

**РАБОЧАЯ ПРОГРАММА ДИСЦИПЛИНЫ
РАСПОЗНАВАНИЕ ОБРАЗОВ И КОМПЬЮТЕРНОЕ ЗРЕНИЕ**

Направление подготовки:
09.03.03 Прикладная информатика

Направленность (профиль)
Искусственный интеллект и анализ данных

Уровень высшего образования: бакалавриат

1. Цели и задачи освоения дисциплины

Цель - изучение процессов обработки и анализа цифровых данных изображений

Для достижения поставленной цели при изучении дисциплины решаются следующие задачи:

изучить возможности анализа изображений и видео "на лету"

изучить алгоритмы распознавания изображений

рассмотреть методы классификации изображений, отслеживания и распознавания объектов

2. Место дисциплины в структуре образовательной программы

Дисциплина «Распознавание образов и компьютерное зрение» относится к части, формируемой участниками образовательного процесса, учебного плана ОПОП ВО по направлению подготовки 09.03.03 Прикладная информатика.

3. Перечень планируемых результатов обучения по дисциплине, соотнесенных с планируемыми результатами освоения образовательной программы

3.1. Компетенции обучающегося, формируемые в результате освоения дисциплины с текстом:

Процесс изучения дисциплины направлен на формирование у обучающихся следующих компетенций (результатов освоения образовательной программы):

Коды компетенций	Содержание компетенций
ПК-7	Способность выполнять интеллектуальный анализ больших данных
ПК-9	Способность применять математические методы моделирования процессов обработки информации с использованием средств интеллектуального анализа данных

3.2. Взаимосвязь планируемых результатов обучения по дисциплине с формируемыми компетенциями образовательной программы

Коды и формулировка компетенции	Индикаторы компетенции	Запланированные результаты обучения
ПК-7 Способность выполнять интеллектуальный анализ больших данных	ПК 7.1 Знать методы и инструментальные средства интеллектуального анализа больших данных ПК 7.2 Уметь выбирать средства представления результатов аналитики больших данных ПК 7.3 Владеть техническими, программными средствами для разработки алгоритмов и программ в области интеллектуального анализа данных	ПК 7.1.1 знать методы и инструментальные средства для машинного обучения и анализа больших данных - предметную область анализа больших данных - теоретические и прикладные основы анализа данных ПК 7.1.2 уметь использовать инструментальные средства машинного обучения и анализа больших данных - проводить сравнительный анализ методов и инструментальных средств анализа больших данных ПК 7.1.3 владеть инструментальными средствами машинного обучения и анализа больших данных ПК 7.2.1 знать средства визуализации результатов решения задач машинного обучения и анализа больших данных методы интерпретации и визуализации решения задач машинного обучения и анализа больших данных ПК 7.2.2 уметь использовать стандартные программные библиотеки для визуализации решений задач машинного обучения и анализа данных ПК 7.2.3 владеть стандартными программными библиотеками для решения задач машинного обучения и анализа данных ПК 7.3.1 знать программные платформы и библиотеки для решения задач машинного обучения и анализа

		<p>больших данных стандарты проведения анализа данных</p> <p>ПК 7.3.2 уметь использовать программные платформы и библиотеки для решения задач машинного обучения и анализа больших данных</p> <p>ПК 7.3.3 владеть программными средствами для разработки алгоритмов машинного обучения, алгоритмами построения искусственных нейронных сетей</p>
<p>ПК-9 Способность применять математические методы моделирования процессов обработки информации с использованием средств интеллектуального анализа данных</p>	<p>ПК 9.1 Знать принципы решения задач машинного обучения и интеллектуального анализа данных</p> <p>ПК 9.2 Уметь создавать алгоритмические и математические модели прикладных задач интеллектуального анализа данных</p> <p>ПК 9.3 Владеть навыками построения описательных и прогнозных аналитических моделей с использованием современных инструментов интеллектуального анализа данных</p>	<p>ПК 9.1.1 знать основные принципы решения задач машинного обучения и анализа данных</p> <p>ПК 9.1.2 уметь проводить спецификацию задачи, реализовывать программы на алгоритмических языках высокого уровня, интерпретировать полученные результаты</p> <p>ПК 9.1.3 владеть методами и алгоритмами машинного обучения</p> <p>ПК 9.2.1 знать математические и алгоритмические модели интеллектуального анализа данных, методы оценки временных и стоимостных характеристик машинного обучения и анализа данных</p> <p>ПК 9.2.2 уметь создавать алгоритмические модели типовых прикладных задач обработки информации, реализовывать программы на алгоритмических языках высокого уровня</p> <p>ПК 9.2.3 владеть навыками использования прикладных библиотек для решения задач машинного обучения</p> <p>ПК 9.3.1 описательные и прогнозные аналитические модели для интеллектуального анализа данных - использовать современные программные инструменты интеллектуального анализа данных</p> <p>9.3.2 разрабатывать и оценивать модели больших данных</p> <p>9.3.3 навыками построения описательных и прогнозных аналитических моделей с использованием современных инструментов интеллектуального анализа данных, навыками выявления требований к результатам анализа больших данных в предметной области</p>

4. Структура и содержание дисциплины

Общая трудоемкость дисциплины составляет 4 зачетных единицы, 144 часов.

4.1. Объем дисциплины

Вид учебной работы	Всего часов		
	Очная форма обучения	Очно-заочная форма обучения	Заочная форма обучения
Общая трудоемкость дисциплины	144	не реализуется	не реализуется
Контактная работа обучающегося с преподавателем (по видам учебных занятий) (всего), в том числе:	32		
<i>Занятия лекционного типа</i>	16		
<i>Занятия семинарского типа (практич., семин., лаборат. и др.)</i>	16		
<i>Самостоятельная работа под руководством преподавателя</i>	76		
Самостоятельная работа обучающихся СРС/подготовка к экзамену (зачету) в соответствии с учебным планом	36		

Вид промежуточной аттестации (зачет/экзамен)	экзамен		
---	---------	--	--

4.2. Разделы и темы дисциплины, их трудоемкость по видам учебных занятий

4.2.1. Темы дисциплины, их трудоемкость по видам учебных занятий для очной формы обучения:

Не реализуется

4.2.2. Темы дисциплины, их трудоемкость по видам учебных занятий для очно-заочной формы обучения:

№ п/п	Наименование разделов, тем учебных занятий	Всего часов	Контактная работа обучающегося с преподавателем (по видам учебных занятий) (всего), в том числе:				Процедура оценивания/ оцениваемые компетенции
			ЛЗ	СЗ	ЛР	СРС	
1	Цифровое изображение	18	2	2		14	Участие в устном опросе, решение задач, выполнение заданий и упражнений для самостоятельной работы / ПК-7, ПК9
2	Классификация изображений	18	2	2		14	Участие в устном опросе, решение задач, выполнение заданий и упражнений для самостоятельной работы / ПК-7, ПК9
3	Нейросетевые методы классификации и поиска похожих изображений	18	4	4		10	Тестирование, решение задач выполнение заданий и упражнений для самостоятельной работы / ПК-7, ПК9
4	Анализ и синтез изображений	18	2	2		14	Участие в устном опросе, решение задач, выполнение заданий и упражнений для самостоятельной работы / ПК-7, ПК9
5	Нейросетевые методы распознавания видео	18	2	2		14	Участие в устном опросе, решение задач, выполнение заданий и упражнений для самостоятельной работы / ПК-7, ПК9
6	Реконструкция изображений	18	4	4		10	Тестирование, решение задач выполнение заданий и упражнений для самостоятельной работы / ПК-7, ПК9
	Экзамен	36				36	Сдача экзамен
	Итого	144	16	16		76	

4.2.3. Темы дисциплины, их трудоемкость по видам учебных занятий для заочной формы обучения:

Не реализуется

4.3. Содержание дисциплины

Тема 1 Цифровое изображение

Цифровое изображение. Глаз человека и цифровая камера. Компьютерное зрение. Свет, цвет и модели цвета.

Коррекция яркости и цвета. Дискретизация изображения. Способы представления цвета. Трихроматическая теория. Модель RGB. Модель CIELAB. Модель CIE XYZ. Модель HSB. Модель Lab.

Обработка изображений. Шумоподавление и свёртка. Виды шума. Усреднение нескольких кадров. Пространственная фильтрация. Временная фильтрация. Свёртка. Свёртка и корреляция. Линейные фильтры и свёртка. Сглаживание.

Фильтр Гаусса. Подавление гауссова шума. Свойства фильтра Гаусса. Компенсация разности освещения.

Медианный фильтр. Алгоритм Single scale retinex (SSR). Билатеральный фильтр. Сглаживание с box-фильтром.

Статистические свойства распределения градиентов в изображении. Выделение краёв. Выделение границ.

Градиент изображения. Дифференцирование и свёртка. Производная фильтра Гаусса. Фильтры: Робертса, Превитт, Собель 3x3, Scharr, Собель 5x5. Сегментация объектов.

Текстура изображения. Анализ текстуры. Форма из текстуры.

Тема 2 Классификация изображений

Совмещение и сопоставление изображений. Алгоритмы сопоставления изображений. Построение мозаики.

Трёхмерная реконструкция по изображениям. Выбор модели преобразования изображений: двумерные и трёхмерные. Построение панорамных изображений. Проблемы совмещения изображений: прямое выравнивание и алгоритм «детектор углов Харриса». Блобы. Deskriptory. Метод SIFT. Робастная оценка. Алгоритм RANSAC. Функции качества

Классификация изображений. Бинарная классификация. Многоклассовая классификация. Распознавание по лицу - верификация. Атрибуты объектов. Распознавание человека по лицу. Квантование признаков. Квантование многомерных признаков. Квантование через кластеризацию. Пространственное распределение. Пространственная пирамида. Гистограммы градиентов. Схема метода «мешок слов». Вычисление признаков фрагментов. Расчёт LBP. Коллекции для распознавания человека. Общая схема поиска изображений. Deskriptory изображений.

Ранжирование результатов. Семантическое хеширование. Алгоритм GISTIS

Тема 3 Нейросетевые методы классификации и поиска похожих изображений

Линейная модель МакКаллока-Питтса. Градиентный метод обучения. Стохастический градиентный спуск.

Представимость функций нейросетями. Задание нейросети. Архитектура нейросети. Многослойная нейросеть.

Линейный перспетрон. Функция перспетрона. Функции активации. Функции потерь. Расчёт градиента. Процедура обратного распространения ошибки. Нейрон как свёртка (S- клетки). Свёрточные сети. Deskriptor HOG/SIFT. Мешок визуальных слов

Нейросетевые признаки. Визуализация работы нейросети. Эволюция признаков. Выход слоя как вектор-признак.

UMAP визуализация. Классификация близких объектов. Дообучение (fine-tuning). Spatial Pyramid Pooling (SPP). Архитектура Inception. Факторизация свёрток (MobileNet). Нейросетевые признаки для поиска похожих

Тема 4 Анализ и синтез изображений

Выделение объектов. Алгоритмы выделения объектов на изображениях. Критерий обнаружения (IoU). Оценка качества детектора. Кривая точности-полноты.

Множественные отклики. Сопоставление шаблонов.

HOG+SVM для поиска пешеходов. Обучение детектора. Нейросетевые методы в детекторах. Каскад (Attentional cascade). Region-based Convolution Networks (R-CNN).

Приведение гипотез к квадратным bbox. Обучение R- CNN. Оценка R-CNN по mAP. Уточнение bbox. Spatial pyramid pooling (SPP). Обучение Fast R-CNN.

Задача сегментации. Извлечение объекта. Сегментация без учителя. Семантическая сегментация. Интерактивная сегментация. Алгоритм Interactive GraphCuts. Графическая модель. «Марковское случайное поле» (MCP).

Факторизация совместного распределения. Conditional Random Field (CRF). Conditional

Random Field. Факторы в модели изображения. Бинарная сегментация. Унарные потенциалы. Парные потенциалы. Итеративные разрезы графов. Визуальная сегментация. Пересегментация через кластеризацию. Алгоритм SLIC. Использование CRF. Нейросетевые модели

Модификация изображений. Синтез изображений. Визуализация вектор-признака. Визуализация изображения по выходам. Реконструкция. Сравнение нейросетевых признаков. Описание стиля. Реконструкция стиля. Детали реконструкции изображения. Стилизация с Perception Loss. Генерация текстур. Модификация метода для фотографий. Общая схема Generative Adversarial Networks (GAN).

Wasserstein GAN. Conditional GAN. CycleGAN. DualGAN. Augmented CycleGAN. Style Loss – функция от признаков, описывающая статистики распределения признаков.

