УНИВЕРСИТЕТ ИТМО

С.Н. Ярышев, В.А. Рыжова, В.В. Коротаев

ВИДЕОАНАЛИЗ В ЦИФРОВЫХ ВИДЕОИНФОРМАЦИОННЫХ СИСТЕМАХ БЕЗОПАСНОСТИ



Санкт-Петербург 2021

МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ РОССИЙСКОЙ ФЕДЕРАЦИИ

УНИВЕРСИТЕТ ИТМО

С.Н. Ярышев, В.А. Рыжова, В.В. Коротаев ВИДЕОАНАЛИЗ В ЦИФРОВЫХ ВИДЕОИНФОРМАЦИОННЫХ СИСТЕМАХ БЕЗОПАСНОСТИ

УЧЕБНОЕ ПОСОБИЕ

РЕКОМЕНДОВАНО К ИСПОЛЬЗОВАНИЮ В УНИВЕРСИТЕТЕ ИТМО по направлению подготовки 12.04.02 Оптотехника в качестве учебного пособия для реализации основных профессиональных образовательных программ высшего образования магистратуры

УНИВЕРСИТЕТ ИТМО

Санкт-Петербург 2021 Ярышев С.Н., Рыжова В.А., Коротаев В.В. Видеоанализ в цифровых видеоинформационных системах безопасности. Учебное пособие. – СПб: Университет ИТМО, 2021. – 110 с.

Рецензент(ы):

Ковриго Александр Олегович, бренд-менеджер оборудования видеонаблюдения, ООО «Легарда»

пособие предназначено для Учебное студентов по направлению подготовки магистров 12.04.02 Оптотехника, по основной образовательной программе "Прикладная оптика" и содержит материалы лабораторных работ по дисциплине "Видеосистемы и видеоаналитика"/"Video systems and video analytics". Обучающимся предлагается для освоения лабораторный комплекс, сформированный на основе библиотек компьютерного зрения с открытым исходным сетевой архитектуры цифровых кодом, a также видеоинформационных систем безопасности. Разработанный авторами лабораторный практикум направлен на формирование у студентов навыков обработки изображений с использованием пакета OpenCV, a также исследования модулей видеоанализа современных интеллектуальных платформ ТАЙФУН. видеонаблюдения TRASSIR К лабораторным работам И сформулированы контрольные вопросы и предложены варианты выполнения. Пособие подготовлено на инженерно-исследовательском факультете.

УНИВЕРСИТЕТ ИТМО

Университет ИТМО – ведущий вуз России в области информационных и фотонных технологий, один из немногих российских вузов, получивших в 2009 году статус национального исследовательского университета. С 2013 года Университет ИТМО – участник программы повышения конкурентоспособности российских университетов среди ведущих мировых научно-образовательных центров, известной как проект «5 в 100». Цель Университета ИТМО – становление исследовательского университета мирового уровня, предпринимательского по типу, ориентированного на интернационализацию всех направлений деятельности.

© Университет ИТМО, 2021 © Ярышев С.Н., Рыжова В.А., Коротаев В.В. 2021

СОДЕРЖАНИЕ

| ВВЕДЕНИЕ4 |
|---|
| Лабораторная работа №1. Методы выделения переднего плана, сегментации и |
| выделения признаков объектов7 |
| Лабораторная работа №2. Методы обнаружения и распознавания QR кодов22 |
| Лабораторная работа №3. Методы выделения и распознавания лиц |
| Лабораторная работа №4. Методы обнаружения и распознавания текста в изображении |
| Лабораторная работа №5. Методы улучшения субъективного восприятия изображений в системе ТАЙФУН59 |
| Лабораторная работа №6. Модуль распознавания лиц в системе TRASSIR66 |
| Лабораторная работа №7. Модуль распознавания автомобильных номеров AutoTRASSIR |
| СПИСОК ИСПОЛЬЗОВАННЫХ ИСТОЧНИКОВ |

ВВЕДЕНИЕ

Одним из важнейших направлений развития видеоинформационных систем безопасности является интеллектуальный анализ наблюдаемой сцены. Системы видеонаблюдения, в которых реализованы интеллектуальные функции, позволяют более эффективно решать поставленные задачи – с меньшими трудозатратами и на более высоком качественном уровне [1, 2].

Развитие видеоаналитики во многом обусловлено тенденциями увеличения количества камер на объектах на единицу площади. Так, например, известен рейтинг городов, сформированный исходя из плотности установки видеооборудования: 1 место занимает г. Ченнай (Индия) - 657 камер на 1 кв.км; 2 место – 480 камер - г. Хайдарабад (Индия); 3 место – 411 камер – г. Харбин (Китай); 4 место – 399 (Лондон); 5 место – 385 (Сямынь). Москва (77 камер) и Санкт-Петербург (38 камер) занимают, соответственно 30 и 46 места [3].

Мониторинг сотен камер существенно усложняет задачу работы оператора, а порой она становится невыполнимой. Выявляя только нужные события и оповещая о них, интеллектуальный анализ изображений позволяет оператору сконцентрироваться исключительно на принятии решений.

Почти все современные цифровые системы видеонаблюдения имеют хотя бы простейшие средства анализа видеоизображения – например, детекторы движения, которых, как правило, недостаточно для решения очень многих актуальных на сегодняшний день задач. Поэтому развитие технологий интеллектуального анализа видеоизображения является одним из приоритетных направлений деятельности известных на международном рынке производителей видеоинформационных систем безопасности (Hikvision Digital Technology, Dahua Honeywell Technology, Bosch Security Systems, Security & Fire. Axis Communications, Flir Systems (Commercial, Government and Defense), IDIS, Hanwha Techwin, Nedap, Milestone Systems).

По мнению известных аналитиков тенденций мирового рынка видеонаблюдения, в условиях пандемии коронавируса <u>COVID-19</u> наиболее важным для компаний становится не создание новых устройств, определяющих ситуацию на рынке, а активное расширение интеллектуального функционала своих продуктов. В этом направлении сейчас активно ведутся исследования, совершенствуются механизмы обработки видеоряда.

особенно современной ситуации Университета B важной для представляется подготовка высококлассных специалистов B области исследования оптико-цифровых, в том числе видеоинформационных, систем на формирования будущих инженеров наиболее основе y значимых профессиональных компетенций [4]:

- способность проектировать и исследовать оптико-электронные цифровые системы с использованием перспективных методов обработки

видеоинформации, для решения профессиональных задач на основе современного развития оптики, электроники и вычислительной техники;

- способность разрабатывать и исследовать новые способы получения, хранения и обработки информации с использованием оптико-электронных приборов и систем на основе разрабатываемых конкурентоспособных технологий;
- способность выполнять теоретические и прикладные научные исследования, моделирование, разработку, экспериментальные исследования оптико-цифровых систем;
- способность использовать современное специализированное программное и материально-техническое обеспечение для моделирования, разработки, экспериментальных исследований оптико-цифровых систем.

Настоящее пособие посвящено освоению и исследованию методов видеоанализа, реализующихся при обработке изображений в видеокамерах и серверах систем видеонаблюдения. Лабораторные работы №№ 1-4 содержат описания методов обработки и задания для выполнения студентами анализа изображений с использованием пакета Open CV. Лабораторные работы №№ 5-7 содержат описание лабораторного практикума для проведения экспериментов с использованием оборудования и программного обеспечения (ПО) российских разработчиков интеллектуальных платформ для видеонаблюдения: ООО «ЭВС» и ООО "ДССЛ-НЕВА" (компания "DSSL") [5].

Лабораторный практикум предназначен для закрепления обучающимися теоретических знаний и приобретения ими практических навыков при изучении дисциплины «Видеосистемы и видеоаналитика». Пособие содержит 7 (семь) лабораторных работ, направленных на формирование основных профессиональных компетенций путем приобретения студентами умений и навыков проектирования и эксплуатации видеоинформационных систем безопасности.

Описание каждой лабораторной работы содержит основные теоретические сведения, порядок выполнения работы с примерами ее выполнения, контрольные вопросы. В конце пособия приводится список рекомендуемых источников литературы.

Выполнение лабораторной работы производится во время занятий в аудитории, оснащенной необходимыми программно-аппаратными средствами, в присутствии преподавателя. Организация лабораторного практикума предполагает в обязательном порядке проведение инструктажа студентов по соблюдению требований техники безопасности при эксплуатации персональных компьютеров и систем сетевого видеонаблюдения, формирующих лабораторные установки.

Внимание!

Лабораторные установки могут включаться и выключаться только преполавателем или лаборантом. или под ИХ непосредственным наблюдением! На персональном компьютере необходимо выполнять только те действия, которые предусматривает порядок выполнения лабораторной работы. Запрешается запускать приложения, не относящиеся К лабораторной работе, и инсталлировать сторонние ПО и модули Python самостоятельно.

В процессе выполнения лабораторной работы студент последовательно выполняет задания в соответствии с указанным порядком. Время, отводимое на выполнение, определяется трудоемкостью лабораторного практикума.

По завершении работы обучающийся демонстрирует преподавателю результаты ее выполнения на персональном компьютере и предъявляет файлы со скриншотами и другими материалами для включения их в отчет.

Отчет о лабораторной работе должен включать в себя изображения экрана (скриншоты), полученные в ходе работы, комментарии к изображениям. Для получения скриншота в ходе работы нужно запоминать соответствующие изображения путем нажатия на кнопку Print Screen для помещения скриншота в буфер обмена и перемещения его в заранее открытый документ MS Word. В дальнейшем этот документ с комплектов скриншотов следует использовать для отчета. Отчет должен быть оформлен в электронном виде и распечатан (при необходимости). За основу отчета должен быть взят прилагаемый шаблон:

Отчет по лабораторной работе № _____«______

(название лабораторной работы)

1. Цель и задачи лабораторной работы: ____

- 2. Методика проведения исследования: _____
- 3. Результаты: ____
- 4. Выводы: ____

В имеющемся шаблоне следует заполнить обязательные поля титульного листа, включающие ФИО и группу студента, дату выполнения работы. В качестве результатов используются скриншоты основных окон программы, их описание и комментарии.

Размер отчета должен быть не менее пяти страниц. При необходимости, вставленные в шаблон рисунки следует дополнить поясняющими графическими элементами и отмасштабировать.

Авторы выражают глубокую благодарность студентам специалитета Ласкавому Н.С., Третьякову А.В., Васильеву Г.В., Яковлеву А.И., НевротовуА.А. за помощь при подготовке материалов для данного пособия.

Лабораторная работа №1. Методы выделения переднего плана, сегментации и выделения признаков объектов

Цели работы

- 1. Изучить методы выделения переднего плана, сегментации и выделения признаков объектов в изображениях на основе открытых библиотек обработки изображений.
- 2. Получить представление о работе в операционной системе Linux, в среде разработки Spyder.
- 3. Получить практические навыки использования языка программирования Python, модулей и пакетов OpenCV [6] и других.

Работа рассчитана на 2 часа в стандартном варианте или на 4 часа в расширенном варианте. В последнем случае преподаватель дает дополнительные задания, связанные с настройкой режимов работы ПО, выделения признаков объектов в изображениях.

Основные теоретические сведения

Выделение переднего плана и сегментация объектов

Под выделением переднего плана обычно подразумевается отделение объекта или группы объектов, представляющих интерес, от заднего фона, который иногда называется задним планом. При этом далеко не всегда разделение производится путем отделения близко расположенных элементов изображения от дальних объектов. Бывает так, что именно дальние объекты представляют интерес. В видеопотоке довольно часто передний план интерпретируется как комбинация движущихся в поле зрения объектов, а задний план, соответственно, как неподвижный фон и группа неподвижных объектов. В этом есть определенный смысл, так как в видеопотоке (в отличие от статических изображений) основная информация сосредоточена именно на движущихся объектах.

Под сегментацией объектов подразумевается операция разделения объектов на отдельные группы по каким-либо признакам, например цвет, форма и размер. Кроме того, при сегментации возможно разделение соприкасающихся объектов.

Довольно часто выделение переднего плана и сегментация выполняются в рамках одного составного алгоритма. Примером такого алгоритма может являться алгоритм водораздела (watershed algorithm) [7].

Для сегментации объектов часто используется разделение объектов по цвету. В этом случае объекты переднего плана (если перед этим использовался алгоритм выделения переднего плана) или исходное изображение подвергаются цветовому анализу. Как известно, исходное изображение чаще всего присутствует в системе RGB или YUV. Однако, в этом случае цвет объекта определить довольно сложно, так как он будет зависеть от уровня яркости и цветовой насыщенности изображения. Поэтому для селекции объектов по цвету чаще всего использует систему цветности HSV (*Hue, Saturation, Value* — тон, насыщенность, значение).

Ние — цветовой тон. Варьируется в пределах 0—360°, однако иногда приводится к диапазону 0—100 или 0—1.

Saturation — насыщенность. Варьируется в пределах 0—100 или 0—1. Чем больше этот параметр, тем «чище» цвет, поэтому этот параметр иногда называют чистотой цвета. А чем ближе этот параметр к нулю, тем ближе цвет к нейтральному серому.

Value (значение цвета) или Brightness — яркость. Также задаётся в пределах 0—100 или 0—1.

Из представленных величин за цвет отвечает параметр *H*. Его значение можно определить по шкале цветов (рис.1).



Рисунок 1 – Шкала цветов для определения цветового тона [8]

Таким образом, селекция объектов по цвету может быть существенно упрощена. Считаем, что:

 $H \in [0; 360]; S, V, R, G, B \in [0; 1]$

Пусть *MAX* – максимальное значение из R, G, B, а *MIN* – минимальное из них. Тогда пересчет из системы RGB в систему HSV производится по формулам [8]:

$$H = \begin{cases} 60 \cdot \frac{G - B}{MAX - MIN} + 0, & \text{если } MAX = R \text{ и } G \ge B \\ 60 \cdot \frac{G - B}{MAX - MIN} + 360, & \text{если } MAX = R \text{ и } G < B \\ 60 \cdot \frac{B - R}{MAX - MIN} + 120, & \text{если } MAX = G \\ 60 \cdot \frac{R - G}{MAX - MIN} + 240, & \text{если } MAX = B \end{cases}$$
$$S = \begin{cases} 0, & \text{если } MAX = 0 \\ 1 - \frac{MIN}{MAX}, & \text{если } MAX \neq 0 \end{cases}$$
$$V = MAX$$

Анализ цветового распределения в изображении удобно производить по трехмерному графику (рис.2) или с использованием визуальных маркеров [9, 10].



Рисунок 2 – График для анализа цветового распределения

Получив диапазон используемых значений для системы HSV (прежде всего значение H), можно использовать их в цветовых фильтрах анализатора цветного изображения, предварительно переведенных в формат HSV.

Выделение признаков объектов

Выделение признаков объектов является одной из важных составляющих алгоритмов выделения и распознавания объектов.

При извлечении признаков объектов создается семейство точек, называемых ключевыми точками и составляющих шаблон объекта, по которому можно выделить и идентифицировать объект. Это возможно при условии, что это семейство ключевых точек действительно соответствует особенностям объекта, по которым его можно отделить от других объектов и затем идентифицировать.

Как правило, ключевые точки находятся на границах областей с резким изменением каких-либо параметров (градиента яркости или цвета, изменении направления, положении тонких линий и пр.), а также при сочетании этих изменений.

точки удобно отыскивать с использованием Ключевые фильтров, выделяющих контуры изображений, и последующим анализом поведения этих контуров. Кроме того, можно использовать апертурные фильтры с сканирующих апертур виде небольших использованием В примитивов, содержащих резкие перепады яркости в различных направлениях, например, признаки Хаара.

Ввиду большого разнообразия исследуемых объектов в настоящее время разработано большое количество алгоритмов выделения признаков. У каждого алгоритма есть особенности, которые используются для более эффективного выделения признаков какого-либо типа объектов.

В лабораторной работе будут рассмотрены следующие алгоритмы выделения признаков объектов:

- детектор углов Харриса;
- детектор углов Ши-Тамази;
- алгоритм SIFT (Scale Invariant Feature Transform);
- алгоритм SURF (Speeded-up robust features);
- алгоритм FAST (Features from accelerated segment test);
- алгоритм BRIEF (Binary Robust Independent Elementary Features).

В детекторе углов Харриса используется поиск ключевых точек, особенность которых в том, что они указывают на фрагмент изображения, содержащий угол. В первом приближении в основе алгоритма имеется матрица размером 2 х 2, в которой путем вычисления градиентов определяется один из элементов, величина которого резко отличается от трех соседних. В настройках алгоритма может изменяться размер матрицы и величина градиента. Кроме того, можно менять величину угла в фрагменте, который будет идентифицироваться. Более подробно см. [11].

Более эффективной версией детектора углов является детектор углов Ши-Тамази. В этом детекторе, в отличие от предыдущего случая, обнаруживается заданное количество наиболее выделяющихся углов в изображении. Количество углов может варьироваться настройками алгоритма. Более подробно см. [12].

Как мы видим, вышеуказанные алгоритмы позволяют получать ключевые точки только одного типа — углы, которые будут иметь только параметры в виде их координат. Но для решения задач выделения и распознавания объектов этих точек недостаточно. Требуется набор точек, которые отличаются между собой особенностями выделяемых фрагментов изображения, включая направления этих особенностей. Для этого необходимо использовать более сложные алгоритмы.

Алгоритм **SIFT** (Scale Invariant Feature Transform) инвариантен к масштабу, смещению, повороту. Алгоритм базируется на анализе изображения после применения фильтра гауссового размытия (гауссианов) с разными значениями размытия. Таким образом получается инвариантность к масштабу изображения. Ключевая точка определяется из условия, что она является локальным экстремумом разности гауссианов. Кроме того, вычисляется положение точки с субпиксельной точностью, определяется контраст в месте перепада яркости. Вычисляется также ориентация ключевой точки. Более подробные сведения об алгоритме можно получить здесь [13].

В результате ключевой точкой (дескриптором) в алгоритме SIFT является вектор. Размерность дескриптора составляет 32 компоненты (2х2х8), но на практике используются дескрипторы размерности 128 компонент (4х4х8). Полученный дескриптор нормализуется, после чего все его компоненты,

значение которых больше 0.2, урезаются до значения 0.2, и затем дескриптор нормализуется ещё раз. В таком виде дескрипторы готовы к использованию. Более подробно см. [13, 14].

Основной недостаток алгоритма SIFT — сложность реализации и, как следствие, небольшое быстродействие.

Существует другой алгоритм — **SURF** (Speeded-up robust features), который вместо гауссовой функции использует более простой фильтр — box filter. В результате быстродействие заметно повышается. Но даже этот ускоренный алгоритм требует больших вычислительных ресурсов и, как следствие, имеет недостаточное быстродействие для обработки видеопотоков в реальном времени. Более подробно см. [15].

Следующий алгоритм **FAST** (Features from accelerated segment test). Фактически этот алгоритм позволяет только обнаружить точки, которые могут быть использованы как ключевые, но никаких параметров, кроме координат этих точек, данный алгоритм не дает. Для дальнейшего построения дескрипторов с полным набором параметров после работы алгоритма FAST следует использовать алгоритмы SIFT или SURF, но уже только для найденных ключевых точек. В целом, такая последовательность работает значительно быстрее.

Алгоритм **BRIEF** позволяет проверить ранее найденные ключевые точки и рассчитать их дополнительные параметры, которые позволяют использовать эти ключевые точки как дескрипторы. Этот алгоритм обладает повышенным быстродействием.

Существенным организационным недостатком алгоритмов SIFT и SURF является тот факт, что они запатентованы и их использование для коммерческих применений затруднено. Альтернативой этим алгоритмам является алгоритм, предложенный командой OpenCV, под названием Oriented FAST and Rotated BRIEF (ORB). Этот алгоритм также позволяет получить дескрипторы из изображения, примерно аналогичные дескрипторам, которые получены с помощью алгоритмов SIFT и SURF. Более подробно см. [16].

Порядок выполнения работы

Работа включает в себя следующие разделы:

- включение лабораторной установки и подготовка ее к работе;
- изучение селекции объектов по цвету в системе цветности HSV;
- изучение алгоритмов детектирования объектов;
- работа с самостоятельно полученным изображением;
- выключение лабораторной установки.

І. Работа выполняется на лабораторной установке, состоящей из персонального компьютера с установленной операционной системой Linux (Ubuntu 20.04 или

старше) и подключенной к нему usb-видеокамерой. Подготовка лабораторной установки содержит следующие этапы:

1. С помощью преподавателя или лаборанта включите компьютер и дождитесь загрузки операционной системы.

2. На рабочем столе найдите папку **LR**, зайдите в нее и скопируйте папку **LR_segmentation_detection** на Рабочий стол. В дальнейшем все действия, предусмотренные порядком выполнения работы, выполняйте в этой папке.

3. Пользуясь боковой панелью, запустите среду разработки Spyder и дождитесь ее загрузки (рис.3):



Рисунок 3 – Окно среды разработки Spyder

II. Изучение селекции объектов по цвету в системе цветности HSV выполняется в следующей последовательности:

1. Загрузите программу hsv1.ру в основное окно среды разработки:

- в меню Файл выберите команду Открыть;
- в появившемся окне выберите Рабочий стол;
- в открывшемся списке выберите папку LR_segmentation_detection;
- в открывшемся списке выберите файл hsv1.py (рис. 4).

| Отмена | Открыть файл | | Q | Открыть |
|------------------|---|---------------------------------|----------------|-------------------------------------|
| 🕚 Недавние | ▲ ப் у227 □ Рабочий стол LR_segmentation_detection ▶ | | | |
| 🕞 Домашняя папка | Имя • | Размер | Тип | Изменён |
| 🔲 Рабочий стол | images fast_detector.py | 1,3 kB | Текст | 23 апр. 15 апр. |
| 🖽 Видео | <pre>fast_detector_with_brief_extractor.py</pre> | 673 байта | Текст | 15 апр. |
| 📄 Документы | good_reatures.py harris_corners.py | 441 байт 1,3 kB | Текст Текст | 14 мая. 14 мая. |
| 🖞 Загрузки | e hsv1.py | 4,1 kB | Текст | 14 мая. |
| 🛋 Изображения | style detector.py site detector.y | 1,3 кв 609 байт 443 байта | Текст | 14 мая. 7 окт 2017 7 окт 2017 |
| Л Музыка | sint_detect.py surf.py | 443 байта 597 байт | Текст | 30 апр. |
| | surf_camera.py | 856 байт | Текст | 30 апр. |
| + Другие места | surf_detect.py | 565 байт | Текст | 7 окт 2017 |
| | test_harris.py | 574 байта | Текст | 19 апр. |
| | e test_vodorazdet.py | 2,4 КВ | текст | rs anp. |
| | | | | |
| | | | | |
| | | | | |
| | | | | |

Рисунок 4 – К выбору рабочей программы

2. Запустите программу hsv1.py, нажав на зеленый треугольник. На рисунке 5 представлены расположенные в окне консоли изображения, иллюстрирующие этапы работы программы селекции объектов по цвету.



Рисунок 5 – Этапы работы программы селекции объектов по цвету

В качестве объекта селекции используется изображение оранжевой рыбки (персонаж Немо) nemo.jpg. Вы увидите выведенные в консоль исходное изображение (рис.5, а), затем результат работы оранжевого (рис.5, б) и белого (рис.5, в) фильтров в отдельности, затем суммарное изображение после применения оранжевого и белого фильтров (рис.5, г). Большое изображение на рисунке 5, д — после суммарной работы двух фильтров с последующим сглаживанием.

