

МИНОБРНАУКИ РОССИИ



Федеральное государственное автономное образовательное учреждение
высшего образования
«Российский государственный гуманитарный университет»
(ФГАОУ ВО «РГГУ»)

ФАКУЛЬТЕТ «ИНФОРМАЦИОННЫХ СИСТЕМ И БЕЗОПАСНОСТИ»
Кафедра «Информационных технологий и систем»

КОМПЬЮТЕРНОЕ ЗРЕНИЕ

РАБОЧАЯ ПРОГРАММА ДИСЦИПЛИНЫ

Направление подготовки 09.03.03 «Прикладная информатика»
Направленность (профиль): Прикладной искусственный интеллект

Уровень высшего образования: бакалавриат

Форма обучения: очная

РПД адаптирована для лиц
с ограниченными возможностями
здоровья и инвалидов

Москва 2026

КОМПЬЮТЕРНОЕ ЗРЕНИЕ

Рабочая программа дисциплины

Составитель:

к.х.н., с.н.с., доцент А.М. Подорожный

УТВЕРЖДЕНО

Протокол заседания кафедры

Информационных технологий и систем

№ 5 от 11.12.2025 г.

ОГЛАВЛЕНИЕ

КОМПЬЮТЕРНОЕ ЗРЕНИЕ	2
1 Пояснительная записка.....	4
1.1. Цель и задачи дисциплины.....	4
1.2. Перечень планируемых результатов обучения по дисциплине, соотнесенных с индикаторами достижения компетенций.....	4
1.3. Место дисциплины в структуре образовательной программы.....	5
2 Структура дисциплины.....	5
3 Содержание дисциплины	5
4 Образовательные технологии.....	6
5 Оценка планируемых результатов обучения.....	6
5.1. Система оценивания.....	6
5.2. Критерии выставления оценки по дисциплине.....	7
5.3. Оценочные средства (материалы) для текущего контроля успеваемости, промежуточной аттестации обучающихся по дисциплине.....	8
6 Учебно-методическое и информационное обеспечение дисциплины.....	9
6.1. Список литературы.....	9
6.2. Перечень ресурсов информационно-телекоммуникационной сети «Интернет».....	10
6.3. Профессиональные базы данных и информационно-справочные системы.....	10
7 Материально-техническое обеспечение дисциплины.....	10
8 Обеспечение образовательного процесса для лиц с ограниченными возможностями здоровья и инвалидов.....	11
9 Методические материалы.....	12
9.1. Планы практических заданий.....	12
Приложение 1. Аннотация дисциплины.....	19

1 Пояснительная записка

1.1. Цель и задачи дисциплины

Дисциплина «Компьютерное зрение» имеет целью теоретическое и практическое освоение методов и технологий компьютерного зрения.

Задачи:

- выработка у студентов знаний о базовых принципах и алгоритмах компьютерного зрения;
- приобретение умений реализовывать основные алгоритмы компьютерного зрения в виде программ и применять их для решения практических задач;
- овладение терминологическим аппаратом и практическими навыками в области компьютерного зрения.

1.2. Перечень планируемых результатов обучения по дисциплине, соотнесенных с индикаторами достижения компетенций

Компетенция	Индикаторы компетенций	Результаты обучения
ПК-1. Способен классифицировать и идентифицировать задачи искусственного интеллекта, выбирать адекватные методы и инструментальные средства решения задач искусственного интеллекта	ПК-1.1. Классифицирует и идентифицирует задачи систем искусственного интеллекта в зависимости от особенностей проблемной и предметной областей	Знать классификацию изображений, приложения и архитектуру нейронных сетей для определения рассматриваемых изображений.
	ПК-1.2. Выбирает методы и инструментальные средства искусственного интеллекта для решения задач в зависимости от особенностей проблемной и предметной областей	Уметь выбирать и реализовывать наилучшие методы для оптимального решения поставленных задач компьютерного зрения.
ПК-2. Способен разрабатывать и тестировать программные компоненты решения задач в системах искусственного интеллекта	ПК-2.1. Разрабатывает приложения систем искусственного интеллекта	Уметь разрабатывать программные компоненты для решения задач в области компьютерного зрения.
	ПК-2.2. Проводит тестирование систем искусственного интеллекта	Владеть навыками тестирования создаваемых систем компьютерного зрения.
ПК-3. Способен разрабатывать и применять методы и алгоритмы машинного обучения для решения задач	ПК-3.1. Проводит анализ требований и определяет необходимые классы задач машинного обучения	Знать методы машинного обучения в области компьютерного зрения, проводить анализ требований, и определения необходимого класса задач.
	ПК-3.2. Принимает участие в оценке, выборе и при необходимости разработке методов и алгоритмов машинного обучения	Уметь осуществлять оценку и выбор методов машинного обучения, разрабатывать некоторые методы и алгоритмы.
ПК-8. Способен создавать и внедрять	ПК-8.2. Участвует в реализации проектов в области сквозной	Владеть навыками, необходимыми для участия в

одну или несколько сквозных цифровых субтехнологий искусственного интеллекта	цифровой субтехнологии «Компьютерное зрение»	реализации проектов сквозной цифровой субтехнологии «Компьютерное зрение»
--	--	---

1.3. Место дисциплины в структуре образовательной программы.