Тема 5 Нейросетевые методы распознавания видео

Видеопоследовательность. Optical flow (оптический поток). Распознавание по движению. Прямолинейное применение свёрточных нейросетей к вычислению оптического потока. Сопоставление пикселей через CNN. Отслеживание объектов в видео. Стандартная схема итеративного трекинга. Multiple object tracking (MOT). Ошибки MOT. Метрики качества. Simple Online and Realtime Tracking (SORT). Распознавание событий

Тема 6 Реконструкция изображений

Перспективная проекция. Матрица проекции и внутренняя калибровка. Уравнение перспективной проекции.

Радиальная дисторсия. Геометрические свойства 2х изображений. Перспективное преобразование плоскости (гомография). Эпиполярная геометрия. Эпиполярное ограничение. Калиброванный случай. Свойства существенной матрицы. Некалиброванный случай.

Структура из движения. Целевая функция. Калибровка камеры (PnP-проблема). Последовательный подход. Метод связок (Bundle adjustment). Фотограмметрия. Граф связанности изображений. Построение «следов». «Скелет» графа. Сопоставление изображений на больших коллекциях.

Бинокулярное стерео. Параллакс. Стереопсис. Плотное стерео. Ректификация изображений. Вычисление соответствующих точек. Ректификация. Радиальная развёртка. Триангуляция. Вычисление карты диспаратета. Стереосопоставление через вычисление диспаратета.

Локальные методы бинокулярного стерео. Обучение с помощью CNN. Проблема сохранения границ. Алгоритм Scanline Optimization. Алгоритм на основе MST. Алгоритм Semi-Global Matching. Алгоритм Simple Tree. Базовый алгоритм с сегментацией. Многовидовая реконструкция. Схема объединения карт глубины. Методы на основе фрагментов. Алгоритм на основе фрагментов. Визуальный тест Тьюринга. Учёт контуров и дополнение реконструкции

Неоднозначность реконструкции. Структурированный мир. Модель и структура. Исторические подходы к реконструкции. Современные подходы к реконструкции.

Итеративная схема с гипотезами. Автоматические vs с взаимодействием с пользователем. Параметрическая модель и детализация реконструкции. Точки и линии схода, горизонт. Свойства точек схода. Вычисление точек схода.

Линия схода (vanishing line). Линия горизонта. Вычисление линии горизонта. Графическая модель. Применение горизонта. Высота камеры. Tour into picture: TIP: Моделирование объектов. TIP: Реконструкция текстуры.

TIP: Результат. Разделение на слои. Моделирование основных слоев. Моделирование глубины. Моделирование сложных фигур. Моделирование лица. Алгоритм Automatic Photo Pop-Up. Приближенная геометрическая модель как контекст. Манхэттенский мир. Действия людей как подсказки. Библиотеки объектов. Обучение по 1 примеру.

4.4. Темы семинарских занятий и лабораторных работ

Тема 1 Цифровое изображение

Решение задач по теме 1.

Тема 2 Классификация изображений

Решение задач по теме 2.

Тема 3 Нейросетевые методы классификации и поиска похожих изображений

Решение задач по теме 3. Тестирование по темам 1,2,3

Тема 4 Анализ и синтез изображений

Решение задач по теме 4.

Тема 5 Нейросетевые методы распознавания видео

Решение задач по теме 5.

Тема 6 Реконструкция изображений

Решение задач по теме 6. Тестирование по темам 1, 2, 3,4, 5, 6

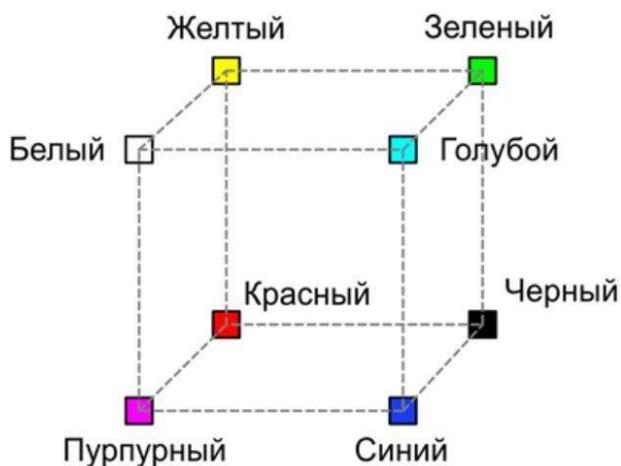
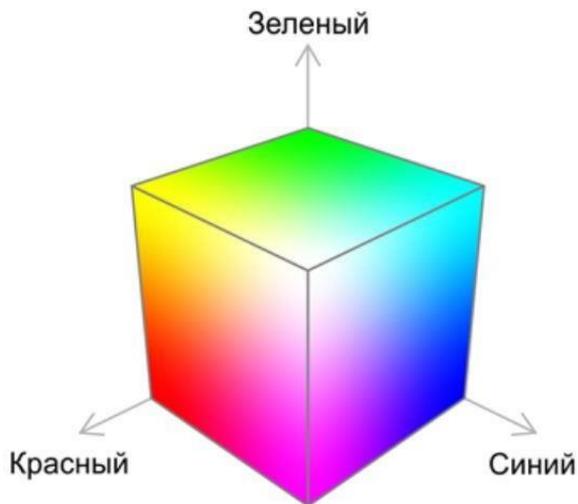
5. Учебно-методическое обеспечение самостоятельной работы

5.1. Задания для самостоятельной подготовки к занятиям семинарского типа

Семинарское занятие №1.

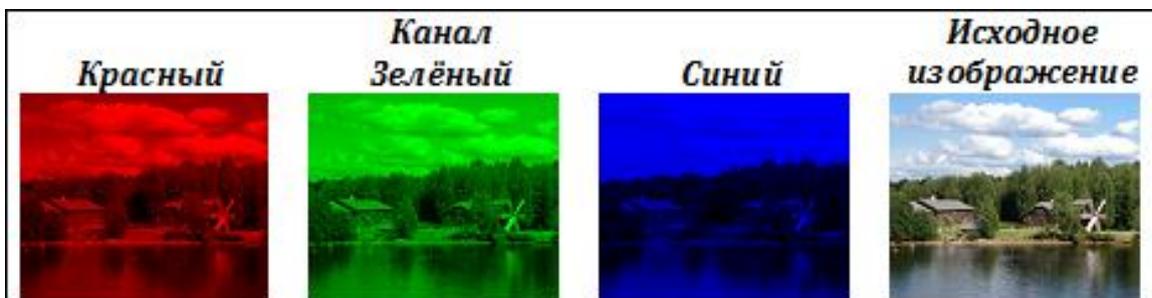
Тема занятия: Цветовые пространства

Цветовой куб RGB

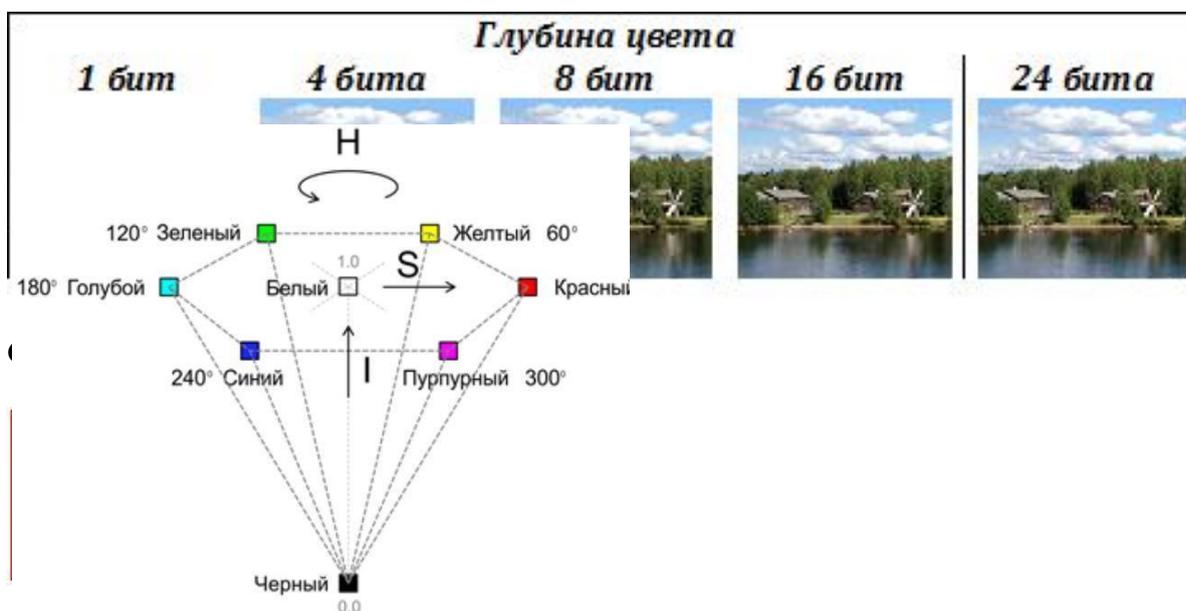


Точки, соответствующие базовым цветам, расположены в вершинах куба, лежащих на осях: красный – $(1;0;0)$, зелёный – $(0;1;0)$, синий – $(0;0;1)$. При этом

вторичные цвета (полученные смешением двух базовых) расположены в других вершинах куба: голубой —(0;1;1), пурпурный — (1;0;1) и жёлтый – (1;1;0). Чёрный и белые цвета расположены в начале координат (0;0;0) и наиболее удалённой от начала координат точке (1;1;1)
 Цветные изображения в модели RGB строятся из трёх отдельных изображений-каналов.

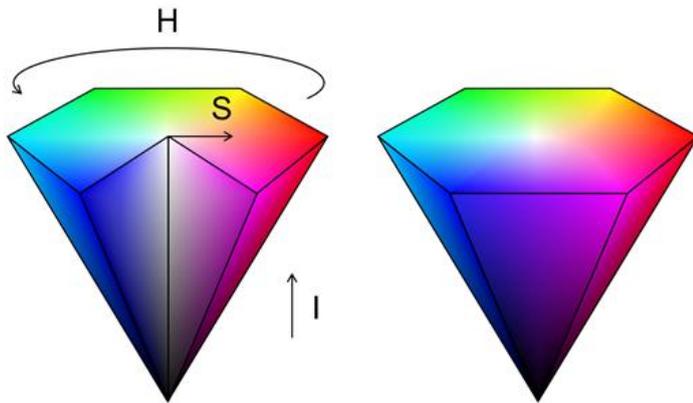


В модели RGB для каждой составляющей цвета отводится определённое количество бит. Количество бит, отводимых на кодирование одного пиксела называется глубиной цвета



Цветовое пространство тон, насыщенность, интенсивность (HSI)

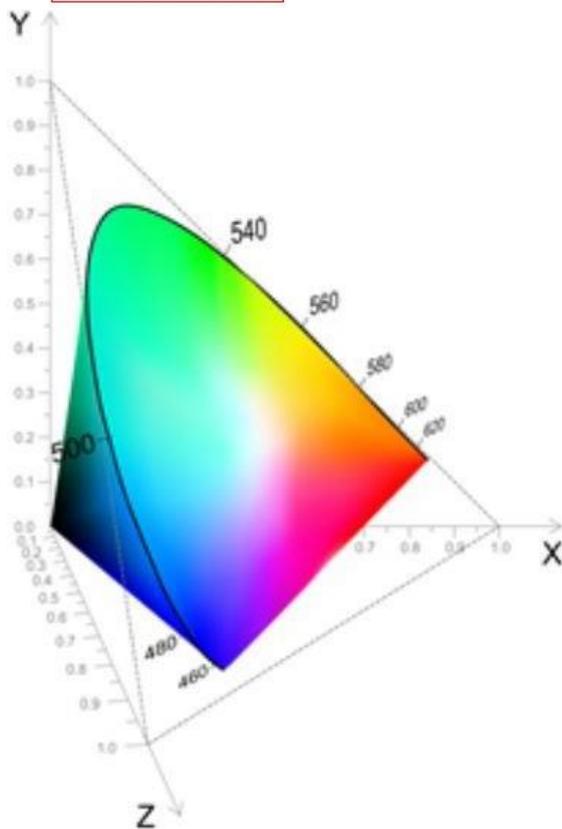
Алгоритм перевода из RGB в HSI:



$$\left\{ \begin{array}{l} H = \begin{cases} \theta; B \leq G \\ 360 - \theta; B > G \end{cases} \text{ где } \theta = \arccos \left(\frac{\frac{1}{2} * ((R - G) + (R - B))}{\sqrt{(R - G)^2 + (R - B)(G - B)}} \right) \\ S = 1 - \frac{3}{(R + G + B)} \min(R, G, B) \\ I = \frac{1}{3}(R + G + B) \end{array} \right.$$

