разрешение видео и количество кадров (свойства объектива), а также освещение.

4.1. Оценка качества детектирования разработанных программ

Протестируем программу для детектирования людей на видео из приложения А, для этого запустим программу на тестовом видеоизображении и представим результат детектирования различных последовательных кадров на рисунке 38.

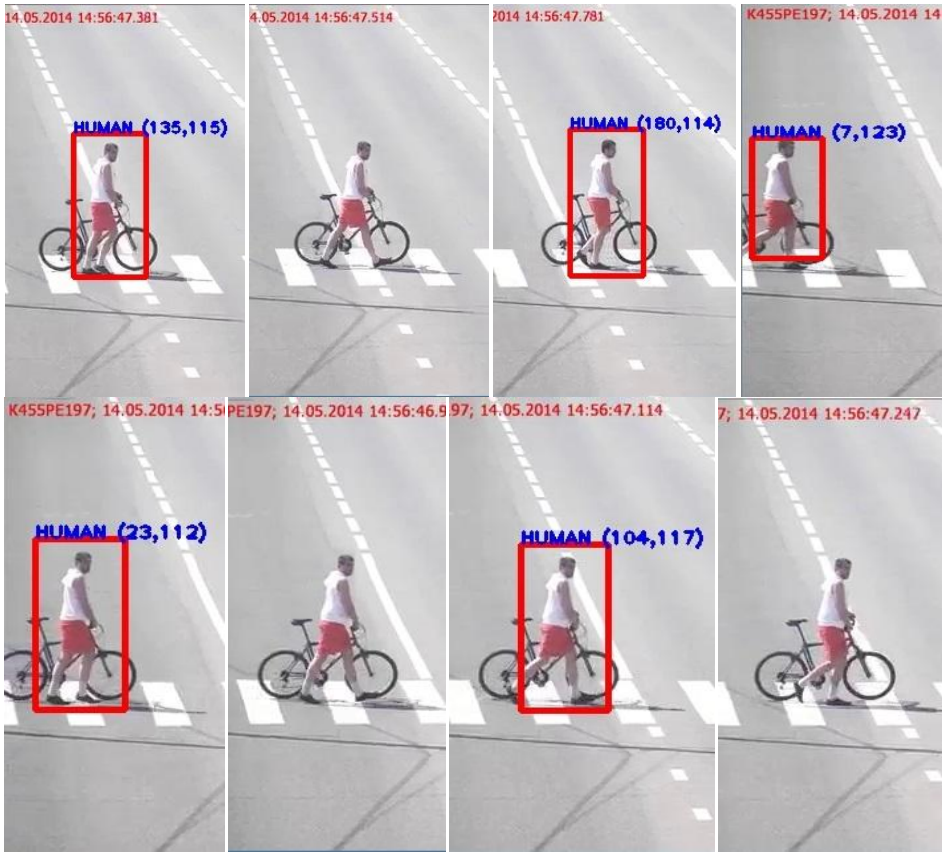

Рисунок 38 – Тестовый прогон программы детектирования людей

В результате тестирования выяснили, что программа детектирует человека и отображает его координаты, но не на всех кадрах. Программа не смогла определить человека на видео во всех позах. Далалом и Триггсом был обучен классификатор с другой фоновой составляющей. Также усложнение для программы привносит то, что человек ведет рядом с собой велосипед.

Вторым экспериментом протестируем программу для детектирования автомобилей на видео из приложения Б, для этого запустим программу на тестовом видеоизображении и представим результат детектирования различных последовательных кадров на рисунке 39.

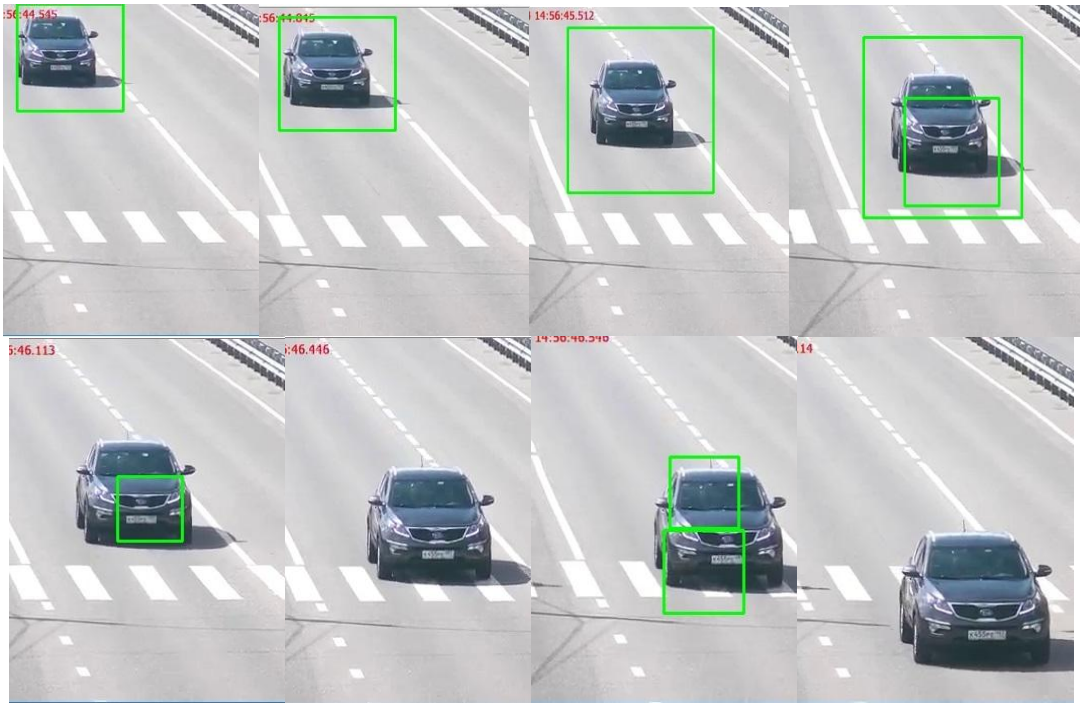

Рисунок 39 – Тестовой прогон программы детектирования автомобилей

В результате можем увидеть, что автомобиль детектируется очень хорошо, но лишь до момента его подъезда к пешеходному переходу, что также свидетельствует, что SVM классификатор необходимо переобучать в условиях дорожной разметки.