Три изображения на рисунке 6 представляют собой трехмерную визуализацию распределения цветов данного изображения в системах RGB (рис. 6, а), HSV (рис. 6, б) и HSV после цветовой фильтрации (рис. 6, в).





3. Замените файл изображения на new_auto1.jpg.

Запустите программу hsv1.py на выполнение. Дождитесь окончания работы программы. Внимательно рассмотрите результаты работы. Какие изображения удалось выделить? Сделайте скриншоты для отчета и анализа.

4. Замените файл изображения на new_auto2.jpg. Запустите программу на выполнение. Дождитесь окончания работы программы. Внимательно рассмотрите результаты работы. Какие изображения удалось выделить? Сделайте скриншоты для отчета и анализа.

5. Загрузите программу hsv2.py в основное окно среды разработки:

- в меню Файл выберите команду Открыть;
- в появившемся окне выберите Рабочий стол;

- в открывшемся списке выберите папку LR_segmentation_detection;
- в открывшемся списке выберите файл hsv2.py.

6. Запустите программу, нажав на зеленый треугольник. В качестве объекта селекции используется изображение new_auto2.jpg. В открытом окне вы увидите изображения автомобилей трех цветов (рис. 7, а).

Щелкните левой кнопкой мыши по изображению *голубого* автомобиля (именно по голубому цвету). Вы увидите появляющиеся в окне консоли (рис. 7, б) параметры выбранного голубого цвета в системе HSV.



Рисунок 7 – Работа с изображением автомобилей разных цветов

Затем щелкните по изображению автомобиля *красного* цвета. Вы увидите параметры этого цвета в окне консоли. Запомните или сохраните комбинации, соответствующие разным оттенкам этих цветов. Выйдите из программы, нажав любую клавишу.

7. Вновь загрузите программу hsv1.py. Найдите фрагмент программы, содержащий раздел, где настраиваются цветовые фильтры:

Здесь настраиваются цветовые фильтры для первого цвета! # Настоящий оранжевый цвет в системе HSV light_orange = (1, 190, 200) dark_orange = (18, 255, 255)

Эти значения можно изменять! #light_orange = (95, 150, 190)

В верхней части закомментируйте строчки для оранжевого цвета, а в нижней части внесите полученные ранее параметры для голубого цвета, при этом в первой строчке эти параметры должны быть уменьшены на 5, а во второй строчке — увеличены на 5. Таким образом, вы зададите параметры голубого цвета в системе HSV с допуском 10. Снимите значки комментариев с измененных нижних строчек.

Аналогично выполните эту операцию для второго (красного) цвета. Для этого найдите фрагмент кода:

```
#______# Здесь настраиваются цветовые фильтры для второго цвета!
# Настоящий белый цвет в системе HSV
light_white = (0, 0, 200)
dark_white = (145, 60, 255)
# Эти значения можно изменять!
#light_white = (0, 0, 200)
#dark_white = (145, 60, 255)
# ______
```

Внесите полученные параметры для красного цвета в нижние строчки кода также в диапазоне -5 и +5. Не забудьте поменять комментарии.

8. Запустите программу hsv1.py. Если ошибок не было, то вы должны получить итоговый результат с выделением автомобилей выбранных цветов (рис. 8). Сделайте скриншоты полученных результатов.



Рисунок 8 – Результат селекции автомобилей по цвету

Ш. Изучение алгоритмов детектирования особых точек.

Угловой детектор Harris

- 1. Загрузите программу harris_corners.py в основное окно среды разработки:
 - в меню Файл выберите команду Открыть;
 - в появившемся окне выберите Рабочий стол;
 - в открывшемся списке выберите папку LR_segmentation_detection;
 - в открывшемся списке выберите файл harris_corners.py.

2. В основном окне среды разработки Spyder появится текст программы. Ознакомьтесь с текстом программы. Пользуясь комментариями в тексте, изучите назначение основных операторов и функций, которые используются в этой программе.

3. Проверьте, чтобы в качестве объекта селекции был выбран файл изображения box.png. Запустите программу, нажав на зеленый треугольник. Вы получите два изображения коробки при разных значениях величины области обнаружения угла (рис. 9). Рассмотрите эти изображения, отметьте для себя их различия.



Рисунок 9 – Изображения коробок с разными областям выделения углов

Поменяйте параметры blockSize в двух строчках программы с 4 и 14 на 8 и 20. Запустите программу при этих новых значениях. Посмотрите, что изменилось. Сделайте скриншоты.

- 4. Выберите изображение face1.jpg. Повторите п. 3 для этого изображения.
- 5. Выберите изображение street1.jpg. Повторите п.3 для этого изображения
- 6. Загрузите программу good_features.ру в основное окно среды разработки:
 - в меню **Файл** выберите команду **Открыть**;
 - в появившемся окне выберите Рабочий стол;
 - в открывшемся списке выберите папку LR_segmentation_detection;
 - в открывшемся списке выберите файл good feature.py.

7. Выберите в качестве файла изображения street1.jpg и запустите программу с помощью зеленого треугольника. Посмотрите на полученное изображение (рис.10). Что выделено синими точками? Сделайте скриншот.



Рисунок 10 – Изображение уличного движения с селекцией углов

Поменяйте параметр программы maxCorners со значения 2000 на значение 200. Запустите программу. Что изменилось в работе программе? Сделайте скриншот.

Поменяйте параметр программы minDistance со значения 20 на значение 100. Запустите программу. Что изменилось в программе? Сделайте скриншот.

8. Измените файл изображения на face1.jpg. Выполните п.7 для этого изображения.

Алгоритм SURF (детектор ORB)

- 9. Загрузите программу surf.py в основное окно среды разработки:
 - в меню Файл выберите команду Открыть;
 - в появившемся окне выберите Рабочий стол;
 - в открывшемся списке выберите папку LR_segmentation_detection;
 - в открывшемся списке выберите файл surf.py.

В этой программе используется детектор ORB.

10. Выберите в качестве объекта селекции файл изображения box.jpg и запустите программу с помощью зеленого треугольника. Внимательно рассмотрите изображение (рис. 11). Какие элементы на нем появились? Что они означают? Сделайте скриншот.



Рисунок 11 – Работа детектора ORB

Попробуйте поменять параметры nfeatures = 100, edgeThreshold = 10, patchSize = 41, fastThreshold = 51, nlevels = 5, firstLevel = 1 (по одному!). Добейтесь, чтобы на изображении были существенные изменения. Сделайте скриншоты и запишите для них указанные параметры.

11. Поменяйте файл изображения на street1.jpg. Выполните п.10 для этого изображения (рис. 12, а).



Рисунок 12 - Работа детектора ORB

12. Поменяйте файл изображение на face1.jpg. Выполните п.10 для этого изображения (рис. 12, б)

IV. Работа с самостоятельно полученным изображением.

1. Загрузите программу surf_camera.py в основное окно среды разработки:

- в меню **Файл** выберите команду **Открыть**;
- в появившемся окне выберите Рабочий стол;
- в открывшемся списке выберите папку LR_segmentation_detection;
- в открывшемся списке выберите файл surf_camera.py.

В этой программе используется детектор ORB.

2. Отрегулируйте камеру таким образом, чтобы в поле ее зрения присутствовало лицо человека, аналогично изображенному на рисунке 13. Внимательно рассмотрите изображение и результаты работы детектора. Какие элементы на нем появились? Что они означают? Сделайте скриншот.



(x=40, y=4) ~ R:5 G:5 B:0



3. Попробуйте поменять параметры nfeatures=100, edgeThreshold = 10, patchSize = 41, fastThreshold = 51, nlevels = 5, firstLevel = 1 (по одному!). Добейтесь, чтобы положение особых точек (дискрипторов) наилучшим образом соответствовало изображению лица. Сделайте скриншоты и запишите для них эти параметры.

4. Скопируйте все полученные скриншоты на собственный носитель для составления отчета.

V. Выключение лабораторной установки.

Обратитесь к преподавателю или лаборанту для проверки результатов и выключения компьютера.

Содержание отчета

1. Краткие сведения об исследуемых алгоритмах.

2. Результаты распознавания особых точек при использовании разных алгоритмов на разных изображениях и с различными настройками алгоритмов.

3. Описание влияния изменения различных параметров алгоритмов на качество формирования особых точек.

4. Анализ работы программы формирования особых точек для изображения, полученного с видеокамеры.

5. Описание положения особых точек и порядок поиска наилучших параметров алгоритма ORB.

Контрольные вопросы

1. Что такое сегментация объектов и какие алгоритмы ее реализуют?

2. Что такое ключевые точки на изображении объекта и для чего они используются?

3. В чем состоит особенность работы алгоритмов детектора углов Харриса и детектора углов Шима-Тамази в поиске ключевых точек?

4. Каковы преимущества алгоритмов SURF и SIFT перед алгоритмами детектора углов Харриса и детектора углов Шима-Тамази?

5. Каковы этапы работы алгоритма ORB по поиску ключевых точек?

Лабораторная работа №2. Методы обнаружения и распознавания QR кодов

Цели работы

- 1. Изучить методы выделения и распознавания изображений QR кодов на основе открытых библиотек обработки изображений.
- 2. Получить представление о работе в операционной системе Linux, в среде разработки Spyder.
- 3. Получить практические навыки использования языка программирования Python, модулей и пакетов OpenCV, Zbar и других.

Работа рассчитана на 2 часа в стандартном варианте или на 4 часа в расширенном варианте. В последнем случае преподаватель дает дополнительные задания, связанные с настройкой режимов работы ПО выделения и распознавания QR кодов.

Основные теоретические сведения

Обнаружение и распознавание QR кодов являются одними из наиболее востребованных функций видеоанализа в технологических телевизионных системах и системах технического зрения. Эти функции используются при маркировании товаров в магазинах, комплектующих и деталей на сборочных предприятиях, на автоматизированных складах и в других применениях. Впервые предложены и использованы на автосборочных заводах Японии.

Обнаружение QR кода на изображении — это процесс обработки изображения с целью поиска в изображении одного или нескольких QR кодов с указанием их координат в изображении. Результатом работы этого процесса может являться база данных выделенных QR кодов с их координатами и небольшими фрагментами изображения, содержащими QR коды.

Распознавание QR кодов — это процесс расшифровки изображения QR кода в соответствии с принятым стандартом путем нахождения областей синхронизации, чтения массива данных и проверки их правильности на основе кодов Рида-Соломона.

QR-код представляет собой квадратное изображение, соответствующее ГОСТ Р ИСО/МЭК 18004-2015 [17].

Изображение QR кода (применяется термин "QR Code символика") включает в себя несколько структурных элементов и состоит из области кодирования и функциональных шаблонов (рис. 14).

Область кодирования включает в себя служебную информацию о формате и версии кода. Остальная площадь, отведенная под область кодирования,

содержит данные и коды исправления ошибок. Минимальный элемент изображения, соответствующий одному биту, носит название модуль.



Рисунок 14 – Структура символа кода версии 7 [17]

Функциональные шаблоны данных не содержат и состоят из следующих элементов: шаблоны поиска, разделители, шаблоны синхронизации, направляющие шаблоны.

Зона вокруг изображения QR-кода должна быть свободной (обычно белым или черным фоном).

В соответствии со стандартом существует 40 возможных вариантов размера символа QR кода, которые обозначаются как версия 1, версия 2 и до версии 40. При этом размер изображения QR кода версии 1 имеет размеры 21 х 21 модуль, версии 2 — 25 х 25 модулей и так далее до версии 40 с размером 177 х 177 модулей. От размера модуля зависит количество записанной в нем информации. Примеры модулей различных размеров приведены на рисунке 15.

В соответствии со стандартом, в углах изображения символа имеются три шаблона поиска (рис. 16), по которым организуется поиск и обнаружение QR-кода в изображении или видеопотоке.

Идентификация шаблона поиска вместе с шаблонами синхронизации однозначно определяет размер, положение и угловую ориентацию символа в поле обзора.



Рисунок 15 – Примеры модулей [17]

По мере увеличения формата в символе появляются дополнительные направляющие шаблоны, которые напоминают шаблон поиска меньшего размера. Они обеспечивают дополнительную синхронизацию для символов большого размера.

Кодируемые данные (кодовые слова) могут быть представлены как в алфавитно-цифровой форме, так и в форме двоичных данных, в частности, в байтовом режиме. Эти данные занимают основную площадь символа, но не могут накладываться на функциональные шаблоны.



Рисунок 16 – Структура шаблона поиска

Расположение модулей и порядок их следования определяется в стандарте, но, как правило, общий поток разбивается на небольшие элементы размером 2 х 4 модуля, образующие байт. И эти элементы, в свою очередь, располагаются в колонках по направлению сверху вниз, а колонки – по направлению слева направо. При кодировании применяется технология исправления ошибок на основе кодов Рида-Соломона.

Применяется также технология маскирования данных с помощью специальных шаблонов таким образом, чтобы темные и светлые модули располагались бы наиболее рациональным образом. Это означает, что в массиве данных не должно быть больших областей только с белыми или только с черными модулями и во всем символе должно быть примерно одинаковое количество белых и черных модулей.

Более подробно с технологией формирования изображения QR кода можно ознакомиться, изучив описание программного пакета ZBar Bar Code Reader с открытым исходным кодом для считывания штрих-кодов из различных источников, таких как видеопотоки, файлы изображений и необработанные датчики интенсивности [18].

Количество закодированной информации зависит от выбранного формата, версии QR, а также от уровня исправления ошибок кода. QR код минимального размера версии 1, содержащий 21 х 21 модуль, позволяет закодировать от 10 до 21 символа. Модуль максимального стандартного размера версии 40 позволяет закодировать от 1852 до 4296 символов. Если длина кодируемой последовательности больше, то может применяться специальная технология структурированного соединения QR кодов.

Общепринятая методика декодирования символа QR кода включает в себя следующие этапы:

1. Определяется место нахождения символа в изображении или видеопотоке. Для этого используют три шаблона поиска. Полученное изображение символа локализуется в соответствующем окне. Выстраивается система координат,

привязанная к шаблонам поиска. Более подробно см. раздел «Рекомендуемый алгоритм декодирования для QR Code» [17].

2. Считывается информация только о формате с соответствующих модулей в бинарном виде. Используя шаблон маски, восстанавливают структуру битов закодированных данных. Считывают данные в виде последовательности и проверяют ее правильность с помощью кодов Рида-Соломона.

3. Считывается информация о номере версии и определяется версия символа (от версии 1 до версии 40).

4. На основе информации о формате и версии получают шаблон маски и, реализуя наложение маски на модули данных путем применения операции «исключающее или», восстанавливают область данных.

5. В соответствии с форматом и версией символа определяют порядок следования данных в символе и восстанавливают кодовые слова.

6. Восстанавливают обнаруженные ошибки с помощью кодов Рида-Соломона.

7. Разделяют кодовые слова на сегменты (байты) в соответствии со структурой данных

8. Декодируют полученные знаки данных в соответствии с применяемым режимом (алфавитно-цифровым, байтовым и др.). Выводится окончательный результат декодирования.

В лабораторной работе применяются две стандартных библиотеки, в состав которых входят готовые функции обнаружения и распознавания QR кодов.

1. Библиотека работы с QR кодами Zbar. В программе Python загружается как внешний модуль командой import pyzbar.

2. Библиотека компьютерного зрения OpenCV. В программе Python загружается в виде внешнего модуля командой import cv2.

Порядок выполнения работы

Работа включает в себя следующие разделы:

- включение лабораторной установки и подготовка ее к работе;
- изучение алгоритмов обнаружения и распознавания QR кодов в среде Python;
- работа с изображением кода, самостоятельно полученного с использованием видеокамеры, на основе библиотеки Zbar;
- работа с изображением кода, самостоятельно полученного с использованием видеокамеры, на основе пакета OpenCV;
- работа с изображением самостоятельно сформированного кода с использованием пакета OpenCV;
- выключение лабораторной установки.

I. Работа выполняется на лабораторной установке, состоящей из персонального компьютера с установленной операционной системой Linux (Ubuntu 20.04 или старше) с подключенной к нему usb-камерой. В работе используются отпечатанные образцы QR кодов. Подготовка лабораторной установки содержит следующие этапы:

1. С помощью преподавателя или лаборанта включите компьютер и дождитесь загрузки операционной системы.

2. На рабочем столе найдите папку LR, зайдите в нее и скопируйте папку LR_QR_code_recognition на Рабочий стол. В дальнейшем все действия, предусмотренные порядком выполнения лабораторной работы, выполняйте в этой папке.

3. Пользуясь боковой панелью (см. рис.3), запустите среду разработки Spyder и дождитесь ее загрузки.

II. Изучение работы алгоритмов обнаружения и распознавания QR кодов в среде Python выполняется в следующей последовательности:

1. Загрузите программу zbar.py в основное окно среды разработки:

- в меню Файл выберите команду Открыть;
- в появившемся окне выберите Рабочий стол;
- в открывшемся списке выберите папку LR_QR_code_recognition;
- в открывшемся списке выберите файл zbar.py.

2. В основном окне среды разработки Spyder появится текст программы (рис. 17). Ознакомьтесь с текстом программы. Пользуясь комментариями в тексте, изучите назначение основных операторов и функций, которые используются в этой программе.

3. Откройте папку с лабораторной работой на рабочем столе. В этой папке находятся несколько тестовых изображений, содержащих QR коды. Обратите внимание, что файлы изображений могут быть в разных форматах и с разными расширениями файлов.

Загрузите первое из этих изображений 1qr.png. Запустите программу, нажав на зеленый треугольник. Вы должны получить изображение 1qr.png с красной рамкой, которая покажет выделенный QR код, а сверху — текст расшифрованного кода.

В ряде случаев при неудачном расположении кода надпись может быть плохо видна. На этот случай дешифрированный код дублируется в терминале программы Spyder.



Рисунок 17 – Интерфейс программы работы с QR кодами

4. Для остановки выполняемой программы нажмите любую клавишу. Сделайте и запомните скриншот изображения. Время выполнения функции распознавания кода, дешифрованный код и название файла с изображением занесите в таблицу 1.

Таблица 1 – К регистрации дешифрованных QR кодов

| Название файла | Дешифрированные | Время выполнения | Примечание |
|----------------|-----------------|------------------|------------|
| | QR коды | | |
| | | | |
| | | | |
| | | | |
| | | | |
| | | | |

5. Измените в тексте программы имя загружаемого файла с изображением. Выполните п.п. 3, 4 для всех тестовых изображений, входящих в папку с лабораторной работой.

III. Работа с изображением QR кода, самостоятельно полученного с использованием видеокамеры.

1. Загрузите программу zbar_camera.py в основное окно среды разработки:

- в меню Файл выберите команду Открыть;
- в появившемся окне выберите Рабочий стол;
- в открывшемся списке выберите папку LR_QR_code_recognition;
- в открывшемся списке выберите файл zbar_camera.py.

2. Запустите программу, нажав на зеленый треугольник. Убедитесь, что при запуске не возникает сообщений об ошибках. Убедитесь, что появилось окно с изображением с камеры.

3. Предъявите камере первое тестовое изображение (рис. 18). Убедитесь, что тестовое изображение обнаружено и расшифровано.



Рисунок 18 – Предъявление камере QR кода

Путем перемещения камеры получите тестовые изображения при условиях, приведенных в таблице 2.

4. Выполните п. 3 для всех остальных тестовых изображениях. Проанализируйте полученные результаты работы программы zbar_camera.py.

| Условия работы | Время | Результат | Примечания |
|-------------------------------------|-----------|-------------|------------|
| (расположение кода в поле зрения | обработки | обнаружения | (скриншот) |
| камеры) | | есть/нет | |
| Прямо по оптической оси | | | |
| камеры | | | |
| Прямо по оптической оси | | | |
| камеры при минимальном размере | | | |
| изображения кода | | | |
| Прямо по оптической оси | | | |
| камеры в условиях расфокусировки | | | |
| (максимально приближено к камере) | | | |
| При максимальном отклонении от | | | |
| оптической оси влево | | | |
| При повороте камеры вокруг | | | |
| оптической оси на 45 градусов | | | |
| При повороте камеры вокруг | | | |
| оптической оси на 45 градусов и при | | | |
| максимальном отклонении влево | | | |

Таблица 2 - К регистрации дешифрованных тестовых QR кодов

IV. Работа с изображением самостоятельно найденного QR кода с использованием пакета OpenCV.

1. Загрузите программу OpenCV_QR_camera.py в основное окно среды разработки:

- в меню Файл выберите команду Открыть;

- в появившемся окне выберите Рабочий стол;
- в открывшемся списке выберите папку LR_QR_code_recognition;
- в открывшемся списке выберите файл OpenCV_QR_camera.py.

2. Запустите программу, нажав на зеленый треугольник. Убедитесь, что при запуске не возникает сообщений об ошибках. Убедитесь, что появилось окно с изображением с камеры.

3. Предъявите камере первое тестовое изображение QR кода. Убедитесь, что тестовое изображение обнаружено и расшифровано.

Путем перемещения камеры получите тестовые изображения кодов при условиях работы, приведенных в таблице 3.

4. Выполните п. 3 еще раз при самом большом тестовом изображении.

5. Скопируйте все полученные скриншоты на собственный носитель для составления отчета.

| | D | | |
|----------------------------------|-----------|-------------|------------|
| Условия работы | Время | Результат | Примечания |
| (расположение кода в поле зрения | обработки | обнаружения | (скриншот) |
| камеры) | | есть/нет | |
| Прямо по оптической оси | | | |
| камеры | | | |
| Прямо по оптической оси | | | |
| камеры при минимальном размере | | | |
| изображения кода | | | |
| Прямо по оптической оси | | | |
| камеры в условиях расфокусировки | | | |
| (максимально приближено к | | | |
| камере) | | | |
| При максимальном отклонении от | | | |
| оптической оси влево | | | |
| При повороте камеры вокруг | | | |
| оптической оси на 45 градусов | | | |
| При повороте камеры вокруг | | | |
| оптической оси на 45 градусов и | | | |
| при максимальном отклонении | | | |
| влево | | | |

Таблица 3 - К регистрации дешифрованных самостоятельно найденных QR кодов

V. Распознавание самостоятельно сформированного QR кода.

1. С помощью любого браузера зайдите на сайт онлайн генератора QR кодов (например, <u>http://qrcoder.ru/</u> или <u>https://qrcode.tec-it.com/ru</u>).

2. Введите в поле генератора QR кодов имена и фамилии студентов, участвующих в лабораторной работе, а также номер группы и получите изображение QR кода.

3. Вновь запустите программу zbar camera.py. Расположите окна программ так, чтобы на экране были бы видны страница браузера, окно среды Spyder и окно с изображением с камеры (рис. 19).

4. Наведите камеру на сформированный QR код в браузере и убедитесь, что код прочитан и расшифрован. Сделайте скриншот полученного окна.

5. Скопируйте все полученные скриншоты на собственный носитель для составления отчета.

VI. Выключение лабораторной установки.

Обратитесь к преподавателю или лаборанту для проверки результатов и выключения компьютера.