Модель CIE XYZ

$$\left\{ \begin{array}{l} x = \frac{X}{X + Y + Z} \\ y = \frac{Y}{X + Y + Z} \\ z = \frac{Z}{X + Y + Z} \end{array} \right.$$



Модель CIELAB

цветовое пространство CIE L*a*b* - международный стандарт

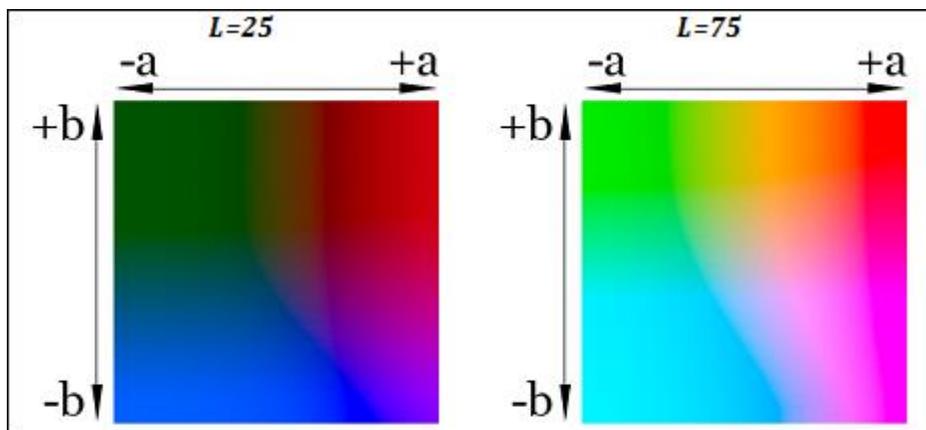
В системе CIE L*a*b* координата L означает светлоту (в диапазоне от 0 до 100), а координаты a,b – означают позицию между зелёным-пурпурным, и синим-жёлтым цветами. Формулы для перевода координат из CIE XYZ в CIE L*a*b* :

$$\begin{cases} L^* = 116 * f\left(\frac{Y}{Y_n}\right) - 16 \\ a^* = 500 * \left(f\left(\frac{X}{X_n}\right) - f\left(\frac{Y}{Y_n}\right)\right) \\ b^* = 200 * \left(f\left(\frac{Y}{Y_n}\right) - f\left(\frac{Z}{Z_n}\right)\right) \end{cases}$$

где (X_n, Y_n, Z_n) – координаты точки белого в пространстве CIE XYZ, а

$$f(x) = \begin{cases} \sqrt[3]{x} \\ \frac{1}{3} * \left(\frac{29}{6}\right)^2 x + \frac{4}{29} \end{cases}$$

срезы цветового тела CIE L*a*b* для двух значений светлоты:



Телевизионные цветоразностные цветовые системы (аналоговое телевидение)

$$\begin{cases} Y = 0.299R + 0.587G + 0.114B \\ I = 0.596R - 0.274G - 0.321B \\ Q = 0.211R - 0.526G + 0.311B \end{cases}$$

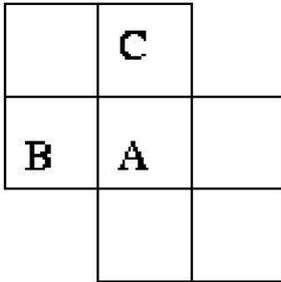
Семинарское занятие по
теме 2Тема: Обработка
изображений

Цель: изучить особенности выделения однородных областей методом последовательного сканирования

Задания (вопросы) для подготовки:

Выделение однородных областей методом последовательного сканирования

Сканируем изображение сверху вниз, слева направо:



1. if $I(A) - I_{avg}(Cl(B)) > \delta$ and $I(A) - I_{avg}(Cl(C)) > \delta$ - создаем новую область, присоединяем к ней пиксел А
2. if $I(A) - I_{avg}(Cl(B)) < \delta$ xor $I(A) - I_{avg}(Cl(C)) < \delta$ - добавить А к одной из областей
3. if $I(A) - I_{avg}(Cl(B)) < \delta$ and $I(A) - I_{avg}(Cl(C)) < \delta$:
 - 3.1. $I_{avg}(Cl(B)) - I_{avg}(Cl(C)) < \delta$ - сливаем области В и С.
 - 3.2. $I_{avg}(Cl(B)) - I_{avg}(Cl(C)) > \delta$ - добавляем пиксел А к тому классу, отклонение от которого минимально.

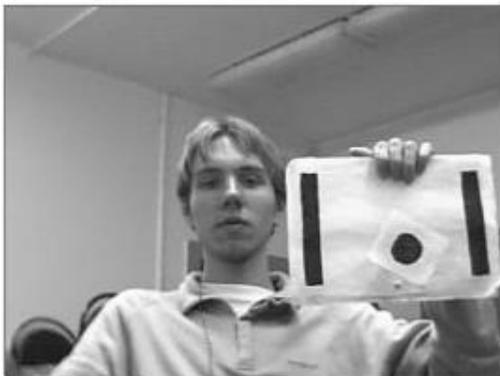
где

$I(A)$ – яркость пиксела А

$Cl(B)$ – область к которой принадлежит пиксел В

$I_{avg}(Cl(B))$ – средняя яркость области к которой принадлежит В

Результат:



Семинарское занятие по теме 3

Тема: Проектирование нейронной сети для распознавания текста

Цель: изучить возможности разработки и обучения нейронной сети для распознавания текста

Задания (вопросы) для подготовки:

формирование графического изображения текста (image_file)

HELLO WORLD

Необходимые библиотеки: OpenCV

Разбиение текста на буквы

Разобьем текст на отдельные буквы. Для этого необходима OpenCV, точнее его функция findContours.

Откроем изображение (cv2.imread), переведем его в ч/б (cv2.cvtColor + cv2.threshold), слегка увеличим (cv2.erode) и найдем контуры:

```
image_file = "text.png"
img = cv2.imread(image_file)
gray = cv2.cvtColor(img, cv2.COLOR_BGR2GRAY)

ret, thresh = cv2.threshold(gray, 0, 255,
cv2.THRESH_BINARY) img_erode = cv2.erode(thresh,
np.ones((3, 3), np.uint8), iterations=1)

# Get contours
contours, hierarchy = cv2.findContours(img_erode,
cv2.RETR_TREE,cv2.CHAIN_APPROX_NONE)

output = img.copy()

for idx, contour in enumerate(contours):
    (x, y, w, h) = cv2.boundingRect(contour)
    # print("R", idx, x, y, w, h, cv2.contourArea(contour),
    hierarchy[0][idx])# hierarchy[i][0]: the index of the next
    contour of the same level
    # hierarchy[i][1]: the index of the previous contour of the
    same level# hierarchy[i][2]: the index of the first child
    # hierarchy[i][3]: the index of
    the parentif
    hierarchy[0][idx][3] == 0:
        cv2.rectangle(output, (x, y), (x + w, y + h), (70, 0, 0), 1)

cv2.imshow("Input", img)
cv2.imshow("Enlarged",
img_erode)
cv2.imshow("Output",
output) cv2.waitKey(0)
```

Результат: получаем иерархическое дерево контуров (параметр `cv2.RETR_TREE`). Первым идет общий контур картинки, затем контуры букв, затем внутренние контуры. Так как нужны только контуры букв, необходимо проверить что «родительским» является общий контур.



Сохраним каждую букву, предварительно отмасштабировав её до квадрата 28x28 (именно в таком формате хранится база MNIST). OpenCV построен на базе numpy, так что можно использовать функции работы с массивами для кропа и масштабирования.

```
def letters_extract(image_file: str,
                    out_size=28) -> List[Any]:
    img = cv2.imread(image_file)
    gray = cv2.cvtColor(img, cv2.COLOR_BGR2GRAY)
    ret, thresh = cv2.threshold(gray, 0, 255,
                                cv2.THRESH_BINARY)
    img_erode = cv2.erode(thresh,
                          np.ones((3, 3), np.uint8),
                          iterations=1)

    # Get contours
    contours, hierarchy = cv2.findContours(img_erode,
                                           cv2.RETR_TREE,
                                           cv2.CHAIN_APPROX_NONE)

    output = img.copy()

    letters = []
    for idx, contour in enumerate(contours):
        (x, y, w, h) = cv2.boundingRect(contour)
        # print("R", idx, x, y, w, h, cv2.contourArea(contour),
        #       hierarchy[0][idx])
        # hierarchy[i][0]: the index of the next
        # contour of the same level
        # hierarchy[i][1]: the index of the previous contour
        # of the same level
        # hierarchy[i][2]: the index of the
        # first child
        # hierarchy[i][3]: the
        # index of the parent if
        hierarchy[0][idx][3] ==
        0:
            cv2.rectangle(output, (x, y), (x + w, y +
            h), (70, 0, 0), 1)
            letter_crop = gray[y:y +
            h, x:x + w]
            # print(letter_crop.shape)

    # Resize letter
    # canvas to
```

```

square_size_max
= max(w, h)
letter_square = 255 * np.ones(shape=[size_max, size_max],
dtype=np.uint8)
if w > h:
    # Enlarge
    image top-
    bottom#
    #
    - - -
    =
    =
    =
    =
    =
    #
    y_pos = size_max//2 - h//2
    letter_square[y_pos:y_pos + h, 0:w]
    = letter_crop
elif w < h:
    # Enlarge
    image left-
    right# -||-
    -
    x_pos = size_max//2 - w//2
    letter_square[0:h, x_pos:x_pos +
w] = letter_crop
else:
    letter_square = letter_crop

# Resize letter to 28x28 and add letter and its X-
coordinate letters.append((x, w,
cv2.resize(letter_square, (out_size, out_size),
interpolation=cv2.INTER_AREA)))

# Sort array in place by X-coordinate
letters.sort(key=lambda x: x[0],
reverse=False)

return letters

```

Необходимо выполнить сортировку буквы по X-координате. Результаты сохраняются в виде tuple (x, w, letter), чтобы из промежутков между буквами потом выделить пробелы.

```

cv2.imshow("0", letters[0][2])
cv2.imshow("1", letters[1][2])
cv2.imshow("2", letters[2][2])
cv2.imshow("3", letters[3][2])
cv2.imshow("4
", letters[4][2])

```

```
cv2.waitKey(0)
)
```



Предварительный этап завешен.

Основной этап. Проектирование сверточной нейронной сети. Нейронная сеть (CNN) для распознавания

Необходимые библиотеки tensorflow

Исходный датасет EMNIST имеет 62 разных символа (A..Z, 0..9 и пр)

```
emnist_labels = [48, 49, 50, 51, 52, 53, 54, 55, 56, 57, 65, 66, 67, 68, 69, 70, 71, 72, 73, 74, 75, 76, 77, 78, 79, 80, 81, 82, 83, 84, 85, 86, 87, 88, 89, 90, 97, 98, 99, 100, 101, 102, 103, 104, 105, 106, 107, 108, 109, 110, 111, 112, 113, 114, 115, 116, 117, 118, 119, 120, 121, 122]
```

Нейронная сеть имеет 62 выхода, на входе она будет получать изображения 28x28, послераспознавания «1» будет на соответствующем выходе сети.

Модель сверточной нейронной сети:

```
from tensorflow import keras
from keras.models
import Sequential
from keras import
optimizers
from keras.layers import Convolution2D, MaxPooling2D, Dropout, Flatten, Dense,
Reshape,
LSTM, BatchNormalization
from keras.optimizers import SGD,
RMSprop, Adamfrom keras import
backend as K
from keras.constraints
import maxnormimport
tensorflow as tf

def
    emnist_
    model()
    : model
    =
    Sequent
    ial()
```

```

model.add(Convolution2D(filters=32, kernel_size=(3, 3), padding='valid',
input_shape=(28,28, 1), activation='relu'))
model.add(Convolution2D(filters=64, kernel_size=(3, 3),
activation='relu'))model.add(MaxPooling2D(pool_size=(2,
2)))
model.add(Dropout(0.25)
) model.add(Flatten())
model.add(Dense(512,
activation='relu'))
model.add(Dropout(0.5))
model.add(Dense(len(emnist_labels), activation='softmax'))
model.compile(loss='categorical_crossentropy', optimizer='adadelat',
metrics=['accuracy'])return model

```

это классическая сверточная сеть, выделяющая определенные признаки изображения (количество фильтров 32 и 64), к «выходу» которой подсоединена «линейная» сеть MLP, формирующая окончательный результат.