Благодаря проведенным экспериментам убедились в необходимости переобучения SVM классификатора в обоих типах приложений детектирования.

Таким образом, можно сделать вывод о том, что программную часть разрабатываемого устройства необходимо каждый раз тестировать и дорабатывать для фактического места его установки при съемке автодороги над пешеходным переходом.

Также необходимо понимать, что без многомасштабной (multiscale) трекерной системы в любом случае не обойтись, и необходим алгоритм, позволяющий «вести» каждый объект в кадре индивидуально.

4.2. Исследование необходимой производительности процессора

Для следующего эксперимента протестируем скорость работы обеих программ на тестовом видео, кадр из видео представлен на рисунке 40.

Рисунок 40 – Кадр из тестового видео

Характеристики видео представлены в таблице 2.

Таблица 2 – Характеристики тестового видео

Формат видео	Mp4
Кодек	AVC1
Разрешение, пк	640x480
Длительность, с	6,1
Частота кадров, к/с	30
Всего кадров, шт	183

Тестирование производилось на ПК с процессором Intel Core i5-2320 с тактовой частотой 3 ГГц, 4 ядра.

Программа по детектированию людей справилась за 2,8 мин. Это означает, что среднем ей понадобилось 0,918 с на обработку одного кадра. Для примера радарный комплекс «Стрелка-СТ» работает с видеоданными частотой 12 к/с.

Детектор автомобилей справился за 1,33 мин, что говорит о том, что в среднем затрачивается 0,437 с на обработку одного кадра. Данные отличаются от тестовых данных для детектора людей, так как программы работают с разными по размеру скользящими окнами, для которых вычисляется HOG-дескриптор.

Четырехъядерный процессор одноплатного компьютера Raspberry pi имеет частоту одного ядра 1,4 ГГц, что говорит о том, что данные будут обрабатываться еще медленнее приблизительно в 2,14 раз.

Рассмотрим несколько возможных решений данной проблемы.

Во-первых, это программное снижение разрешения видео до 320x240 пк, что не является проблемой для HOG-дескриптора, и он сможет производить детектирование и при таком разрешении.

Во-вторых, существует вариант программного снижение числа кадров, либо можно воспользоваться автоматическим пропуском в процессе слежения за дорожной ситуацией, так как частота в 30 к/с слишком высока.

В-третьих, имеется аппаратной решение вопроса, связанное с выбором более мощного одноплатного компьютера, либо разработка собственной электронной платы на базе процессорного модуля.

Также возможно решение с применением MicroSDHC карты, в которой потоковое видео будет порционно сохраняться и обрабатываться с последующей очисткой долговременной памяти устройства. Объема в 256 Гб хватит для того, чтобы не происходило переполнения внутренней памяти системы.

4.3.Исследование влияния освещения на качество детектирования

Для исследования влияния освещения при детектировании, произведем поиск автомобилей на тестовом изображении на рисунке 41. Далее будем программно снижать освещенность на 5% за тест, и смотреть на работу дескриптора. Результаты теста сведем в таблицу 3.

На первом кадре представленного изображения – оригинальная фотография с автомобилями, примем освещение в этом кадре, как при нормальных условиях. В качестве визуализации тестирования продемонстрируем на втором кадре снижение освещения на 35%, а на третьем кадре – на 65%.