Дисциплина «Компьютерное зрение» является дисциплиной части учебного плана, реализуемой участниками образовательных отношений, по направлению подготовки 09.03.03 Прикладная информатика, профиль: Прикладной искусственный интеллект. Дисциплина реализуется на факультете Информационных систем и безопасности кафедрой Информационных технологий и систем.

Для освоения дисциплины необходимы знания, умения и владения, сформированные в ходе изучения следующих дисциплин: «Линейная алгебра и аналитическая геометрия», «Дискретная математика и математическая логика», «Программирование», «Основы обработки изображений».

В результате освоения дисциплины формируются знания, умения и владения, необходимые для изучения следующих дисциплин: «Моделирование и оптимизация на основе искусственного интеллекта», «Мультимедиа технологии в сфере искусственного интеллекта», «Производственная (технологическая (проектно-технологическая)) практика», «Производственная (преддипломная) практика».

2 Структура дисциплины

Общая трудоёмкость дисциплины составляет 4 з.е., 144 академических часов.

Структура дисциплины для очной формы обучения

Объем дисциплины в форме контактной работы обучающихся с педагогическими работниками и (или) лицами, привлекаемыми к реализации образовательной программы на иных условиях, при проведении учебных занятий:

Семестр	Тип учебных занятий	Количество часов
6	Лекции	20
6	Практические занятия	36
Всего:		56

Объем дисциплины в форме самостоятельной работы обучающихся составляет 88 академических часов.

3 Содержание дисциплины

№	Наименование раздела дисциплины	Содержание
1.	Компьютерное Зрение и классификация изображений.	Задачи компьютерного зрения. Приложения компьютерного зрения. Применение машинного обучения в компьютерном зрении. Классификация изображений. Постановка задачи классификации изображений. Архитектура нейронной сети для классификации изображений.

2.	Локализация и определение границ объекта на изображении.	Постановка задачи локализации объекта на изображении. Архитектура нейронной сети для локализации объекта на изображении. Постановка задачи определения границ объекта на изображении. Фильтр Собеля. Определение границ объектов на изображении с помощью нейронных сетей.
3.	Методы оптимизации изображений.	Метод Non-Maximum Suppression. Эффективный алгоритм подавления локальных немаксимумов. Оценки вычислительной сложности для метода NMS. Преобразование Хафа. Обнаружение геометрических примитивов на изображении с помощью преобразования Хафа. Оптический поток. Метод оценки оптического потока Лукаса-Канаде. Применения методов оценки оптического потока для отслеживания объекта на видео. Сегментация изображений. Постановка задачи сегментации изображений. Сегментация изображений с помощью сверточных нейронных сетей. Архитектура U-Net.
4.	Генерация реалистичных изображений.	Генеративно-состязательные сети для генерации реалистичных изображений. Архитектуры генератора и дискриминатора. Повышение стабильности обучения генеративно-состязательных сетей.

4 Образовательные технологии

Для проведения учебных занятий по дисциплине используются различные образовательные технологии. Для организации учебного процесса может быть использовано электронное обучение и (или) дистанционные образовательные технологии.

5 Оценка планируемых результатов обучения

5.1. Система оценивания

Форма контроля	Макс. количество баллов	
	За одну работу	Всего
Текущий контроль:		
- опрос-коллоквиум	5 баллов	20 баллов
- практические задания	5 баллов	20 баллов
- самостоятельные работы	5 баллов	20 баллов
Зачет с оценкой по билетам		40 баллов
Итого за семестр		100 баллов

Полученный совокупный результат конвертируется в традиционную шкалу оценок и в шкалу оценок Европейской системы переноса и накопления кредитов (European Credit Transfer System; далее – ECTS) в соответствии с таблицей:

100-балльная шкала	Традиционная шкала	Шкала ECTS	
95 – 100	Отлично Хорошо Удовлетворительно	зачтено	A
83 – 94			B
68 – 82			C
56 – 67			D