Обучение сверточной нейронной сети

Этап — обучение сети. Для этого необходимо воспользоваться базой EMNIST, скачать которую можно по ссылке (https://www.itl.nist.gov/iaui/vip/cs_links/EMNIST/gzip.zip).

Для чтения базы воспользуемся библиотекой **idx2numpy**. Подготовим данные для обучения и валидации.

```
import idx2numpy
```

```

emnist_path = '/home/Documents/TestApps/keras/emnist/'
X_train = idx2numpy.convert_from_file(emnist_path + 'emnist-byclass-
train-images-idx3-ubyte')
y_train = idx2numpy.convert_from_file(emnist_path + 'emnist-byclass-train-labels-idx1-
ubyte')

```

```

X_test = idx2numpy.convert_from_file(emnist_path + 'emnist-byclass-test-
images-idx3-ubyte')y_test = idx2numpy.convert_from_file(emnist_path +
'emnist-byclass-test-labels-idx1-ubyte')

```

```

X_train = np.reshape(X_train, (X_train.shape[0], 28, 28, 1))
X_test = np.reshape(X_test, (X_test.shape[0], 28, 28, 1))

```

```
print(X_train.shape, y_train.shape, X_test.shape, y_test.shape,
```

```

len(emnist_labels))k = 10
X_train =
X_train[:X_train.shape[0]
// k]y_train =
y_train[:y_train.shape[0]

```

```

// k] X_test =
X_test[:X_test.shape[0] //
k] y_test =
y_test[:y_test.shape[0] //
k]

# Normalize
X_train =
X_train.astype(np.flo
at32)X_train /= 255.0
X_test = X_test.astype(np.float32)
X_test /= 255.0

x_train_cat = keras.utils.to_categorical(y_train,
len(ernist_labels))y_test_cat =
keras.utils.to_categorical(y_test,
len(ernist_labels))

```

В результате подготовлено 2 набора: для обучения и валидации

При выполнении практического задания рекомендуется использовать 1/10 датасета для обучения (параметр k), в противном случае процесс займет не менее 10 часов.

Запускаем обучение сети, в конце процесса сохраняем обученную

```

модель на диск.# Set a learning rate reduction
learning_rate_reduction = keras.callbacks.ReduceLROnPlateau(monitor='val_acc',
patience=3,
verbose=1, factor=0.5, min_lr=0.00001)

```

```

# Required for learning_rate_reduction:
keras.backend.get_session().run(tf.global_variables_initializer())

```

```

model.fit(X_train, x_train_cat,
validation_data=(X_test, y_test_cat),
callbacks=[learning_rate_reduction], batch_size=64,
epochs=30)

```

```

model.save('ernist_letters.h5')

```

процесс обучения занимает около получаса на компьютере уровня core i9

Распознавание

Для распознавания загружаем модель и вызываем функцию predict_classes:model = keras.models.load_model('ernist_letters.h5')

```

def
ernist_predict_img(mod
el, img): img_arr =

```

```

np.expand_dims(img,
axis=0)img_arr = 1 -
img_arr/255.0
img_arr[0] =
np.rot90(img_arr[0], 3)
img_arr[0] =
np.fliplr(img_arr[0])
img_arr =
img_arr.reshape((1, 28,
28, 1))

result =
model.predict_classes([im
g_arr])return
chr(ernist_labels[result[0
]])

```

Окончательная функция, которая на входе получает файл с изображением, а на выходе выдает строку:

```

def img_to_str(model: Any,
image_file: str):letters =
letters_extract(image_file)
s_out = ""
for i in range(len(letters)):
dn = letters[i+1][0] - letters[i][0] - letters[i][1] if i <
len(letters) - 1 else 0s_out += ernist_predict_img(model,
letters[i][2])
if (dn >
letter
s[i][1
]/4):
s_out
+= ''
return s_out

```

Пример:

```

model =
keras.models.load_model('ernist_lette
rs.h5')s_out = img_to_str(model,
"hello_world.png") print(s_out)

```

Результат:

```

PS C:\Python> python .\keras_ernist.py
Using TensorFlow backend.

HELLØ WØRLD

```

Тема: Обработка изображений и видео с помощью библиотеки компьютерного зрения OpenCV

Цель: Изучить базовые операции обработки изображений с использованием реализаций соответствующих функций библиотеки компьютерного зрения OpenCV

Задания (вопросы) для подготовки:

Рассмотрим прототипы функций, реализующих базовые операции по обработке видео с помощью библиотеки OpenCV

В библиотеке OpenCV для Python работа с видео может проходить либо путем обработки видеофайлов из файловой системы, либо же путем обработки потокового видео, например, с **Web-камеры**.

Для работы с видео из файловой системы необходимо поместить видео файл в директорию общую с файлом кода.

Вариант для Web камеры подразумевает простое указание индекса камеры, здесь это - 0, если же на ПК установлено несколько камер, то указываем 1, и так далее.

```
import cv2 # Импорт модуля OpenCV

cap = cv2.VideoCapture(0); # Видео вывод с веб камеры
# компьютера, подключенной камере
cap = cv2.VideoCapture("VIDEO0102.mp4"); # Вывод с видео файла

print(cap.get(cv2.CAP_PROP_FRAME_WIDTH)) # Вывод в консоли размера нашего
окна.

cap.set(3,1280) #
Установка длины окна
cap.set(4,700) # Ширина
окна

p
r
i
n
t
(
с
а
p
.
g
e
t
(
3
```

```
)  
)  
p  
r  
i  
n  
t  
(  
c  
a  
p  
.  
g  
e  
t  
(  
4  
)  
)
```

Объект **cap** будет содержать метод **read**, который возвращает два параметра **ret** и **frame**. Первый параметр - булево значение и принимает 1 или 0.

Второй параметр **frame** - содержит данные в виде массива кадров. Можно вывести их на экран:

```
while(True): # Вывод кадров  
  
    производится в цикле ret, frame =  
  
    cap.read()  
    frame = cv2.rectangle(frame,(384,0),(510,128),(0,0,255),10) # добавление к  
    видео выводу объекта прямоугольника  
  
    print(frame) # Вывод массива в консоль  
  
    cv2.imshow("frame",frame) # Метод для визуализации массива кадров  
  
cap.release  
(  
cv2.destro  
yAllWind  
ows()
```

В **Python** модуле **cv2** есть фильтры. Рассмотрим **Гауссовский** фильтр и фильтр **Canny**, который будет выдавать черно-белые кадры в виде белых контуров и черных областей.

Код:

```

while(True):
    ret, frame = cap.read()
    frame = cv2.rectangle(frame,(384,0),(510,128),(0,0,255),10)
    gray = cv2.cvtColor(frame, cv2.COLOR_BGR2GRAY) # Перевод массива
кадров в черно-белую градацию

    edges = cv2.Canny(gray, 1, 50) #
Нахождение контуров
cv2.imshow("edges",edges) #
обработанный вариант

print(frame)

cv2.imshow("frame", frame) # оригинальный вариант

cap.release
()
cv2.destro
yAllWindows
ows()

```

Анализируя обработанное видео можно заметить большое количество контуров или шумов, мешающим правильно распознать образ нашего изначального варианта. Их необходимо устранить. Используем для этого гауссовский фильтр:

```

gray = cv2.cvtColor(frame, cv2.COLOR_BGR2GRAY)
gray = cv2.GaussianBlur(gray, (7, 7), 1.5) # Параметры позволяют регулировать
шумность

edges = cv2.Canny(gray, 1, 50) #
Нахождение контуров
cv2.imshow("edges",edges) # вывод на
экран

```

Анализ:

Объект **frame** передается в метод модуля cv2 **GaussianBlur**, из которого создается другой объект **gray**, передающийся в метод **Canny**. Т.е. сначала черно-белый массив фильтруется от шумов, а затем создается его контурная модель, которая выводится на экран методом **cv2.imshow("edges",edges)**. Подобные операции являются этапами в детектировании различных объектов и их распознавании.

Семинарское занятие по теме 5

Тема: Обнаружение и распознавание объектов

Цель: изучить возможности обнаружения и распознавание объектов в Python с помощью OpenCV

Задания (вопросы) для подготовки:

Задание для практической работы: Обнаружение и распознавание объектов в Python с помощью OpenCV

Обнаружение объекта это современная компьютерная технология, которая связана обработкой изображений, глубоким обучением и компьютерным зрением для обнаружения объектов, присутствующих в файле изображения. Все технологии, используемые в методе обнаружения объектов (как мы упоминали ранее), связаны обнаружением экземпляров объекта на изображении или видео

Требования для использования техники обнаружения объектов - каскадная техника Хаара

Каскадная техника Хаара – это подход, основанный на машинном обучении, при котором мы используем множество положительных и отрицательных изображений, чтобы научиться классифицировать изображения. Каскадные классификаторы Хаара считаются эффективным способом обнаружения объектов с помощью библиотеки OpenCV

Необходимые библиотеки:

- библиотека OpenCV
- библиотека matplotlib

Необходимо написать программу в знершт для обнаружения объектов. В качестве примера можно взять любое изображение или стоп-кадр видеопоследовательности(opencv-od.png):



Открытие изображения:

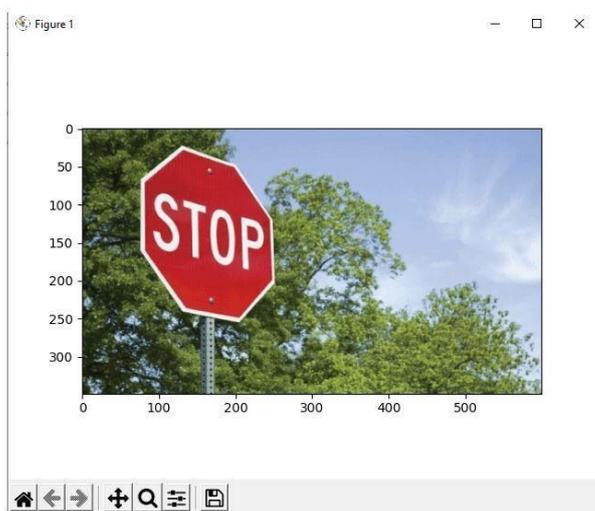
```
# Import
OpenCV
module
import cv2
# Import pyplot from
matplotlib as pltdfrom
matplotlib import
pyplot as pltd
# Opening the image from files
imaging =
cv2.imread("opencv-
```

```

od.png")# Altering
properties of image with
cv2
img_gray = cv2.cvtColor(imaging,
cv2.COLOR_BGR2GRAY) imaging_rgb =
cv2.cvtColor(imaging, cv2.COLOR_BGR2RGB)
# Plotting image with subplot() from plt
plt.subplot(1, 1, 1)
# Displaying
image in the
output
plt.imshow(ima
ging_rgb)
plt.show()

```

Результат:



Распознавание или обнаружение объекта

Для успешного поиска объекта воспользуемся каскадным классификатором, егонеобходимо импортировать в программу в виде XML-файла.

Затем необходимо использовать функцию `detectMultiScale()` с импортированным каскадным файлом, чтобы определить, присутствует ли объект на изображении или нет

Обнаружение объекта на изображении с помощью метода

```

detectMultiScale()xml_data = cv2.CascadeClassifier('XML-data.xml')
detecting = xml_data.detectMultiScale(imaging_gray, minSize =(30, 30))

```

Код:

```

# Import
OpenCV
module
import cv2

```

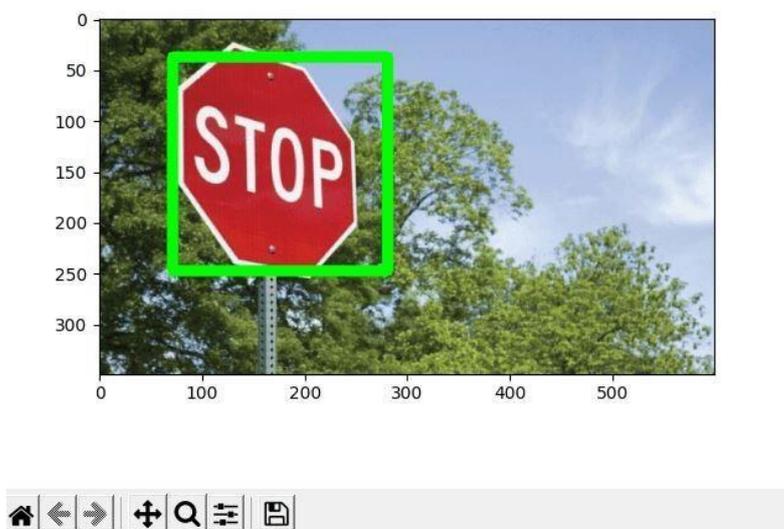
```

# Import pyplot from matplotlib as plt

from matplotlib
import pyplot as pltd#
Opening the image
from files
imaging =
cv2.imread("opencv-
od.png")# Altering
properties of image with
cv2
imaging_gray = cv2.cvtColor(imaging,
cv2.COLOR_BGR2GRAY)imaging_rgb =
cv2.cvtColor(imaging, cv2.COLOR_BGR2RGB)
# Importing Haar cascade classifier
xml_data xml_data =
cv2.CascadeClassifier('XML-
data.xml')
# Detecting object in the image with Haar
cascade classifierdetecting =
xml_data.detectMultiScale(imaging_gray,
min
Size =(30, 30))# Amount
of object detected
amountDetecting =
len(detecting)
# Using if condition to highlight the
object detectedif amountDetecting
!= 0:
    for(a, b, width, height) in detecting:
        cv2.rectangle(imaging_rgb,(a, b), # Highlighting detected
            object with rectangle(a + height, b + width),
            (0, 275, 0), 9)
# Plotting image with
subplot() from plt
pltd.subplot(1, 1, 1)
# Displaying
image in the
output
pltd.imshow(ima
ging_rgb)
pltd.show()

```

В случае правильного решения задачи будет получен следующий результат:



Семинарское занятие 6.