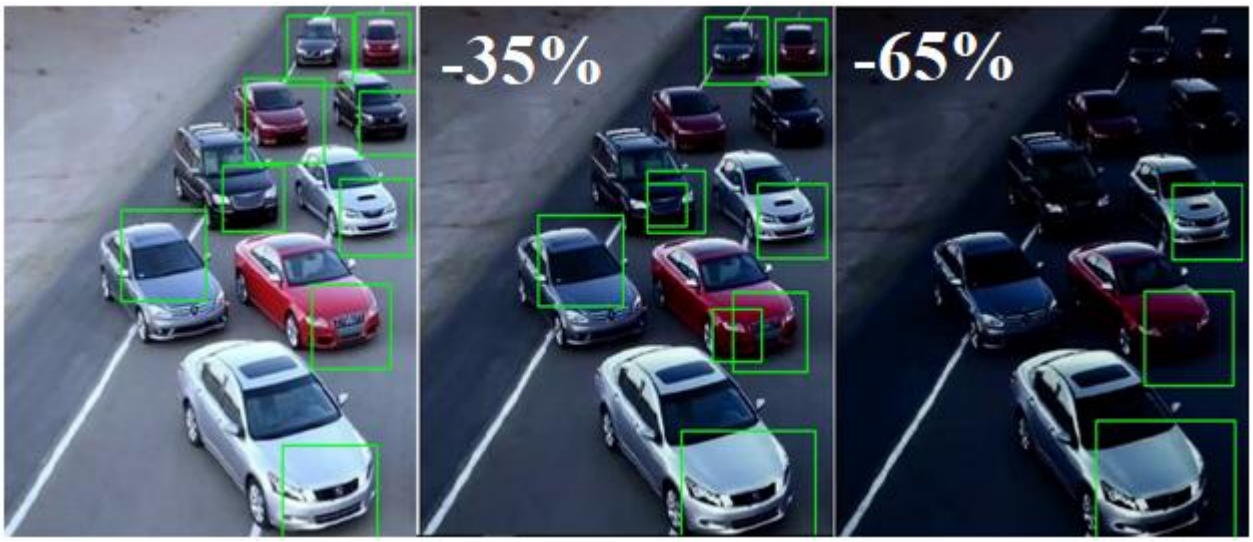

Рисунок 41 – Тестируемые кадры влияния освещения на детектирование

Как видно из тестируемых изображений, снижение освещения на 35% от оригинального заставляет программу ошибаться и пропускать некоторые автомобили, а в некоторых случаях она «становится не уверена» в принадлежности объекта к исследуемому классу. При доработке программы можно избавиться от «двойных прямоугольников детектирования» сложив их вместе, но таким образом будет теряться более точное определение координаты местоположения объекта в кадре.

Таблица 3 – Тестирование влияния освещения на качество детектирования при поиске автомобилей на изображении

Степень снижения освещения, %	Количество детектированных автомобилей, шт	Количество детектированных автомобилей, %
Оригинальное изображение	9	100
5	9	100
10	9	100
15	9	100
20	8	88,9
25	7	77,8
30	9	100
35	5	55,6
40	7	77,8
45	5	55,6
50	6	66,7
55	5	55,6
60	3	33,3
65	2	22,2
70	1	11,1
75	1	11,1
80	1	11,1
85	1	11,1
90	1	11,1
95	0	0
100	0	0

Снижение освещения на 60% от оригинального способствует снижению качества детектирования настолько, что программа смогла классифицировать лишь три автомобиля из 9, представленные на тестовом изображении.

Представим на рисунке 42 график зависимости N(E) количества детектированных объектов от степени снижения освещенности.

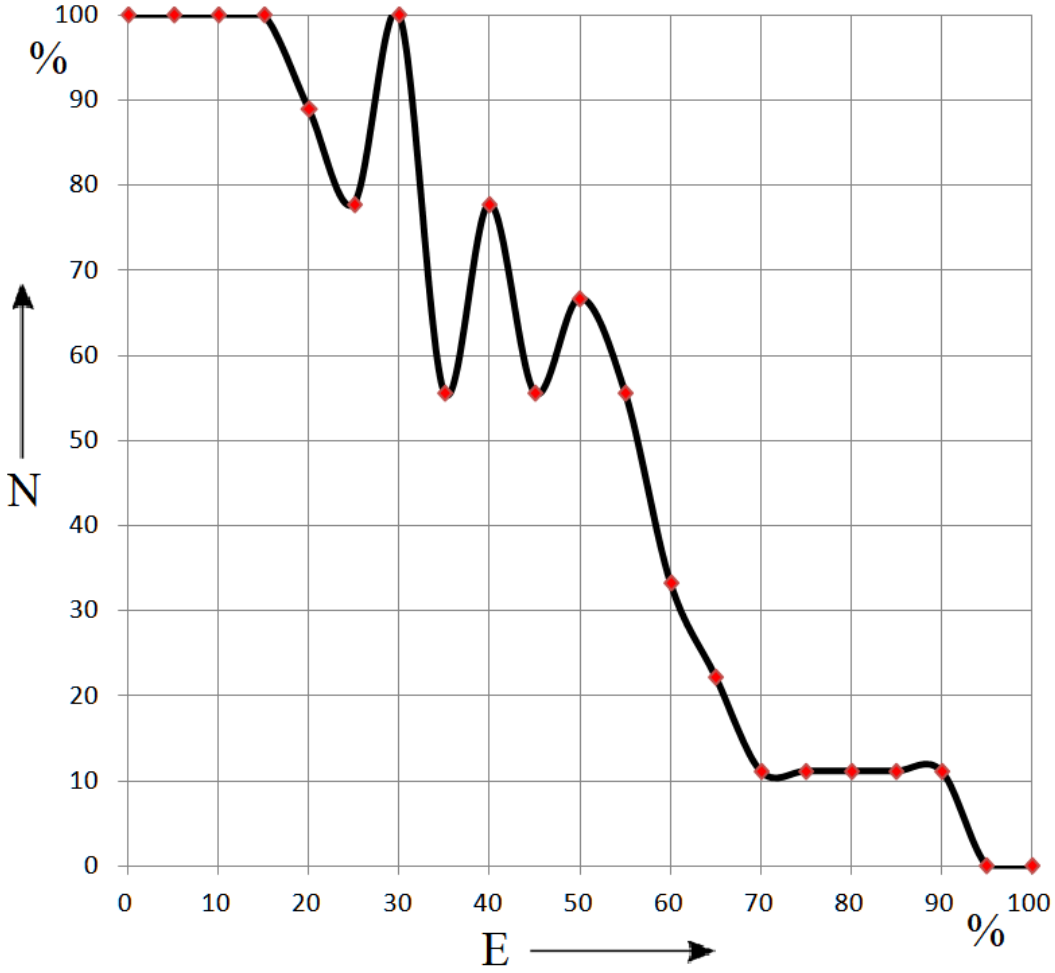

Рисунок 42 – График зависимости качества детектирования от освещения

Стоит сказать, что отмечается интересный эффект при снижении освещения на 80% от оригинала. Данный эффект представлен на рисунке 43.

HOG-дескриптор пытаясь отыскать автомобиль детектирует единственные светлые участки на автомобилях – их фары, при этом игнорируя автомобиль на переднем плане.