50 – 55			E
20 – 49	Неудовлетворительно	не зачтено	FX
0 – 19			F

5.2. Критерии выставления оценки по дисциплине

Баллы/ Шкала ECTS	Оценка по дисциплине	Критерии оценки результатов обучения по дисциплине
100-83/ A,B	«отлично»	<p>Выставляется обучающемуся, если он глубоко и прочно усвоил теоретический и практический материал, может продемонстрировать это на занятиях и в ходе промежуточной аттестации.</p> <p>Обучающийся исчерпывающе и логически стройно излагает учебный материал, умеет увязывать теорию с практикой, справляется с решением задач профессиональной направленности высокого уровня сложности, правильно обосновывает принятые решения.</p> <p>Свободно ориентируется в учебной и профессиональной литературе.</p> <p>Оценка по дисциплине выставляется обучающемуся с учётом результатов текущей и промежуточной аттестации.</p> <p>Компетенции, закреплённые за дисциплиной, сформированы на уровне – «высокий».</p>
82-68/ C	«хорошо»	<p>Выставляется обучающемуся, если он знает теоретический и практический материал, грамотно и по существу излагает его на занятиях и в ходе промежуточной аттестации, не допуская существенных неточностей.</p> <p>Обучающийся правильно применяет теоретические положения при решении практических задач профессиональной направленности разного уровня сложности, владеет необходимыми для этого навыками и приёмами.</p> <p>Достаточно хорошо ориентируется в учебной и профессиональной литературе.</p> <p>Оценка по дисциплине выставляется обучающемуся с учётом результатов текущей и промежуточной аттестации.</p> <p>Компетенции, закреплённые за дисциплиной, сформированы на уровне – «хороший».</p>
67-50/ D,E	«удовлетворительно»	<p>Выставляется обучающемуся, если он знает на базовом уровне теоретический и практический материал, допускает отдельные ошибки при его изложении на занятиях и в ходе промежуточной аттестации.</p> <p>Обучающийся испытывает определённые затруднения в применении теоретических положений при решении практических задач профессиональной направленности стандартного уровня сложности, владеет</p>

Баллы/ Шкала ECTS	Оценка по дисциплине	Критерии оценки результатов обучения по дисциплине
		<p>необходимыми для этого базовыми навыками и приёмами. Демонстрирует достаточный уровень знания учебной литературы по дисциплине. Оценка по дисциплине выставляются обучающемуся с учётом результатов текущей и промежуточной аттестации. Компетенции, закреплённые за дисциплиной, сформированы на уровне – «достаточный».</p>
49-0/ F,FX	«неудовлетворительно»	<p>Выставляется обучающемуся, если он не знает на базовом уровне теоретический и практический материал, допускает грубые ошибки при его изложении на занятиях и в ходе промежуточной аттестации. Обучающийся испытывает серьёзные затруднения в применении теоретических положений при решении практических задач профессиональной направленности стандартного уровня сложности, не владеет необходимыми для этого навыками и приёмами. Демонстрирует фрагментарные знания учебной литературы по дисциплине. Оценка по дисциплине выставляются обучающемуся с учётом результатов текущей и промежуточной аттестации. Компетенции на уровне «достаточный», закреплённые за дисциплиной, не сформированы.</p>

5.3. Оценочные средства (материалы) для текущего контроля успеваемости, промежуточной аттестации обучающихся по дисциплине

Вопросы к опросам-коллоквиумам

Раздел 1. Компьютерное зрение и классификация изображений.

1. Обзор прикладных задач.
2. Основные алгоритмы машинного обучения.
3. Алгоритмы не глубокого обучения.
4. State-of-the-art методы, которые применяются в компьютерном зрении.
5. Нейронные сети для задачи классификации изображений.
6. Построения процесса обучения и тестирования сети.
7. Публичные данные для обучения нейронной сети на задачу классификации изображений

Раздел 2. Локализация и определение границ объекта на изображении.

1. Переход от классификации изображения к локализации объектов в одной нейронной сети.
2. Методы локализации объектов, основные подходы, и state-of-the-art решения.
3. Демонстрация локализации объектов на уровне меток класса в аннотации обучающей выборки
4. Фильтры для выделения границ на изображении.
5. Вертикальный и горизонтальный фильтры Собеля.
6. Вычисление вектора градиента яркости с помощью фильтра Собеля.
7. Алгоритмы определения границ объектов на базе нейронных сетей

Раздел 3. Методы оптимизации изображений.

1. Формирование окончательного результата алгоритма детектора.
2. Описание алгоритма Non-Maximum Suppression,
3. Вычислительная сложность и эффективность Non-Maximum Suppression.
4. Формирование окончательного результата с помощью NMS.
5. Описание алгоритма преобразования Хафа.
6. Алгоритмы Хафа для обнаружения линий и окружностей на изображении.
7. Алгоритм Хафа для обнаружения простых форм на изображении.
8. Понятие оптического потока.
9. Оценка оптического потока методом Лукаса-Канаде.
10. Оценка оптического потока в задаче отслеживания объектов на видео.
11. Прикладные задачи, основанные на оптическом потоке.
12. Существующие подходы к задаче сегментации изображений.
13. Сверточная нейронная сеть для сегментации изображений.
14. Архитектура U-Net, обучение и тестирование нейронной сети для сегментации изображений.

Раздел 4. Генерация реалистичных изображений.