Рисунок 19 – Пример расположения окон браузеров

Содержание отчета

1. Краткие сведения об исследуемых алгоритмах.

2. Результаты распознавания QR кодов в виде скриншотов и пояснений работы алгоритма (почему не было распознавания в каждом конкретном случае и что нужно предпринять для улучшения процесса распознавания).

3. Результаты в виде заполненных таблиц с анализом полученных данных.

4. Результаты должны быть представлены для изображений QR кодов, как полученных из файлов, так и снятых с помощью камеры при разных условиях съемки и при использовании двух разных методов распознавания (на основе библиотеки Zbar и на основе библиотеки OpenCV).

Контрольные вопросы

1. В чем особенности обнаружения и распознавания QR кодов на изображении?

2. Какова структура QR кода и по каким шаблонам организуется его поиск?

3. Опишите функции библиотеки Zbar, используемой для поиска и распознавания QR кодов на изображении.

4. Опишите функции библиотек пакета OpenCV для работы с QR кодами.

5. От чего зависит время обработки кода на изображении?

Лабораторная работа №3. Методы выделения и распознавания лиц

Цели работы

- 1. Изучить работу алгоритма выделения лиц на основе метода Виолы-Джонса
- 2. Изучить работу алгоритма распознавания лиц на основе сверточной нейронной сети.
- 3. Получить представление о работе в операционной системе Linux, в среде разработки Spyder.
- 4. Получить практические навыки использования языка программирования Python, модулей и пакетов OpenCV, Face Racognition и других.

Работа рассчитана на 2 часа в стандартном варианте или на 4 часа в расширенном варианте. В последнем случае преподаватель дает дополнительные задания, связанные с настройкой режимов работы программного обеспечения выделения и распознавания лиц.

Основные теоретические сведения

Выделение и распознавание лиц являются одними из наиболее востребованных функций видеоаналитики. Эти функции используются в системах охранного и технологического видеонаблюдения, системах контроля доступа, бизнес-аналитике и других применениях [21]. Выделение изображения лица на фотографии или в кадре является одной из самых распространенных задач распознавания объектов в системах технического зрения.

Выделение лиц на изображении реализуется в процессе обработки изображения с целью поиска в изображении одного или нескольких человеческих лиц с указанием их координат в изображении. Результатом этого процесса может являться база данных выделенных лиц с их координатами и небольшими фрагментами изображения, содержащими лица.

Распознавание лиц предполагает сравнение изображения лица с лицами, содержащимися в базе данных, которая формируется заранее и содержит данные лиц, которые требуется распознать. Как правило, сравниваются не сами изображения лиц (фотографии), а специальные шаблоны, которые содержат в себе набор характерных признаков и занимают значительно меньший объем, чем исходные изображения. Поэтому сравнение шаблонов производится быстрее, а ошибки сравнения значительно меньше.

Такие шаблоны стандартизированы и описаны в ГОСТ Р ИСО/МЭК 19794-5-2006 Автоматическая идентификация. Идентификация биометрическая. Форматы обмена биометрическими данными. Часть 5. Данные изображения лица [21]. Характерные признаки, содержащиеся в шаблонах, получают из геометрических параметров взаимного расположения некоторого количества характерных контрольных точек лица (рис. 20).



Рисунок 20 – Расположение контрольных точек лица в соответствии с ГОСТ Р ИСО/МЭК 19794-5-2006

Шаблоны создаются на основе 2D изображения или 3D модели лица. В рамках данной работы используются только 2D изображения.

Существуют методы распознавания лиц, основанные на сравнении геометрических признаков изображений лиц. В этом случае для получения шаблона достаточно одного изображения лица. Но такой метод будет чувствителен к изменению положения лица в кадре и изменению некоторых других признаков лица, например в силу изменения возраста. Для более устойчивого распознавания требуется введение нескольких шаблонов, соответствующих нескольких положениям лица или другим особенностям.

Наиболее часто для решения этой задачи используется метод Виолы-Джонса, разработанный в 2001 г. П. Виолой и М. Джонсом и базирующийся на системе признаков Хаара (предложены в 1909 году) [22, 23]. Эти признаки дают хорошее описание перепадов яркости в изображении (рис. 21).



Рисунок 21 – Каскады Хаара

Основная идея метода заключается в использовании скользящего окна, содержащему один из признаков Хаара. Это скользящее окно сканирует исследуемое изображение.

Теоретически, алгоритм Виолы-Джонса можно использовать для обнаружения в изображении любых объектов. Для этого требуется провести процедуру обучения алгоритма с использованием тестовых изображений нужного типа объектов. На рисунке 22 показан пример обучающих изображений лиц. Алгоритм обучения работает с изображениями в оттенках серого [22].



Рисунок 22 – Примеры изображений лиц, которые используются для обучения алгоритма выделения лиц
Применительно к задаче поиска и выделения изображений лиц процесс обучения с учителем выглядит так [23, 24]. Используется тестовая выборка, содержащая около 10 тысяч изображений лиц в оттенках серого.

В процессе процедуры обучения создается база данных из слабых классификаторов, для каждого классификатора установлены следующие параметры: тип признака Хаара, его положение внутри окна и значение порога.

Задача слабого классификатора – угадывать присутствие объекта в более, чем 50% случаев. С использованием процедуры обучения AdaBoost создается сильный классификатор, состоящий из слабых классификаторов. После проведенного обучения полученная база данных сохраняется в виде файла, обычно с расширением *.hml

Распознавание объектов производится для 11 масштабов изображения. Алгоритм сканирует исследуемое изображение, начиная с размера окна сканирования 24х24 и затем увеличивая масштаб в 1,25 раза. Для того, чтобы классификатор вынес решение о том, что в окне присутствует искомый шаблон, нужно, чтобы выполнилось условие сильного классификатора.

Базовый алгоритм Виолы-Джонса (далее базовый алгоритм) имеет ряд недостатков:

- длительное время работы алгоритма обучения. В ходе обучения алгоритму необходимо проанализировать большое количество тестовых изображений;
- большое количество близко расположенных друг к другу результатов из-за применения различных масштабов и скользящего окна.

В данной работе мы будем использовать базы каскадов Хаара, заранее подготовленные с помощью алгоритма Виолы-Джонса.

В составе пакета Open CV имеются функции детектирования объектов на изображении detectMultiScale(), которая использует базу признаков распознаваемых объектов на основе каскадов Хаара для каждого из типов объектов CascadeClassifier(), включая объект — лицо человека. Как правило, такие базы подключаются в виде файлов.

Как вариант, можно использовать распознаватель на основе каскадов Хаара и создать базу для каждого распознаваемого лица как систему признаков на основе алгоритма Виолы-Джонса, для этого может потребоваться некоторое количество изображений лица каждого из людей, входящих в базу данных. Распознавание при этом можно производить по максимальному совпадению признаков, которое можно оценить, например, с использованием метода SURF. Расширение базы данных лиц достигается простым добавлением нового шаблона.

Другая группа методов основана на использовании нейронных сетей и технологии глубокого обучения. В этом случае используется сверточная нейронная сеть. Для обучения используются изображения лиц, которые впоследствии предстоит распознать. В этом случае добавление нового лица в базу потребует процесса обучения сети.

В данной работе мы воспользуемся модулем Face recognition, который в свою очередь базируется на библиотеке Dlib. Эта библиотека содержит алгоритмы машинного обучения и различные инструменты для работы с изображениями лиц. На Dlib распространяется тип лицензии <u>Boost</u>, позволяющий использовать ее в коммерческих проектах, что способствовало ее широкому распространению.

Как следует из описания модуля Face Recognition, он является наиболее простым программным средством для распознавания лиц. Использование глубокого обучения на основе библиотеки Dlib позволило достигнуть вероятности распознавания более 99% на общедоступных тестах для проверки лиц Labeled Faces in the Wild, известных как парное сопоставление. В библиотеке Dlib присутствует заранее обученный под задачу распознавания лиц алгоритм на основе сверточной нейронной сети ResNet. Используется одна ИЗ многочисленных модификаций ResNet. которая была натренирована по датасету, (набору данных), содержащему около 3 млн. изображений лиц.

Непосредственно перед процессом распознавания в модуле Face Recognition производится поиск изображений лиц с использованием контрольных точек. В целом алгоритм поиска похож на алгоритм, используемый в пакете OpenCV, но используется другой файл классификатора на базе алгоритма Виолы-Джонса.

Порядок выполнения работы

Работа включает в себя следующие разделы:

- включение лабораторной установки и подготовка ее к работе;
- исследование алгоритма обнаружения лиц в среде Python на основе файлов изображений из рабочей папки;
- исследование алгоритма распознавания лиц в среде Python на основе файлов изображений из рабочей папки;
- исследование алгоритмов обнаружения и распознавания лиц в среде Python на основе файлов самостоятельно созданных изображений;
- выключение лабораторной установки.

І. Работа выполняется на лабораторной установке, состоящей из персонального компьютера с установленной операционной системой Linux (Ubuntu 20.04 или старше) с подключенной к нему usb-камерой. Подготовка лабораторной установки содержит следующие этапы:

1. С помощью преподавателя или лаборанта включите компьютер и дождитесь загрузки операционной системы.

2. На рабочем столе найдите папку LR, зайдите в нее и скопируйте папку LR_face_recognition на Рабочий стол. В дальнейшем все действия,

предусмотренные порядком выполнения лабораторной работы, выполняйте в этой папке.

3. Пользуясь боковой панелью (см. рис.3), запустите среду разработки Spyder и дождитесь ее загрузки.

II. Изучение алгоритмов обнаружения лиц в среде Python.

- 1. Загрузите программу face_detection.ру в основное окно среды разработки:
 - в меню Файл выберите команду Открыть;
 - в появившемся окне выберите Рабочий стол;
 - в открывшемся списке выберите папку LR face recognition;
 - в открывшемся списке выберите файл face_detection.py.

2. В основном окне среды разработки Spyder появится текст программы (рис. 23). Ознакомьтесь с текстом программы. Пользуясь комментариями в тексте, изучите назначение основных операторов и функций, которые используются в этой программе.



Рисунок 23 – Окно среды разработки Spyder

3. Запустите программу, нажав на зеленый треугольник в верхней части окна Spyder. В случае появления ошибок в окне **Консоль IPython** обратитесь к преподавателю или лаборанту. В большинстве случаев ошибка связана с

неправильным подключением камеры. Обычно номер камеры определяется командой:

cap = cv2.VideoCapture(0)

В некоторых случаях следует поменять номер камеры с 0 но 1.

После запуска программы может появиться небольшое окно, предлагающее открыть всплывающее окно, но обычно сразу выводится изображение с камеры.

4. Наведите камеру на группу студентов, участвующих в выполнении лабораторной работы. В случае обнаружения лиц программа выделит каждое обнаруженное лицо зеленым прямоугольником (рис. 24). Добейтесь обнаружения всех лиц путем изменения положения студентов в поле зрения камеры, дистанции до камеры, подстройки фокусировки камеры, если такая регулировка есть.



Рисунок 24 – Пример выделения обнаруженных на изображении лиц

5. Проведите исследование зависимости эффективности работы алгоритма от положения лица в кадре. При этом лицо следует отклонять в трех направлениях (по горизонтали, вертикали и наклону), как показано на рисунке 25. Зафиксируйте предельные значения отклонений для каждого студента, участвующего в лабораторной работе. Для иллюстрации результатов можно пользоваться скриншотами.





В Ubuntu имеется возможность получить скриншот не всего экрана, а интересующей области Для этого необходимо воспользоваться комбинацией клавиш Ctrl + Shift + PrintScreen, затем после появления изображения курсора в виде креста выделить нужную область мышкой.

Скопированное изображение можно вставить во временный файл, открытый с помощью LibreOffice Writer (аналог редактора Word). Запустить его можно с боковой панели Ubuntu.

6. Исследуйте зависимость работоспособности алгоритма от величины объекта (лица). Для этого следует постепенно отодвигаться (отходить) от камеры и добиться такого положения, при котором процесс выделения лица станет неустойчивым. Выполните указанную процедуру для каждого студента, участвующего в выполнении лабораторной работы.

7. По полученным изображениям рассчитайте минимально необходимый размер изображения лица в пикселах для каждого студента, зная формат изображения, получаемого с камеры. Этот параметр уточните у преподавателя или лаборанта, но как правило, камеры для данной лабораторной работы используется с форматом кадра 4х3 (с разрешением камеры 640 x 480).

8. Исследуйте зависимость работоспособности алгоритма от качества оптической системы (степени фокусировки объектива). Для этого следует всем студентам, участвующим в лабораторной работе, расположиться перед камерой

таким образом, чтобы изображения их лиц занимали по вертикали около 1/5 от размера кадра. Сначала следует хорошо сфокусировать камеру, используя регулировку объектива. Убедиться, что все лица обнаруживаются, и сделать скриншот. Затем, медленно вращая объектив, добиться неустойчивого выделения лиц в кадре. В таком положении также сохранить скриншот.

9. На этом исследование алгоритма обнаружения лиц можно завершить. Выход из программы производится путем нажатия клавиши Esc.

III. Исследование алгоритма распознавания лиц производится с помощью программы test_face_recognition_from_camera.py.

1. Загрузите программу test_face_recognition _from_camera.py в основное окно среды разработки:

- в меню Файл выберите команду Открыть;

- в появившемся окне выберите Рабочий стол;
- в открывшемся списке выберите папку LR face recognition;
- в открывшемся списке выберите файл test_face_recognition _from_camera.py.

2. В основном окне среды разработки Spyder появится текст программы. Ознакомьтесь с ним (рис. 26).



Рисунок 26 – Окно среды разработки Spyder

Пользуясь комментариями в тексте, изучите назначение основных операторов и функций, которые используются в этой программе. Проверьте, что номер камеры тот же самый, что и в предыдущей программе.

3. Запустите программу, нажав на зеленый треугольник. При этом вы увидите изображение, получаемое с камеры. Переместите это окно в правую часть экрана монитора.

Найдите на рабочем столе вашу папку с лабораторной работой LR_face_recognition, которую вы скопировали на рабочий стол. Откройте эту папку и изучите ее содержимое (рис. 27).

| Рабочий стол | LR LR_face_re | ecognition 🔻 | Q | III III | | 0 |
|---|-------------------|--------------|-----------|---------------------------------------|-----------------------------|------------------------------------|
| Недавние Избранные | | | 200 | and the second | SAS | P |
| 🕜 Домашняя папка | cascade_ files | 1.png | 2.jpg | 3.jpg | 4.jpg | ad1.jpg |
| 🔲 Рабочий стол | N | 2 | B | - | | |
| 🗏 Видео | ad2.jpg | ad3.jpg | ad4.jpg | face_ | gd1.jpg | gd2.jpeg |
| 📄 Документы | | | | detection. | | |
| ∄ Загрузки | 9 | | State . | | | |
| Изображения | gd3.jpg | gd4.jpg | gpb1.jpg | gpb2.jpg | gpb3.jpg | gpb4.jpg |
| јј музыка | 9 | | 2 | A | 2 | |
| | ar1.ipg | | ar3.ipg | ar4.ipg | Lf1.ipg | lf2.ipg |
| + Другие места | lf3.jpg | lf4.jpg | my_image. | oh1.jpg | oh2.jpg | oh3.jpg |
| | oh4.jpg | pr1.jpg | jpg | pr3.jpg | pr4.jpeg | sl1.jpg |
| | sl2.jpg | sl3.jpg | sl4.jpg | test_face_ recognition _from_ca | лр Выделени е и распо | .~lock.ЛР Выделени е и распо |
| | | | | | | |

Рисунок 27 – Содержимое папки для изучения алгоритма распознавания лиц

В этой папке, помимо исходных текстов программ, которые вы запускаете, и файла с описанием лабораторной работы, вы найдете изображения лиц людей (известных зарубежных киноартистов), сгруппированных по названию. Для каждого из этих людей имеется по четыре фотографии, которые можно использовать для изучения процесса распознавания лиц.

Обратите внимание, что в тексте программы есть этап формирования шаблонов для выбранных людей, например, для изображения Одри Хепберн — это файл с изображением oh1.jpg.

4. Откройте первое изображение с помощью программы просмотра фотографий. Расположите открывшееся окно в левой части экрана. Настройте положение камеры таким образом, чтобы открывающееся окно с фотографией целиком бы вписывалось в изображение, формируемое камерой.

В случае распознания лица данной программой поверх изображения будет присутствовать красный прямоугольник. Внизу прямоугольника будет присутствовать надпись с фамилией или именем распознанного человека. Если человек не распознан или отсутствует в базе данных, то появится надпись Unknown (рис. 28).



Рисунок 28 – Пример распознания лица программой

Результат распознавания может быть четырех видов: правильное распознавание, неправильное распознавание, неизвестное лицо, лицо в кадре не обнаружено.

5. Занесите результаты распознавания лиц в таблицу 4, где имя файла состоит из двух или трех букв, а номер файла — цифра перед расширением .jpg:

| Начало имени | Номер файла 1 | Номер файла 2 | Номер файла 3 | Номер файла 4 |
|---------------|---------------|---------------|---------------|---------------|
| файла/человек | | | | |
| oh | | | | |
| sl | правильно | | | |
| pr | | | | |
| lf | | неправильно | | |
| gr | | | неизвестное | |
| gd | | | | не обнаружено |
| ad | | | | не обнаружено |
| gpb | | | | |

Таблица 4 – Пример внесения результатов распознавания лиц

6. Обратите внимание, что два человека являются для программы неизвестными. В таблице 3 это шестой и седьмой файлы с названиями gd и ad соответственно. Их лица отсутствуют в базе данных.

Внесите этих людей в базу данных. Для этого необходимо создать шаблон на основе фотографии человека. Остановите программу, нажав на клавишу Q. Найдите в тексте программы фрагмент создания шаблона для шестого человека:

Загрузка эталонного изображения для шестого человека depardie_image = face_recognition.load_image_file("gd1.jpg") # Вычисление шаблона для шестого изображения depardie_face_encoding = face_recognition.face_encodings(depardie_image)[0]

Напишите или частично скопируйте этот фрагмент для седьмого человека. Изменения выделены **жирным шрифтом**:

Загрузка эталонного изображения для **седьмого** человека **delon_**image = face_recognition.load_image_file("**ad1.jpg**") # Вычисление шаблона для шестого изображения **delon_**face_encoding = face_recognition.face_encodings(**delon_**image)[0]

Теперь шаблон лица седьмого человека также внесен в базу данных. Найдите фрагмент кода с указанием списка имен людей в базе данных:

Построение матрицы известных изображений лиц и соответствующих им имен

known_face_encodings = [
 odri_face_encoding,
 loren_face_encoding,
 rishar_face_encoding,
 funes_face_encoding,
 reno_face_encoding,

```
depardie_face_encoding

]

known_face_names = [

"Hepburn",

"Loren",

"Rishar",

"Funes",

"Reno",

"Depardie"

]
```

Внесите в этот список нового человека. Изменения выделены жирным шрифтом:

Построение матрицы известных изображений лиц и соответствующих им имен

```
known_face_encodings = [
  odri_face_encoding,
  loren_face_encoding,
  rishar_face_encoding,
  funes face encoding,
  reno_face_encoding,
  depardie_face_encoding,
  delon_face_encoding
1
known_face_names = [
  "Hepburn",
  "Loren".
  "Rishar".
  "Funes".
  "Reno",
  "Depardie",
  "Delon"
1
```

Теперь после запуска программы будет сформирован шаблон и для седьмого человека.

7. Выполните внесение в базу данных восьмого человека, изображения которого хранятся в файлах gpb1.jpg — gpb4.jpg. Для этого выполните действия, аналогичные п. б., для восьмого человека. Запустите программу и убедитесь, что она работает.

8. Проделайте п.п. 4, 5 еще раз и заполните таблицу 4 уже для восьми людей, внесенных в базу данных.

9. (Дополнительный) Выполните п.п. 4, 5 с другими шаблонами лиц. Для этого в тексте программы измените все файлы фотографий для шаблонов с номера 1 на номер 2 (например, ad1.jpg на ad2.jpg). Запустите программу и заполните таблицу 4.

Проделайте то же самое для файлов с номерами 3 и 4.

10. Сделайте скриншоты с распознанными изображениями лиц для отчета. Остановите работу программы, нажав на кнопку Q.

IV. Исследование алгоритмов обнаружения и распознавания лиц на самостоятельно полученных изображениях.

1. Для этого следует предварительно получить фотографии с изображениями лиц студентов, занятых в лабораторной работе, и сформировать базу данных. Загрузите программу face_detection (по п.п. 1, 2, раздел **III**) и запустите ее. Появится окно изображения с камеры.

Каждому студенту следует расположиться перед камерой таким образом, чтобы изображение его лица составляло бы примерно 1/3 от вертикального формата камеры. Изображение сохраняется в файле my_image.jpg, который размещается в скопированной папке с лабораторной работой в момент нажатия клавиши Esc выхода из программы.

Чтобы сформировать изображения всех студентов, участвующих в лабораторной работе, следует переименовать полученный файл my_image.jpg и вновь запустить программу face_detection.py.

2. Теперь следует внести полученные файлы изображений для формирования шаблона и внести имена студентов в список по аналогии с действиями из п. 6, раздел **III**.

3. Запустите программу и убедитесь, что в открывшемся окне при попадании студентов в поле зрения камеры их лица обнаруживаются и распознаются.

Сделайте скриншоты с распознанными изображениями лиц всех студентов (как раздельно каждого студента, так и всех студентов сразу).

4. Скопируйте все полученные скриншоты на собственный носитель для составления отчета. Обратитесь к преподавателю или лаборанту для выключения компьютера.

V. Выключение лабораторной установки.

Обратитесь к преподавателю или лаборанту для проверки результатов и выключения компьютера.

Содержание отчета

1. Краткие сведения об исследуемых алгоритмах.

2. Результаты обнаружения и распознавания лиц в виде скриншотов и пояснений особенностей работы алгоритма (почему не было распознавания в каждом конкретном случае и что нужно предпринять для улучшения процесса распознавания).

3. Результаты распознавания лиц из предварительно созданной базы данных и фотографий лиц. Результаты должны быть в виде заполненных таблиц и наиболее характерных скриншотов с результатами распознавания.

4. Результаты распознавания лиц студентов после занесения их фотографий в базу данных с наиболее характерными скриншотами и пояснениями особенностей работы алгоритма распознавания.

Контрольные вопросы

1. Что такое выделение и распознавание лиц на изображении? Чем эти процессы отличаются друг от друга?

2. Что такое шаблон для распознавания лица и как он формируется?

3. Какие существуют алгоритмы распознавания лиц и на каких методах они основаны?

4. Каковы функциональные особенности работы модуля Face Recognition на основе библиотеки dlib?

5. Каковы функции библиотек OpenCV для распознавания лиц и какие алгоритмы для работы с изображениями лиц они содержат?

6. От каких факторов зависит результат работы модуля распознавания лиц?

Лабораторная работа №4. Методы обнаружения и распознавания текста в изображении

Цели работы

- 1. Изучить методы обнаружения и распознавания изображений текста на основе открытых библиотек обработки изображений.
- 2. Получить представление о работе в операционной системе Linux, в среде разработки Spyder.
- 3. Получить практические навыки использования языка программирования Python, модулей и пакетов OpenCV, Tesseract, Easyocr и других.