Тема занятия: Построение карты глубины

Алгоритм построения карты глубины.

Формулировка задания:

Реализовать алгоритм построения карты глубины путем анализа движения камеры.

Входными данными алгоритма является последовательность кадров обного объекта.

Результатом работы алгоритма должна являться последовательность карт глубины, соответствующих кадрам исходной последовательности.

Предлагается реализовать и попробовать комбинировать следующие подходы:

1. Вычисление модуля оптического потока
2. Билатеральная фильтрация карт глубины
3. Сопоставление гистограмм яркости карт глубины соседних кадров
 1. Различные метрики: Минковского, Евклидова, Манхэттенская, Махаланобиса, Canberra distance, Хэмминга, косинусное, расстояние Джаккарда, DTW, Левенштейна.
 2. Приложения метрического подхода: нечёткий матчинг таблиц, Ленкор, в DL, классификация текстов.

5.2. Задания и упражнения для самостоятельной работы

№ п/п	Наименование темы дисциплины	Задания и упражнения для самостоятельной работы
1	Цифровое изображение Классификация изображений Нейросетевые методы классификации и поиска похожих изображений Анализ и синтез изображений	1. Реализовать алгоритм для реконструкции фотографий Прокудина-Горского на основе сопоставления изображений и методов обработки изображений 2. Реализовать алгоритм интеллектуального изменения пропорций изображений на основе поиска путей минимальной стоимости 3. Реализовать модуль классификации изображений дорожных знаков с использованием признаков типа

	Нейросетевые методы распознавания видео Реконструкция изображений	гистограммы ориентаций градиентов и метода опорных векторов 4. Разработать нейросетевой алгоритм для регрессии ключевых точек лица человека 5. Реализовать процедуру построения классификатора изображений на основе дообучения нейронной сети 6. Реализация детектора лиц людей на основе полносвязной нейросети
--	--	--

5.4. Перечень тем (задания) для курсовой работы

Не предусмотрено

6. Оценочные средства для проведения текущего контроля и промежуточной аттестации обучающихся

6.1 Примерные оценочные средства, включая тестовые оценочные задания для проведения текущего контроля и промежуточной аттестации обучающихся по дисциплине (модулю) приведены в Приложении 1 к рабочей программе дисциплины.

6.2 Оценочные средства для проведения промежуточной аттестации обучающихся по дисциплине (модулю) включают следующие разделы:

- перечень компетенций, формируемых в процессе освоения учебной дисциплины;
- описание показателей и критериев оценивания компетенций, описание шкал оценивания;
- типовые контрольные задания или иные материалы, необходимые для оценки результатов обучения по учебной дисциплине, обеспечивающих достижение планируемых результатов освоения образовательной программы;
- методические материалы, определяющие процедуры оценивания результатов обучения по учебной дисциплине, обеспечивающих достижение планируемых результатов освоения образовательной программы.

7. Учебно-методическое и информационное обеспечение дисциплины

а) основная литература:

1. Прохоров, С. А. Компьютерный инструментарий информационных технологий и цветографические интерпретации живописи в архитектурном образовании : учебное пособие / С. А. Прохоров. — Москва : Ай Пи Ар Медиа, 2022. — 134 с. — ISBN 978-5-4497-1513-5. — Текст : электронный // Цифровой образовательный ресурс IPR SMART : [сайт]. — URL: <https://www.iprbookshop.ru/117032.html> (дата обращения: 03.02.2022). — Режим доступа: для авторизир. пользователей
2. Катунин, Г. П. Компьютерные технологии в фотографии. Усиление резкости цифровых изображений : учебное пособие для СПО / Г. П. Катунин. — Саратов, Москва : Профобразование, Ай Пи Ар Медиа, 2022. — 375 с. — ISBN 978-5-4488-1350-4, 978-5-4497-1564-7. — Текст : электронный // Цифровой образовательный ресурс IPR SMART : [сайт]. — URL: <https://www.iprbookshop.ru/118466.html> (дата обращения: 03.02.2022). — Режим доступа: для авторизир. пользователей

б) дополнительная литература:

1. Шакирьянов, Э. Д. Компьютерное зрение на PythonOR . Первые шаги / Э. Д. Шакирьянов. — Москва : Лаборатория знаний, 2021. — 161 с. — ISBN 978-5-00101-944-2. — Текст : электронный // Цифровой образовательный ресурс IPR SMART : [сайт]. — URL: <https://www.iprbookshop.ru/103032.html> (дата обращения: 03.02.2022). — Режим доступа: для авторизир. пользователей

в) интернет-ресурсы:

1. ЭБС IPR Books <http://iprbookshop.ru> (учебники и учебные пособия, монографии, сборники научных трудов, научная периодика, профильные журналы, справочники, энциклопедии);

2. ООО «ИВИС» <https://dlib.eastview.com> (электронные версии периодических изданий ООО «ИВИС»);
3. Web of Science <http://webofknowledge.com/> (обширная международная универсальная реферативная база данных);
4. Scopus <https://www.scopus.com> (международная универсальная реферативная база данных, индексирующая более 21 тыс. наименований научно-технических, гуманитарных и медицинских журналов, материалов конференций примерно 5000 международных издательств);
5. Научная электронная библиотека eLIBRARY.RU <https://elibrary.ru> (крупнейший российский информационный портал в области науки, технологии, медицины и образования);
6. Единая межведомственная информационно-статистическая система (ЕМИСС) <https://fedstat.ru/indicators/>
7. Федеральная служба государственной статистики <http://www.gks.ru/>
8. Министерство науки и высшего образования Российской Федерации <https://minobrnauki.gov.ru/>
9. Министерство экономического развития Российской Федерации <http://economy.gov.ru/minec/about/structure>
10. Федеральное агентство по печати и массовым коммуникациям <http://www.fapmc.ru/rospechat.html>
11. Министерство цифрового развития, связи и массовых коммуникаций Российской Федерации <https://digital.gov.ru/ru>
12. <http://www.mathnet.ru/> Общероссийский математический портал (информационная система)

8. Материально-техническое обеспечение дисциплины

№	Наименование учебных аудиторий (лабораторий) и помещений для самостоятельной работы	Перечень оборудования и технических средств обучения	Программное обеспечение, в т.ч. отечественного производства
1	Учебные аудитории для проведения занятий лекционного типа	<p>Преподавательский стол; столы обучающихся; стулья; классная доска; мультимедийный комплекс; наглядные пособия (плакаты)</p> <p><i>Место, оборудованное для лиц с ограниченными возможностями и.</i></p> <p>Лицензионное программное обеспечение, подключенное к Интернет</p>	<ol style="list-style-type: none"> 1. 7-Zip (Бесплатное ПО); 2. 10-Strike Network Inventory ПО РФ (ПО) 3. Ductor Academic ПО РФ (Бесплатное ПО); https://basegroup.ru/deductor/manual/licence-deductor-academic 4. Eset Endpoint security (Платное ПО) https://help.eset.com/eula/GIMP (Бесплатное ПО); https://docs.gimp.org/2.10/ru/ 5. microsoft office профессиональный плюс 2016 (ПО) https://www.microsoft.com/en-us/Useterms/Retail/Office/2016Professional/Useterms_Retail_Office_2016Professional_RUS.htm 6. Microsoft power Bi (Бесплатное ПО); https://powerbi.microsoft.com/ru-ru/windows-license-terms/icrosoft Visual Studio (Бесплатное ПО); https://visualstudio.microsoft.com/ru/license-terms/mt171584/ 7. Notepad ++ (Бесплатное ПО); https://www.gnu.org/licenses/old-licenses/gpl-2.0.html 8. Zoom (Бесплатное ПО); https://explore.zoom.us/ru/terms/ 9. Anaconda3 2019 (Бесплатное ПО); 10. https://www.anaconda.com/eula-anaconda-individual-edition Android studio (Бесплатное ПО); https://developer.android.com/studio/terms 11. Brackets (Бесплатное ПО); https://github.com/brackets-cont/brackets/blob/master/LICENSE 12. CodeBlocks (Бесплатное ПО); https://www.codeblocks.org/license/Firebird (Бесплатное ПО); https://firebirdsql.org/en/licensing/

			<p>13. KNIME analytics platform (Бесплатное ПО); https://www.knime.com/downloads/full-license</p> <p>14. Loginom community РФ ПО (Бесплатное ПО); https://loginom.ru/legal</p> <p>15. Monogame SDK (Бесплатное ПО); https://github.com/MonoGame/MonoGame/blob/develop/LICENSE.txt</p> <p>Openproj (Бесплатное ПО); https://opensource.org/licenses/CPAL-1.0</p> <p>16. tableau 2019 (Бесплатное ПО); https://www.tableau.com/legal</p> <p>17. Visual studio community 2017 (Бесплатное ПО); https://www.google.com/url?sa=t&rct=j&q=&esrc=s&source=web&cd=&ved=2ahUKEwi-qZeRxxv7zAhXhsYsKHZoRBAsQFnoECBgQAQ&url=https%3A%2F%2Fvisualstudio.microsoft.com%2Fwp-content%2Fuploads%2F2017%2F01%2FVS2017_COMMUNITY_RC_RUS_Eula.1049-1.docx&usg=AOvVaw0tLx1QA4E2McNypfRn9tTo</p> <p>18. Visual studio community 2019 (Бесплатное ПО); https://visualstudio.microsoft.com/ru/license-terms/mlt110718/</p> <p>19. Консультант плюс</p>
2	Компьютерный класс	<p>Преподавательский стол; столы обучающихся; стулья; классная доска; мультимедийный комплекс; ПК преподавателя; ПК обучающихся; наглядные пособия (плакаты) <i>Место, оборудованное для лиц с ограниченными возможностями и.</i> Лицензионное программное обеспечение, подключенное к Интернет</p>	<p>1. 7-Zip (Бесплатное ПО);</p> <p>2. 10-Strike Network Inventory ПО РФ (ПО)</p> <p>3. Ductor Academic ПО РФ (Бесплатное ПО); https://basegroup.ru/deductor/manual/licence-deductor-academic</p> <p>4. Eset Endpoint security (Платное ПО) https://help.eset.com/eula/GIMP (Бесплатное ПО); https://docs.gimp.org/2.10/ru/</p> <p>5. microsoft office профессиональный плюс 2016 (ПО) https://www.microsoft.com/en-us/Useterms/Retail/Office/2016Professional/Useterms_Retail_Office_2016Professional_RUS.htm</p> <p>6. Microsoft power Bi (Бесплатное ПО); https://powerbi.microsoft.com/ru-ru/windows-license-terms/ Microsoft Visual Studio (Бесплатное ПО); https://visualstudio.microsoft.com/ru/license-terms/mt171584/</p> <p>7. Notepad ++ (Бесплатное ПО); https://www.gnu.org/licenses/old-licenses/gpl-2.0.html</p> <p>8. Zoom (Бесплатное ПО); https://explore.zoom.us/ru/terms/</p> <p>9. Anaconda3 2019 (Бесплатное ПО);</p> <p>10. https://www.anaconda.com/eula-anaconda-individual-edition Android studio (Бесплатное ПО); https://developer.android.com/studio/terms</p> <p>11. Brackets (Бесплатное ПО); https://github.com/brackets-cont/brackets/blob/master/LICENSE</p> <p>12. CodeBlocks (Бесплатное ПО);https://www.codeblocks.org/license/ Firebird (Бесплатное ПО); https://firebirdsql.org/en/licensing/</p> <p>13. KNIME analytics platform (Бесплатное ПО); https://www.knime.com/downloads/full-license</p> <p>14. Loginom community РФ ПО (Бесплатное ПО); https://loginom.ru/legal</p> <p>15. Monogame SDK (Бесплатное ПО); https://github.com/MonoGame/MonoGame/blob/develop/LICENSE.txt</p> <p>Openproj (Бесплатное ПО); https://opensource.org/licenses/CPAL-1.0</p> <p>16. tableau 2019 (Бесплатное ПО); https://www.tableau.com/legal</p> <p>17. Visual studio community 2017 (Бесплатное ПО); https://www.google.com/url?sa=t&rct=j&q=&esrc=s&source=web&cd=&ved=2ahUKEwi-qZeRxxv7zAhXhsYsKHZoRBAsQFnoECBgQAQ&url=https%3A%2F%2Fvisualstudio.microsoft.com%2Fwp-content%2Fuploads%2F2017%2F01%2FVS2017_COMMUNITY_RC_RUS_Eula.1049-1.docx&usg=AOvVaw0tLx1QA4E2McNypfRn9tTo</p> <p>18. Visual studio community 2019 (Бесплатное ПО); https://visualstudio.microsoft.com/ru/license-terms/mlt110718/</p> <p>19. Консультант плюс</p>