1. Генеративно-сопоставительные сети, архитектура генератора и дискриминатора.
2. Целевые функции генератора и дискриминатора
3. Условия стабильного обучения генеративно-сопоставительных сетей.
4. Схема процесса обучения и тестирования генеративно-сопоставительных сетей.

Из вопросов коллоквиумов к зачету составляются билеты, по два вопроса в билете, обязательно из разных разделов.

6 Учебно-методическое и информационное обеспечение дисциплины

6.1. Список литературы

Основная

1. Сацюк, А. В. Компьютерное зрение и нейронные сети. Практика : учебное пособие / А. В. Сацюк. – Москва ; Вологда : Инфра-Инженерия, 2025. - 364 с. – ISBN 978-5-9729-2706-7. - Текст : электронный. - URL: <https://znanium.ru/catalog/product/2225332>
2. Шапиро, Л. Компьютерное зрение : учебник / Л. Шапиро, Д. Стокман. - 5-е изд. - Москва : Лаборатория знаний, 2024. - 762 с. - ISBN 978-5-93208-725-1. - Текст : электронный. - URL: <https://znanium.ru/catalog/product/2167351>.
3. Солем, Я. Э. Программирование компьютерного зрения на языке Python : практическое руководство / Я. Э. Солем ; пер. с англ. А. А. Слинкина. - 2-е изд. - Москва : ДМК Пресс, 2023. - 313 с. - ISBN 978-5-89818-310-3. - Текст : электронный. - URL: <https://znanium.ru/catalog/product/2102597>.
4. Компьютерное зрение. Современные методы и перспективы развития : монография / ред. Р. Дэвис, М. Терк ; пер. с англ. В. С. Яценкова. - Москва : ДМК Пресс, 2022. - 690 с. - ISBN 978-5-93700-148-1. - Текст : электронный. - URL: <https://znanium.com/catalog/product/2109506>.

Дополнительная

1. Ростовцев, В. С. Искусственные нейронные сети : учебник / В. С. Ростовцев. - Санкт-Петербург : Лань, 2019. - 216 с. - ISBN 978-5-8114-3768-9. - Текст : электронный // Лань : электронно-библиотечная система. - URL: <https://e.lanbook.com/book/122180>.
2. Аллен, Б. Д. Think DSP. Цифровая обработка сигналов на Python / Б. Д. Аллен ; перевод с английского А. Э. Бряндинский. - Москва : ДМК Пресс, 2017. - 160 с. - ISBN 978-5-97060-454-0. - Текст : электронный // Лань : электронно-библиотечная система. - URL: <https://e.lanbook.com/book/93566>
3. Жданов, А. А. Автономный искусственный интеллект : учебное пособие / А. А. Жданов. - 5-е изд. - Москва : Лаборатория знаний, 2020. - 362 с. - ISBN 978-5-00101-655-7. - Текст : электронный // Лань : электронно-библиотечная система. - URL: <https://e.lanbook.com/book/135544>

6.2. Перечень ресурсов информационно-телекоммуникационной сети «Интернет»

1. Smashing Magazine – крупнейший ресурс, посвященный Веб-разработкам. Статьи, книги, сведения о вакансиях, заказах и пр. <https://www.smashingmagazine.com>.
2. Хабр – крупнейший русскоязычный ресурс, посвященный Веб-разработкам. Разработки, администрирование, дизайн, менеджмент, маркетинг и др. <https://habr.com/ru/>
3. <https://www.scopus.com> – Международная реферативная наукометрическая БД, Scopus, доступна по подписке.
4. <http://www.consultant.ru/> - КонсультантПлюс компьютерная справочная правовая система.

6.3. Профессиональные базы данных и информационно-справочные системы

Доступ к профессиональным базам данных: <https://liber.rsuh.ru/ru/bases>

Информационные справочные системы:

1. Консультант Плюс
2. Гарант

7 Материально-техническое обеспечение дисциплины

Для реализации программы дисциплины требуется:

– компьютеры, с предустановленным базовым программным обеспечением в составе ОС Windows 10 и MS Office;

- объединение компьютеров в локальную сеть компьютерного класса с высокоскоростным выходом в интернет;
- для лекционного курса – посадочные места по количеству обучающихся, доска, мультимедиа проектор с экраном;
- рабочее место преподавателя, оснащенное компьютером с выходом на проектор.

Используемое программное обеспечение:

1. Windows 10
2. Microsoft Office 2013 Pro
3. Mozilla Firefox 52.8.1 ESR
4. Adobe Acrobat Reader
5. Kaspersky Endpoint Security
6. Платформа Сбер Джаз

8 Обеспечение образовательного процесса для лиц с ограниченными возможностями здоровья и инвалидов

В ходе реализации дисциплины используются следующие дополнительные методы обучения, текущего контроля успеваемости и промежуточной аттестации обучающихся в зависимости от их индивидуальных особенностей:

- для слепых и слабовидящих: лекции оформляются в виде электронного документа, доступного с помощью компьютера со специализированным программным обеспечением; письменные задания выполняются на компьютере со специализированным программным обеспечением или могут быть заменены устным ответом; обеспечивается индивидуальное равномерное освещение не менее 300 люкс; для выполнения задания при необходимости предоставляется увеличивающее устройство; возможно также использование собственных увеличивающих устройств; письменные задания оформляются увеличенным шрифтом; экзамен и зачёт проводятся в устной форме или выполняются в письменной форме на компьютере.

