

Министерство науки и высшего образования Российской Федерации
Федеральное государственное бюджетное образовательное учреждение высшего
образования
«ИРКУТСКИЙ НАЦИОНАЛЬНЫЙ ИССЛЕДОВАТЕЛЬСКИЙ ТЕХНИЧЕСКИЙ
УНИВЕРСИТЕТ»

Структурное подразделение «Брикс кафедры (205)»

УТВЕРЖДЕНА:
на заседании кафедры
Протокол №6 от 25 февраля 2026 г.

Рабочая программа дисциплины

**«ГЛУБОКОЕ ОБУЧЕНИЕ В ТЕХНОЛОГИЯХ КОМПЬЮТЕРНОГО ЗРЕНИЯ / DEEP
LEARNING IN COMPUTER VISION TECHNOLOGY»**

Направление: 09.03.01 Информатика и вычислительная техника

Искусственный интеллект и компьютерные науки /Artificial Intelligence and Computer
Science

Квалификация: Бакалавр

Форма обучения: очная

Документ подписан простой
электронной подписью
Составитель программы:
Эррахел Назих
Дата подписания: 10.06.2026

Документ подписан простой
электронной подписью
Утвердил: Киреев Анна
Павловна
Дата подписания: 12.06.2026

Документ подписан простой
электронной подписью
Согласовал: Афанасьев
Александр Диомидович
Дата подписания: 10.06.2026

Год набора – 2026

Иркутск, 2026 г.

1 Перечень планируемых результатов обучения по дисциплине, соотнесённых с планируемыми результатами освоения образовательной программы

1.1 Дисциплина «Глубокое обучение в технологиях компьютерного зрения / Deep Learning in Computer Vision Technology» обеспечивает формирование следующих компетенций с учётом индикаторов их достижения

Код, наименование компетенции	Код индикатора компетенции
ПКС-3 Способность разрабатывать программные продукты, используя современные технологии в области искусственного интеллекта	ПКС-3.6
ПКС-5 Способен реализовывать проекты в сфере искусственного интеллекта	ПКС-5.5
ПКС-6 Способен выполнять процедуры сборки программных модулей и компонент в программный продукт	ПКС-6.1

1.2 В результате освоения дисциплины у обучающихся должны быть сформированы

Код индикатора	Содержание индикатора	Результат обучения
ПКС-3.6	Способен проводить практические исследования глубокого обучения для решения задач компьютерного зрения	Знать Основные типы задач компьютерного зрения: классификация, семантическая сегментация, сегментация экземпляра, обнаружение объектов. Архитектуры современных сверточных нейронных сетей (MobileNet, ResNet, EfficientNet, VGG, Inception, Xception). Методы предварительного обучения нейронных сетей: выделение признаков (feature extraction) и дообучение (fine-tuning). Алгоритмы детектирования объектов (R-CNN, Fast R-CNN, Faster R-CNN, YOLO, RetinaNet, Mask R-CNN) и их принципы работы. Методы визуализации работы сверточных нейронных сетей (промежуточные активации, тепловые карты активации). Основы работы с библиотекой OpenCV для обработки изображений и распознавания объектов. Принципы работы алгоритмов DeepDream и нейронной передачи стиля. Методы кластеризации

		изображений (К-средних, иерархическая кластеризация) и генерации изображений (вариационные автокодировщики). Основные метрики и наборы данных для оценки моделей компьютерного зрения (например, ImageNet, Kaggle datasets). Уметь Применять сверточные нейронные сети для решения задач классификации изображений. Выполнять предварительную обработку данных и аугментацию с использованием инструментов, таких как ImageDataGenerator в Keras. Использовать предварительно обученные модели для выделения признаков и дообучения на небольших наборах данных. Реализовать алгоритмы визуализации (например, тепловые карты активации, промежуточные активации) для анализа работы нейронных сетей. Применять алгоритмы DeepDream и нейронной передачи стиля для генерации и трансформации изображений. Реализовать детектирование и сегментацию объектов с использованием TensorFlow Object Detection API, YOLO, и RetinaNet. Использовать библиотеку OpenCV для выполнения задач обработки изображений, таких как поиск фигур, распознавание лиц и отслеживание объектов. Проводить кластеризацию изображений и реализовывать поиск изображений по содержанию (content-based image retrieval). Выполнять практические исследования, включая постановку задачи, выбор подходящей модели, обучение и анализ результатов. Владеть Навыками программирования на Python для реализации алгоритмов глубокого обучения с использованием
--	--	---