3	Учебные аудитории для проведения занятий семинарского типа, курсового проектирования (выполнения курсовых работ), групповых и индивидуальных консультаций, текущего контроля и промежуточной аттестации, а также самостоятельной работы обучающихся	Преподавательский стол; столы обучающихся; стулья; классная доска; мультимедийный комплекс; ПК преподавателя; ПК обучающихся; наглядные пособия (плакаты) <i>Место, оборудованное для лиц с ограниченными возможностями и.</i> Лицензионное программное обеспечение, подключенное к Интернет	<ol style="list-style-type: none"> 1. 7-Zip (Бесплатное ПО); 2. 10-Strike Network Inventory ПО РФ (ПО) 3. Ductor Academic ПО РФ (Бесплатное ПО); https://basegroup.ru/deductor/manual/licence-deductor-academic 4. Eset Endpoint security (Платное ПО) https://help.eset.com/eula/GIMP (Бесплатное ПО); https://docs.gimp.org/2.10/ru/ 5. microsoft office профессиональный плюс 2016 (ПО) https://www.microsoft.com/en-us/Useterms/Retail/Office/2016Professional/Useterms_Retail_Office_2016Professional_RUS.htm 6. Microsoft power Bi (Бесплатное ПО); https://powerbi.microsoft.com/ru-ru/windows-license-terms/ icrosoft Visual Studio (Бесплатное ПО); https://visualstudio.microsoft.com/ru/license-terms/mt171584/ 7. Notepad ++ (Бесплатное ПО); https://www.gnu.org/licenses/old-licenses/gpl-2.0.html 8. Zoom (Бесплатное ПО); https://explore.zoom.us/ru/terms/ 9. Anaconda3 2019 (Бесплатное ПО); 10. https://www.anaconda.com/eula-anaconda-individual-edition Android studio (Бесплатное ПО); https://developer.android.com/studio/terms 11. Brackets (Бесплатное ПО); https://github.com/brackets-cont/brackets/blob/master/LICENSE 12. CodeBlocks (Бесплатное ПО); https://www.codeblocks.org/license/Firebird (Бесплатное ПО); https://firebirdsql.org/en/licensing/ 13. KNIME analytics platform (Бесплатное ПО); https://www.knime.com/downloads/full-license 14. Loginom community РФ ПО (Бесплатное ПО); https://loginom.ru/legal 15. Monogame SDK (Бесплатное ПО); https://github.com/MonoGame/MonoGame/blob/develop/LICENSE.txt Openproj (Бесплатное ПО); https://opensource.org/licenses/CPAL-1.0 16. tableau 2019 (Бесплатное ПО); https://www.tableau.com/legal 17. Visual studio community 2017 (Бесплатное ПО); https://www.google.com/url?sa=t&rct=j&q=&esrc=s&source=web&cd=&ved=2ahUKEwi-qZeRxv7zAhXhsYsKHZoRBAsQFnoECBgQAQ&url=https%3A%2F%2Fvisualstudio.microsoft.com%2Fwp-content%2Fuploads%2F2017%2F01%2FVS2017_COMMUNITY_RC_RUS_Eula.1049-1.docx&usg=AOvVaw0tLx1QA4E2McNypfRn9tTo 18. Visual studio community 2019 (Бесплатное ПО); https://visualstudio.microsoft.com/ru/license-terms/mlt110718/ 19. Консультант плюс
4	Библиотека с читальным залом	Стол обучающегося, стулья, ПК обучающегося, принтер Электронная библиотечная система и библиотечное абонентное обслуживание (учебная литература на бумажных носителях) Лицензионное программное обеспечение, подключенное к Интернет	<ol style="list-style-type: none"> 1. 7-Zip (Бесплатное ПО); 2. microsoft office профессиональный плюс 2016 (ПО) https://www.microsoft.com/en-us/Useterms/Retail/Office/2016Professional/Useterms_Retail_Office_2016Professional_RUS.htm 3. Microsoft power Bi (Бесплатное ПО); https://powerbi.microsoft.com/ru-ru/windows-license-terms/ icrosoft Visual Studio (Бесплатное ПО); https://visualstudio.microsoft.com/ru/license-terms/mt171584/ 4. Антиплагиат 5. Консультант плюс

9. Перечень информационных технологий

Образовательный процесс по дисциплине поддерживается средствами электронной информационно-образовательной среды Университета, которая обеспечивает:

- доступ к учебным планам, рабочим программам дисциплин (модулей), практик, к изданиям электронных библиотечных систем и электронным образовательным ресурсам, указанным в рабочей программе, через личный кабинет студента и преподавателя;
- фиксацию хода образовательного процесса, результатов промежуточной аттестации и результатов освоения основной образовательной программы;
- проведение всех видов занятий, процедур оценки результатов обучения, реализация которых предусмотрена с применением дистанционных образовательных технологий;
- формирование электронного портфолио обучающегося, в том числе сохранение работ обучающегося, рецензий и оценок на эти работы со стороны любых участников образовательного процесса;
- взаимодействие между участниками образовательного процесса, в том числе синхронное и (или) асинхронное взаимодействие посредством сети Интернет.

Каждый обучающийся обеспечен индивидуальным неограниченным доступом к электронно-библиотечной системе (ЭБС университета), содержащей издания учебной, учебно-методической и иной литературы по основным изучаемым дисциплинам и сформированной на основании прямых договоров с правообладателями.

Перечень программного обеспечения:

1. Adobe flash player 31;
2. Adobe reader 10;
3. Java 6.0;
4. K-Lite Codec Pack;
5. Win rar;
6. Microsoft Office 10;
7. Microsoft Visio 10;
8. Microsoft Visual studio;

Профессиональные базы данных и информационно-справочные системы:

1. Kaspersky Endpoint Security для бизнеса <http://inion.ru/resources/bazy-dannykh-inion-ran/> - библиографические базы данных ИНИОН РАН по социальным и гуманитарным наукам;
2. <http://inion.ru/resources/bazy-dannykh-inion-ran/> - библиографические базы данных ИНИОН РАН по социальным и гуманитарным наукам;
3. <http://www.scopus.com/> - реферативная база данных Scopus – международная универсальная реферативная база данных;
4. <http://elibrary.ru/defaultx.asp> - крупнейший российский информационный портал электронных журналов и баз данных по всем отраслям наук.
5. <https://www.cfin.ru/finanalysis/math/>, База данных «Финансовая математика – Библиотека управления» - Корпоративный менеджмент

10. Методические указания для обучающихся

10.1. Преподавание дисциплины осуществляется в соответствии с Федеральным государственным образовательным стандартом высшего образования

Основными формами получения и закрепления знаний по данной дисциплине являются занятия лекционного и семинарского типа, самостоятельная работа обучающегося, в том числе под руководством преподавателя, прохождение рубежного контроля.

Основной объем часов по изучению дисциплины согласно учебным планам приходится на самостоятельную работу обучающихся. Самостоятельная работа включает в себя изучение

учебной, учебно-методической и специальной литературы, её конспектирование, подготовку к занятиям семинарского типа, текущему контролю и промежуточной аттестации (зачету или (и) экзамену).

Текущий контроль успеваемости по учебной дисциплине и промежуточная аттестация осуществляются в соответствии с Положением о текущем контроле успеваемости и промежуточной аттестации обучающихся по образовательным программам высшего образования: программам бакалавриата, программам специалитета, программам магистратуры.

Наличие в Университете электронной информационно-образовательной среды, а также электронных образовательных ресурсов позволяет осваивать курс инвалидам и лицам с ОВЗ.

10.2. Особенности освоения учебной дисциплины инвалидами и лицами с ограниченными возможностями здоровья.

Особенности освоения учебной дисциплины инвалидами и лицами с ОВЗ определены в Положении об организации обучения студентов-инвалидов и студентов с ограниченными возможностями здоровья, утвержденным приказом ректора.

Обучение инвалидов и лиц с ОВЗ может осуществляться индивидуально, а также с применением электронного обучения, дистанционных образовательных технологий.

Выбор методов и средств обучения, образовательных технологий и учебно-методического обеспечения реализации образовательной программы осуществляется Университетом самостоятельно, исходя из необходимости достижения обучающимися планируемых результатов освоения образовательной программы, а также с учетом индивидуальных возможностей обучающихся из числа инвалидов и лиц с ОВЗ.

Форма проведения промежуточной аттестации для студентов-инвалидов и лиц с ОВЗ устанавливается с учетом индивидуальных психофизических особенностей (устно, письменно на бумаге, письменно на компьютере, в форме тестирования и т.п.). При необходимости инвалидам и лицам с ОВЗ предоставляется дополнительное время для подготовки ответа на зачете или экзамене.

В группах, в состав которых входят студенты с ОВЗ, с целью реализации индивидуального подхода, а также принципа индивидуализации и дифференциации, рекомендуется использовать технологию нелинейной конструкции учебных занятий, предусматривающую одновременное сочетание фронтальных, групповых и индивидуальных форм работы с различными категориями студентов, в т.ч. имеющих ОВЗ.

В случае наличия обучающихся с нарушением функций опорно-двигательного аппарата, зрения и слуха, они обеспечиваются необходимым оборудованием, имеющимся в Университете, а также предоставляемым в рамках Соглашения с РУМЦ РГСУ от 14 ноября 2019 года.

11. Методические рекомендации преподавателю по организации учебного процесса по дисциплине

При изучении дисциплины рекомендуется использовать следующий набор средств и способов обучения:

- рекомендуемую основную и дополнительную литературу;
 - задания для подготовки к занятиям семинарского типа (вопросы для обсуждения, кейс задания, расчетные задачи и др.);
 - задания для текущего контроля успеваемости (задания для самостоятельной работы обучающихся, тестовые задания в рамках электронной системы тестирования);
 - вопросы и задания для подготовки к промежуточной аттестации по итогам освоения дисциплины, позволяющие оценить знания, умения и уровень приобретенных компетенций.
- При проведении занятий лекционного и семинарского типа, в том числе в форме вебинаров и on-line курсов необходимо строго придерживаться тематического плана дисциплины,

приведенного в РПД. Необходимо уделить внимание рассмотрению вопросов и заданий, включенных в тестовые оценочные задания, при необходимости, решить аналогичные задачи с объяснением алгоритма решения.

Следует обратить внимание обучающихся на то, что для успешной подготовки к текущему контролю (выполнению ТОЗ) и промежуточной аттестации (зачету или экзамену) недостаточно прочитать рабочий учебник, размещенный в личном кабинете. Нужно изучить материалы основной и дополнительной литературы, список которой приведен в РПД, законодательные и нормативные акты, а также материалы, рекомендованные в разделе «Перечень ресурсов информационно-телекоммуникационной сети «Интернет», необходимых для освоения дисциплины».

Текущий контроль успеваемости по учебной дисциплине и промежуточная аттестация осуществляются в соответствии с Положением о текущем контроле успеваемости и промежуточной аттестации обучающихся по образовательным программам высшего образования: программам бакалавриата, программам специалитета, программам магистратуры.

Программа разработана Замега Э.Н.