- для глухих и слабослышащих: лекции оформляются в виде электронного документа, либо предоставляется звукоусиливающая аппаратура индивидуального пользования; письменные задания выполняются на компьютере в письменной форме; экзамен и зачёт проводятся в письменной форме на компьютере; возможно проведение в форме тестирования.

- для лиц с нарушениями опорно-двигательного аппарата: лекции оформляются в виде электронного документа, доступного с помощью компьютера со специализированным программным обеспечением; письменные задания выполняются на компьютере со специализированным программным обеспечением; экзамен и зачёт проводятся в устной форме или выполняются в письменной форме на компьютере.

При необходимости предусматривается увеличение времени для подготовки ответа.

Процедура проведения промежуточной аттестации для обучающихся устанавливается с учётом их индивидуальных психофизических особенностей. Промежуточная аттестация может проводиться в несколько этапов.

При проведении процедуры оценивания результатов обучения предусматривается использование технических средств, необходимых в связи с индивидуальными особенностями обучающихся. Эти средства могут быть предоставлены университетом, или могут использоваться собственные технические средства.

Проведение процедуры оценивания результатов обучения допускается с использованием дистанционных образовательных технологий.

Обеспечивается доступ к информационным и библиографическим ресурсам в сети Интернет для каждого обучающегося в формах, адаптированных к ограничениям их здоровья и восприятия информации:

- для слепых и слабовидящих: в печатной форме увеличенным шрифтом, в форме электронного документа, в форме аудиофайла.
- для глухих и слабослышащих: в печатной форме, в форме электронного документа.
- для обучающихся с нарушениями опорно-двигательного аппарата: в печатной форме, в форме электронного документа, в форме аудиофайла.

Учебные аудитории для всех видов контактной и самостоятельной работы, научная библиотека и иные помещения для обучения оснащены специальным оборудованием и учебными местами с техническими средствами обучения:

- для слепых и слабовидящих: устройством для сканирования и чтения с камерой SARA CE; дисплеем Брайля PAC Mate 20; принтером Брайля EmBraille ViewPlus;
- для глухих и слабослышащих: автоматизированным рабочим местом для людей с нарушением слуха и слабослышащих; акустический усилитель и колонки;
- для обучающихся с нарушениями опорно-двигательного аппарата: передвижными, регулируемые эргономическими партами СИ-1; компьютерной техникой со специальным программным обеспечением.

9 Методические материалы

9.1. Планы практических заданий

Агеев А.В., Богуславский А.А., Власов С.О, Соколов С.М. Компьютерное зрение. Лабораторный практикум. М.: ИПМ им. М.В. Келдыша, 2024. 60 с. doi: 10.20948/mono-2024-ageev URL: <https://keldysh.ru/e-biblio/ageev>

Практическая работа №1. «Представления изображения в памяти компьютера и алгоритм выравнивания гистограммы»

Цель: Знакомство с базовыми понятиями компьютерного зрения, с представлением изображения в компьютере и анализ его частотных характеристик.

Теоретическая часть. Стр. 3-12.

Практическая часть.

1. В интернете найти 3 изображения, которые соответствуют следующим требованиям:

- Первое изображение должно быть недоэкспонированным;
- Второе изображение должно быть нормально экспонированным;
- Третье изображение должно быть переэкспонированным.

2. Над каждым из трех изображений необходимо выполнить операции:

- Загружаем изображение с помощью `opencv: cv.imread`;
- Переводим изображение из цветного пространства RGB в градацию оттенков серого: `cv.cvtColor(img, cv.COLOR_BGR2GRAY)`;
- С помощью функции `plt.hist()` – библиотеки `matplotlib` построить гистограмму черно-белого изображения;
- К черно-белому изображению применить операцию эквализации гистограммы (`Opencv: cv.equalizeHist`);
- Вывести эквализированное изображение и его гистограмму; f. Для исходного и эквализированного изображения построить эмпирическую функцию распределения.

3. Подготовить отчет, который должен содержать:

- Ответы на вопросы из пункта «Вопросы для самопроверки»;
- Описание операции эквализации гистограммы;
- Описание результатов ваших экспериментов;
- Исходный код программы на языке программирования python.

4. У каждого студента должны быть свой набор изображений.

Практическая работа №2. «Фильтрация и Бинаризация изображений».

Цель: Изучение подхода бинаризации и оператора свертки для извлечения информации из изображений.