Работа рассчитана на 2 часа в стандартном варианте или на 4 часа в расширенном варианте. В последнем случае преподаватель дает дополнительные задания, связанные с настройкой режимов работы программного обеспечения выделения и распознавания текста в изображении.

Основные теоретические сведения

Выделение (на основе поиска и обнаружения) и распознавание текста является одной из классических задач технического зрения. Эта технология широко известна на примере обработки отсканированных документов и приведения их от изображения к текстовому формату. Однако в настоящее время эта технология широко востребована в системах видеонаблюдения и технологического телевидения. В частности, на ней основаны системы распознавания номеров автомобилей, железнодорожных вагонов, контейнеров в системах транспортной безопасности и логистики, распознавания номеров банковских купюр в банковских системах, чтения дорожной информации в системах управления беспилотными автомобилями.

Как и во многих подобных системах, эта технология содержит два этапа [25]. Этап поиска, обнаружения и выделения текста в изображении основан на сканировании изображения и обнаружении его участков, содержащих текст.

Этап распознавания изображения текста включает в себя процесс сегментации изображения на отдельные изображения букв и слов. Затем подключаются алгоритмы распознавания отдельных символов с использованием баз данных алфавитов и шрифтов. Далее выполняется декодирование текстовой информации путем совместной обработки полученных символов, объединения их в слова и проверки текста на уровне слов с применением словарной базы данных, а также проверки правильности использования слов в предложении на базе правил того или иного языка.

Процесс поиска участков изображения, содержащих текст, а также процесс выделения и распознавания отдельных символов производится методами глубокого обучения на основе сверточной нейронной сети.

Процесс обработки текста на уровне слов и предложений производится также на основе нейронных сетей и глубокого обучения, но в этом случае используется рекуррентная нейронная сеть, а именно ее разновидность LTSM. Долгая краткосрочная память (Long short-term memory; LSTM) – особая разновидность архитектуры рекуррентных нейронных сетей, способная к обучению долговременным зависимостям [25, 26].

Библиотека Tesseract

С точки зрения информационных технологий обработки изображения текст представляет собой упорядоченный набор изображений отдельных символов, число которых ограничено тем или иным алфавитом или другим набором данных (иероглифами). При этом совместное использование этих символов также ограничено и подчиняется определенным правилам того или иного языка. Таким образом, общее решение задачи распознавания текста может быть сведено к последовательности следующих действий:

- предварительная обработка изображения с целью повысить обнаружительную способность алгоритмов поиска текста;
- обнаружение на изображении фрагментов, содержащих текст;
- разделение обнаруженного текста на отдельные символы или буквы;
- распознавание отдельных символов или букв с использованием баз данных алфавитов и шрифтов;
- объединение распознанных символов в группы (слова) на основе баз данных словарей;
- объединение слов в предложения на основе базы данных правил языка;
- по результатам последних двух пунктов выполнение проверки правильности распознавания символов и, при необходимости, повторное распознавание сомнительного символа с его заменой в соответствии с правилами написания слов и предложений конкретного языка.

Подобная технология впервые успешно использована в коммерческих применениях фирмой Hewlett Packard и использована в программном обеспечении сканеров в середине 1980-х годов. В дальнейшем это программное обеспечение стало открытым и в настоящее время известно под названием **Tesseract**. Библиотека Tesseract с открытым исходным кодом написана на языке C и доработана на языке C++.

В настоящее время имеется много фреймворков, которые позволяют использовать эту библиотеку в составе программ, написанных и на других языках программирования. В частности, в данной лабораторной работе используется программа, написанная на языке Python. Поддержка библиотеки Tesseract, которая устанавливается независимо от языка Python, обеспечивается с помощью модуля **pytesseract**. Так как библиотека свободно распространяется, любой пользователь может в своих разработках использовать эту библиотеку [25-27].

По умолчанию библиотека Tesseract использует английский алфавит и словарь. Однако при необходимости можно легко подключить более ста дополнительных алфавитов и языков. Реализуется также поддержка русского алфавита и языка.

Поскольку исходный вариант библиотеки Tesseract предназначен прежде всего для распознавания текстовых документов, то по умолчанию без дополнительных настроек библиотека хорошо распознает традиционный машинописный текст на белом фоне, например, данный текст методических указаний к лабораторной работе. Это связано с тем, что такой текст содержит стандартный набор шрифтов, качество изображения с текстом достаточно высокое (достаточное разрешение, высокий контраст изображения и высокое отношение сигнал/шум), а также в полной мере используются возможности проверки текста с использованием словаря и базы данных правил языка. В этом легко убедиться, выполнив далее пункты данной лабораторной работы.

Но в случае других применений возможны затруднения с использованием данной библиотеки. При использовании технологии распознавания текста на реальных изображениях с камеры могут проявлять себя следующие факторы, снижающие эффективность процесса распознавания текста:

- ограниченное разрешение изображения в зоне обнаружения и распознавания текста, усугубленное процессом расфокусировки и возможного смаза вследствие движения объекта или дрожания камеры;
- ограниченное отношение сигнал/шум, особенно в условиях недостаточной освещенности;
- ограниченный контраст изображения вследствие загрязнения поверхности объекта, плохих метеорологических условий наблюдения, недостаточного освещения;
- пространственное расположение поверхности с текстом или под произвольным углом к оптической оси камеры (по горизонтали, вертикали или вокруг оптической оси) или на криволинейной поверхности;
- нанесение распознаваемого текста нестандартными шрифтами (с изменяемыми размерами букв и символов), граффити;
- использование неклассического (жаргонного) написания слов, не предусмотренных словарем;
- несоответствие набора символов словам (серийные номера, торговые идентификаторы, номера автомобилей, вагонов, контейнеров и т. п.).

В библиотеке Tesseract имеется ряд параметров, которые можно настроить под конкретное применение. Последние версии библиотеки включают в себя

языковые модели и словари, которые построены на основе предварительно обученных нейронных сетей.

Предусмотрена также возможность самостоятельного обучения или дообучения языковых моделей дополнительным символам. При этом можно создать свои собственные базы словарей и символов под конкретные нужды. Для этого в библиотеке имеются соответствующие утилиты. Однако это требует больших усилий при подготовке базы данных и последующего процесса обучения. Поэтому такой метод рекомендуется в том случае, когда не удается достичь приемлемого результата распознавания с использованием стандартного набора настроек.

Рассмотрим библиотеку Tesseract версии 4.1.1, которая имеет следующие варианты настроек:

- 1. По ключу -psm выставляется режим сегментации обнаруженных фрагментов текста *Page segmentation modes*:
 - 0 Orientation and script detection (OSD) only.
 - 1 Automatic page segmentation with OSD.
 - 2 Automatic page segmentation, but no OSD, or OCR. (not implemented)
 - 3 Fully automatic page segmentation, but no OSD. (Default)
 - 4 Assume a single column of text of variable sizes.
 - 5 Assume a single uniform block of vertically aligned text.
 - 6 Assume a single uniform block of text.
 - 7 Treat the image as a single text line.
 - 8 Treat the image as a single word.
 - 9 Treat the image as a single word in a circle.
 - 10 Treat the image as a single character.
 - 11 Sparse text. Find as much text as possible in no particular order.
 - 12 Sparse text with OSD.
 - 13 Raw line. Treat the image as a single text line, bypassing hacks that are Tesseract-specific.
- 2. По ключу -оет устанавливается основной алгоритм детектирования *OCR Engine modes*:
 - 0 Legacy engine only.
 - 1 Neural nets LSTM engine only.
 - 2 Legacy + LSTM engines.
 - 3 Default, based on what is available.

Кроме того, специальными ключами можно установить путь к пользовательскому словарю, пользовательскому набору символов, указать величину символов в DPI, указать используемый язык. Для более точной настройки можно использовать параметры файла конфигурации, в котором имеется несколько сотен параметров. Список этих параметров можно посмотреть в файле, имеющемся в папке к лабораторной работе.

Порядок выполнения работы

Работа включает в себя следующие разделы:

- включение лабораторной установки и подготовка ее к работе;
- изучение работы алгоритмов обнаружения и распознавания изображений текста в среде Python;
- изучение настроек алгоритмов обнаружения и распознавания изображений текста в среде Python;
- работа с символьной информацией при выделении и распознавании номеров автомобилей с использованием библиотеки Tesseract;
- работа с самостоятельно найденным изображением с номером автомобиля и его распознавание с использованием библиотеки Tesseract;
- выключение лабораторной установки.

І. Работа выполняется на лабораторной установке, состоящей из персонального компьютера с установленной операционной системой Linux (Ubuntu 20.04 или старше) с подключенной к нему usb-камерой. В работе используются отпечатанные изображения, содержащие текстовую и символьную информацию.

1. С помощью преподавателя или лаборанта включите компьютер и дождитесь загрузки операционной системы.

2. На рабочем столе найдите папку **LR**, зайдите в нее и скопируйте папку **LR_text_recognition** на Рабочий стол. В дальнейшем все действия, предусмотренные порядком выполнения лабораторной работы, выполняйте в этой папке.

3. Пользуясь боковой панелью (см. рис. 3), запустите среду разработки Spyder и дождитесь ее загрузки.

II. Изучение работы алгоритмов обнаружения и распознавания изображений текста в среде Python.

1. Загрузите программу tesseract_text.py в основное окно среды разработки:

- в меню Файл выберите команду Открыть;
- и появившемся окне выберите Рабочий стол;
- в открывшемся списке выберите папку LR_text_recognition;
- в открывшемся списке выберите файл tesseract_text.py.

2. В основном окне среды разработки Spyder появится текст программы. Ознакомьтесь с текстом программы (рис. 29). Пользуясь комментариями в тексте, изучите назначение основных операторов и функций, которые используются в этой программе.



Рисунок 29 – Окно среды разработки

3. В загруженной программе есть строки, которые используются для введения изображения из файла (см. текст программы на рис. 29, начиная с 12 строки):

Подключение фото img = cv2.imread('engtext.jpg') #img = cv2.imread('rustext.jpg')

В данном случае вводится файл engtext.jpg, в котором присутствует картинка, содержащая английский текст. Проверьте, что вводится именно этот файл, а следующая строчка программы закомментирована.

4. Запустите программу, нажав на зеленый треугольник. Будет выведено окно с картинкой, на которой выделены строки текста. В окне монитора программы Spyder будет выведен распознанный текст (рис. 30).



Рисунок 30 – Пример распознавания текста на изображении

5. Сделайте и сохраните для отчета скриншоты окон с картинкой и с распознанным текстом. Остановите программу, нажав клавишу Esc.

Ш. Изучение настроек алгоритмов обнаружения и распознавания изображений неанглоязычного текста в среде Python.

1. Подключите картинку с русским текстом. Для этого в тексте программы закомментируйте строчку:

img = *cv2.imread*(*'engtext.jpg'*) и откройте строчку:

#img = cv2.imread('rustext.jpg')

2. Запустите программу, нажав на зеленый треугольник. Вы увидите окно с картинкой, на которой выделены строки текста. В окне монитора программы Spyder также будет выведен распознанный текст, но с использованием английского алфавита.

3. Сделайте скриншоты окна с картинкой и с результатами распознавания текста. Остановите программу, нажав клавишу Esc.

Как вы уже поняли, Tesseract по умолчанию работает с английским текстом.

4. Выполните настройку Tesseract под русский текст. Для этого в тексте программы закомментируйте строчку (рис. 30):

config = *r'--оет 3 --рsm 6 -l еng'* и откройте строчку: #config = r'--oem 3 --psm 6 -l rus'

5. Запустите программу, нажав на зеленый треугольник. Вы увидите окно с картинкой, на которой выделены строки текста. В окне монитора программы Spyder также будет выведен распознанный текст, но уже на русском языке.

6. Сделайте и сохраните для отчета скриншоты окна с картинкой и с распознанным текстом. Остановите программу, нажав клавишу Esc.

7. Аналогично действиям по п.4 данного раздела работы подключите картинку с текстом на двух языках — rusengtext.jpg, отредактировав строчку ввода файла. Запустите программу. Посмотрите результат распознавания.

8. Пользуясь описанием команд Tesseract из файла «Все команды Tesseract.pdf», добейтесь наилучшего распознавания двуязычного текста. Сделайте и сохраните скриншоты окна с картинкой и с распознанным текстом.

IV. Выделение и распознавание номеров автомобилей.

1. Сделаем попытку выполнить распознавание номеров автомобилей с помощью Tesseract. Для этого введем в тексте программы картинку с изображением автомобиля, например 708.jpg. Посмотрите результат распознавания. Сделайте скриншоты окна с картинкой и с распознанным текстом. Остановите программу, нажав клавишу Esc. Объясните результат работы программы в стандартном режиме.

2. Загрузите программу tesseract_plates.py в основное окно среды разработки:

- в меню Файл выберите команду Открыть;

- в появившемся окне выберите Рабочий стол;
- в открывшемся списке выберите папку LR_text_recognition;
- в открывшемся списке выберите файл tesseract_plates.py.

3. В основном окне среды разработки Spyder появится текст программы. Ознакомьтесь с текстом программы. Пользуясь комментариями в тексте, изучите назначение основных операторов и функций, которые используются в этой программе. В данной программе предусмотрено два независимых режима.

Сначала в программе производится поиск номера автомобиля с использованием алгоритма Виолы-Джонса. Для этого потребовалось загрузить файл каскадного классификатора Xaapa haarcascade_russian_plate_number2.xml, выполненного специально для поиска российских автомобильных номеров. В результате работы этого алгоритма была выделена зона номера, а затем эта зона была сохранена в виде отдельного изображения.

На втором этапе с помощью библиотеки Tesseract изображение номера распознано как текст со специально введенными ограничениями.

4. Запустите программу, нажав на зеленый треугольник. Вы увидите несколько окон с картинкой, на которой выделены строки текста (рис. 31). В окне монитора программы Spyder также будет выведен распознанный текст номера.

5. Сделайте скриншоты окон с картинкой и с распознанным текстом. Остановите программу, нажав клавишу Esc.



Рисунок 31 – Распознавание автомобильного номера

V. Выделение и распознавание номеров автомобилей на самостоятельно найденных изображениях.

1. Выполните п.п. 3-5 предыдущего раздела работы для другого изображения car_image.png. Для этого загрузите программу tesseract_plates_camera.py в основное окно среды разработки:

- в меню Файл выберите команду Открыть;
- в появившемся окне выберите Рабочий стол;
- в открывшемся списке выберите папку LR_text_recognition;
- в открывшемся списке выберите файл tesseract_plates_camera.py.

2. В основном окне среды разработки Spyder появится текст программы. Ознакомьтесь с текстом программы. Пользуясь комментариями в тексте, изучите назначение основных операторов и функций, которые используются в этой программе. В основном эта программа аналогична предыдущей, но здесь предусмотрено получение изображения с камеры с последующим выделением номера и его распознаванием.

3. Откройте окно браузера и в поисковой системе Яндекс наберите «автомобиль с номером», откройте найденное изображение в режиме «картинки». Расположите окно с браузером таким образом, чтобы на мониторе компьютера были бы видно одновременно окна браузера и Spyder с открытой программой.

4. Направьте камеру на открытое окно браузера с картинкой автомобиля.

5. Запустите программу, нажав на зеленый треугольник. При запуске предусмотрена пятисекундная задержка для выхода камеры в рабочий режим. Возможно, при неудачном направлении визирования, программа запустится не с первого раза.

Полученное изображения с камеры автомобиля с обнаруженным номером, а также обнаруженный номер отобразятся в открывшихся окнах. Распознанный номер, а также время обнаружения и распознавания будут отображаться в окне терминала (рис. 32).



Рисунок 32 – Результат работы с самостоятельно найденным изображением

6. Немного подвигайте камеру, имитируя движение, получите и рассмотрите результат распознавания. Для остановки работы программы нажмите Esc. Посмотрите также результат в остановившемся терминале. Сделайте основные скриншоты.

7. Измените картинку в браузере и вновь запустите программу. Выполните действия по п.п. 4-6 для новой картинки.

8. Рассмотрите еще раз текст программы. Поясните основные этапы ее работы. Назовите основные недостатки программы. Предложите пути их устранения.

9. Скопируйте все полученные скриншоты на собственный носитель для составления отчета. Обратитесь к преподавателю или лаборанту для выключения компьютера.

VI. Выключение лабораторной установки.

Обратитесь к преподавателю или лаборанту для проверки результатов и выключения компьютера.

Содержание отчета

1. Краткие сведения об исследуемых алгоритмах.

2. Результаты распознавания текста из изображений. Описание особенностей работы программы и путей ее улучшения.

3. Результаты обнаружения номеров в виде скриншотов и пояснений особенностей работы алгоритма (почему не было распознавания в каждом конкретном случае и что нужно предпринять для улучшения процесса распознавания).

4. Анализ работы программы обнаружения и распознавания номеров с камеры. Поясните основные этапы ее работы. Назовите основные недостатки программы. Предложите пути их устранения.

Контрольные вопросы

1. В чем состоит процедура распознавания текстовой информации на изображении?

2. Какие методы используются для обработки текстов на уровне слов и предложений?

3. Какие функции библиотеки Tesseract позволяют ее использовать для распознавания текстов без дополнительных настроек?

4. В чем особенности распознавания текстовой информации на изображениях от камер систем видеонаблюдения?

5. Какие настройки библиотеки Tesseract используются для работы с реальными изображениями?

Лабораторная работа №5. Методы улучшения субъективного восприятия изображений в системе ТАЙФУН

Цели работы

- 1. Изучить структуру телевизионной системы безопасности на базе ПО «Тайфун», ее составных частей и интерфейса.
- 2. Изучить методы улучшения качества изображения в видеоинформационных системах безопасности.
- 3. Получить практические навыки применения приемов работы, настройки оборудования и определения основных характеристик.

Основные теоретические сведения

Работа систем видеонаблюдения и видеорегистрации применительно к системам безопасности характерна тем, что условия работы видеокамер могут быть далеко не идеальны. В этом случае требуется повышение качества изображения, сформированного камерой, для улучшения его субъективного восприятия. Так как видеосигнал на выходе камеры уже не может быть улучшен, то требуется применение методов посткоррекции изображений перед их выводом на монитор оператора. Возможны методы аппаратной и программной коррекции качества изображения.

К аппаратным можно отнести методы, при которых используются аппаратные средства, входящие в тракт обработки видеосигнала. В частности, речь идет о подстройке видеосигнала в видеоусилителе перед его аналогоцифровым преобразованием. Обычно для черно-белого видеосигнала используется три основных регулировки:

- регулировка яркости, которая сводится к изменению постоянной составляющей видеосигнала,

- регулировка контрастности, при которой изменяется коэффициент усиления видеосигнала,

- регулировки гамма-коррекции, при которой корректируется градационная характеристика.

К программным средствам коррекции относятся методы обработки уже оцифрованного видеосигнала. Чаще всего используется коррекция цифрового видеосигнала на основе построения гистограммы вероятностей распределения видеосигнала по уровням квантования.

Телевизионная система безопасности на базе ПО «Тайфун»

В данной лабораторной работе изучение методов повышения качества изображения в видеоинформационных системах безопасности выполняется с

использованием Телевизионной системы оперативного наблюдения и регистрации (ТСНР) на базе ПО «Тайфун».

ТСНР представляет собой аппаратно-программный комплекс для прямого видеонаблюдения, а также записи и воспроизведения видеосигнала от одного или многих источников видеосигналов с одновременной их записью и возможностью воспроизведения записанных видеосигналов на одном или нескольких автоматизированных рабочих местах (АРМ) операторов.

ПО «Тайфун» – демонстрационная версия коммерческой разработки компании ООО «ЭВС». В отличие от коммерческой версии, демонстрационное ПО имеет ограничения по времени работы на компьютере и требует периодического запуска программы [30, 31].

Лабораторная установка состоит из персонального компьютера класса Pentium4 с жестким диском 200 Гбайт и установленных в PCI слоты (*Peripheral component interconnect* - «взаимосвязь периферийных компонентов» -шина вводавывода для подключения периферийных устройств к материнской плате компьютера) одной или нескольких плат видеозахвата. В работе используются четыре аналоговых источника видеосигнала разного качества [32]:

- черно-белая камера с числом элементов 256х288 элементов с низкой разрешающей способностью;

- черно-белая камера с числом элементов 500х576 элементов со средней разрешающей способностью;

- цветная камера с числом элементов 750х576 элементов с высокой разрешающей способностью;

- выход телевизионного тюнера видеомагнитофона с видеосигналом стандарта SECAM.

Видеосигнал с каждого из этих источников с помощью специальных усилителей-корректоров распределяется на два выхода. Таким образом, общее число видеовыходов равно восьми.

Максимальное количество подключенных телевизионных камер соответствует числу телевизионных входов системы видеонаблюдения и регистрации. В нашем случае используется четырехканальная плата видеозахвата с одним устройством ввода видеосигнала (рис. 33). Это устройство позволяет вводить один телевизионный сигнал в реальном масштабе времени с максимальной частотой смены кадров 25 в секунду.

Четыре сигнала подключаются к такому устройству с помощью аналогового коммутатора телевизионных сигналов. Таким образом, в любой момент времени к устройству ввода видеосигнала может быть подключена только один видеосигнал из четырех. При вводе нескольких видеосигналов неизбежно их прореживание, связанное с переключением. В результате число кадров в секунду по каждому каналу заметно меньше, чем при работе с одним видеосигналом.



Рисунок 33 – Четырехканальная плата видеозахвата

Формат изображения связан с преобразованием сигнала из аналоговой формы в цифровую. При этом преобразовании возможно прореживание сигнала как по горизонтали, так и по вертикали.

Для исследуемых аналоговых сигналов с видеокамер максимальное качество достигается при формате изображения 720х576 элементов изображения или пикселов. Форматы изображения с меньшим качеством имеют значения, в два или в четыре раза меньшие, чем при максимальном формате.

Порядок выполнения работы

Работа рассчитана на 4 часа и включает в себя следующие разделы:

- включение лабораторной установки и подготовка ее к работе;
- изучение основных функциональных элементов программы «Тайфун»;
- изучение методов аппаратной коррекции изображений;
- изучение методов программной коррекции изображений;
- выключение лабораторной установки.
- **І.** Включение лабораторной установки выполняется следующим образом:
 - включить блок питания телевизионных камер;
 - включить компьютер с помощью кнопки на его передней стенке;
 - после включения компьютера автоматически запустится операционная система Windows и автоматически запустится программа «Тайфун».

ІІ. Изучение основных функциональных элементов программы «Тайфун»

Программа состоит из нескольких окон. На рисунке 34 представлено Основное окно программы, которое появляется всегда, даже когда программа запущена первый раз и не настроена.



Рисунок 34 – Основное окно ПО Тайфун

1. Через основное окно программы доступны самые общие настройки программы. Для доступа к настройкам ПО необходимо открыть доступ к ним. Для этого следует щелкнуть по кнопке с изображением закрытого замка. Закрытый замок изменится на открытый и доступ к настройкам станет возможен.

2. Наведите на изображение любой телевизионной камеры указатель мыши. Произойдет выделение этой камеры и будет подсвечен ее номер или название в системе. Нажмите кнопку мыши. Вы увидите список возможных действий с выбранной камерой. Перечислите эти действия. Сделайте скриншот для отчета.