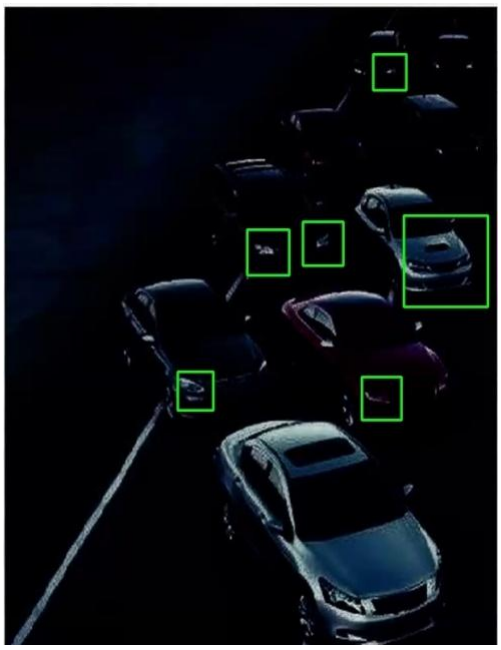

Рисунок 43 – Эффект детектирования при сверхнизкой освещенности

Заключение

В результате выполнения магистерской диссертационной работы выполнены основные задачи, поставленные индивидуальным планом научно-исследовательской работы. Последовательное выполнение поставленных задач привело к достижению основной цели диссертационной работы: реализации программно-аппаратного комплекса устройства для детектирования нарушений на пешеходном переходе.

Основываясь на существующем опыте реализации устройств видеофиксации нарушения ПДД по соблюдению скоростного режима, проезду на запрещающий сигнал светофора, пересечению транспортными средствами запрещающих полос, а также учитывая ошибки детектирования, возникающие у современных комплексов, предложена модель устройства, позволяющего при должной доработке стать бюджетной альтернативной существующим системам.

Применение технологий компьютерного зрения в современном мире позволяет не только упростить жизнь человека, исключив достаточно рутинные процессы, например при производстве какой-либо продукции, но и обезопасить её, будь то при обыкновенной прогулке по улице или нахождении на станциях метро, или же переходе через дорогу.

Существующие методы и алгоритмы детектирования, классификации и отслеживания перемещения объектов еще далеки от совершенства. Метод гистограммы направленных градиентов с применением классификатора машины опорных векторов на сегодняшний день один из качественнейших алгоритмов для детектирования и классификации объектов в видеопотоке. Таким образом, выбор данного алгоритма для реализации в представленной работе оправдывает своё применение в связи с высоким качеством детектирования и низким числом ложноположительных срабатываний.

Также, выбранная библиотека компьютерного зрения с открытым исходным кодом OpenCV позволяет полностью реализовать все поставленные задачи в программной части разрабатываемого устройства.

С помощью построения структурной схемы создаваемого комплекса удалось произвести подбор необходимых электронных составляющих, которые отличаются небольшой стоимостью и компактностью, что позволило разработать схемотехническую часть системы.

В третьей главе описывается программная часть устройства и предлагается блок-схема алгоритма программы. В данной главе описываются некоторые мысли и идеи, которые помогут при дальнейшей доработке программы. На текущий момент разработан листинг программы по детектированию людей в видеопотоке, а также программа по детектированию автомобилей, в которой встроена функция по обучению классификатора машины опорных векторов. Для её применения необходимо предоставить для программы набор большого числа изображений, где есть необходимый для детектирования объект, и число изображений, где данного объекта нет вовсе.

В завершающей главе проведены исследования работы разработанных программ по детектированию движущихся объектов в видеопотоке. Исследовано качество работы по поиску и классификации объектов и выявлено, что классификатор необходимо обучать именно для того места, где будет использоваться разрабатываемый комплекс, так как необходимо учитывать все рельефные и геометрические особенности данного конкретного места. Проведено исследование скорости работы программ и предложены операции по улучшению скорости их работы, такие как снижение разрешения видео и уменьшение числа кадров используемых при детектировании.

Последнее исследование работы программ включает в себя сравнение количества детектированных автомобилей при различной освещенности.

Было выявлено, что при снижении освещенности области детектирования на 50% от основного резко снижает качество детектирования искомых объектов. Это говорит о том, что использование подобных комплексов в вечернее время может привести к резкому возрастанию ложноположительных срабатываний, что вызовет неправильную работу систему по детектированию нарушений ПДД на пешеходном переходе, и, соответственно, к возрастанию числу ошибочно отправленных штрафов владельцам транспортных средств попавших в поле зрения камеры.

Полученные результаты работы программ и исследований говорят о том, что подобные комплексы, детектирующие нарушения ПДД водителями транспортных средств, можно создавать с намного более дешевыми компонентами, а технологии компьютерного зрения требуют дальнейших разработок и применений в повседневности для улучшения качества жизни людей.

Список используемой литературы

1.E.R. Davies. Machine Vision : Theory, Algorithms, Practicalities. — Morgan Kaufmann, 2004.

2.Demant C., Streicher-Abel B. and Waszkewitz P. Industrial Image Processing: Visual Quality Control in Manufacturing. — Springer-Verlag, 1999. — ISBN 3-540-66410-6.

3.Batchelor B.G. and Whelan P.F. Intelligent Vision Systems for Industry. — Springer-Verlag, 1997

4.Gonzales R. C. and Wintz P. A. Digital Image Processing. — Longman Higher Education, 2001. — ISBN 978-0201110265.

5.Berthold K.P. Horn. Robot Vision. — MIT Press, 1986. — ISBN 0-262-08159-8 (Б.К.П. Хорн, Зрение роботов: перевод с англ. — М.: Мир, 1989).

6.И.В. Степанов, Ю.А. Грачев, Управление деятельностью по обеспечению безопасности дорожного движения. Состояние, проблемы, пути совершенствования, Вестник Санкт-Петербургского университета МВД России, 2015. – 4 с.