		библиотек Keras, TensorFlow, PyTorch и OpenCV. Инструментами подготовки и аугментации данных для задач компьютерного зрения. Методиками анализа и интерпретации результатов работы нейронных сетей (например, визуализация активаций, оценка метрик). Навыками работы с архитектурами нейронных сетей (R-CNN, YOLO, Mask R-CNN) для задач детектирования и сегментации. Техниками проектирования и реализации курсовых проектов в области глубокого обучения и компьютерного зрения. Навыками использования онлайн-платформ (например, Kaggle) для тестирования и проверки моделей. Методами оценки качества моделей с использованием метрик (например, precision, recall, IoU) и анализа ошибок.
ПКС-5.5	Понимает технологии глубокого обучения, их применимость и ограничения в области обработки изображений	Знать Основные принципы и архитектуры технологий глубокого обучения, применяемых в обработке изображений (сверточные нейронные сети, такие как VGG, ResNet, Inception, MobileNet, EfficientNet). Области применения технологий глубокого обучения в компьютерном зрении: классификация изображений, обнаружение объектов, семантическая и экземплярная сегментация, нейронная передача стиля, генерация изображений. Ограничения технологий глубокого обучения: необходимость больших объемов данных, вычислительная сложность, переобучение, ограниченная интерпретируемость моделей. Метрики оценки качества моделей в задачах обработки изображений (например, accuracy, precision, recall, IoU, F1-score). Типы наборов данных,

		используемых в компьютерном зрении (например, ImageNet, COCO), и их особенности. Особенности предварительно обученных моделей и их ограничения при применении к специфическим наборам данных. Принципы работы алгоритмов DeepDream и нейронной передачи стиля, их возможности и ограничения. Уметь Определять подходящие технологии глубокого обучения для конкретных задач обработки изображений (например, выбор между YOLO и Mask R-CNN для детектирования объектов). Анализировать применимость технологий глубокого обучения с учетом ограничений, таких как объем данных, вычислительные ресурсы и требования к точности. Оценивать ограничения моделей на основе результатов их работы (например, анализ переобучения или низкой обобщающей способности). Использовать метрики для оценки производительности моделей в задачах обработки изображений. Применять предварительно обученные модели для задач обработки изображений с учетом их ограничений (например, необходимость дообучения для специфических данных). Интерпретировать результаты визуализации (например, тепловые карты активации) для понимания работы моделей и их ограничений. Владеть Навыками анализа и сравнения различных архитектур глубокого обучения (например, ResNet vs. EfficientNet) с точки зрения их применимости и ограничений. Техниками выбора подходящих моделей и их адаптации для задач обработки изображений. Навыками интерпретации метрик и результатов работы моделей для
--	--	---

		выявления их сильных и слабых сторон. Методами визуализации (например, промежуточные активации, тепловые карты) для оценки ограничений моделей. Навыками работы с инструментами (Keras, TensorFlow) для реализации и тестирования моделей обработки изображений. Способностью документировать анализ применимости и ограничений технологий глубокого обучения в рамках курсового проекта.
ПКС-6.1	Использует выбранную среду программирования для разработки процедур интеграции программных модулей	Знать Основы программирования на Python и его применения в задачах компьютерного зрения. Основные библиотеки и фреймворки для глубокого обучения: Keras, TensorFlow, PyTorch, OpenCV. Принципы интеграции программных модулей в средах программирования (например, использование API для работы с моделями глубокого обучения). Структуру и функциональность TensorFlow Object Detection API для интеграции моделей детектирования объектов. Принципы работы с ImageDataGenerator в Keras для подготовки данных и интеграции в процесс обучения. Основы автоматизации задач с использованием библиотек, таких как ruwinauto, для интеграции программных модулей. Форматы данных и их обработку для интеграции в пайплайны глубокого обучения (например, загрузка изображений в OpenCV). Принципы модульного программирования и организации кода для обеспечения совместимости модулей. Уметь Разрабатывать процедуры интеграции программных модулей в Python с использованием библиотек Keras, TensorFlow, PyTorch и

		OpenCV. Настраивать пайплайны обработки данных для интеграции с моделями глубокого обучения (например, использование ImageDataGenerator для аугментации данных). Реализовать интеграцию моделей детектирования объектов (например, YOLO, RetinaNet) с использованием TensorFlow Object Detection API. Использовать OpenCV для интеграции модулей обработки изображений (например, загрузка, предобработка и анализ изображений). Настраивать взаимодействие между различными программными модулями (например, связка визуализации активаций с моделью классификации). Автоматизировать процессы интеграции с использованием библиотек, таких как ruwinauto, для управления GUI или тестирования. Выполнять отладку и тестирование интегрированных программных модулей для обеспечения их корректной работы. Владеть Навыками программирования на Python для разработки и интеграции модулей в задачах компьютерного зрения. Техниками настройки и использования библиотек Keras, TensorFlow, PyTorch и OpenCV для создания интегрированных решений. Навыками работы с TensorFlow Object Detection API для интеграции моделей детектирования в программные системы. Методами организации кода для обеспечения модульности и повторного использования. Навыками тестирования и отладки интегрированных программных модулей с использованием сред программирования. Способностью документировать процесс интеграции модулей в
--	--	--