Рабочая программа дисциплины рассмотрена и принята на заседании кафедры от 27.01.2022 г., протокол №6.

**Лист регистрации изменений и дополнений
в рабочую учебную программу**

Составителем внесены следующие изменения:

Содержание изменений	Номер протокола и дата заседания кафедры, по утверждению изменений
Рабочая программа дисциплины дополнена и утверждена	№ 1 от 28.08.2023

ОЦЕНОЧНЫЕ СРЕДСТВА ПО ДИСЦИПЛИНЫ
РАСПОЗНАВАНИЕ ОБРАЗОВ И КОМПЬЮТЕРНОЕ ЗРЕНИЕ

Направление подготовки:
09.03.03 Прикладная информатика

Направленность (профиль)
Искусственный интеллект и анализ данных

Уровень высшего образования: бакалавриат

Москва – 2023

СОДЕРЖАНИЕ

1. Перечень компетенций, формируемых в процессе освоения учебной дисциплины
2. Описание показателей и критериев оценивания компетенций, описание шкал оценивания
3. Типовые контрольные задания или иные материалы.
4. Методические материалы, определяющие процедуры оценивания результатов обучения по учебной дисциплине

1. Перечень компетенций, формируемых в процессе освоения учебной дисциплины

1.1. Планируемые результаты освоения образовательной программы

В результате изучения дисциплины обучающийся должен обладать следующими компетенциями (результатами освоения образовательной программы):

Коды компетенций	Содержание компетенций
ПК-7	Способность выполнять интеллектуальный анализ больших данных
ПК-9	Способность применять математические методы моделирования процессов обработки информации с использованием средств интеллектуального анализа данных

1.2. Взаимосвязь планируемых результатов обучения по дисциплине с планируемыми результатами освоения образовательной программы

Коды и формулировка компетенции	Индикаторы компетенции	Запланированные результаты обучения
ПК-7 Способность выполнять интеллектуальный анализ больших данных	<p>ПК 7.1 Знать методы и инструментальные средства интеллектуального анализа больших данных</p> <p>ПК 7.2 Уметь выбирать средства представления результатов аналитики больших данных</p> <p>ПК 7.3 Владеть техническими, программными средствами для разработки алгоритмов и программ в области интеллектуального анализа данных</p>	<p>ПК 7.1.1 знать методы и инструментальные средства для машинного обучения и анализа больших данных - предметную область анализа больших данных - теоретические и прикладные основы анализа данных</p> <p>ПК 7.1.2 уметь использовать инструментальные средства машинного обучения и анализа больших данных - проводить сравнительный анализ методов и инструментальных средств анализа больших данных</p> <p>ПК 7.1.3 владеть инструментальными средствами машинного обучения и анализа больших данных</p> <p>ПК 7.2.1 знать средства визуализации результатов решения задач машинного обучения и анализа больших данных методы интерпретации и визуализации решения задач машинного обучения и анализа больших данных</p> <p>ПК 7.2.2 уметь использовать стандартные программные библиотеки для визуализации решений задач машинного обучения и анализа данных</p> <p>ПК 7.2.3 владеть стандартными программными библиотеками для решения задач машинного обучения и анализа данных</p> <p>ПК 7.3.1 знать программные платформы и библиотеки для решения задач машинного обучения и анализа больших данных стандарты проведения анализа данных</p> <p>ПК 7.3.2 уметь использовать программные платформы и библиотеки для решения задач машинного обучения и анализа больших данных</p> <p>ПК 7.3.3 владеть программными средствами для разработки алгоритмов машинного обучения, алгоритмами построения искусственных нейронных сетей</p>
ПК-9 Способность применять математические методы моделирования процессов обработки информации с использованием средств интеллектуального анализа данных	<p>ПК 9.1 Знать принципы решения задач машинного обучения и интеллектуального анализа данных</p> <p>ПК 9.2 Уметь создавать алгоритмические и математические модели прикладных задач интеллектуального анализа данных</p> <p>ПК 9.3 Владеть</p>	<p>ПК 9.1.1 знать основные принципы решения задач машинного обучения и анализа данных</p> <p>ПК 9.1.2 уметь проводить спецификацию задачи, реализовывать программы на алгоритмических языках высокого уровня, интерпретировать полученные результаты</p> <p>ПК 9.1.3 владеть методами и алгоритмами машинного обучения</p> <p>ПК 9.2.1 знать математические и алгоритмические модели интеллектуального анализа данных, методы оценки временных и стоимостных характеристик машинного обучения и анализа данных</p> <p>ПК 9.2.2 уметь создавать алгоритмические модели типовых прикладных задач обработки информации,</p>

	навыками построения описательных и прогнозных аналитических моделей с использованием современных инструментов интеллектуального анализа данных	реализовывать программы на алгоритмических языках высокого уровня ПК 9.2.3 владеть навыками использования прикладных библиотек для решения задач машинного обучения ПК 9.3.1 описательные и прогнозные аналитические модели для интеллектуального анализа данных - использовать современные программные инструменты интеллектуального анализа данных 9.3.2 разрабатывать и оценивать модели больших данных 9.3.3 навыками построения описательных и прогнозных аналитических моделей с использованием современных инструментов интеллектуального анализа данных, навыками выявления требований к результатам анализа больших данных в предметной области
--	--	--

2. Описание показателей и критериев оценивания компетенций, описание шкал оценивания

2.1. Текущий контроль успеваемости по учебной дисциплине и промежуточная аттестация осуществляются в соответствии с Положением о текущем контроле успеваемости и промежуточной аттестации обучающихся по образовательным программам высшего образования: программам бакалавриата, программам специалитета, программам магистратуры.

2.2. В семестре степень освоения компетенций оценивается по 100-балльной шкале в форме тестирования два раза в семестр. В зачетно-экзаменационный период баллы приводят к среднеарифметическому значению и переводятся в традиционную четырехбалльную систему. Данная оценка может повлиять на итоговую.

Этапы формирования компетенций и критериев оценивания компетенций на различных этапах формирования:

Коды и формулировка компетенции	Индикаторы достижения компетенции	Уровень выраженности и критерии оценивания	Этапы формирования
ПК-7 Способность выполнять интеллектуальный анализ больших данных	ПК 7.1 Знать методы и инструментальные средства интеллектуального анализа больших данных	Низкий (пороговый): Демонстрирует поверхностные знания методов и инструментальных средств для машинного обучения и анализа больших данных - предметную область анализа больших данных - теоретические и прикладные основы анализа данных Средний: Уверенно показывает умение использовать инструментальные средства машинного обучения и анализа больших данных - проводить сравнительный анализ методов и инструментальных средств анализа больших данных показывает Высокий: На отличном уровне показывает владение инструментальными средствами машинного обучения и анализа больших данных	Первый этап: Проведение текущего контроля успеваемости по дисциплине Второй этап: Проведение промежуточной аттестации по учебной дисциплине.
	ПК 7.2 Уметь выбирать средства представления	Низкий (пороговый): Демонстрирует поверхностные знания средств визуализации результатов решения задач машинного обучения и анализа больших данных методы	Первый этап: Проведение текущего контроля

	результатов аналитики больших данных	интерпретации и визуализации решения задач машинного обучения и анализа больших данных Средний: Уверенно показывает умение использовать стандартные программные библиотеки для визуализации решений задач машинного обучения и анализа данных Высокий: На отличном уровне показывает владение стандартными программными библиотеками для решения задач машинного обучения и анализа данных	успеваемости по дисциплине Второй этап: Проведение промежуточной аттестации по учебной дисциплине.
	ПК 7.3 Владеть техническими, программными средствами для разработки алгоритмов и программ в области интеллектуального анализа данных	Низкий (пороговый): Демонстрирует поверхностные знания программных платформ и библиотеки для решения задач машинного обучения и анализа больших данных стандарты проведения анализа данных Средний: Уверенно показывает умение использовать программные платформы и библиотеки для решения задач машинного обучения и анализа больших данных Высокий: На отличном уровне показывает владение программными средствами для разработки алгоритмов машинного обучения, алгоритмами построения искусственных нейронных сетей	Первый этап: Проведение текущего контроля успеваемости по дисциплине Второй этап: Проведение промежуточной аттестации по учебной дисциплине.
ПК-9 Способность применять математические методы моделирования процессов обработки информации с использованием средств интеллектуального анализа данных	ПК 9.1 Знать принципы решения задач машинного обучения и интеллектуального анализа данных	Низкий (пороговый): Демонстрирует поверхностные знания основных принципов решения задач машинного обучения и анализа данных Средний: Умеет проводить спецификацию задачи, реализовывать программы на алгоритмических языках высокого уровня, интерпретировать полученные результаты Высокий: На отличном уровне показывает владение методами и алгоритмами машинного обучения	Первый этап: Проведение текущего контроля успеваемости по дисциплине Второй этап: Проведение промежуточной аттестации по учебной дисциплине.
	ПК 9.2 Уметь создавать алгоритмические и математические модели прикладных задач интеллектуального анализа данных	Низкий (пороговый): Демонстрирует поверхностные знания математических и алгоритмических моделей интеллектуального анализа данных, методы оценки временных и стоимостных характеристик машинного обучения и анализа данных Средний: Умеет создавать алгоритмические модели типовых прикладных задач обработки информации, реализовывать программы на алгоритмических языках высокого уровня Высокий: На отличном уровне владеет навыками использования прикладных библиотек для решения задач машинного обучения	

	<p>ПК 9.3 Владеть навыками построения описательных и прогнозных аналитических моделей с использованием современных инструментов интеллектуального анализа данных</p>	<p>Низкий (пороговый): Демонстрирует поверхностные знания в области описательных и прогнозных аналитических моделей для интеллектуального анализа данных Средний: Умеет разрабатывать и оценивать модели больших данных, использовать современные программные инструменты интеллектуального анализа данных Высокий: На отличном уровне владеет навыками построения описательных и прогнозных аналитических моделей с использованием современных инструментов интеллектуального анализа данных, навыками выявления требований к результатам анализа больших данных в предметной области</p>	<p>Первый этап: Проведение текущего контроля успеваемости по дисциплине Второй этап: Проведение промежуточной аттестации по учебной дисциплине.</p>
--	--	---	--

В ходе текущего контроля успеваемости при ответах на семинарских и практических занятиях, промежуточной аттестации в форме экзамена (зачет с оценкой) обучающиеся оцениваются по четырёхбалльной шкале оценивания: «отлично», «хорошо», «удовлетворительно», «неудовлетворительно»:

шкала оценки	описание
оценка "отлично"	выставляется обучающимся, показавшим всестороннее, систематическое и глубокое знание учебно-программного материала, умение свободно выполнять задания, предусмотренные программой, усвоивших основную и дополнительную литературу, рекомендованную программой. Оценка "отлично" выставляется студентам, усвоившим взаимосвязь основных понятий дисциплины в их значении для приобретаемой профессии, проявившим творческие способности в понимании, изложении и использовании учебно-программного материала.
оценка "хорошо"	выставляется обучающимся, показавшим полное знание учебно-программного материала, успешно выполняющим предусмотренные в программе задания, усвоившим основную литературу, рекомендованную в программе. Оценка "хорошо" выставляется студентам, продемонстрировавшим систематический характер знаний по дисциплине и способным к их самостоятельному пополнению и обновлению в ходе дальнейшей учебной работы и профессиональной деятельности.
оценка "удовлетворительно"	выставляется обучающимся, показавшим знания основного учебно-программного материала в объеме, необходимом для дальнейшей учебы и предстоящей работы по специальности, справившимся с выполнением заданий, предусмотренных программой, ориентирующимся в основной литературе, рекомендованной программой. Оценка "удовлетворительно" выставляется студентам, допустившим погрешности в ответе на экзамене и при выполнении экзаменационных заданий, но обладающим необходимыми знаниями для их устранения под руководством преподавателя.
оценка "неудовлетворительно"	выставляется обучающимся, имеющим пробелы в знаниях основного учебно-программного материала, допустившим принципиальные ошибки в выполнении предусмотренных

	программой заданий. Оценка "неудовлетворительно" ставится студентам, которые не могут продолжить обучение или приступить к профессиональной деятельности по окончании вуза без дополнительных занятий по соответствующей дисциплине.
--	--

В ходе промежуточной аттестации в форме зачёта обучающиеся оцениваются «зачтено» или «не зачтено»:

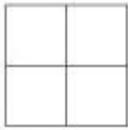
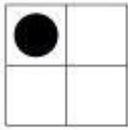
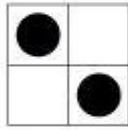
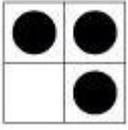
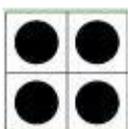
шкала оценки	описание
оценка "зачтено"	выставляется обучающимся, показавшим знания основного учебно-программного материала, справившимся с выполнением заданий, предусмотренных программой, ориентирующимся в основной и дополнительной литературе, рекомендованной программой.
оценка "не зачтено"	выставляется обучающимся, имеющим пробелы в знаниях основного учебно-программного материала, допустившим принципиальные ошибки в выполнении предусмотренных программой заданий

3. Типовые контрольные задания или иные материалы, необходимые для оценки результатов обучения по учебной дисциплине.