7.М.В. Яшина, А.А. Толмачев, Методы распознавания образов для оценки характеристик пешеходных потоков, T-Comm - Телекоммуникации и Транспорт, 2017. – 7 с.

8.Л. Шапиро, Дж. Стокман. Компьютерное зрение = Computer Vision. — М.: Бином. Лаборатория знаний, 2006. — 752 с. — ISBN 5-94774-384-1.

9. Д. Форсайт, Ж. Понс. Компьютерное зрение. Современный подход = Computer Vision: A Modern Approach. — М.: «Вильямс», 2004. — 928 с. — ISBN 5-8459-0542-7.

10.А.В. Кузнецов, Привлечение к административной ответственности за нарушения ПДД при использовании средств автоматической фиксации, Психопедагогика в правоохранительных органах, 2013. – 6 с.

11.А.И. Смоляков, Использование результатов частной видеофиксации как неотъемлемая составляющая при реализации принципа неотвратимости наказания за нарушения Правил дорожного движения, Юридическая наука и правоохранительная практика, 2013. – 8 с.

12.Schuster P. Moving the Stars. Christian Doppler, His Life, His Works and Principle and the World After. — Living Edition Publishers, 2005. — 232 с.

13.А. Афанасьев, Стрелка-СТ — космическое оружие ГИБДД [Электронный ресурс]. //27.10.2010// URL: https://www.drive2.ru/b/288230376151783221/. (Дата обращения: 01.05.2017).

14.АПК «Автоураган» [Электронный ресурс]. //29.03.2017// URL: https://www.drive2.ru/b/469561560000364865/. (Дата обращения: 28.05.2017).

15.Все о камерах ГИБДД [Электронный ресурс]. //17.05.2018// URL: http://albz.ru/kamery-gibdd. (Дата обращения: 21.05.2018).

16.А. Нечаев, Как ошибаются камеры видеофиксации [Электронный ресурс]. // 08.09.2016 // URL: http://www.avto25.ru/news/articles/2016/09/08/32226.html (Дата обращения: 10.02.2018).

17.Ту Дж., Гонсалес Р. Принципы распознавания образов, М. 1978

18.Журавлев Ю.И. Об алгебраическом подходе к решению задач распознавания и классификации // Проблемы кибернетики. – М.: Наука, 1978, вып. 33. – С. 5-68.

19."Method and apparatus for identifying scale invariant features in an image and use of same for locating an object in an image", David Lowe's patent for the SIFT algorithm, March 23, 2004

20.Владимир Вьюгин. Математические основы теории машинного обучения и прогнозирования. — МЦМНО, 2013. — 390 с.

21.N. Dalal, B. Triggs, Histograms of Oriented Gradients for Human Detection/INRIA Rhàone-Alps,655 avenue de l'Europe,Montbonnot 38334,France/ 2005.– 8 с.

22.Viola and Jones, «Rapid object detection using a boosted cascade of simple features», Computer Vision and Pattern Recognition, Cambridge, 2001, – 8 с.
23.Кэлер А., Брэдски Г. Изучаем OpenCV 3 = Learning OpenCV 3. — М.: ДМК-Пресс, 2017. — 826 с.
24.Computer Vision System Toolbox [Электронный ресурс]. // MathWorks // URL: https://matlab.ru/products/computer-vision-system-toolbox/computer-vision-system-toolbox_rus.pdf (Дата обращения: 12.10.2017).
25.COM - Based SBCs : The Superior Architecture for Small Form Factor Embedded Systems". Diamond Systems Corp. Retrieved 27 December 2016.
26."Foundation Strategy 2016 - 2018". Raspberry Pi. Raspberry Pi Foundation. pp. 3–5. Retrieved 26 November 2016.
27.Хоровиц П., Хилл У. Искусство схемотехники: В 2-х томах = The Art of Electronics: Second Edition (© Cambridge University Press, 1980) / Пер. с англ. под ред. М. В. Гальперина, редакторы: Н. В. Серегина, Ю. Л. Евдокимова. — М.: Мир, 1983. — т. 1: 568 с., т. 2: 590 с.
28.О.П. Соловей, Н.Н. Иванов, Алгоритм трекинга объектов в реальном времени с обработкой ошибок, доклады Белорусского государственного университета информатики и радиоэлектроники, 2013. – 5 с.
29.Правила дорожного движения по состоянию на 2018 год / Законы и кодексы / Эксмо / 2018. – 96 с.
30.Бьёрн Страуструп. Язык программирования C++ = The C++ Programming Language / Пер. с англ. — 3-е изд. — СПб.; М.: Невский диалект — Бином, 1999. — 991 с. — 3000 экз. — ISBN 5-7940-0031-7.