Теоретическая часть. Стр. 13-22.

Практическая часть.

Необходимо реализовать программное обеспечение, которое будет определять отверстия на фотографиях деталей. Для реализации необходимо использовать алгоритмы бинаризации и метод низкочастотного фильтра (размытие изображений).

1. Загружаем изображение (cv.imread).

Рисунок: Результат загрузки изображения и вывода изображения

2. Переводим изображение из цветового пространства BGR в оттенки градации серого (cv.cvtColor).

Рисунок. Результат перевода изображения в оттенки градации серого

3. Перед бинаризацией, возможно, потребуется размыть изображение, медианным фильтром или фильтром Гаусса.

Экспериментируйте (cv.medianBlur; cv.GaussianBlur).

Рисунок. Результат размытия изображения

4. Для бинаризации необходимо воспользоваться одним из методов, и необходимо достичь того, чтобы отверстия четко выделялись и были белого цвета.

1. Глобальная ручная бинаризация — для изображений необходимо вручную подобрать пороги бинаризации (cv.threshold, cv.THRESH_BINARY).

2. Глобальная бинаризация с автоматическим выбором порога по методу Отцу (cv.threshold, cv.THRESH_BINARY+cv.THRESH_OTSU).

3. Адаптивная бинаризация (cv.adaptiveThreshold):

1. cv.ADAPTIVE_THRESH_MEAN_C.

2. cv.ADAPTIVE_THRESH_GAUSSIAN_C.

Для одного изображения работает один метод, для другого другой, экспериментируйте для получения результата.

Мы воспользуемся бинаризацией по методу Отцу: Рисунок. Результат бинаризации изображения по методу Отцу

Нашей конечной целью является определение отверстий. Метод Отцу определил отверстия как фон (черные цвет), а все остальное как объект. Поэтому нам необходимо инвертировать бинарное изображение. Выполним это с помощью функции cv.bitwise_not.

Получим следующий результат: Рисунок. Инвертированное бинарное изображение

5. После бинаризации, с помощью алгоритма поиска связанных компонент (cv.connectedComponents) необходимо промаркировать каждый объект, который был выделен алгоритмом бинаризации.

Функция cv.connectedComponents — вернет матрицу такого же размера, как и матрица бинарного изображения, но элементы этой матрицы будут содержать метки связанных компонент. Для того, чтобы вывести результат маркировки на экран (с целью убедиться в корректности работы), необходимо каждой метки присвоить случайный RGB цвет и с помощью созданной палитры нарисовать изображение и вывести его:

Листинг 5. Запуск алгоритма поиска связанных компонент и вывод результатов его работы

```
num_labels, labels = cv.connectedComponents(otsu)
label_color = [rgb_random() for i in range(num_labels)]
```

```
for i in range(img.shape[0]):
    for j in range(img.shape[1]):
        objects[i, j] = label_color[labels[i, j]]
cv.imshow("labels", objects)
```

Рисунок. Результат маркировки бинарного изображения (Листинг 5)

6. Для каждого объекта необходимо построить Bounding Box. OpenCv предоставляет функцию рисования прямоугольников cv.rectangle.

Эта функция на вход принимает координаты, где рисовать. Вам необходимо написать функцию, которая выполнит сканирование матрицы меток, и для каждой метки вычислит координаты прямоугольника, который затем необходимо нарисовать на исходном изображении с помощью функции cv.rectangle.

7. Алгоритм поиска координат размеченных объектов:

1. Заводим массив координат *bounding_boxes* для каждого размеченного объекта. Количество объектов нам известно – это количество меток, которые вернул нам алгоритм cv.connectedComponents. Каждый элемент массива *bounding_boxes* – это координаты прямоугольника, в который вписан некоторый объект. Координаты прямоугольника описываются двумя точками. Верхней левой точкой и правой нижней точкой.

1. *bounding_boxes* [k] – к который вписан объект с меткой k.

2. *bounding_boxes* [k] точки прямоугольника.

3. *bounding_boxes* [k] правой точки прямоугольника.

2. Цикл по матрице меток, строка за строкой:

1. Пусть *l* – это текущая метка. *i, j* – текущие координат матрице меток.

2. Сравниваем координаты текущей позиции для метки *l* с координатами для этой метки в массиве *bounding_boxes*.