3. Изучите последние элементы интерфейса - окна просмотра изображений камер. Если эти изображения имеются на экране, то это означает, что при загрузке программы по умолчанию эти камеры уже были выбраны для отображения на экране монитора.

Как правило, число одновременно открытых окон на экране одного монитора не должно превышать десяти. При этом изображения окон камер не должны закрывать важные элементы программы, например, поэтажный план.

Программа «Тайфун» позволяет настроить изображения телевизионных камер оптимальным образом.

4. Каждое из изображений с камер можно индивидуально настроить. Можно переместить эти окна в любое место на экране. Поместив курсор мыши на границу окна, можно изменить его размер и форму. При нажатии кнопки максимизации окна его изображение будет развернуто во весь экран. При нажатии кнопки сворачивания окна его изображение примет прежний размер.

Последовательно перемещая изображения с камер и изменяя их размеры, добейтесь наиболее удобного их размещения. По окончании этого следует запомнить скриншот экрана.

5. Как правило, число камер в системе превышает число открытых окон. Если требуется открыть окно камеры, которого нет на экране, то следует найти это окно на поэтажном плане объекта.

Найдите камеру, изображения которой нет на экране. Наведите на нее курсор мыши. Нажмите правую кнопку мыши. Откроется новое окно просмотра с изображением выбранной камеры. Расположите это окно на экране наиболее удобным образом.

6. Если на экране присутствует много изображений камер, то для повышения информативности предусмотрен режим изменения вида окон просмотра. Для

этого следует в главном окне программы выбрать в меню «Вид» опцию «Показывать заголовки окон просмотра видеосигналов» и снять напротив нее птичку. Вид окон изменится. Теперь окно содержит только изображение камеры, а надписи экран не загромождают. Измените вид окон просмотра и затем запомните скриншот экрана.

7. Для работы системы необходимо выбрать требуемый формат. Для этого существует специальное окно, которое открывается при настройке платы видеозахвата (рис. 35, а).

| a | б |
|--|---|
| Настройка параметров карты ввода Общие Тревоги Тип Цунами Обозначение 1 Автоматический опрос камер 1 Номер драйвера 1 Настройка параметров драйвера Формат видеосигнала | Video Format Driver 1 Image Dimensions 0K 768 x 288 1/4 1/4 1/4 1/4 1/4 1/4 1/4 1/4 1/2 1/2 1/4 1/2 |
| Настроить Настроить Задержка переключения (мс) Режекторный фильтр Отключить | Video Source Driver 1 Video Connector • MUX0 • MUX1 • MUX2 • MUX3 • Enable 6V/H6 Video Standard: PAL-B,D,G,H,J |
| 🗸 OK 🗶 Cancel | OK Cancel Default |

Рисунок 35 – Вид окон для настройки параметров изображения

В нем можно выбрать формат изображения (рис. 35, б), а также основные параметры видеосигнала и настройки стандарта цветности (рис. 35, в).

Ш. Изучение методов аппаратной коррекции изображений.

Аппаратная регулировка параметров изображения производится во вкладке «Видео», в окне свойств видеокамеры (рис. 36).

| Настройка параметров ТВ камеры: Резерв 16 | | | | | | |
|---|---------------------------|----------|------------------------------------|-------------------------------------|----------------|--|
| Общие | Видео | Тревоги | Детек | тор движения | Запись на диск | |
| - Настр Динаг Яркос | ойка па инческий ть | раметров | з виде н АЦП 0 128 | осигнала Матомат Насыщенности | Стандартные | |
| Контр | астность | | 128 | Цветовой тон | | |
| Передача по сети Качество изображения 50 ГГ Использовать сжатие | | | | | | |
| | | | | | | |
| | | | | V 0 | K 🗙 Cancel | |

Рисунок 36 – Окно аппаратной коррекции изображения

Отключите режим «автомат» и отрегулируйте вручную качество изображения используя подвижки регулировки «яркость» и «контрастность». Наблюдайте, как изменяется изображение камеры при изменении этих регулировок. Сделайте скриншоты и отметьте в отчете визуальные изменения изображений при разных настройках.

IV. Изучение методов программной коррекции изображений.

Программная коррекция изображения видеокамеры основана на выборе одного из стандартных алгоритмов или выполняется вручную по гистограмме видеосигнала. Поставьте галочку для свойства «Использовать» и снимите ее для свойства «Автоматическая коррекция» (рис. 37).

Рассмотрите внимательно внешний вид гистограммы сигнала. Отметьте, какие ее элементы соответствуют наибольшему и наименьшему значениям видеосигнала, наибольшему и наименьшему значениям градационной характеристики монитора.

Используя зеленые точки, установите наилучшие значения динамических диапазонов видеосигнала и монитора.

Используя среднюю точку, установите наилучшее значение для гамма-коррекции видеосигнала.

Сделайте скриншоты окна с параметрами и гистограммой, а также с изображением от камеры и сохраните для отчета.

| Настройка параметров ТВ камеры: 330 | | | | | | |
|--|----------------------------|--|--|--|--|--|
| Тревоги Детектор движения | Запись на диск Коррекция 🔳 | | | | | |
| Использование коррекции Параметры коррекции Параметры коррекции О Использовать О Использовать О Использовать О Использовать О Использовать О Использовать О Метод 1 О Метод 2 О Метод 3 | | | | | | |
| Использование апертурной коррекции ГИспользовать 3 | | | | | | |
| Использовать эти параметры для всех ТВ камер | | | | | | |

Рисунок 37 – Окно программной коррекции изображения

IV. Выключение лабораторной установки выполняется в последовательности, обратной ее включению.

Контрольные вопросы

1. Что входит в состав аппаратной и программной частей лабораторной установки?

2. Перечислите основные элементы интерфейса программы "Тайфун".

3. Какие методы используются для регулировки качества изображений в видеосистемах безопасности?

- 4. Каков порядок аппаратной коррекции изображений в ПО "Тайфун"?
- 5. Каков порядок настройки модулей программной коррекции изображений?

Лабораторная работа №6. Модуль распознавания лиц в системе TRASSIR

Цели работы

- 1. Изучить структуру телевизионной системы безопасности на базе ПО TRASSIR, ее составных частей и интерфейса.
- 2. Изучить функциональные возможности и основные режимы работы модуля *Трекер/Распознаватель лиц* системы на базе ПО TRASSIR.
- 3. Получить практические навыки применения приемов работы, настройки оборудования и определения основных характеристик.

Основные теоретические сведения

Программное обеспечение TRASSIR - это современная автоматизированная организация система, назначением которой является видеонаблюдения, видеоинформации, интеллектуальная обработка И хранение a также предоставление средств доступа к видеоинформации оперативно-диспетчерскому персоналу. ПО выполнено в сетевой распределенной архитектуре: оно может работать как на одиночном сервере, так и в составе многомашинных комплексов.

Структура обмена данными построена по простой схеме. Существуют средства получения видео — камеры различного разрешения, выбор которых может варьироваться в зависимости от поставленных задач. Видеорегистраторы позволяют как сегментировать структуру обмена данными для построения масштабных систем, так и облегчать подключение нескольких камер в один узел. Возможности системы хранения данных гибко настраиваются, от установки накопителей различного объема до хранения видео в облаке.

Доступ к системе осуществляется как с удаленной рабочей станции, так и с любого компьютера локальной сети. Имеется возможность оперативного обзора через Internet, а также доступ извне локальной сети к данным архива. Для этого используется любой браузер либо собственный web-клиент Trassir.

Программа видеонаблюдения TRASSIR обеспечивает максимальную совместимость для оборудования различных брендов и возможность подключения аналоговых и IP-камер любого разрешения, а также HDD/SSD любого объема. Это позволяет получать наилучшее качество изображения и не ограничивать себя в размерах архива [33].

Лабораторные работы 6 и 7 выполняются на программно-аппаратном комплексе, с использованием коммерческой разработки ПО, предоставленной компанией DSSL (ООО «ДССЛ-НЕВА»).

Для работы в сети Internet используется университетская сеть, подключение к которой осуществляется через сетевой коммутатор, к которому компьютеры

учебно-лабораторного комплекса подключены посредством второй сетевой карты [34].

ПО TRASSIR можно запускать в двух режимах: в штатном режиме и в режиме "без перезапуска при сбоях".

• В штатном режиме - запуск ярлыка "Trassir 4 Server/Client", созданный при установке в меню Пуск (будет запущен файл watchdog-vc120.exe из корневой папки TRASSIR). В этом случае состояние сервера будет отслеживать специальный модуль - *Watchdog*.

• В режиме "без перезапуска при сбоях" - запуск ярлыка "Trassir 4 Server/Client (без перезапуска при сбоях)", созданный при установке в меню Пуск (будет запущен файл t1server-vc120.exe/t1client-vc120.exe из корневой папки TRASSIR).

Признаком успешного запуска является отображение в верхней части экрана значка главной панели управления и наличие иконки в панели задач:

Программный комплекс TRASSIR включает целый спектр модулей видеоанализа для удовлетворения различных потребностей при обеспечении безопасности объектов.

Назначение и состав модуля Трекер/Распознаватель лиц

В данной работе изучается *Трекер/Распознаватель лиц*, содержащий модули Face Recognition и Face Analytics. Модуль предназначен для автоматического обнаружения и распознавания лиц на изображении с камеры.

Возможности модуля:

- обнаружение лица человека (Поиск лиц на видео с камеры и их выделение. Возможно обнаружение лица на видео, снятого с любого ракурса, включая лица, изображённые в профиль);
- идентификация людей по лицу и оценка качества (Сравнение найденного лица с сохраненным в базе данных и определение степени совпадения);
- определение пола и возраста по лицу;
- распознавание определённых атрибутов внешности человека (Помимо пола и возраста модуль может распознать и выполнить поиск по отдельным атрибутам внешности, таким как цвет волос, наличие очков или головного убора и другим отличительным признакам);
- возможность распознать использование фотоснимка в кадре вместо реального человека (Сопоставление различных характеристик лица человека со статичным изображением с целью предупреждения о попытках использования фото в кадре);
- поиск в архиве по лицу;

 использование модуля в СКУД (системах контроля и управления доступом).

Рекомендации по выбору камеры и ее настройке:

- для работы с модулем рекомендуется использовать камеру с вариофокальным объективом, который позволит приблизить или удалить область съемки без изменения положения камеры; не рекомендуется использовать камеры с объективами fish-eye;
- на камере должно быть настроено минимальное значение выдержки и минимальное значение GOP (группа кадров Group of Pictures наименьшая независимая структура кодирования в видеопоследовательности);
- использование камеры с любым разрешением, но из расчета, чтобы размер лица в кадре был больше 128 рх.

<u>Рекомендации по выбору места установки камеры и освещению области</u> <u>съемки:</u>

- область съемки, в которой производится распознавание лиц, должна быть хорошо освещена; наличие теней на лице или пересвета значительно снизит вероятность распознавания этого человека;
- направление съемки должно быть таким, чтобы лица людей смотрели непосредственно в объектив камеры; допускается небольшой поворот камеры по горизонтали или вертикали, но не более 30 градусов.

Настройки модуля обнаружения и распознавания лиц можно применять индивидуально для каждого канала. Одновременно модуль может обрабатывать изображения со всех подключенных к нему камер. Максимальное количество одновременно включенных детекторов определяется приобретенной пользователем лицензией. Более подобно по модулю видеоанализа см. [35, 36].

Порядок выполнения работы

Работа рассчитана на 4 часа и включает в себя следующие разделы:

- включение лабораторной установки и подготовка ее к работе;
- изучение функциональных возможностей и основных режимов работы модуля *Трекер / Распознаватель лиц* системы на базе ПО TRASSIR:
 - настройка базы данных лиц (процесс создания персон);
 - управление каналами (настройки распознавателя лиц для каждого видеоканала);
 - настройка распознавателя лиц для каждого канала;
 - проверка корректности настроек и пример настроек для отображения работы модуля на экране;
- создание собственного шаблона для контроля работы модуля;
- выключение лабораторной установки.

І. Включение лабораторной установки и подготовка ее к работе производится в следующей последовательности:

- подать питание на видеокамеры;
- включить компьютер с помощью кнопки на его передней стенке;
- после включения компьютера автоматически запустится операционная система Windows и автоматически запускается предварительно инсталлированное ПО Trassir Server.

Для работы IP-камер с компьютерами Trassir все оборудование должно находиться в одной локальной подсети. Выбранная и настроенная подсеть имеет диапазон адресов 192.168.1.1-192.168.1.254 (соответствует маске 255.255.255.0).

II. Изучение функциональных возможностей и основных режимов работы модуля *Трекер / Распознаватель лиц*

1. Работа с основными настройками модуля

Основные настройки модуля расположены во вкладке «Модули» => «Распознавание лиц» (рис. 38).

| < Владелец лицензии: ИТМО ключ для лаборатории > Настро | ойки - Admin - А457-8/Модули/Расп | ознавание лиц | - T | | _ | | | |
|---|-----------------------------------|-----------------|----------------|-------------------|-----------------|----------------|------------------|---|
| 🛢 A457-8 🔹 🔹 | Настройка Справка | | | | | | | |
| | โลงค์วอน แน่นด์ และสามอดังหน | | | | | | | |
| 🛟 Настройки сервера | Глубина хранения, дней | 3 | ÷ | Максима | льное количеств | о потоков 4 | ÷ | |
| 📩 Модули | 🔲 Режим для Системы Конт | роля и Управлен | ия Доступом За | дать настройки по | умолчанию | | | |
| | Порог качества | 50% | ÷ | Период д | етектирования | 250 мс | ÷ | |
| Аналитика | Порог уверенности | 50% | ÷ | Алгорит | м детекции | АЛГ1 | • | |
| AutoIrassir | Минимальный размер лица | 64 пикс | ÷ | Алгорит | м распознавания | АЛГ1 | • | |
| | Максимальный размер лица | 4000 пикс | ÷ | Порог жі | ивости | 50 | ÷ | |
| ыр телефон | Объединять короткие тре | ки | | Алгорит | м эмоций | Распозн | навать всё 🔻 | |
| | Время жизни трека | 2 сек | ÷ | | | | | |
| Tuenoshubunite iling | Порог похожести | 90% | ÷ | | | | | |
| 📕 Устройства | | | | | 4.0 | | | |
| • | | | | | • Отменить изм | ленения 📋 Прим | енить изменения | _ |
| '📑 Каналы | База данных лиц | | | | | | | |
| Corri | Расположение: | | | | Лон | альный сервер | • | |
| Celb | Статус: готово | | | | | посмо | треть содержимое | |
| 🔛 Автоматизация | Управление каналами | | | | | | | |
| | Имя канала | Пол/Возраст | Атрибуты | Живость | Аналитика | Поиск по лицу | Распознавать | |
| | | | ,, | | | | | |
| | AC-D5024.2 | | | | | | | |
| | AC-D5024.3 | \checkmark | | \checkmark | | \checkmark | \checkmark | |
| | | | | | | | | |
| | | | | | | | | |
| ыстрый поиск] ¥ | | | | | | | | |
| neipen noneng n | | | | | | | | Γ |

Рисунок 38 – Основное меню настроек модуля распознавания лиц

Используемые базы информации в модуле распознавания лиц:

• Временная база лиц для хранения всех распознанных лиц. Ее размер определяется в Глобальных настройках (рис. 38) и определяется параметром «Глубина хранения», который задается в днях.

• База лиц, содержащая информацию о персоне и его антропометрические данные, которые используются для сравнения с лицом, обнаруженным на видео. Максимальный размер этой базы определяется лицензией и отображается в поле «Размер базы данных».

Максимальное количество видеопотоков, с которыми работает модуль, определяет количество "очередей", в которых производится обнаружение лиц. В каждом полученном кадре модуль пытается обнаружить лицо, и при увеличении числа потоков увеличивается скорость обнаружения. Максимальное значение потоков ограничено количеством ядер процессора на сервере. Параметр «Максимальное количество потоков» также задается в панели Глобальных настроек (рис. 38).

Внимание: увеличение числа потоков приведет к увеличению нагрузки на сервер. Модуль может искать лица во всех кадрах. Однако не на всех кадрах лицо человека отображается в хорошем качестве. Надо помнить, что у модуля распознавания лиц может произойти ложное срабатывание. Для исключения этого рекомендуется сформировать режим работы детектора в качестве СКУД, активировав флаг «Режим для Системы Контроля и Управления Доступом» (рис. 38). При этом по умолчанию задаются оптимальные настройки детектора, позволяющие контролировать эффективность его работы в многопотоковом режиме.

Описание функционала настроек по умолчанию (рис. 39):

• Порог качества исключает лица с плохим качеством: смазанные, частично скрытые и другие.

• Порог уверенности – граница, определяющая степень соответствия обнаруженного лица и персоны в базе данных лиц.

• Минимальный размер лица и Максимальный размер лица определяют диапазон размеров лиц, с которыми работает модуль.

• Период детектирования – интервал между кадрами, которые будут использоваться для обнаружения лиц, чем он меньше, тем чаще производится поиск лиц на видео.

• Алгоритм детекции и Алгоритм распознавания – внутренние наборы правил, при помощи которых производится обнаружение лиц на видео и их распознавание. Если в результате выбранных настроек лица не детектируются или не распознаются, то выберите другой алгоритм.

• Порог живости – граница, определяющая степень живости лица или насколько обнаруженное лицо похоже на фотографию.

• Алгоритм эмоций – набор правил, позволяющий среди всех обнаруженных лиц отображать людей только со счастливым выражением лица.

| 📝 Режим для Системы Контроля и Управления Доступом | | Задать настройки по умолчанию | |
|--|-------------|-------------------------------|-----------------------|
| Порог качества | 65% 🛨 | Период детектирования | 250 мс 🚊 |
| Порог уверенности | 65% 🛨 | Алгоритм детекции | АЛГ2 - |
| Минимальный размер лица | 160 пикс 🛨 | Алгоритм распознавания | АЛГ2 - |
| Максимальный размер лица | 4096 пикс 🛨 | Порог живости | 50 🛨 |
| 📃 Объединять короткие трек | и | Алгоритм эмоций | Только счастливые 🔻 |
| Время жизни трека | 2 сек | | |
| Порог похожести | 90% | | |
| | | < Отменить изменения | 💾 Применить изменения |

Рисунок 39 – Настройки по умолчанию

Во время движения человек может поворачивать голову или лицо, может скрываться за естественными препятствиями. Очередное появление лица в кадре будет восприниматься модулем как новое. При установке флага «Объединять короткие треки» модуль будет объединять эти движения в одно, в зависимости от следующих параметров: «Время жизни трека» и «Порог похожести» (рис. 40).

| 📝 Объединять короткие треки | | | | | |
|-----------------------------|-------|---|--|--|--|
| Время жизни трека | 2 сек | ÷ | | | |
| Порог похожести | 90% | ÷ | | | |

Рисунок 40 – Параметры треков

• Время жизни трека – это время, в течение которого модуль хранит лицо одного человека, обнаруженное в разных кадрах. Например, время жизни трека - 5 сек, модуль обнаружил лицо и человек отвернулся от камеры. Если он повернется через 4 секунды, то информация о лице дополнит существующую запись. А если через 6 секунд, то создаст новую.

• Порог похожести – граница, определяющая степень похожести обнаруженного лица человека и сохраненного ранее. Если обнаруженное в кадре лицо похоже на сохраненное ранее, то информация о нем дополнит текущую запись в базе данных. Если степень похожести лиц не превышает заданной границы, то создается новая запись.

2. Работа с Базой данных лиц

База лиц используются для сравнения с лицами, которые обнаруживает модуль *Трекер/распознаватель лиц* на изображении с камер. База данных лиц (рис. 38) может храниться как локально, так и на любом сервере TRASSIR с соответствующей лицензией. Для того чтобы подключиться к базе данных, необходимо настроить соединение с сервером и указать его в настройке «Расположение». Для перехода в базу данных лиц необходимо нажать на ссылку «Посмотреть содержимое».
Алгоритм настройки подключения к базе данных:

1) Открыть окно Настройки сервера (рис. 41).

| 🐡 < Владелец лицензии: ИТМО ключ для лаборатории > Настрой | ки - Admin - А457-8/Настройки сервера | |
|---|---|-----------------|
| S A457-8 | Настройка Справка | |
| Настройки сервера | Версия ПО: Service Pack 1110366, Grabber Pack 1110366 | a |
| Облако Трассир Архив Запись сетевых каналов Синхронизация архива | Хеш-функция SSL-сертификата: Управление сертификатами 4A F1 25 60 98 41 A2 9D 75 A7 A8 C4 9E 50 83 26 02 9E 26 E3 | |
| Скриншоты Веб-сервер Карта | Соединено через <u>на 57-а (132.103.1.20)</u> Индикаторы здоровья Б. с. с. 23 с. | |
| Отчеты База Данных Персоны | Глубина архива: 22 Сеть: ОК Алтаратическия вкод под пользователем: не использовате Диски: ОК Скрипты: ОК Алпаратное ускорение видео: OpenGL База данных: ОК Облако: и/д Устройство воспроизведения звука: По умолчанию Камероь: 4 / 4 Модули: ОК Устройство воспроизведения звука: | • • |
| Пользователи Аудит | Загрузка СРU: 45.9% Допустимый период бездействия: 30 мин Аптайм: 1 час, 7 мин. Выход из системы при бездействия: С//DSL/Trassir-4.1/shots | • |
| Модули | Язык: по имолчанию | • |
| > 🗰 Устройства | Слицензия | |
| > 📲 Каналы | Владелец: ИТМО ключ для лаборатории TCP/IP порт сервера (управление): 3080 (требуется перезапуск) | |
| > 🚠 Сеть | Создана: 25.9.2019 ТСР/ІР порт сервера (видео): 3081 | |
| [быстрый поиск] х | Показать лицензию 🛆 Обновить лицензию - | |
| | | <u>З</u> акрыть |

Рисунок 41 – Окно Настройки сервера

2) В списке настроек выбрать пункт База данных (рис. 42).

| 😨 < Владелец лицензии: ИТМО ключ для лаборатории > Настрой | ки - Admin - А457-8/Настройки сервера/База Данных | _ 0 X |
|--|---|-----------------|
| SA457-8 | Настройка Справка | |
| Настройки сервера | Тип сервера: PostgreSQL - | |
| > Облако Трассир | Состояние: Соединение установлено | |
| Архив | Параметры соединения | |
| Запись сетевых каналов | Agpec: localhost | |
| Синхронизация архива | Порт: 5432 📩 | |
| Скриншоты | Имя базы данных postgres | |
| Веб-сервер | Пользователь: postgres | |
| Карта | Пароль: | |
| Отчеты | | |
| База Данных | Срок хранения записей 170 📩 дней | |
| Персоны | | |
| > Пользователи | | |
| Аудит | | |
| > 掉 Модули | | |
| > 🏥 Устройства | | |
| > 📑 Каналы | | |
| > 💂 Сеть | | |
| > 🤯 Автоматизация | | |
| [быстрый поиск] ж | | |
| | | <u>З</u> акрыть |

Рисунок 42 – Настройки базы данных

- 3) Указать настройки подключения:
 - Тип сервера оставить значение "PostgreSQL".