3.1. Примерные варианты оценочных заданий (ТОЗ) для контрольного рубежа в рамках текущего контроля

Задание	Оцениваемая компетенция
При использовании какого алгоритма может наблюдаться распад изображения на блоки 8x8 пикселей? +: при использовании рекурсивного алгоритма -: при использовании JPEG -: при использовании фрактального алгоритма	ПК-7
В каких из ниже приведенных форматах используется алгоритм RLE? +: TIFF +: BMP +: PCX -: нет правильного ответа	ПК-7
Укажите верный порядок действий при сжатии изображения в алгоритме JPEG-2000? +: сдвиг по яркости, переход в цветовое пространство YUV, дискретное вэйвлет-преобразование -: сдвиг по яркости, дискретное вэйвлет-преобразование, переход в цветовое пространство YUV -: переход в цветовое пространство YUV, дискретное вэйвлет- преобразование, сдвиг по яркости	ПК-7
Чем алгоритм со списком активных ребер отличается от алгоритма со списком реберных точек? -: он использует динамические структуры данных +: он не хранит в памяти точки пересечения контура с каждой строкой раstra -: при переходе к новой строке требуется полностью переформировывать CAP в отличие от списка реберных точек	ПК-7

-: сохраняет упорядоченность CAP по возрастанию x в отличие от списка реберных точек, который не сохраняет упорядоченность по возрастанию x	
Какой основной алгоритм сжатия используется в формате JPEG- 2000? -: алгоритм RLE -: фрактальный алгоритм +: рекурсивный алгоритм	ПК-7
Обеспечение возможности показать изображение низкого разрешения, используя только начало файла - это актуальное требование для алгоритмов, применяемых: +: для сжатия изображений в сетевых приложениях -: к фотореалистичным изображениям -: к деловой графике	ПК-9
Выберите верное утверждение: -: алгоритм Хаффмана не увеличивает размера исходных данных в худшем случае -: алгоритм Хаффмана требует помещения в файл со сжатыми данными таблицы соответствия кодируемых символов и кодирующих цепочек -: алгоритм Хаффмана требует двух проходов по массиву сжимаемых данных +: все ответы верны	ПК-9
Какой алгоритм сжатия реализован в форматах GIF и TIFF? -: RLE +: LZW -: LZ	ПК-9
На какой класс изображений ориентирован алгоритм Хаффмана? +: двуцветные черно-белые изображения, в которых преобладают большие пространства, заполненные белым цветом -: одноцветные изображения размером кратным примерно 7 Мб -: изображения с небольшим количеством цветов: деловую и научную графику	ПК-9
Как можно разрешить проблему фильтрации пикселей, находящихся в окрестности краев изображения? -: нет такой проблемы +: не проводить фильтрацию для таких пикселей +: доопределить значения пикселей за границами изображения, при помощи зеркального отражения	ПК-9
Что из приведенного ниже является цветовой моделью? -: RGB -: CMYK -: L*a*b* -: CIE XYZ +: все ответы верны	ПК-9
Какому оттенку 8-битного полутонового изображения соответствует нулевой оттенок палитры 4-битного изображения, составленной алгоритмом равномерного разбиения цветового пространства? (индексация палитры начинается с нуля) +: 0 -: 16 -: 32 -: 64 -: 128	ПК-9

<p>-: 192</p> <p>-: 240</p>	
<p>Какая последовательность закодирована алгоритмом кодирования длины повторения RLE на битовом уровне кодом: 111 000 111 101 001 110 010 111 000 111 101 при кодировании числа повторений тремя битами?</p> <p>-: 14 единиц, 6 нулей, 7 единиц, 6 нулей, 14 единиц, 14 нулей 5 единиц</p> <p>+: 14 единиц, 5 нулей, 1 единица, 6 нулей, 2 единицы, 14 нулей 5 единиц</p> <p>-: 3 единицы, 10 нулей, 27 единиц, 5 нулей</p> <p>-: 3 единицы, 11 нулей, 28 единиц, 3 нуля</p>	ПК-7
<p>Какие пиксели будут закрашены алгоритмом аппроксимации полутонов с использованием усечения по порогу если порог равен 150, а атрибуты пикселей 0,0,0,0?</p> <p>-: </p> <p>-: </p> <p>-: </p> <p>-: </p> <p>+: </p>	ПК-7

3.2. Вопросы для подготовки к промежуточной аттестации (к зачету)

1. Цифровое полутоновое изображение
2. Дискретизация изображения
3. Способы представления цвета
4. Трихроматическая теория
5. Модель RGB
6. Модель CMYK
7. Модель RGB
8. Модель Lab
9. Модель HSB

10. Модель CIE XYZ
11. Модель CIELAB
12. Статистические свойства распределения градиентов в изображении
13. Пространственная фильтрация
14. Фильтр Гаусса
15. Медианный фильтр
16. Билатеральный фильтр
17. Сегментация объектов
18. Текстура изображения
19. Свёртка с фильтром Гаусса
20. Шумоподавление и свёртка
21. Сглаживание с box-фильтром
22. Выделение краёв
23. Текстура изображения
24. Анализ текстуры
25. Двумерные модели преобразования изображений: перенос
26. Двумерные модели преобразования изображений: подобие
27. Двумерные модели преобразования изображений: аффинное
28. Двумерные модели преобразования изображений:
проективное преобразование
29. Перспективное преобразование плоскости (гомография)
30. Алгоритм RANSAC
31. 2D преобразования изображений
32. Алгоритм детектора Харриса
33. Метод SIFT
34. Классификация изображений
35. Распознавание по лицу - верификация
36. Бинарная классификация
37. Многоклассовая классификация
38. Распознавание человека по лицу
39. Квантование признаков
40. Квантование многомерных признаков
41. Квантование через кластеризацию
42. Гистограммы градиентов
43. Схема метода «мешок слов»
44. Расчёт LBP
45. Дескрипторы изображений
46. Ранжирование результатов
47. Алгоритм GISTIS
48. Линейная модель МакКаллока-Питтса
49. Градиентный метод обучения
50. Стохастический градиентный спуск
51. Представимость функций нейросетями
52. Линейный перспетрон
53. Функция перспетрона
54. Расчёт градиента
55. Процедура обратного распространения ошибки
56. Свёрточные сети

57. Визуализация работы нейросети
58. Эволюция признаков
59. Выход слоя как вектор-признак
60. UMAP визуализация
61. Классификация близких объектов
62. Дообучение (fine-tuning)
63. Spatial Pyramid Pooling (SPP)
64. Факторизация свёрток (MobileNet)
65. Нейросетевые признаки для поиска похожих
66. Выделение объектов
67. Алгоритмы выделения объектов на изображениях
68. Оценка качества детектора
69. Кривая точности-полноты
70. Множественные отклики
71. Сопоставление шаблонов
72. Обучение детектора
73. Нейросетевые методы в детекторах
74. Обучение R-CNN
75. Ключевые методы – SSD, Faster R-CNN
76. Извлечение объекта
77. Задача сегментации
78. Сегментация без учителя
79. Семантическая сегментация
80. Алгоритм Interactive GraphCuts
81. Графическая модель
82. Факторы в модели изображения
83. Парные потенциалы
84. Унарные потенциалы
85. Бинарная сегментация
86. Итеративные разрезы графов
87. Визуальная сегментация
88. Пересегментация через кластеризацию
89. Визуализация вектор-признака
90. Визуализация изображения по выходам
91. Реконструкция
92. Сравнение нейросетевых признаков
93. Детали реконструкции изображения
94. Генерация текстур
95. Видеопоследовательность
96. Optical flow (оптический поток)
97. Распознавание по движению
98. Отслеживание объектов в видео
99. Стандартная схема итеративного трекинга
100. Распознавание событий
101. Перспективная проекция
102. Уравнение перспективной проекции
103. Радиальная дисторсия
104. Геометрические свойства 2х изображений

105. Эпиполярная геометрия
106. Эпиполярное ограничение
107. Структура из движения
108. Поиск похожих изображений: Метод «Мешок слов»
109. Локальные методы бинокулярного стерео
110. Обучение с помощью CNN
111. Алгоритм Scanline Optimization
112. Алгоритм на основе MST
113. Алгоритм Semi-Global Matching
114. Алгоритм Simple Tree
115. Базовый алгоритм с сегментацией
116. Алгоритм на основе фрагментов
117. Визуальный тест Тьюринга
118. Учёт контуров и дополнение реконструкции
119. Исторические подходы к реконструкции
120. Параметрическая модель и детализация реконструкции
121. Точки и линии схода, горизонт
122. Вычисление линии горизонта
123. Моделирование глубины
124. Моделирование лица
125. Приближенная геометрическая модель как контекст

4. Методические материалы, определяющие процедуры оценивания результатов обучения по учебной дисциплине.

Процедура оценивания результатов обучения по учебной дисциплине осуществляется на основе Положением о текущем контроле и промежуточной аттестации обучающихся, утвержденными приказом ректора.

4.1 Первый этап: Проведение текущего контроля успеваемости по дисциплине

Проведение текущего контроля успеваемости по дисциплине осуществляется в ходе контактной работы с преподавателем в рамках аудиторных занятий и в ходе самостоятельной работы студента.

Текущий контроль в ходе контактной работы осуществляется по следующим видам:

1) Вид контроля: проверка сформированности компетенций в ходе самостоятельной работы обучающихся; текущий опрос, проводимый во время аудиторных (семинарских/практических/лабораторных) занятий; оценивание подготовленных докладов, сообщений, презентаций, домашних заданий.

Порядок проведения: в ходе подготовки к занятиям оценивается выполнение задания, рекомендованного к самостоятельной работе обучающихся, путем выборочной проверки.

Фиксируются результаты работы студентов в ходе проведения семинарских и практических занятий (активность, полнота ответов, способность поддерживать дискуссию, профессиональный язык и др.).

В ходе отдельных занятий обеспечивается проведение письменных опросов по тематике прошедших занятий. В ходе выполнения заданий обучающийся должен в меру имеющихся знаний, умений, навыков, сформированности компетенции дать развернутые ответы на поставленные в задании открытые вопросы и ответить на вопросы закрытого типа в установленное преподавателем время.

Продолжительность проведения процедуры определяется преподавателем самостоятельно, исходя из сложности индивидуальных заданий, количества вопросов, объема оцениваемого учебного материала.

Задания по подготовке докладов, сообщений, презентаций, домашних заданий выдаются заранее при подготовке к семинарским и практическим занятиям; подготовленные работы оцениваются с фиксацией в журнале учета посещаемости и успеваемости обучающихся.

2) Вид контроля: Контроль с использованием тестовых оценочных заданий по итогам освоения тем дисциплины (текущий (рубежный) контроль).

Порядок проведения: До начала проведения процедуры преподавателем подготавливаются необходимые оценочные материалы для оценки знаний, умений, навыков.

Оценка знаний, умений и навыков, характеризующих сформированность компетенций, осуществляется с помощью тестовых оценочных заданий (ТОЗ), сформированные в соответствии с Требованиями по подготовке тестовых оценочных заданий

Внеаудиторная контактная работа преподавателя с обучающимся осуществляется в ходе выполнения рейтинговой работы и контроля со стороны преподавателя за самостоятельной работой студента. Текущий контроль в ходе самостоятельной работы осуществляется в следующем виде:

3) Вид контроля: Подготовка курсовой работы (при наличии в учебном плане).

Технология проведения: За каждым обучающимся, принимающим участие в процедуре преподавателем закрепляется тема курсовой работы. После получения задания и в процессе его подготовки обучающийся должен в меру имеющихся знаний, умений, навыков, сформированности компетенции дать развернутое раскрытие темы, выполнить расчетное или иное задание.

4.2 Второй этап: Проведение промежуточной аттестации по учебной дисциплине.

В соответствии с учебным планом по учебной дисциплине предусмотрена подготовка и сдача зачета.

Порядок проведения промежуточной аттестации регламентируется Положением о текущем контроле и промежуточной аттестации, утвержденным приказом ректора Университета.