– Если *bounding_boxes* [k] . To *bbbbbbbbbbbbbbggbbbbbbb* [kk]

– Если *bounding_boxes* [k]. *l*

To *bounding_boxes* [k] [kk]. *l*

– Если *b bounding_boxes* [k] [k]. *rr*

To *bounding_boxes* [k]. *r*

– Если *bounding_boxes* [k] [kk]. *rrr*

To *bounding_boxes* [k] [kk]. *rr*

Листинг 6. Реализация алгоритма поиска обрамляющих прямоугольников для размеченных объектов

```
import dataclasses
import math
```

```
@dataclasses.dataclass
```

```
class Point:
```

```
    x: int = 0,
```

```
    y: int = 0,
```

```
class Rectangle:
```

```
    def __init__
```

```
        (self, left: Point, right: Point):
```

```
        self._up_left_point = left
```

```
        self._down_right_point = right
```

```
        @property
```

```
        return bounding_boxes
```

8. Результат построения Bounding Box для всех объектов, представленных на изображении:

Рисунок. Результат построения bounding boxes вокруг промаркированных элементов изображения

Рисунок. Результат построения bounding boxes вокруг промаркированных элементов изображения с фильтрацией по площади прямоугольника

9. Перед построением необходимо фильтровать прямоугольники по площади, чтобы не рисовать их для паразитных объектов.

В данном случае при фильтрации по площади прямоугольника, мы потеряем центральное отверстие.

10. Необходимо разобраться, с алгоритмом работы:

1. Глобальной Бинаризации.
2. Автоматического выбора порога бинаризации по методу Отцу.
3. Адаптивной бинаризации (cv.ADAPTIVE_THRESH_MEAN_C).

11. Необходимо описать алгоритм поиска Bounding Box для каждой метки, который вы реализуете.

12. Подготовить отчет, который будет содержать теоретическую справку по алгоритмам бинаризации, построения Bounding Box и описание вашей программы, которая должна иметь следующий интерфейс:

1. На вход подается путь до изображения.
2. Программа отображает каждый этап модификации изображения и результирующее изображение с ограничивающими прямоугольниками, которые показывают отверстия в деталях.

Практическая работа №3. «Сегментации изображений»

Цель: Знакомство с алгоритмом сегментации водоразделов.

Теоретическая часть. Стр. 29-31.

Практическая часть.

Необходимо реализовать программное обеспечение для ручной сегментации изображения методом watershed. Программа должна предоставлять GUI для ручного рисования множества маркеров, которые затем будут использоваться в алгоритме сегментации.

Необходимо самостоятельно выбрать цветное изображение из интернета, и выполнить его сегментацию на несколько сегментов с помощью разработанного ПО.

Для реализации данного ПО необходимо выполнить следующую последовательность действий:

1. Выполнить загрузку изображения:

- 1: img = cv.imread(filename)
- 2: img = cv.resize(img, (600, 400))
- 3: imgCopy = np.copy(img)
- 4: cv.imshow('img', img)

2. Настроить обработку событий мыши окна img, для реализации механизма рисования маркеров:

- 1: mask = np.zeros(shape = (img.shape[0], img.shape[1]),
- 2: dtype = np.uint8)
- 3: cv.setMouseCallback("img", onMouseCallback)
- 4: while True:
- 5: c = cv.waitKey()
- 6: print("c =", c)
- 7: if c == 27: # esc
- 8: break
- 9: if c == 114: # r
- 10: mask = np.zeros(shape = (img.shape[0],
- 11: img.shape[1]),
- 12: dtype = np.uint8)
- 13: img = np.copy(imgCopy)
- 14: cv.imshow('img', img)
- 15: if c == 32: # space 32
- 16: pass # здесь необходимо реализовать логику
- 17: # работы

```
18: # с алгоритмом сегментации
```

```
19: # cv.watershed
```

В строке 3 мы устанавливаем обратный вызов, который будет вызываться каждый раз, когда происходит событие, связанное с мышью в окне `img`. Начиная со строки 4 мы запускаем цикл событий, который ждет нажатия клавиш клавиатуры и событий мыши. Когда фокус находится на окне `img` и мы нажимаем «esc», то мы выходим из цикла (строки 7-8), когда мы нажимаем на букву «g», то выполняется операция очистки нарисованных маркеров (строки 9- 14). Когда нажимаем на «пробел» то запускается механизм сегментации, который будет описан ниже.

3. Необходимо реализовать callback на события мыши, с целью создания механизма рисования маркеров:

```
1: mousePressed = False
```

```
2: prevPoint = (-1, -1)
```

```
3: mask = None
```

```
4: img = None
```

```
5:
```

```
6: def onMouseCallback(event,x,y,flags,param):
```

```
7: global mousePressed
```

```
8: global prevPoint
```

```
9: if event == cv.EVENT_LBUTTONDOWN:
```

```
10: mousePressed = True
```

```
11: print("EVENT_LBUTTONDOWN")
```

```
12: elif event == cv.EVENT_LBUTTONUP:
```

```
13: mousePressed = False
```

```
14: prevPoint = (-1, -1)
```

```
15: print("EVENT_LBUTTONUP")
```

```
16: if mousePressed:
```

```
17: pt = (x, y)
```

```
18: if prevPoint == (-1, -1):
```

```
19: prevPoint = pt
```

```
20: return
```

```
21: cv.line(mask,
```

```
22: prevPoint,
```

```
23: pt,
```

```
24: (255, 255, 255),
```

```
25: 5, 33
```

```
26: 8,
```

```
27: 0)
```

```
28: cv.line(img,
```

```
29: prevPoint,
```

```
30: pt,
```

```
31: (255, 255, 255),
```

```
32: 5,
```

```
33: 8,
```

```
34: 0)
```

```
35: cv.imshow("img", img)
```

```
36: prevPoint = pt
```