• Адрес и Порт – IP-адрес или DNS-имя сервера, где установлена база данных. Если база данных установлена локально, оставьте значение localhost. Если база данных установлена на другом сервере, убедитесь, что ваш IP-адрес находится в списке *разрешенных адресов* для внешних подключений.

• Имя базы данных, Пользователь, Пароль – параметры, которые были указаны для базы данных на этапе ее установки.

• Срок хранения записей – срок, по истечении которого старые события будут перезаписываться новыми событиями.

4) Убедиться, что подключение успешно установлено (в поле «Состояние» появится строка «Соединение установлено») (рис.43):

| Настройка Спра | вка |
|---------------------|------------------------|
| Тип сервера: | PostgreSQL 💌 |
| Состояние: | Соединение установлено |
| Параметры соедин | яния |
| Адрес: | localhost |
| Порт: | 5432 |
| Имя базы данных: | trassir |
| Пользователь: | postgres |
| Пароль: | |
| Срок хранения запис | сей 180 🗢 дней |
| | |

Рисунок 43 – Успешное подключение к серверу

Если подключение не будет установлено, то строка *Состояние* будет содержать ошибку с описанием причины, по которой подключение невозможно. Например, в данном случае подключение невозможно, так как неправильно указано имя базы данных (рис. 44):

| Настройка Справи | ka | |
|------------------|---|--|
| Тип сервера: | PostgreSQL 💌 | |
| Состояние: | ОШИБКА Код ошибки: ВАЖНО: база данных "trassir3" не существует | |

Рисунок 44 – Ошибка подключения к серверу

TRASSIR для распознавания использует локальный кэш базы данных лиц. Поэтому при потерях связи с сервером, на котором расположена база данных лиц, распознавание будет продолжено. Локальный кэш базы данных лиц будет обновлен сразу после восстановления связи.

3. Управление каналами с модулем Трекер/Распознаватель лиц

В нижней части основного окна настроек модуля отображается список каналов, на которых включен модуль *Трекер/Распознаватель лиц* (рис. 38). Нажав на ссылку, вы перейдете к настройкам модуля на выбранном канале (рис. 45):

| Управление каналами | | | | | | |
|---------------------|--------------|--------------|---------|-----------|---------------|--------------|
| Имя канала | Пол/Возраст | Атрибуты | Живость | Аналитика | Поиск по лицу | Распознавать |
| Faces | \checkmark | \checkmark | | \square | | \square |
| TR-D2111IR3W 1ch | \checkmark | | | | | \checkmark |

Рисунок 45 – Управление каналами

Описание функций модуля по каналам:

• Пол/Возраст – отображение в интерфейсе оператора пола и возраста человека, определенных по антропометрическим характеристикам его лица;

• Атрибуты – поиск лиц по определённым атрибутам внешности;

• Живость – функция, позволяющая отличить человека в кадре от фото или изображения;

• Аналитика – включает передачу данных о распознанном лице скрипту "Аналитика";

• Поиск по лицу – функции поиска по лицу в кадре и фото;

• Распознавать – функция распознавания лиц по базе данных лиц.

4. Настройка распознавателя лиц для канала

Для активации модуля необходимо в настройках канала в области «Программные детекторы» выбрать пункт «Трекер/распознаватель лиц» и выбрать Сервер, который будет производить расчет аналитики (рис. 46).

Для открытия окна настройки нажмите на активную ссылку «Настройки распознавателя лиц». В открывшемся окне необходимо определить размер Зоны детектирования – области изображения, в которой будет производиться распознавание лиц.

Размер и расположение зоны детектирования могут быть заданы непосредственно на изображении. Для этого необходимо потянуть курсором за желтую рамку, определяющую границу зоны детектирования, как показано на рисунке 47.



Рисунок 46 – Выбор сервера для расчета аналитики

Пример формирования зоны детектирования на изображении лаборатории показан на рисунке 47.



Рисунок 47 – Выбор зоны детектирования

Если общие параметры детектирования не подходят для работы детектора на данном канале, необходимо снять флаг «Использовать глобальные параметры» и изменить эти параметры вручную (рис. 48).

| 🔲 Использовать глобальные параметры | | | Определять пол/возраст |
|-------------------------------------|-----------------------------------|---|---|
| Порог качества | 51% 📩 🗹 Объединять короткие треки | | 🔽 Определять атрибуты лица |
| Минимальный размер лица | 64 пикс 📩 Время жизни трека 2 сек | ÷ | 🔽 Определять живость лица |
| Максимальный размер лица | 1024 пикс 📩 Порог похожести 90% | ÷ | 📝 Включить аналитику |
| Период детектирования | 250 мс 📩 | | 🔽 Распознавать лица, используя: |
| Статус: Ок | | | Перейти на страницу настроек распознавателя лиц Отменить изменения |
| | | | <u>З</u> акрыть |

Рисунок 48 – Пользовательские настройки параметров детектирования

Нажав на активную ссылку «Перейти на страницу настроек распознавателя лиц», можно снова перейти к общим настройкам детектора для канала.

5. Проверка корректности настроек детектирования

Для того чтобы проверить настройки детектирования, нужно нажать правой клавишей мыши на области с изображением, выбрать в выпадающем меню пункт «Вид...», поставить флаг рядом с пунктом «Показывать фигуры» и выбрать в выпадающем списке «Распознаватель лиц» (рис. 49).



Рисунок 49 – Настройка отображения фигур на лицах

В результате работы модуля распознанные лица будут выделены на изображении. Пример такого выделения показан на рисунке 50.



Рисунок 50 – Распознанные лица

Ш. Создание собственного шаблона для распознавания лиц.

Полноценную работу модуля можно видеть в интерфейсе оператора. Для этого вы можете создать простой шаблон, содержащий собственную Базу лиц для работы модуля с заданным контингентом.

В данном случае База лиц – это часть базы персон, в которую включены выбранные Вами люди и их антропометрические данные. На рисунке 51 показан пример создания папки «ОЭПиС», содержащей данные по студентам, находящимся в лаборатории.

Размер ее отображается в поле «Размер» базы данных лиц в окне «Настройки» => «Распознавание лиц» и определяется лицензией TRASSIR.

Рекомендации к фотографиям, используемым для распознавания:

Для распознавания лиц TRASSIR использует все фотографии, находящиеся в Базе лиц. Вероятность распознать по ним человека, попавшего в объектив камеры, зависит от качества загруженных в TRASSIR фотографий. Чтобы повысить вероятность распознавания, необходимо придерживаться следующих рекомендаций:

• для одной персоны можно загрузить несколько фотографий, одна из которых должна быть снята в анфас, а на остальных допускается поворот на не более чем 30 градусов по вертикали или горизонтали (фотографии, на которых человек изображен в профиль, значительно снизят вероятность распознавания);

• если человек носит очки, то для улучшения распознавания загрузите фото в очках;

• фотографии, на которых лицо человека размыто, засвечено или находится в тени, значительно снизят вероятность распознавания;

• лицо на фотографиях должно быть не менее 128х128 рх.

| 🔹 < Владелец лицензии: ИТМО ключ для лаборатор | ии > Настройки - Admin - А457-8/Модули/Распознавание лиц/База лиц | |
|--|---|-----------------------|
| A457-8 | 🔹 🗸 Настройка Справка | |
| Настройки сервера | 🔶 🕂 Добавить персону 🔚 Добавить папку | |
| ~ 🏚 Модули | Введите ФИО или примечание | × 👻 Показать БД лиц 👻 |
| Аналитика AutoTrassir IntelliVicion | ЭПИС Александр Николаевмч | лицо в базе |
| Sip телефон Нейронный детектор | Артем | лицо в базе |
| Распознавание лиц База лиц | Виктория Александровна | Лицо |
| > 🗮 Устройства | Кирилл | лицо |
| ∼ 📑 Каналы | Коля | в базе |
| Потерянные каналы AC-D4121IR1 1 | Санчез | Лицо в базе |
| AC-D4121IR1 3 | Саша | Лицо |
| Модель оптики Распознавание лиц | - | |
| Распознавание номеров Саботаж | • | |
| [быстрый поиск] | × | |

Рисунок 51 – Пример создания базы лиц

Сделайте и сохраните для отчета скриншоты, отображающие последовательность Ваших действий по изучению функциональных возможностей и основных режимов работы модуля *Трекер/Распознаватель лиц* и результаты создания шаблона для распознавания лиц.

IV. Выключение лабораторной установки выполняется в последовательности, обратной ее включению.

Контрольные вопросы

1. Что входит в состав аппаратной и программной части лабораторной установки?

2. Перечислите состав и возможности модулей Face Recognition и Face Analytics в программе Trassir.

3. Какие требования предъявляются к видеосистеме для эффективной работы модулей распознавания лиц?

4. Для какого количества каналов можно настроить модуль распознавания лиц в видеосистеме и чем это количество определяется?

5. Каковы рекомендации к изображениям для формирования базы данных?

Лабораторная работа №7. Модуль распознавания автомобильных номеров AutoTRASSIR

Цели работы

- 1. Изучить назначение и особенности использования систем автоматического распознавания автомобильных номеров.
- 2. Изучить функциональные возможности и основные режимы работы модуля AutoTRASSIR.
- 3. Получить практические навыки применения приемов работы, настройки оборудования и определения основных характеристик.

Основные теоретические сведения

B настоящее время системы автоматического распознавания автомобильных номеров востребованы в самых различных областях. Они применяются работе автотранспортных предприятий, станций В техобслуживания, автомобильных парковок и др. Подобные системы позволяют контролировать наличие автомобилей в зоне обслуживания, определять время обслуживания автомобилей клиентов, количество свободных мест на парковке, фиксировать время пребывания автомобиля в конкретной зоне, организовывать автоматический въезд и выезд автомобилей и т.д. Кроме того, возможность автоматического распознавания номера автомобиля является важным аспектом контроля и обеспечения безопасности дорожного движения ввиду постоянно увеличивающегося на дорогах количества транспортных средств.

Выделим конкретные задачи, решение которых требует использования системы автоматического распознавания автомобильных номеров:

- контроль и ограничение доступа;
- организация платного доступа для автомобилей;
- управление потоками автотранспорта;
- управление временем нахождения автотранспортного средства на территории;
- регистрация автотранспорта;
- отслеживание автотранспорта внесенного в список наблюдения.

Под системами автоматического распознавания автомобильных номеров подразумеваются программный или аппаратно-программный комплекс, который реализует алгоритмы автоматического распознавания номерных знаков для регистрации событий, связанных с перемещением автомобилей, т.е. для автоматизации ввода данных и их последующей обработки [37].

В основе автоматического распознавания автомобильных номеров лежат следующие процедуры цифровой обработки изображений [38, 39]: локализация, нормализация, сегментация, распознавание, синтаксический анализ.

Локализация предназначена для обнаружения и выделения на изображении области с регистрационным номером автомобиля.

Для локализации области номера на изображении применяются различные алгоритмы. В большинстве своем они базируются на бинаризации, выделении контуров, морфологической обработке изображений для устранения мелких деталей и разрывов [39]. При этом предполагается, что исходное цветное изображение преобразуется в полутоновую форму.

Методы глобальной и локальной бинаризации изображений заключаются в разделении всех пикселей полутонового изображения по яркости на два класса – объект и фон. Однако адаптивные подходы являются более предпочтительными ввиду возможности компенсировать влияние помех на различные участки изображения, например, распределение теней из-за неоднородности освещения.

В результате формируются связанные последовательности точек бинарного изображения — контуры. На заключительном этапе процедуры локализации определяется, какой из полученных контуров является границей области автомобильного номера.

Подходы, основанные на анализе контуров, позволяют находить номер различного размера и под различным наклоном. Однако у них есть несколько недостатков:

- 1) на изображении автомобиля может быть много прямоугольных объектов, похожих своими очертаниями на автомобильный номер;
- относительно высокая трудоемкость вычислений даже на изображении небольшого размера время обнаружения может достигать нескольких секунд;
- 3) они основаны на анализе границ номера, что не всегда возможно в реальных условиях. Например, на изображениях запыленных автомобилей могут отсутствовать четко выраженные границы.

Более эффективным для локализации области номера на изображении представляется альтернативный подход на основе метода Виолы – Джонса, который позволяет находить область номера в сложных и нетипичных условиях в реальном времени [23].

Метод Виолы – Джонса основан на применении набора признаков Хаара. Признак Хаара (рис. 21) состоит из смежных прямоугольных областей, которые позиционируются на изображении, далее происходит суммирование интенсивности пикселей в областях, затем между суммами вычисляется разность. На этапе обнаружения заданной области в методе Виолы – Джонса используется окно определенного размера, которое перемещается по изображению. Начальное значение сканирующего окна можно задать исходя из размеров автомобильного номера 520×115 мм [29]. Недостатком данного подхода является относительно невысокая степень инвариантности к аффинным и проекционным искажениям объектов на изображениях.

После локализации области изображения с автомобильным номером необходимо сформировать изображение, содержащее только номер, и выполнить его **нормализацию**. Нормализация заключается в приведении размеров и ориентации полученного изображения с номером к требуемому виду. Здесь выполняются геометрические преобразования, определение границ и обрезание рамки номера по горизонтали и по вертикали, фильтрация с целью шумоподавления или повышения контраста.

Как правило, выделенная область с номером имеет сравнительно небольшой размер, содержит границы номера, горизонтальные полосы очертания бампера и радиаторной решетки. Поэтому для определения угла поворота рамки номера используется алгоритм на основе преобразования Хафа для линий [40]. Алгоритм заключается в следующем:

1) определяются линии, длина которых больше половины ширины области номера;

2) формируется прямая из средних значений всех точек полученных линий;

3) вычисляется угол между полученной прямой и линией горизонта.

Далее поиск границ рамки номерной платы автомобильного номера выполняется с использованием гистограммы интенсивности по горизонтали и вертикали. Для построения гистограмм значения всех пикселей бинарного изображения суммируются по строкам или столбцам, затем выделяется максимум и отсеиваются все значения меньше 20% от максимума.

Следующая процедура распознавания номера (сегментация) обеспечивает разделение изображения на знакоместа, то есть выделение областей отдельных символов. Самым простым подходом для сегментации символов является использование заранее заданного шаблона [4].

Шаблон можно представить в виде изображения темных прямоугольников, соответствующих символам, на светлом фоне, как показано на рисунке 52 [41]. Основным требованием для эффективной сегментации здесь является правильное определение рамки номера на этапе нормализации, а любое отклонение от реальных габаритов будет способствовать ухудшению сегментации символов номера.



Рисунок 52 – Шаблон расположения символов на номерной плате автомобиля

Следующий подход к сегментации основан на построении горизонтальной проекции средней интенсивности [40, 41]. При этом вычисляется средняя интенсивность в каждом столбце изображения номера и определяются столбцы, в которых средняя интенсивность значительно отличается от порогового значения. На рисунке 53 показан пример работы метода гистограммы.



Рисунок 53 – Гистограмма распределения яркости пикселей номерной пластины

Еще один способ **сегментации** базируется на проведении контурного анализа [38]. После получения контуров, которые представляют собой связанные последовательности точек бинарного изображения, определяются те из них, которые являются границами областей символов номера автомобиля. Это достигается путем фильтрации контуров, удовлетворяющих определенным требованиям к соотношению геометрических характеристик [40]. Пример работы соответствующего алгоритма показан на рисунке 54.



Рисунок 54 – Результат сегментации символов автомобильного номера

Итак, подход, основанный на использовании шаблонов, очень прост в реализации и не требует сложных операций, связанных с анализом изображения для поиска символов, но для его работы необходимо точное выделение границ рамки номера, что в реальных условиях не всегда выполнимо. Гистограммный анализ изображения имеет высокую чувствительность к шумам и дефектам на изображении, особенно в промежутках между символами. В результате этого гистограммы могут не дать выявить ярко выраженные максимумы в промежутках между символами. В данном методе могут возникнуть также затруднения при выделении символов региона, так как под ними находится символы принадлежности к стране. Сегментация на основе контурного анализа является менее требовательной к условиям реализации, поскольку здесь используются обобщенные геометрические признаки.

Процедура **распознавания символов** автомобильного номера предназначена для формирования строки символов на основе приведения каждого из них к заранее определенному стандартному виду. После предварительной обработки к соответствующим выделенным знакоместам можно применить различные методы распознавания: методы сопоставления с шаблонами, методы на основе анализа моментов изображения, метода опорных векторов (support vector machine, SVM) [42, 43].

Преимущества шаблонных методов заключаются в простоте реализации и устойчивости к дефектам изображения символов. Основной недостаток шаблонных методов заключается в невозможности распознавания символов, которые подвержены аффинным и проекционным искажениям и отличаются от заложенных в системе шаблонов.

Достоинство использования моментов изображения для выделения признаков символов заключается в высокой устойчивости к изменению масштаба изображения и другим геометрическим преобразованиям. Недостаток их использования заключается в высокой чувствительности к шумам и дефектам на изображении, в результате чего символы могут неправильно классифицироваться.

Достоинство метода SVM заключается в том, что для построения классификатора для распознавания символов достаточно обучающей выборки сравнительно небольшого размера. Кроме того, он имеет низкую вероятность ошибки. Метод опорных векторов представляется наиболее эффективным для распознавания символов автомобильного номера [41].

Последняя процедура (**процедура синтаксического анализа**) выполняется для определения элементов строки, содержащей символы номера. Данные элементы могут различаться согласно стандартам стран регистрации автотранспортных средств. Например, в Российской Федерации используется стандарт, определяемый ГОСТ Р 50577-93 (с изменениями, внесенными приказами Федерального агентства по техническому регулированию и метрологии).

Существующие методы распознавания регистрационных номеров автомобиля имеют низкую устойчивость к изменениям масштаба, смещениям, поворотам, смене ракурса и прочим искажениям. Поэтому в последнее время широкое распространение получили алгоритмы с применением сверточных нейронных сетей, ИЗ преимуществ которых является слабая одним чувствительность к искажениям входного сигнала.

Назначение, состав и функциональные возможности модуля AutoTRASSIR

AutoTRASSIR - система автоматического распознавания автомобильных государственных регистрационных знаков, может быть использована для контроля въезда/выезда автотранспорта с территории предприятий, а также службами автоинспекции: на пропускных пунктах и в других контрольных точках. AutoTRASSIR обеспечивает взаимодействие с системами контроля доступа, видео- и аудиоконтроля и исполнительными устройствами (например, шлагбаумами) [35, 36].

Программа AutoTRASSIR является одним из пионеров в линейке отечественных продуктов данного направления, создана разработчиками Российской компании DSSL, является клиентским приложением системы видеонаблюдения TRASSIR и отличается простотой рабочего интерфейса и настройки.

Максимальное количество камер, подключаемых к рабочей станции с модулем распознавания, не ограничено, однако чрезмерная нагрузка камерами снизит скорость его работы (это касается любого модуля, где присутствуют алгоритмы обработки изображений). Модуль поддерживает работу с IP и аналоговыми видеокамерами.

Модуль AutoTRASSIR поставляется в двух вариантах: "быстрый" (до 200 км/час), и "медленный" (до 30 км/час). Данный параметр определяется лицензией, настройка обоих типов модулей в TRASSIR идентична.

В TRASSIR встроены несколько версий модуля AutoTRASSIR, которые имеют ряд особенностей:

- Модули AutoTRASSIR (LPR1) и AutoTRASSIR (LPR3) работают только локально на всех серверах с TRASSIR.
- Модуль AutoTRASSIR (LPR4) работает:
 - локально или удаленно на всех видеорегистраторах с TRASSIR OS;
 - только удаленно на серверах с TRASSIR для Windows.
- Модуль AutoTRASSIR (LPR5) работает:
 - локально на сервере с TRASSIR OS версии NeuroStation;
 - удаленно на всех серверах с TRASSIR.

В данной работе нейросетевые возможности ПО не исследуются ввиду ограничений лицензии. Поэтому особенности настроек для версии LPR5 не рассматриваются.

Возможности модуля AutoTRASSIR:

- Распознавание автомобильных номеров по шаблонам и без, модуль работает с номерами следующих стран: Россия, Украина, Турция, Тайвань, Молдова, Кыргызстан, Казахстан, Испания, Грузия, Беларусь, Китай и др. Использование шаблонов для распознавания зависит от выбранной версии модуля AutoTRASSIR:
 - LPR5 используются нейросетевые возможности TRASSIR, одновременно по нескольким шаблонам и без, если страны отсутствует в списке шаблонов AutoTRASSIR;
 - LPR4 и LPR3 одновременно по нескольким шаблонам и без, если страна отсутствует в списке шаблонов AutoTRASSIR;
 - LPR1 по одному шаблону, который определяется лицензией на TRASSIR;
- Работа с внутренней и внешними базами данных

- Поиск распознанных номеров в реальном времени. Использование баз данных в качестве белого ("свой"), черного ("чужой") и/или информационного списков.
- Сохранение распознанных номеров во внутренней базе данных с указанием времени и даты проезда, ссылки на видеоинформацию и т.п.
- Расширенный поиск и редактирование номеров во внутренней базе данных.
- Интерфейс оператора с возможностью вывода на экран информации о транспортном средстве и его номере одновременно с нескольких камер.
- Наличие журнала распознанных номеров с возможностью поиска.

Возможности интерфейса

В системе существует установка тревожных событий на отдельный номер или группу номеров. Тревожными реакциями системы могут быть текстовая информация любого содержания (задержать, досмотреть, пропустить и т.д.), звуковые сигналы, световая индикация (к контроллеру подключаются сигнальные лампы или светодиоды), включение любого внешнего устройства, отправление сообщений во внешние системы (например, рассылка СМС сообщений на указанные номера по SSMP протоколу).

С целью объективной идентификации автомобиля в базу данных можно также добавить следующую информацию: марку, цвет, ФИО владельца, контактные данные. По результатам распознавания эта информация может отображаться на мониторе рабочей станции. При некорректном распознавании предусмотрена возможность ручной корректировки номера автомобиля, такое действие отобразится в журнале событий, как корректировка.

Регистрационные номерные знаки транспортных средств в России

Регистрационные номерные знаки Российской Федерации представляют собой специальные символические знаки, которые используются для специальной учёта легковых И грузовых автомобилей, мотоциклов, И строительной техники, вооружения, прицепов. Регистрационный номерной знак может наноситься непосредственно на транспортное средство или на пластины (металлические или изготовленные из другого материала), которые закрепляются на транспортное средство.