Приложение А

```
#include <iostream>
#include "objdetect.hpp" //подключение вспомогательных библиотек
#include "opencv2/highgui.hpp"
#include "opencv2/pictproc.hpp"
#include <stdio.h>
using namespace cv;
using namespace std;
//int to string вспомогательная функция
string intToString(int number) {
std::stringstream ds;
ds << number;
return ds.str();
}
int main() {
//переключатель для включения системы детектирования
bool detectEnabled = false;
Mat kadr;
//захват видеокадра
VideoCapture capture;
IplImage* image = 0;
IplImage* templ = 0;
IplImage* dst = 0;
Mat grad, angleOfs;
while (1) {
//зацикливание видео с последнего кадра на первый
capture.open("2.mp4");
unsigned int i, j;
HOGDescriptor HOGdefault;
HOGdefault.setSVMDetector(HOGDescriptor::getDefaultPeopleDetector());
while (capture.get(CV_CAP_PROP_POS_KADRS)<capture.get(CV_CAP_PROP_KADR_COUNT) - 1) {
//проверка достижения последнего кадра видео, иначе получил ошибку памяти
vector<Rect> Detect, Detect_filter;
capture.read(kadr);
//читаем первый кажр видео
int x = 0;
int y = 0;
// задаём весовые коэффициенты
//если детектирование включено, поиск людей в кадре
if (detectEnabled) {
HOGdefault.detectMultiScale(kadr, Detect);
for (i = 0; i < Detect.size(); i++)
{
Rect r = Detect[i];
//следующий оператор for вычисляет все прямоугольники от детектированных объектов
//и помещает их в Detect_filter
for (j = 0; j <Detect.size(); j++)
if (j != i && (r&Detect[j]) == r)
break;
if (j == Detect.size())
Detect_filter.push_back(r);
} //произведем корректировки размера прямоугольника, так как он несколько больше чем
искомый объект
for (i = 0; i <Detect_filter.size(); i++)
{
Rect r = Detect_filter[i];
r.x += cvRound(r.width*0.09);
r.width = cvRound(r.width*0.78);
r.y += cvRound(r.height*0.067);
r.height = cvRound(r.height*0.78);
rectangle(kadr, r.tl(), r.br(), cv::Scalar(255, 0, 255), 3);
```

```
putText(kadr, "HUMAN (" + intToString(r.x) + "," + intToString(r.y) + ")", Point(r.x,
r.y), 1, 1, Scalar(255, 255, 255), 2);
}
}
//демонстрация нашего видео с прямоугольниками детектирования
imshow("Kadr", kadr);
kadr = kadr + 111;
switch (waitKey(1)) {
case 27: //нажмите esc, чтобы выйти из программы
return 0;
case 116: //нажмите 't', чтобы производить детектирование
detectEnabled = !detectEnabled;
if (detectEnabled == false) cout << "Detecting disabled." << endl;
else cout << "Detecting enabled." << endl;
break;
}
}
//отпускаем захват видео перед следующим захватом (циклом повторения видео).
capture.release();
}
return 0;
}
```