В строках 1-4 объявляем глобальные переменные. Callback вызывается на различные события мыши, мы будем использовать, такие события как: движение мыши, нажатие и отпускание левой кнопки. На первое событие нажатия левой кнопки мыши, мы будем запоминать позицию курсора (строки 16-20), пока мы держим левую кнопку мыши и двигаем мышь, то между последовательными позициями курсора мыши мы рисуем

прямые линии (строки 21-36). Прямые линии рисуются, как на изображении, которое отображается в окне, так и в матрице `mask`, которая будет использоваться для формирования маркеров.

4. Реализация механизма ручной сегментации, который должен быть вставлен на место строчки 16 в листинге пункта 2.

```

1: print("start watershed")
2: contours, hierarchy = cv.findContours(
3: mask,
4: cv.RETR_TREE,
5: cv.CHAIN_APPROX_SIMPLE)
6: if len(contours) == 0:
7: continue
8: markers = np.zeros(
9: shape = (img.shape[0], img.shape[1]),
10: dtype = np.int32)
11: color = 1
12: for i in range(len(contours)):
13: cv.drawContours(markers, contours, i,
14: color, 2, cv.LINE_8, hierarchy,
15: 0)
16: color = color + 1
17: imgGray = cv.cvtColor(img, cv.COLOR_BGR2GRAY) 34
18: colorTab = [rgb_random() for _ in range(len(contours))]
19: wmarkers = cv.watershed(imgCopy, markers)
20: wshed = np.zeros_like(img)
21: for i in range(img.shape[0]):
22: for j in range(img.shape[1]):
23: index = wmarkers[i, j] 24: if index == -1:
25: wshed[i, j] = [255, 255, 255]
26: else:
27: wshed[i, j] = 0.5 * colorTab[index-1]
28: + 0.5 * imgGray[i, j]
29: cv.imshow('img', wshed)

```

Когда мы рисовали кривые в окне `img`, то они рисовались, как на изображении, которые отображалось в окне, так и матрице `mask`, которую теперь необходимо преобразовать в маркеры. Каждый маркер должен обладать собственным идентификатором, поэтому нам необходимо разделить маркеры: не пересекающиеся кривые (несвязанные прямым соседством наборы точек) являются разными маркерами. С помощью функции поиска контуров, мы определим все нарисованные кривые (строка 2). После чего мы можем перейти к формированию маркеров:

1. (Строка 8) Сначала мы выделим память для маркеров.

2. (Строка 13) Затем нарисуем все кривые, в качестве цвета для рисования будем использовать идентификатор кривой.

3. (Строка 17) Для каждого будущего сегмента необходимо выбрать цвет, которым мы будем обозначать этот сегмент. Для этого воспользуемся генерацией случайного цвета для каждого идентификатора сегмента.

4. (Строка 18) Запускаем сегментацию.

5. (Строки 19-27) Рисуем сегменты, используем линейную комбинацию цвета сегмента, к которому относится конкретный пиксель, и яркость пикселя.

6. (Строка 28) Отображаем полученный результат.

Для формирования случайного цвета предлагается использовать следующий метод:

```

1: def rgb_random():

```

```
2: return np.random.randint(0,  
3: 255,  
4: size=3,  
5: dtype=np.uint8)
```

АННОТАЦИЯ РАБОЧЕЙ ПРОГРАММЫ ДИСЦИПЛИНЫ

Цель дисциплины: Дисциплина «Компьютерное зрение» имеет целью теоретическое и практическое освоение методов и технологий компьютерного зрения.

Задачи:

- выработка у студентов знаний о базовых принципах и алгоритмах компьютерного зрения;
- приобретение умений реализовывать основные алгоритмы компьютерного зрения в виде программ и применять их для решения практических задач;
- овладение терминологическим аппаратом и практическими навыками в области компьютерного зрения.

В результате освоения дисциплины обучающийся должен:

Знать: классификацию изображений, приложения и архитектуру нейронных сетей для определения рассматриваемых изображений; методы машинного обучения в области компьютерного зрения, проводить анализ требований, и определения необходимого класса задач.

Уметь: выбирать и реализовывать наилучшие методы для оптимального решения поставленных задач компьютерного зрения; разрабатывать программные компоненты для решения задач в области компьютерного зрения; осуществлять оценку и выбор методов машинного обучения, разрабатывать некоторые методы и алгоритмы.

Владеть: навыками тестирования создаваемых систем компьютерного зрения; навыками, необходимыми для участия в реализации проектов сквозной цифровой субтехнологии «Компьютерное зрение».