Типы регистрационных номеров для транспортных средств, постоянно участвующих в дорожном движении РФ [29], и их сопоставление с шаблонами автомобильных номеров модуля AutoTRASSIR приведены в таблице 5.

| <u></u> | | | |
|---|------------------------------|----------------------|--|
| Изображение номерного | Описание | Шаблон в | |
| знака | Описание | AutoTRASSIR | |
| Группа 1 - транспортные средства, принадлежащие юридическим лицам и | | | |
| гражданам РФ, юридич | еским лицам и гражданам инос | странных государств, | |

Таблица 5 – Типы регистрационных номерных знаков

| Изображение номерного | Описание | Шаблон в |
|-----------------------|--------------------------------|--------------------|
| знака | 2 | AutoIRASSIR |
| кроме транспортных | средств группы 3, а также тран | спортные средства, |
| принадлежа | щие физическим лицам без гра | жданства |
| с227на 69 | тип I - для легковых, | м976 мм 34 |
| | грузовых автомобилей и | |
| D 555 | автобусов (кроме типа 1Б); | |
| D 222 | тип 1А - для легковых, | |
| PY 39 | грузовых автомобилей и | |
| RUS | автобусов с нестандартным | |
| | местом крепления (кроме | |
| | типа 1Б) | |
| AO Z65 78 | тип 1Б - для легковых | A A OOO 50 |
| | такси, транспортных | |
| | средств, оборудованных для | |
| | перевозок более восьми | |
| | человек, осуществляющих | |
| | перевозку с лицензией; | |
| 7771 47 | тип 2 - для автомобильных | |
| | прицепов и полуприцепов; | |
| 7777 7777 | тип 3 - для тракторов, | |
| 3/33 3/73 | самоходных дорожно- | |
| MM 55 MM 55 | строительных машин и | |
| | прицепов (полуприцепов); | |
| 077(4477) | тип 4 - для мотоциклов; | |
| 0//0 1133 | тип 4А - для внедорожных | 9329 |
| | мототранспортных средств. | |
| | (снегоболотоходы, | РН 61 |
| [MM 55] | мотовездеходы); | |
| RUS | тип 4Б - для мопедов. | |
| AA 23 | | |
| | | |

Группа 2 - транспортные средства воинских частей, организаций и учреждений федеральных органов исполнительной власти РФ, в которых федеральным законом предусмотрена военная служба, а также опытные (испытательные) образцы военной и специальной техники, изготовленные предприятиями промышленности по государственному оборонному заказу для нужд федеральных органов исполнительной власти;



тип 5 - для легковых, грузовых автомобилей и автобусов; **тип 6** - для прицепов и



| Изображение номерного знака | Описание | Шаблон в AutoTRASSIR |
|-----------------------------|---------------------------|-------------------------|
| 1895 | полуприцепов; | |
| 10 23 | тип 7 - для тракторов, | |
| TC 25 | самоходных дорожно- | |
| 1100 | строительных машин и | |
| 4400 | прицепов к ним; | |
| PHISO | тип 8 - для мотоциклов, | |
| | внедорожных | |
| | мототранспортных средств. | |

Группа 3 - транспортные средства дипломатических представительств, консульских учреждений, в том числе возглавляемых почетными консульскими должностными лицами, международных (межгосударственных) организаций и их сотрудников, аккредитованных при Министерстве иностранных дел РФ;

| | 1 1 | |
|------------------|----------------------------|--|
| | тип 9 - для легковых | |
| | автомобилей глав | |
| | дипломатических | |
| | представительств; | |
| | тип 10 - для легковых, | |
| 002 D 0 0 78 | грузовых автомобилей и | |
| | автобусов дипломатических | |
| 002 ± 007 78 | представительств, | |
| | консульских учреждений, в | |
| | том числе возглавляемых | |
| | почетными консульскими | |
| | должностными лицами, | |
| | международных | |
| D 876 | (межгосударственных) | |
| | организаций и сотрудников, | |
| 27 77 | аккредитованных при МИД | |
| 23/1 | РФ; | |
| | тип 11 - для мотоциклов. | |
| | | |

Группа 5 - транспортные средства, принадлежащие органам внутренних дел Российской Федерации и Федеральной службе войск национальной гвардии Российской Федерации, кроме транспортных средств, отнесенных к группе 2;

| з; цля автомобильных и полуприцепов; | |
|--|--|
| | з; иля автомобильных и полуприцепов; |

| Изображение номерного знака | Описание | Шаблон в AutoTRASSIR |
|-----------------------------|---|-------------------------|
| 55 <u>37</u> A 99 | тип 22 - для мотоциклов, внедорожных мототранспортных средств | |
| Группа 6 - классичес | кие (ретро) и спортивные тран | спортные средства |
| K A0002 78 RUS | тип 23 - для классических (ретро) легковых и грузовых | |
| K MM | автомобилей; тип 24 - для классических | |
| 976 39 RUS | (ретро) легковых и грузовых автомобилей с | |
| K 133 | нестандартным местом крепления; | |
| [AA]77] | тип 25 - для классических (ретро) мотоциклов | |
| CA0002 78 RUS | тип 26 - для спортивных легковых и грузовых | |
| CMM | автомобилей; тип 27 - для спортивных | |
| 976 39 RUS | легковых и грузовых автомобилей с | |
| | нестандартным местом крепления; | |
| | тип 28 - для спортивных мотоциклов. | |

AutoTRASSIR работает как без шаблонов, так и с использованием собственных шаблонов автомобильных номеров, построенных на внутренних списках, а также шаблонов разных стран, построенных на внешних списках. Имеется возможность настроить внутренние списки, подключить внешние списки, хранящиеся во внешних текстовых файлах или базах данных.

AutoTRASSIR распознает номера автомобилей вне зависимости от подключения шаблона той или иной страны. Шаблоны используются только для отображения в интерфейсе оператора. То есть, если распознанный номер соответствует включенному шаблону страны, то в интерфейсе оператора он будет отображаться в формате этой страны:



В противном случае распознанный номер будет отображаться как набор распознанных знаков:

B663KT777

Подбор, установка и настройка камер для работы с модулем AutoTRASSIR

Рекомендуется использовать черно-белые камеры, так как они обладают большей (по сравнению с цветными камерами) разрешающей способностью и чувствительностью (изображение цветной камеры при распознавании преобразуется в черно-белое). Аналоговая камера видеонаблюдения должна обладать разрешением от 500 ТВЛ.

Основная проблема качества картинки при распознавании автомобильных номеров – скоростное смазывание (motion blur). Для того чтобы смазывания не происходило, выдержка (время экспозиции кадра, shutter) должна быть достаточно малой. Максимально допустимая выдержка зависит от скорости автомобиля и угла установки камеры (см. рис. 55). Под углом установки камеры понимается угол между оптической осью камеры и направлением движения автомобилей.



Рисунок 55 – Зависимость допустимой выдержки от скорости автомобиля и угла установки камеры [35]

Выдержку необходимо установить в фиксированное положение, либо (если позволяет камера) установить ограничение выдержки. Камера обязательно должна обладать функцией ручной установкой выдержки (shutter)! К примеру, для угла установки камеры в 20° и скорости 80 км/ч необходимо установить выдержку 1/1000 секунды (см. график).

При значительных углах установки камеры также нужно учитывать время проезда автомобиля через поле зрения камеры. Для достижения высокого качества распознавания камера должна снять 10 и более кадров с читаемым номером.

Прочие настройки камеры:

• автоматическая фокусировка должна быть отключена;

• любая информация, которая выводится на изображение (дата, имя камеры и т.д.), должна быть отключена;

• фокусное расстояние должно быть настроено так, чтобы номерной знак автомобиля на анализируемом видео занимал в ширину около 120-140 пикселей по горизонтали (см рис. 56).



Рисунок 56 – Пример изображения для работы модуля AutoTRASSIR

Необходимо убедиться в наличии достаточной освещенности. Для этого нужно записать небольшой фрагмент видео, при воспроизведении номер должен легко читаться. Если изображение номера оказывается слишком шумным или темным, нужно увеличить освещенность либо заменить объектив на более светосильный. Убедитесь также, что диафрагма объектива полностью открыта.

В реальных условиях работы модуля не рекомендуется устанавливать камеру на малой высоте, чтобы исключить ее засветку фарами проезжающих автомобилей. Если для контроля въезда/выезда используется шлагбаум, то камеру рекомендуется устанавливать таким образом, чтобы шлагбаум не доходил до низа экрана. В противном случае возможны ложные срабатывания на шлагбаум.

Порядок выполнения работы

Работа рассчитана на 4 часа и включает в себя следующие разделы:

- включение лабораторной установки и подготовка ее к работе;
- изучение функциональных возможностей и основных режимов работы модуля AutoTRASSIR:
 - настройка соединения с сервером для работы модуля;
 - настройка модуля распознавания номеров для каждого канала
- создание внутреннего списка номеров (базы данных) для работы модуля с собственным шаблоном;
- создание собственного шаблона для контроля работы модуля;
- проверка корректности настроек модуля для отображения на экране;
- выключение лабораторной установки.

І. Включение лабораторной установки и подготовка ее к работе производится в следующей последовательности:

- подать питание на видеокамеры;
- включить компьютер с помощью кнопки на его передней стенке;
- после включения компьютера автоматически запустится операционная система Windows и автоматически запускается предварительно инсталлированное ПО Trassir Server.

Для работы IP-камер с компьютерами Trassir все оборудование должно находиться в одной локальной подсети. Выбранная и настроенная подсеть имеет диапазон адресов 192.168.1.1-192.168.1.254 (соответствует маске 255.255.255.0).

II. Изучение функциональных возможностей и основных режимов работы модуля AutoTRASSIR

1. Настройка соединения с сервером для работы модуля

В настройках пользователя (клиента) должно быть разрешение сервера для работы аналитики через сеть. Рекомендуется следующая последовательность действий на сервере для разрешения клиенту использовать аналитику через сеть:

- открыть главную панель управления, нажав значок запуска главной панели управления ———, который находится по центру верхней границы экрана;
- открыть настройки TRASSIR, нажатием кнопки «Настройки» (рис. 57);



Рисунок 57 – Открытие окна настроек

• нажать стрелку напротив названия сервера, чтобы раскрыть дерево проекта (рис. 58);



Рисунок 58 – Активация дерева проекта

 перейти по адресу: Настройки сервера=> Облако Трассир=>Пользователи, где выбрать пользователя (клиента), которому необходимо разрешить аналитику через сеть (рис. 59);



Рисунок 59 – Пример выбора пользователя Artem

• поставить галочку напротив пункта «Разрешить аналитику через сеть» в рабочем окне (рис. 60).

| | | <u>И</u> мя пользователя: | |
|-------|---|--|-----------|
| | | Artem | |
| | | 📝 Разрешить локальный вход | |
| Co Co | > | 📝 Разрешить подключение с Trassir Server/Client | — Vasauri |
| | | 📝 Разрешить подключение с мобильного / из браузера | удалить |
| | | 🔽 Разрешить аналитику через сеть | |

Рисунок 60 – Делегирование полномочий пользователю

<

Общая настройка модуля AutoTRASSIR производится на сервере, к которому подключены камеры, а на сервере аналитики выбирается только Версия LPR. При использовании AutoTRASSIR версий LPR1 и LPR3, выбор сервера аналитики не производится.

2. Настройки модуля AutoTRASSIR для работы с каналами

Общая настройка на сервере включает несколько этапов (рис. 61):

- перед тем, как производить настройку модуля, необходимо убедиться, что подбор, установка и настройка камер для работы с модулем AutoTRASSIR были произведены корректно, а также корректно настроено подключение к Базе данных;
- на сервере перейти в пункт *Каналы* дерева настроек, выбрать нужный канал из списка и включить на выбранном канале *Распознаватель автомобильных номеров* в области *Программные детекторы*;
- обязательно проверить, чтобы в области Запись архива параметр На диски сервера имел значение Постоянная запись, либо По детектору;
- для активации модуля необходимо в Настройках канала в области Программные детекторы выбрать пункт Распознаватель автомобильных номеров, выбрать Сервер, который будет производить расчет аналитики.

| Запись архива | Программные детекторы | |
|---|--|---|
| На диски сервера: | Распаковывать: Автоматический выбор 💌 | • |
| Обычный канал 🔻 | Детектор движения: | |
| По детектору 👻 | Детектор активности 👻 Настроить зоны | |
| Перейти к расписанию 🕶 | ActiveSearch Kалибровка пола Kалибровка пола M Распознаватель автомобильных номеров <u>Настройка AutoTrassir</u> | |
| Состояние: Работает | В Детектор оставленных предметов ☐ Детектор огня/дыма ☐ Детектор саботажа | |
| Генерировать события о появлении движения | Детектор лиц | |

Рисунок 61 – Настройка модуля на сервере

3. Настройка основных параметров модуля на выбранном канале

Перейдите по активной ссылке *Настройка AutoTRASSIR* (рис. 61).

Настройки AutoTRASSIR на канале зависят от версии модуля, которая выбирается в настройке *Bepcus LPR* (рис. 62). После изменения версии модуля сервер необходимо перезагрузить.

Пункт Использование лицензий показывает количество подключенных на данный момент каналов (видеокамер) из максимально доступных (ограничивается лицензией). В графе Каналы отображается список каналов, на которых активирован модуль AutoTRASSIR.

| Атрана: Россия Каналы: ип ОСR: стандартный С Јаблоны для отображения Украина Турция Тайвань Россия Молдова Кыргызстан Казахстан Испания Грузия | Страна: Россия Каналы: Тип ОСR: стандартный Каналы: Шаблоны для отображения Ухраина Ухраина Тайвань Россия Молдова Кыргызстан Казахстан Испания Грузия Беларусь | Страна: Россия Каналы: Тип ОСR: стандартный Каналы: Шаблоны для отображения Украина Турция Тайвань Россия Молдова Кыргызстан Казахстан Испания Грузия Беларусь Внешние списки Добавить текстовый файл Добавить источник данных ODBC Протестировать все ODBC источники | Страна: Россия Каналы: Тип ОСR: стандартный Шаблоны для отображения Украина Турция Пайвань Россия Молдова Кыргызстан Калания Колария Грузия Беларусь Внешние списки Добавить текстовый файл Добавить источник данных ODBC Протестировать все ODBC источники | Версия LPR: | lpr4 | Только удаленно | Использование лицензии: 0/2 | |
|--|---|---|---|---|--|-----------------|-----------------------------|---|
| ип ОСR: стандартный Јаблоны для отображения Украина Турция Тайвань Россия Молдова Кыргызстан Казахстан Испания Грузия Бараксь | Гип ОСR: стандартный Шаблоны для отображения Уукраина Уукраина Лурция Тайвань Р Россия Молдова Кыргызстан Кыргызстан Кыргызстан Казахстан Испания Грузия Беларусь Внешние списки | Гип ОСR: стандартный Шаблоны для отображения Ухраина Уукраина Гурция Тайвань Россия Молдова Кыргызстан Колания Грузия Беларусь Внешние списки Добавить текстовый файл | Гип ОСR: стандартный Шаблоны для отображения Уукраина Турция Тайвань Россия Молдова Кыргызстан Кыргызстан Горзия Беларусь Внешние списки Добавить текстовый файл Добавить источник данных ODBC Протестировать все ODBC источники | Страна: | Россия | • | Каналы: | |
| Шаблоны для отображения Украина Турция Тайеань Россия Молдова Кыргызстан Казахстан Испания Грузия Грузия | Шаблоны для отображения Ухраина Турция Тайвань Россия Молдова Кыргызстан Казахстан Испания Грузия Беларусь Внешние списки | Шаблоны для отображения Ухраина Турция Тайвань Россия Молдова Кыргызстан Кыргызстан Гспания Грузия Беларусь Внешние списки Добавить текстовый файл Добавить источник данных ODBC Протестировать все ODBC источники | Шаблоны для отображения Ухраина Турция Тайвань Россия Молдова Кыргызстан Казахстан Испания Грузия Беларусь Внешние списки Добавить текстовый файл Добавить источник данных ODBC Протестировать все ODBC источники | Гип OCR: | стандартный 🔻 | 1 · · · · | | - |
| Украина Турция Тайвань Россия Молдова Кыртызстан Казахстан Испания Грузия Бараксь | Украина Турция Тайвань Россия Молдова Кыргызстан Казахстан Испания Грузия Беларусь Внешние списки | Украина Турция Тайвань Россия Молдова Кыргызстан Казахстан Испания Грузия Беларусь Внешние списки Добавить текстовый файл Добавить источник данных ODBC Протестировать все ODBC источники | Украина Турция Тайвань Россия Молдова Кыргызстан Казахстан Испания Грузия Беларусь Внешние списки Добавить текстовый файл Добавить источник данных ODBC Протестировать все ODBC источники | Шаблоны дл | ая отображения | | | |
| - Ferendary | Внешние списки Добавить текстовый файл Добавить историцик заниных ОДВС Поотестировать все ОДВС историцики | ✓ Беларусь Внешние списки Добавить текстовый файл Добавить источник данных ODBC Протестировать все ODBC источники | ✓ Беларусь Внешние списки Добавить текстовый файл Добавить источник данных ODBC Протестировать все ODBC источники | Укра Турц Тайв У Росс Мол Кырг Каза Испа Груз | ила ань ия дова гызстан хстан хстан ыня | | | |

Рисунок 62 – Настройка параметров версий модуля на сервере

<u>Настройка AutoTRASSIR (LPR3/LPR4)</u>

Настройка сводится к выбору количества зон распознавания и определение их границ. Во время настройки используйте следующие рекомендации:

• Количество полос. Выбирайте количество полос исходя из реальной ширины проезжей части, указывая ближайшее возможное значение.



Рисунок 62 – Настройка количества полос

Стандартная ширина полосы дороги принимается за 3,5 метра. К примеру, камера захватывает ширину дороги, равную восьми метрам (в этот размер включаются вся реальная ширина изображения в метрах, а не только дорожное полотно). В этом случае необходимо выбрать ближайшее значение количества полос «2».

• Определение границ зон распознавания. Выделяйте обособленные зоны, чтобы получать информацию о проезде автомобиля с привязкой к той или иной полосе дороги (контроль выделенных полос для маршрутного транспорта, детекция проезда автомобиля по тротуару и т.п.). Кроме этого, это уменьшит количество ложных срабатываний детектора и позволит экономить ресурсы сервера, анализируя только реально интересующие и пригодные для этого области изображения.

При выделении областей распознавания необходимо учитывать глубину резкости и то, какое количество кадров успеет снять камера за время проезда автомобиля внутри зоны. Необходимо помнить, что не все кадры смогут подойти для распознавания, изображение номера автомобиля должно быть чётким и хорошо различимым. В большинстве случаев достаточно получить 4-5 пригодных для распознавания кадров.

Проверить корректность настройки можно, включив *отображение* Фигур AutoTRASSIR (рис. 63). Для этого нажмите правой клавишей мыши на изображении, выберите в выпадающем меню пункт Bud..., поставьте флаг рядом с пунктом Показывать фигуры и выберите в выпадающем списке Детектор номеров.

| Общие настройки АТ | | | | |
|--------------------------------|-----------------|--|--|--|
| 🗌 Использовать общие настройки | | | | |
| 🗹 Имя канала 🔲 Жирным | | | | |
| 🗹 Показывать фигуры 🛛 Дете | ктор номеров 🔻 | | | |
| 🗌 Показывать фигуры в архиве | | | | |
| Видео поток | Автоматически 🔻 | | | |
| Порог размера окна | 400 пикс 🛨 | | | |
| Показывать границы | Только фокус 🔻 | | | |
| 🗌 Показывать OSD (кодек, разре | ешение, FPS) | | | |
| 🗹 Сохранять соотношение сторон | | | | |
| 🖂 Включить предпросмотр шка | лы архива | | | |
| Режим картинка в картинке п | ри увеличении | | | |

Рисунок 63 – Настройка отображения фигур

При правильных настройках модуля на экране отобразятся *фигуры AutoTRASSIR* как результат поиска и распознания номера (рис. 64).



Рисунок 64 – Пример отображения работы модуля

На данном рисунке изображение столбика при индикаторе «1» - *Очередь* обработки - отображает очередь обработки номеров. Если столбик заполняется и становится красным, то это значит, что CPU сервера сильно нагружен и AutoTRASSIR не успевает обрабатывать кадры и начинает пропускать номера.

Изображение прямоугольника при индикаторе «2» - *Качество* распознавания в кадре – формируется для каждого кадра, который был использован для определения номера. В зависимости от того, подошел кадр для распознавания или нет, цвет прямоугольника будет изменяться от зеленого ("хороший" кадр) до красного ("плохой" кадр).

<u>Настройка AutoTRASSIR (LPR1)</u>

Во время настройки Вы будете пользоваться следующими инструментами:

– *Предполагаемый размер номера* - расчетная область изображения (синий прямоугольник на рисунке), размеры которой будут использованы для распознавания номера;

- Зоны распознавания - области изображения, в которых будет происходить распознавания номеров.

В зависимости от сцены выберите в выпадающем списке *Количество полос* необходимое значение (рис. 65). На экране появится соответствующее количество областей Зон распознавания. Выбор количества полос производится по тем же рекомендациям, что и для версии LPR3/LPR4. В настройке *Разрешение детектора номеров* оставьте значение по умолчанию - Полный размер!

Сопоставьте *предполагаемый размер номера* (вы можете свободно перемещать иконку по экрану) с реальным изображением номера автомобиля в кадре. Могут быть три возможные ситуации при сопоставлении.

1) Если реальный размер номера автомобиля на изображении не сильно отличается от *предполагаемого*, можно увеличить или уменьшить изображение автомобиля в кадре, изменив фокусное расстояние объектива камеры. Дополнительно можно попробовать изменить ракурс обзора за счет изменения угла наклона камеры или высоты\места ее установки.



Рисунок 65 – Окно для настройки модуля

- 2) Если реальный размер номера автомобиля на изображении намного больше предполагаемого размера номера, можно использовать опшию Предустановка к настройке Разрешение детектора номеров. В этом случае картинки будет пережато оптимальным способом. с разрешение минимальными потерями качества и минимальной дополнительной нагрузкой на CPU сервера. Данная ситуация возможна, если для отслеживания довольно узкого участка дороги используется камера с высоким разрешением. Обратите внимание, что предполагаемый размер номера в этом случае зависит от значения параметра Количество полос.
- 3) Если реальный размер номера автомобиля на изображении намного меньше *предполагаемого размера номера*, а настройками фокусного расстояния объектива и изменением места установки/угла наклона камеры это не устраняется, то возможно, что для распознавания номеров используется камера с недостаточными для данной сцены разрешением.

В случае, если на предыдущих этапах настройки не удалось добиться полного сопоставления реального размера номера автомобиля с *предполагаемым размером номера*, то можно задать предполагаемый размер номера вручную.

Для этого в настройке *Разрешение детектора* номеров выберите значение *Пользовательский (указать размер)* (рис. 66). На выбранном кадре измените *предполагаемый размер номера* так, чтобы он точно совпал с реальным изображением номера автомобиля.



Рисунок 66 – Пример выбора пользовательского номера

Использование параметра *Пользовательский (указать размер)* увеличивает нагрузку на CPU сервера. Кроме того, при пережатии изображения в произвольный размер могут возникать артефакты сжатия, что отрицательно скажется на качестве распознавания. Используйте данный вариант настройки только если предыдущие варианты не помогли.

Ш. Создание внутренней базы данных автомобильных номеров

1. Выбор страны для формирования шаблона номеров

В общей настройке Страна (рис. 67) можно выбрать название страны, в которой будет производиться распознавание автомобильных номеров. Для повышения качества распознавания китайских автомобильных номеров в настройке *Tun OCR* должен быть выбран **Китайский**. Во всех остальных случаях можно использовать **стандартный** тип OCR.

| Настройка Справка | |
|--|--|
| Версия LPR: Ірг4 Только удаленно Страна: Россия Т Тип OCR: стандартный Т | Использование лицензии: 0 / 2 Каналы: |
| Шаблоны для отображения Украина Турция Тайвань Россия Молдова Кыргызстан Казахстан Испания Грузия Беларусь | |
| Внешние списки Добавить текстовый файл Добавить источник данных ODBC | Протестировать все ODBC источники |

Рисунок 67 – Настройка шаблона для отображения номера

При этом в списке Шаблоны для отображения будут выбраны шаблон этой страны и шаблоны стран, граничащих с ней. Модуль AutoTRASSIR может использовать как встроенные списки номеров, хранящиеся в собственной базе данных, так и внешние списки автомобильных номеров из баз данных.