Приложение Б

```
#define POSITIVE_TRAINING_SET_PATH "DTST\\POS\\"
#define NEGATIVE_TRAINING_SET_PATH "DTST\\NEG\\"
#define WINDOW_NAME "WIN"
#include <opencv2/opencv.hpp>
#include <vector>
#include <iostream>
#include <ctime>
// если используется система на базе Linux
#ifndef __linux
#include <io.h>
#define access _access_s
#else
#include <unistd.h>
#include <memory>
#endif
#include <string>
#include <fstream>
#define TRAFFIC_VIDEO_FILE "2.avi"
#define TRAINED_SVM "car_detect.yml"
#defineIMAGE_SIZE Size(40, 40)
using namespace cv;
using namespace std;
using namespace cv::ml;
bool pack_lv(const string &pack);
void image_input(string directory, vector<Mat>& image_list);
vector<string> obj_in_directory(string directory);
void supMach_detect(const Ptr<SVM>& supMach, vector< float > & HystGr_detector);
void conv_ml(const std::vector< cv::Mat > & train_samples, cv::Mat& trainData);
void sample_negat(const vector< Mat > & fullnegls, vector< Mat > & negls, const Size &
size);
Mat hogdescvis(const Mat& color_origImg, vector<float>& descriptorValues, const Size &
size);
void compute_HystGr(const vector< Mat > & pict_lst, vector< Mat > & gradlt, const Size &
size);
void train_supMach(const vector< Mat > & gradlt, const vector< int > & labl);
void locat_drawing(Mat & pict, const vector< Rect > & locs, const Scalar & color);
void testing(const Size & size);
int main(int argc, char** argv)
{
if (!pack_lv(TRAINED_SVM)) {
vector< Mat > poslist;
vector< Mat > fullnegls;
vector< Mat > negls;
vector< Mat > gradlt;
vector< int > labl;
image_input(POSITIVE_TRAINING_SET_PATH, poslist);
labl.assign(poslist.size(), +1);
image_input(NEGATIVE_TRAINING_SET_PATH, fullnegls);
labl.insert(labl.end(), fullnegls.size(), -1);
compute_HystGr(poslist, gradlt, IMAGE_SIZE);
compute_HystGr(fullnegls, gradlt, IMAGE_SIZE);
train_supMach(gradlt, labl);
}

testing(IMAGE_SIZE);
return 0;
}
bool pack_lv(const string &pack)
{
return access(pack.c_str(), 0) == 0;
```

```
}
vector<string> obj_in_directory(string directory)
{
vector<string> obj;
char buf[256];
string comm;
#ifdef __linux__
comm = "ls " + directory;
shared_ptr<FILE> pipe(popen(comm.c_str(), "r"), pclose);

char cw[256];
getcw(cw, sizeof(cw));

while (!feof(pipe.get()))
if (fgets(buf, 256, pipe.get()) != NULL) {
string pack(cw);
pack.append("/");
pack.append(buf);
pack.pop_back();
obj.push_back(pack);
}
#else
comm = "dir /b /s " + directory;
FILE* pipe = NULL;
if (pipe = _popen(comm.c_str(), "rt"))
while (!feof(pipe))
if (fgets(buf, 256, pipe) != NULL) {
string pack(buf);
pack.pop_back();
obj.push_back(pack);
}
_pclose(pipe);
#endif
return obj;
}
void image_input(string directory, vector<Mat>& image_list) {
Mat pict;
vector<string> obj;
obj = obj_in_directory(directory);

for (int i = 0; i < obj.size(); ++i) {

pict = imread(obj.at(i));
if (pict.empty())
continue;
#ifdef _DEBUG
imshow("image", pict);
waitKey(10);
#endif
resize(pict, pict, IMAGE_SIZE);
image_list.push_back(pict.clone());
}
}
void supMach_detect(const Ptr<SVM>& supMach, vector< float > & HystGr_detector)
{
// получить вектора поддержки
Mat supp = supMach->getSupportVectors();
const int supp_total = supp.sers;
// получить функцию решения
Mat alpha, svidx;
double rho = supMach->getDecisionFunction(0, alpha, svidx);
CV_Assert(alpha.total() == 1 && svidx.total() == 1 && supp_total == 1);
CV_Assert((alpha.type() == CV_64F && alpha.at<double>(0) == 1.) ||
```

```
(alpha.type() == CV_32F && alpha.at<float>(0) == 1.f));
CV_Assert(supp.type() == CV_32F);
HystGr_detector.clear();
HystGr_detector.resize(supp.columns + 1);
memcpy(&HystGr_detector[0], supp.ptr(), supp.columns*sizeof(HystGr_detector[0]));
HystGr_detector[supp.columns] = (float)-rho;
}
/*
* Преобразование набора обучения / тестирования, который будет использоваться алгоритмами
OpenCV Machine Learning.
* TrainData - это матрица размера выборки
* Транспонирование образцов производится при необходимости.
*/
void conv_ml(const std::vector< cv::Mat > & train_samples, cv::Mat& trainData)
{
//--преобразование данных
const int sers = (int)train_samples.size();
const int columns = (int)std::max(train_samples[0].columns, train_samples[0].sers);
cv::Mat tmp(1, columns, CV_32FC1); //< использование транспонирования если необходимо
trainData = cv::Mat(sers, columns, CV_32FC1);
vector< Mat >::const_iterator itr = train_samples.begin();
vector< Mat >::const_iterator end = train_samples.end();
for (int i = 0; itr != end; ++itr, ++i)
{
CV_Assert(itr->columns == 1 ||
itr->sers == 1);
if (itr->columns == 1)
{
transpose(*(itr), tmp);
tmp.copyTo(trainData.row(i));
}
else if (itr->sers == 1)
{
itr->copyTo(trainData.row(i));
}
}
}
void sample_negat(const vector< Mat > & fullnegls, vector< Mat > & negls, const Size &
size)
{
Rect cup;
cup.width = size.width;
cup.height = size.height;
const int size_x = cup.width;
const int size_y = cup.height;
srand((unsigned int)time(NULL));
vector< Mat >::const_iterator pict = fullnegls.begin();
vector< Mat >::const_iterator end = fullnegls.end();
for (; pict != end; ++pict)
{
cup.x = rand() % (pict->columns - size_x);
cup.y = rand() % (pict->sers - size_y);
Mat roi = (*pict)(cup);
negls.push_back(roi.clone());
#ifdef _DEBUG
imshow("pict", roi.clone());
waitKey(10);
#endif
}
}
Mat hogdescvis(const Mat& color_origImg, vector<float>& descriptorValues, const Size &
size)
{
```

```
const int DIM_X = size.width;
const int DIM_Y = size.height;
float zoomFac = 3;
Mat visu;
resize(color_origImg, visu, Size((int)(color_origImg.columns*zoomFac),
(int)(color_origImg.sers*zoomFac)));
int nuclSize = 8;
int gradBinSize = 9;