В области *Внешние списки* отображается список всех подключенных внешних списков AutoTRASSIR. Для их формирования используются специальным образом составленные внешние текстовые файлы или программный интерфейс ODBC (Open Database Connectivity) для доступа к базам данных.

2. Ведение внутреннего списка номеров

Модуль AutoTRASSIR может использовать встроенные списки номеров, хранящиеся в собственной базе данных. Если хранящийся во встроенном списке номер будет распознан, то AutoTRASSIR сформирует сообщение (рис. 68) в соответствии с настройками, указанными для данного номера.



Рисунок 68 – Сообщение при распознавании мотоциклетного номера

Существует 2 способа добавления номера во внутренний список AutoTRASSIR:

1). Добавление распознанного номера из журнала AutoTRASSIR (рис. 69): В области Журнал AutoTRASSIR кликнуть правой клавишей на нужный номер и выбрать пункт Добавить во встроенный список. Откроется закладка Списки.

| Режим р | аботы: | Живой Архив | | |
|-------------------|--|-------------------|-----------------|------------|
| ¢ | ильтр: [| Показать все | | • 🔊 🖻 |
| Номер | | Время | Канал | |
| -1A5 | - 197 | 18-71-11 | AT-highur | au 2 |
| Н59 Найт | | ги "о185bo197" | | |
| B18 | Испр | равить номер | | |
| H & 1 | Print | | | |
| . 2 | | | | 6 |
| 1 3 | Откр | рыть архив за 26. | 12.2016 18:21:1 | 1 |
| | Открыть архив во весь экран за 26.12.2016 18:21:11 | | | |
| . 5 | UIK | | | |
| · 5 • 2 | Unq | | | |
| + 5 + 2 < 5 | Доба | ввить "0185b019 | 7" во встроени | чый список |

Рисунок 69 – Пример добавления номера из журнала

2). Добавление нового номера непосредственно в список (рис. 70): В области Журнал AutoTRASSIR открыть закладку *Списки* и выполнить двойной щелчок мышью на поле *Добавить*.

| | Номер | Реакция | Комментарий |
|---|-------------|---------------|-------------|
| 1 | o185bo197 | чёрный список | |
| 2 | y685th36 | чёрный список | |
| 3 | добавить но | | |

Рисунок 70 – Пример прямой записи номера список

В списке можно изменить следующие параметры номера (рис. 70, 71):

• Номер – автомобильный номер с использованием только латинских букв;

• Реакция – тип реакции из выпадающего списка ("черный список", "белый список" или "информация");

• Комментарий – сообщение, выдаваемое оператору при распознавании данного номера.

| | Номер | Реакция | Комментарий |
|---|-------------|--------------|------------------------|
| 1 | 5555ok?3 | информация | Военный |
| 2 | 9329ph61 | белый список | Мотоциклетный |
| 3 | a000077 | информация | Полиция |
| 4 | aa00050 | информация | Общественный транспорт |
| 5 | B448PM178 | белый список | Машина на улице |
| 6 | m976mm34 | белый список | Тестовый номер |
| 7 | добавить но | | |

Встроенный список:

Рисунок 71 – Пример встроенного списка

Чтобы удалить номер из списка, выделите его и нажмите кнопку Delete на клавиатуре.

IV. Формирование собственного шаблона AutoTRASSIR

Для того, чтобы начать работать со встроенными списками модуля, Вам необходимо предварительно сформировать простой шаблон AutoTRASSIR. Это

позволит проверить корректность настройки и работоспособность модуля. Для создания собственного шаблона:

1. Откройте Главную панель управления и отобразите Видеомонитор на одном из экранов сервера.

2. Нажмите на кнопку Редактор шаблонов и выберите пункт Новый шаблон.

3. Нажмите кнопку +Добавить AutoTRASIR.

4. В свободную область перетащите из списка каналов сигнал с камеры, который обрабатывается модулем AutoTRASSIR.

5. Нажмите кнопку Сохранить как....

В открывшемся окошке введите имя нового шаблона и нажмите кнопку ОК. При предъявлении камере изображения с номером (рис. 72) распознанный номер появится в журнале сформированного шаблона.



Рисунок 72 – Пример созданного шаблона AutoTRASSIR

V. Проверка правильности настройки и корректности работы модуля AutoTRASSIR с помощью составленного шаблона.

1. Скопируйте из Таблицы 5 на смартфон и последовательно предъявите камере изображения регистрационных номеров на экране монитора.

2. Настройте камеру таким образом, чтобы номер на экране был различаем.

3. Выберете номер типа 1 - регистрационные знаки для легковых, грузовых автомобилей и автобусов.

4. Проверьте правильность настройки модуля AutoTRASSIR: если все настроено верно, то при распознавании номер появится в журнале AutoTRASSIR (рис. 73), соответствующий изображению, установленному шаблоном страны для данного типа номеров.

| TRASSI | ₹ 👓 📼 💷 🎢 📩 | Охрана 🖺 | |
|---|---|---|--|
| Журнал Поиск Режим работы: Фильтр: Показат | Списки Жи <u>е</u> ой <u>А</u> рхив в все "?" при неузеренной детекции [] | Архив Фильтр по времени: © Предустановка: Цас © Конкретное время: | |
| Номер Время м976 мм 34 12 м976 мм 34 13 М976 мм 34 12 М976 мм 34 12 мм 34 1 | Kanan - 53:57 10:00:00:00:00:00:00:00:00:00:00:00:00:0 | Начало: 6.03.2020 1 14.45.40 1 Конец: 6.03.20 1 15.45.40 1 Фильтр по вызовам: Входящие D Звонка: Длительность: 00:00:00 1 Свернуть 0 Гоиск Все выодящие № Исходящие № Протущенные От Кому Дата Длительность | |
| Номер: М 976 Канал: <u>hDQ1owG0_in</u> Шаблон: ти/тих3хх2-v | Найден в списках: Белый списка: Номер Эмклус Время: 06 Mar 2020 13:53:02 1 Качество: 99.9 | Управление отображением ↑ 30 14 30 15 30 16 № 1 1453:12.320 ¥ | |

Рисунок 73 – Пример правильного отображения на экране номера типа 1

5. Если при проезде автотранспорта (или другом способе предъявления номера) распознанные номера не появляются, то проверьте:

• стоит ли галочка Распознаватель автомобильных номеров в области Программные детекторы настроек канала;

• настроено ли подключение к Базе данных.

6. Если в журнале номер отображен правильно (в блоке с информацией отображается правильный шаблон), сделайте скриншот экрана с исследуемым номером и журналом, в котором видно правильно отображенный шаблон.

7. Повторите п.п. 3 – 6 для номеров типов 1Б, 4, 5, 20 из Таблицы 5.

На рисунках 74–76 приведены примеры скриншотов экрана, полученных при проверке правильности настройки шаблона модуля AutoTRASSIR:

| Журнал Поиск Списки Ас-D4121R1 (Поголоч | ная справа) @ А457-11 🔍 |
|--|-------------------------|
| Режим работы: Жидой Дрхив Фильтр по времени: | |
| Фильтр: Показать все 🔹 🖍 💼 🐵 Предустановка: Цас 💌 | |
| "?" при неуверенной детекции 📃 💿 Конкретное время: | |
| Начало: 06.03.2020 🕂 14.48.53 🕂 | |
| Номер Время Канал А Ка | |
| HEI HOMEPA T 152/30 hDU1ow/dumminkuvc KOHEL; 00.05.20 - 1346035 - | |
| ИНЕТ НОМЕРА 1 Б25363 ПООТОИМО ПЛЯМКИИ ФИЛЬТР ПО ВЫЗОВАМ: | |
| HET HOMEPA 15:24:57 hD01ov(60 im3mknvc | vr vrakter |
| HET HOMEPA ↑ 15:24:57 hDQ1owG0_im3mknvc | |
| 5555 ок 43 ↑ 15:22:28 hDQ1 окG0 im3mkmvc Исходящие: | |
| 55550 К43 ↓ 15:22:18 hDQ1owG0_im3mknvc ID Звонка: | |
| НЕТ НОМЕРА 15:22:16 hDQ1owG0_im3mknvc Длительность: 00:00:00 ÷ | A Sec |
| SSSS0K43 + IS2211 hDQ1.owG0_im3mknvc | |
| нет номеря т 152203 подолжитис Свернуть Свернуть Свернуть | 55 OK 13 |
| | |
| Все С Входящие С Исходящие С Пропущенные | |
| Управлять отображением + Лоравить солытие 1 От Кому Дата Длительность | |
| Номер: Найден в списках: | |
| | |
| информация | |
| | |
| | |
| СПС С с с с 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 | |
| 55550K ⁻¹³ | |
| | |
| | |
| | |
| | |
| Kanar: hDO1owG0 im3mknyc BoeM8: 06 Mar 2020 15:22:28 | |
| 30 16 | 30 17 30 18 |
| Wabnoh: Tu/Tu4xx2-v2 Kavecras: 0.0 | Пауза 16:22:28.794 🗱 |



| TRASSIR | 00 📖 💷 🏂 📩 | Охрана 🖺 | |
|--|--|---|-----------------------|
| Журнал Поиск Списи Режим работы: Жив Жив Фильтр: Показать все Номер Время | ки Дрхив 🖉 📅 🧭 📅 ??" при неуверенной детекции | Архив Фильтр по времени: © Предустановка: Час © Конкретное время: Начало: 06.03.2020 - 1 14-51:17 - 2 по по по составля с на по составля | Q 2 2 2 3 |
| HET HOMEPA ↓ 143953 HET HOMEPA ↓ 143803 HET HOMEPA ↓ 143739 HET HOMEPA ↓ 143727 HET HOMEPA ↓ 143727 HET HOMEPA ↓ 143659 HET HOMEPA ↓ 1436512 HET HOMEPA ↓ 1436512 HET HOMEPA ↑ 143345 55550K43 ↓ 142730 55550K43 ↓ 1427325 | hDQ1owG0_im3mknvc hDQ1owG0_im3mknvc hDQ1owG0_im3mknvc hDQ1owG0_im3mknvc hDQ1owG0_im3mknvc hDQ1owG0_im3mknvc hDQ1owG0_im3mknvc hDQ1owG0_im3mknvc hDQ1owG0_im3mknvc hDQ1owG0_im3mknvc hDQ1owG0_im3mknvc hDQ1owG0_im3mknvc hDQ1owG0_im3mknvc hDQ1owG0_im3mknvc | Конец: 06.03.20 = 15:51:17 = Фильтр по вызовам: Входящие Исходящие 10 Звонка: Длительность: 00:00:00 = Свернуть Поиск Все Кодящие К Исходящие К Пропущенные | |
| Номер: АОС Канал: hDQ1owG0_im3mkm Шаблон: гu / rux6-v1 | ооо 77 к Время: Об Маг 2020 14:36:59 Качество: 100.0 | ОГ КОМУ Дата Длительность Управление отображением | |

Рисунок 75 – Пример правильного отображения номера типа 20



Рисунок 76 – Пример неверного отображения номера (шаблон дипломатического номера не распознан)

Сделайте и сохраните для отчета скриншоты, отображающие последовательность Ваших действий по изучению функциональных возможностей и основных режимов работы модуля AutoTRASSIR и результаты создания шаблона для распознавания номеров.

IV. Выключение лабораторной установки выполняется в последовательности, обратной ее включению.

Контрольные вопросы

1. В чем состоят особенности обработки изображений для решения задач поиска и распознавания автомобильных номеров?

2. Какова структура и функционал модуля распознавания регистрационных знаков транспортных средств в программе Trassir?

3. Каковы требования к параметрам и условиям эксплуатации видеокамер при организации работы с модулем AutoTRASSIR?

4. Поясните алгоритм настройки основных режимов работы модуля видеоанализа в системе Trassir. Как распределяются функции между сервером и компьютером пользователя?

5. От чего зависят конфигурация и размер зон детекции и распознавания, формируемых при настройках модуля AutoTRASSIR на канале?

СПИСОК ИСПОЛЬЗОВАННЫХ ИСТОЧНИКОВ

- 1. Дамьяновски Владо Библия видеонаблюдения, 3-е издание: Пер. с англ. / Владо Дамьяновски. М.: «Секьюрити фокус», 2017. 422с.: ил. (Серия «Энциклопедия безопасности»). ISBN 978-5-9901176-7-9
- Кругль Герман Профессиональное видеонаблюдение. Практика и технологии аналогового и цифрового ССТV, 2-е изд.: Пер. с англ. - М.: Секьюрити Фокус (Security Focus), 2019. – 626 с.: ил. ISBN 978-5-9901176-2-4
- 3. TADVISER- портал выбора технологий и поставщиков. Государство. Бизнес. IT. Статья «Видеонаблюдение (мировой рынок)» [Электронный pecypc]. URL: <u>https://www.tadviser.ru/</u> (дата обращения 01.11.2020).
- 4. Образовательный стандарт высшего образования Санкт-Петербургского национального исследовательского университета информационных технологий, механики и оптики магистратура, по объединенной группе направлений подготовки «Фотоника». Утвержден с изменениями и дополнениями приказом ректора от 3 сентября 2019г. №939-од [Электронный ресурс]. URL: <u>https://edu.itmo.ru/files/41 /</u> (дата обращения 01.11.2020).
- 5. Возможности систем видеоаналитики Trassir [Электронный ресурс]. URL: <u>https://www.dssl.ru/possibility</u> / (дата обращения 18.11.2020).
- OpenCV 3.x with Python By Example. Second Edition. Make the most of OpenCV and Python to build applications for object recognition and augmented reality. Gabriel Garrido Prateek Joshi BIRMINGHAM — MUMBAI 2018 Packt Publishing
- 7. OpenCV-Python-Глава 22: Алгоритм водораздела для сегментации изображений [Электронный ресурс]. URL: https://russianblogs.com/article/6325822971/ (дата обращения 28.11.2020).
- 8. HSV (цветовая модель) [Электронный ресурс]. URL: <u>https://ru.wikipedia.org/wiki/HSV_/</u> (дата обращения 28.11.2020).
- Image Segmentation Using Color Spaces in OpenCV + Python [Электронный pecypc]. URL: <u>https://realpython.com/python-opencv-color-spaces/</u> (дата обращения 28.11.2020).
- 10. Как определить HSV цвет в OpenCV (python) [Электронный ресурс]. URL: <u>http://zabaykin.ru/?p=31(д</u>ата обращения 30.11.2020).
- 11. Harris Corner Detection [Электронный ресурс]. URL: <u>https://opencv-python-tutroals.readthedocs.io/en/latest/py_tutorials/py_feature2d/py_features_harris/py_fe</u>
- 12. Shi-Tomasi Corner Detector & Good Features to Track [Электронный ресурс]. – URL: <u>https://opencv-python-</u> <u>tutroals.readthedocs.io/en/latest/py_tutorials/py_feature2d/py_shi_tomasi/py_shi</u> <u>_tomasi.html#shi-tomasi (дата обращения 12.12.2020).</u>

- 13. Построение SIFT дескрипторов и задача сопоставления изображений [Электронный ресурс]. URL: <u>https://habr.com/ru/post/106302/(дата обращения 21.12.2020)</u>.
- 14. Introduction to SIFT (Scale-Invariant Feature Transform) [Электронный
pecypc].–URL:<u>https://opencv-python-
tutroals.readthedocs.io/en/latest/py_tutorials/py_feature2d/py_sift_intro/py_sift_i
ntro.html#sift-intro(дата обращения 21.12.2020).</u>
- 15. Introduction to SURF (Speeded-Up Robust Features). [Электронный ресурс]. URL: <u>https://opencv-python-</u> <u>tutroals.readthedocs.io/en/latest/py_tutorials/py_feature2d/py_surf_intro/py_surf</u> <u>intro.html (д</u>ата обращения 21.12.2020).
- 16. ORB (Oriented FAST and Rotated BRIEF) [Электронный ресурс]. URL: <u>https://opencv-python-</u> <u>tutroals.readthedocs.io/en/latest/py_tutorials/py_feature2d/py_orb/py_orb.html#o</u> rb(дата обращения 18.01.2021).
- 17. ГОСТ Р ИСО/МЭК 18004-2015 Группа П85 Национальный стандарт Российской Федерации. Информационные технологии. Технологии автоматической идентификации и сбора данных. Спецификация символики штрихового кода QR Code
- 18. Zbar 0.10. Project description [Электронный ресурс]. URL: <u>https://pypi.org/project/zbar/</u> (дата обращения 18.04.2021).
- 19. Adrian Rosebrock An OpenCV barcode and QR code scanner with ZBar. May 21, 2018 [Электронный pecypc]. URL: <u>https://www.pyimagesearch.com/2018/05/21/an-opencv-barcode-and-qr-code-scanner-with-zbar/</u> (дата обращения 18.04.2021).
- 20. Open Source Computer Vision. QRCodeDetector Class Reference. [Электронный pecypc]. – URL: <u>https://docs.opencv.org/master/de/dc3/classcv_1_1QRCodeDetector.html</u> (дата обращения 18.04.2021).
- 21. ГОСТ Р ИСО/МЭК 19794-5-2006 Автоматическая идентификация. Идентификация биометрическая. Форматы обмена биометрическими данными. Часть 5. Данные изображения лица.
- 22. Дмитрий Азаров Метод распознавания лиц Виолы-Джонса (Viola-Jones) Sat, Apr 11, 2015 [Электронный ресурс]. – URL: <u>https://oxozle.com/2015/04/11/metod-raspoznavaniya-lic-violy-dzhonsa-viola-jones/(дата обращения 28.04.2021)</u>.
- 23. P Viola P., Jones M.J. Rapid Object Detection using a Boosted Cascade of Simple Features. Proc. IEEE Conf. on Computer Vision and Pattern Recognition (CVPR 2001), 2001, vol. 1, p. 511–518.
- 24. Анализ существующих подходов к распознаванию лиц. Блог компании Синезис. Работа с видео. 2014 [Электронный ресурс]. URL:
<u>https://habr.com/ru/company/synesis/blog/238129/</u> (дата обращения 08.05.2021).

- 25. Обучаем вместе с Tesseract OCR [Электронный ресурс]. URL: <u>https://favicon.tech/obuchaem-vmeste-s-tesseract-ocr/(дата</u> обращения 18.05.2021).
- 26. tesseract Documentation [Электронный ресурс]. URL: <u>https://tesseract-ocr.github.io/tessapi/4.0.0/index.html(дата обращения 18.05.2021)</u>.
- 27. Russian Car Plate Detection with OpenCV and Tesseract OCR [Электронный pecypc]. URL: <u>https://towardsdatascience.com/russian-car-plate-detection-with-opencv-and-tesseractocr-dce3d3f9ff5c(дата обращения 25.05.2021).</u>
- 28. Разработка библиотеки распознавания российских автомобильных номеров [Электронный pecypc]. URL: <u>https://kostyakulakov.ru/opencv/(дата обращения 28.05.2021)</u>.
- 29. ГОСТ Р 50577-2018 Знаки государственные регистрационных транспортных средств. Типы и основные размеры. Технические требования Licence plates for vehicles. Types and main sizes. Technical [Электронный pecypc]. URL: <u>https://docs.cntd.ru/document/1200160380</u> (дата обращения 28.05.2021).
- З0. Телевизионная система наблюдения и регистрации Тайфун. Описания процессов для поддержание жизненного цикла ПО. г. Санкт-Петербург, 2016. [Электронный ресурс]. – URL: <u>https://evs.ru/kat_podr.php?kat=1_1</u> (дата обращения 01.06.2021).
- 31. Телевизионная система наблюдения и регистрации «Тайфун». Руководство по установке и настройке программного обеспечения. ЭВС1.131.779-РУ. СПб, 2016. [Электронный ресурс]. URL: <u>https://evs.ru/dload/manual.pdf(дата обращения 01.06.2021).</u>
- 32. С.Н. Ярышев Видеоаналитика и цифровая обработка видеосигнала Методические указания по выполнению лабораторных работ Санкт-Петербург 2011.
- 33. Возможности систем видеоаналитики Trassir [Электронный ресурс]. URL: <u>https://www.dssl.ru/possibility</u> / (дата обращения 18.06.2021).
- 34. Рыжова В.А., Ярышев С.Н., Коротаев В.В. Интеллектуальные системы видеонаблюдения. Учебное пособие. СПб: Университет ИТМО, 2020. 112 с.
- 35. Руководство администратора Trassir [Электронный pecypc]. URL: <u>https://www.dssl.ru/bitrix/redirect.php?event1=countFiles&event2=TrassirManu</u> <u>al_ru_Administrator_4.1.pdf&event3=dsslru&goto=/upload/iblock/f7d/TrassirM</u> <u>anual_ru_Administrator_4.1.pdf</u> (дата обращения 03.06.2021).
- 36. Руководство оператора Trassir [Электронный pecypc]. URL: <u>https://www.dssl.ru/bitrix/redirect.php?event1=countFiles&event2=TrassirManu</u>

al_ru_Operator_4.1.pdf&event3=dsslru&goto=/upload/iblock/fa6/TrassirManual _ru_Operator_4.1.pdf (дата обращения 03.06.2021).

- 37. Свирин И., Ханин А. Некоторые аспекты автоматического распознавания автомобильных номеров // Алгоритмы безопасности. 2010. № 3. С. 26–29.
- 38. Shin Shi. Emgu CV Essentials. Packt Publishing, 2013, 105 p.
- 39. Гонсалес Р., Вудс Р. Цифровая обработка изображений. 3-е изд., испр. и доп. М.: Техносфера, 2012. 1104 с.
- 40. Кирпичников А.П., Ляшева С.А., Обухов А.В., Шлеймович М.П. Автоматическое распознавание автомобильных номеров // Вестник Казанского технологического университета. 2015. Т. 18, № 4. С. 218–222.
- 41. *Le Cun Y., Bengio Y.* Convolutional networks for images, speech and time series // The handbook of brain theory and neural networks. 1998. Vol. 7. P. 255–258.
- 42. *Мурыгин К.В.* Нормализация изображения автомобильного номера и сегментация символов для последующего распознавания // Искусственный интеллект. 2010. № 3. С. 364–369.
- 43. А.В. Обухов, С.А. Ляшева, М.П. Шлеймович Методы распознавания автомобильных номеров // Вестник Чувашского университета. 2016. № 3 с. 201-208.

Ярышев Сергей Николаевич Рыжова Виктория Александровна Коротаев Валерий Викторович

Видеоанализ в цифровых видеоинформационных системах безопасности

Учебное пособие

В авторской редакции Редакционно-издательский отдел Университета ИТМО Зав. РИО Н.Ф. Гусарова Подписано к печати Заказ № Тираж Отпечатано на ризографе

Редакционно-издательский отдел Университета ИТМО 197101, Санкт-Петербург, Кронверкский пр., 49, литер А