float radRangeForOneBin = (float)(CV_PI / (float)gradBinSize); // разделение 180° на 9
бинов
     // подготовка структуры данных
int nucls_in_x_dir = DIM_X / nuclSize;
int nucls_in_y_dir = DIM_Y / nuclSize;
float*** StrGrads = new float**[nucls_in_y_dir];
int** nuclUpdateCounter = new int*[nucls_in_y_dir];
for (int y = 0; y<nucls_in_y_dir; y++)
{
StrGrads[y] = new float*[nucls_in_x_dir];
nuclUpdateCounter[y] = new int[nucls_in_x_dir];
for (int x = 0; x<nucls_in_x_dir; x++)
{
StrGrads[y][x] = new float[gradBinSize];
nuclUpdateCounter[y][x] = 0;
for (int bin = 0; bin<gradBinSize; bin++)
StrGrads[y][x][bin] = 0.0;
}
}
int bloc_in_x_dir = nucls_in_x_dir - 1;
int bloc_in_y_dir = nucls_in_y_dir - 1;
int descriptorDataIdx = 0;
int nucl_X = 0;
int nucl_Y = 0;
for (int blockx = 0; blockx<bloc_in_x_dir; blockx++)
{
for (int blocky = 0; blocky<bloc_in_y_dir; blocky++)
{
for (int nuclNr = 0; nuclNr<4; nuclNr++)
{
nucl_X = blockx;
nucl_Y = blocky;
if (nuclNr == 1) nucl_Y++;
if (nuclNr == 2) nucl_X++;
if (nuclNr == 3)
{
nucl_X++;
nucl_Y++;
}
for (int bin = 0; bin<gradBinSize; bin++)
{
float StrGrad = descriptorValues[descriptorDataIdx];
descriptorDataIdx++;
StrGrads[nucl_Y][nucl_X][bin] += StrGrad;
}
nuclUpdateCounter[nucl_Y][nucl_X]++;
}
}
}
  // рассчитываем среднюю величину градиента
for (nucl_Y = 0; nucl_Y<nucls_in_y_dir; nucl_Y++)
{
for (nucl_X = 0; nucl_X<nucls_in_x_dir; nucl_X++)
{
float NrUpdatesForThisNucl = (float)nuclUpdateCounter[nucl_Y][nucl_X];
```

```
// рассчитываем среднюю величину градиента для каждого бина
for (int bin = 0; bin<gradBinSize; bin++)
{
StrGrads[nucl_Y][nucl_X][bin] /= NrUpdatesForThisNucl;
}
}
}
// рисуем ячейки
for (nucl_Y = 0; nucl_Y<nucls_in_y_dir; nucl_Y++)
{
for (nucl_X = 0; nucl_X<nucls_in_x_dir; nucl_X++)
{
int draw_X = nucl_X * nuclSize;
int draw_Y = nucl_Y * nuclSize;
int mx = draw_X + nuclSize / 2;
int my = draw_Y + nuclSize / 2;
rectangle(visu, Point((int)(draw_X*zoomFac), (int)(draw_Y*zoomFac)), Point((int)((draw_X
+ nuclSize)*zoomFac), (int)((draw_Y + nuclSize)*zoomFac)), Scalar(100, 100, 100), 1);
// изображение для каждой ячейки величины градиента
for (int bin = 0; bin<gradBinSize; bin++)
{
float currentGradStrength = StrGrads[nucl_Y][nucl_X][bin];
if (currentGradStrength == 0)
continue;
float RadCurr = bin * radRangeForOneBin + radRangeForOneBin / 2;
float dirVecX = cos(RadCurr);
float dirVecY = sin(RadCurr);
float maxVecLen = (float)(nuclSize / 2.f);
float scale = 2.5;    // вычисление координат линий
float x1 = mx - dirVecX * currentGradStrength * maxVecLen * scale;
float y1 = my - dirVecY * currentGradStrength * maxVecLen * scale;
float x2 = mx + dirVecX * currentGradStrength * maxVecLen * scale;
float y2 = my + dirVecY * currentGradStrength * maxVecLen * scale;
// визуализация градиента
line(visu, Point((int)(x1*zoomFac), (int)(y1*zoomFac)), Point((int)(x2*zoomFac),
(int)(y2*zoomFac)), Scalar(0, 255, 0), 1);
}
}
}
  // очистка памяти
for (int y = 0; y<nucls_in_y_dir; y++)
{
for (int x = 0; x<nucls_in_x_dir; x++)
{
delete[] StrGrads[y][x];
}
delete[] StrGrads[y];
delete[] nuclUpdateCounter[y];
}
delete[] StrGrads;
delete[] nuclUpdateCounter;
return visu;
}
void compute_HystGr(const vector< Mat > & pict_lst, vector< Mat > & gradlt, const Size &
size)
{
HOGDescriptor HystGr;
HystGr.winSize = size;
Mat gray;
vector< Point > loc;
vector< float > descr;
vector< Mat >::const_iterator pict = pict_lst.begin();
vector< Mat >::const_iterator end = pict_lst.end();
```

```
for (; pict != end; ++pict)
{
cvtColor(*pict, gray, COLOR_BGR2GRAY);
HystGr.compute(gray, descr, Size(8, 8), Size(0, 0), loc);
gradlt.push_back(Mat(descr).clone());
#ifdef _DEBUG
imshow("grad", hogdescvis(pict->clone(), descr, size));
waitKey(10);
#endif
}
}
void train_supMach(const vector< Mat > & gradlt, const vector< int > & labl)
{
/* стандартные строки для обучения SVM */
Ptr<SVM> supMach = SVM::create();
supMach->setCoef0(0.0);
supMach->setDegree(3);
supMach->setTermCriteria(TermCriteria(TermCriteria::MAX_ITER, 100, 1e-3));
supMach->setGamma(0);
supMach->setKernel(SVM::LINEAR);
supMach->setNu(0.5);
supMach->setP(0.1);
supMach->setC(0.01);
supMach->setType(SVM::EPS_SVR);
Mat train_data;
conv_ml(gradlt, train_data);
clog << "Start training...";
supMach->train(train_data, ROW_SAMPLE, Mat(labl));
clog << "...[done]" << endl;

supMach->save(TRAINED_SVM);
}
void locat_drawing(Mat & pict, const vector< Rect > & locs, const Scalar & color)
{
if (!locs.empty())
{
vector< Rect >::const_iterator loc = locs.begin();
vector< Rect >::const_iterator end = locs.end();
for (; loc != end; ++loc)
{
rectangle(pict, *loc, color, 2);
}
}
}
void testing(const Size & size)
{
char key = 27;
Mat pict, draw;
Ptr<SVM> supMach;
HOGDescriptor HystGr;
HystGr.winSize = size;
VideoCapture vis;
vector< Rect > locs;
// загрузка обученного классификатора
supMach = StatModel::load<SVM>(TRAINED_SVM);
vector< float > HystGr_detector;
supMach_detect(supMach, HystGr_detector);
HystGr.setSVMDetector(HystGr_detector);
// запуск камеры
vis.open(TRAFFIC_VIDEO_FILE);
if (!vis.isOpened())
{
cerr << "Unable to open the device" << endl;
```

```
exit(-1);
}
int num_of_vehicles = 0;
bool end_of_process = false;
while (!end_of_process)
{
vis >> pict;
if (pict.empty())
break;
draw = pict.clone();
for (int pi = 0; pi < pict.sers; ++pi)
for (int pj = 0; pj < pict.columns; ++pj)
if (pj > pict.columns / 2) {
pict.at<Vec3b>(pi, pj)[0] = 0;
pict.at<Vec3b>(pi, pj)[1] = 0;
pict.at<Vec3b>(pi, pj)[2] = 0;
}
locs.clear();
HystGr.detectMultiScale(pict, locs);
locat_drawing(draw, locs, Scalar(0, 255, 0));
imshow(WINDOW_NAME, draw);
key = (char)waitKey(10);
if (27 == key)
end_of_process = true;
}
}
```