		рамках курсового проекта. Техниками интеграции данных и моделей в единый пайплайн обработки изображений (например, от предобработки до вывода результатов).
--	--	---

2 Место дисциплины в структуре ООП

Изучение дисциплины «Глубокое обучение в технологиях компьютерного зрения / Deep Learning in Computer Vision Technology» базируется на результатах освоения следующих дисциплин/практик: «Алгоритмы ML / Machine Learning Algorithms», «Математика / Mathematics», «Программирование / Programming», «Теоретические основы искусственного интеллекта / Theoretical Foundations of Artificial Intelligence»

Дисциплина является предшествующей для дисциплин/практик: «Глубокое обучение в технологиях обработки естественного языка (NLP) / Deep Learning in Natural Language Processing Technologies (NLP)», «Технологии внедрения решений искусственного интеллекта / Artificial Intelligence Implementation Technologies»

3 Объем дисциплины

Объем дисциплины составляет – 5 ЗЕТ

Вид учебной работы	Трудоемкость в академических часах (Один академический час соответствует 45 минутам астрономического часа)	
	Всего	Семестр № 5
Общая трудоемкость дисциплины	180	180
Аудиторные занятия, в том числе:	75	75
лекции	30	30
лабораторные работы	0	0
практические/семинарские занятия	45	45
Самостоятельная работа (в т.ч. курсовое проектирование)	69	69
Трудоемкость промежуточной аттестации	36	36
Вид промежуточной аттестации (итогового контроля по дисциплине)	Экзамен, Курсовая работа	Экзамен, Курсовая работа

4 Структура и содержание дисциплины

4.1 Сводные данные по содержанию дисциплины

Семестр № 5

№ п/п	Наименование раздела и темы дисциплины	Виды контактной работы						СРС		Форма текущего контроля
		Лекции		ЛР		ПЗ(СЕМ)				
		№	Кол. Час.	№	Кол. Час.	№	Кол. Час.	№	Кол. Час.	
1	2	3	4	5	6	7	8	9	10	11
1	Обзор задач и архитектур компьютерного	1	4							Тест

	зрения									
2	Обучение сверточной нейронной сети	2	4			1	2	3	8	Решение задач
3	Использование предварительно обученных моделей	3	4			2	6	3	8	Решение задач
4	Визуализация работы нейронных сетей	4	4			3	4	3	6	Решение задач
5	Алгоритмы DeepDream и нейронная передача стиля	5	4			4	6	3	7	Решение задач
6	Детектирование и сегментация объектов	6	6			5	8			Решение задач
7	Основы работы с библиотекой OpenCV	7	4			6	4			Решение задач
8	Распознавание лиц с использованием OpenCV					7	4			Решение задач
9	Распознавание лиц с использованием сверточных нейронных сетей					8	4			Решение задач
10	Отслеживание объектов в видеопотоке					9	3			Решение задач
11	Кластеризация изображений					10	2			Решение задач
12	Генерация изображений с использованием вариационных автокодировщиков					11	2			Решение задач
13	Поиск изображений по содержанию									Решение задач
14	Курсовой проект							1, 2	40	Отчет по лабораторной работе
	Промежуточная аттестация								36	Экзамен, Курсовая работа
	Всего		30				45		105	

4.2 Краткое содержание разделов и тем занятий

Семестр № 5

№	Тема	Краткое содержание
1	Обзор задач и архитектур компьютерного зрения	Обзор сфер применения компьютерного зрения: классификация, семантическая сегментация, сегментация экземпляра, обнаружение объектов.

		Изучение метрик, известных наборов данных (ImageNet, COCO). Обзор современных сверточных сетей: MobileNet, ResNet, EfficientNet.
2	Обучение сверточной нейронной сети	Изучение операции свертки, параметров свертки, карт признаков, pooling-слоев. Применение сверточной модели для классификации изображений. Использование ImageDataGenerator в Keras для аугментации данных. Практика: классификация изображений на наборе данных "Dogs vs. Cats" (Kaggle).
3	Использование предварительно обученных моделей	Изучение подходов к использованию предварительно обученных сетей: выделение признаков и дообучение. Обзор архитектур: VGG, ResNet, Inception, Xception. Практика: применение моделей VGG-16, ResNet для классификации изображений.
4	Визуализация работы нейронных сетей	Изучение методов визуализации: промежуточные активации, фильтры, тепловые карты активации класса. Практика: визуализация активаций и тепловых карт с использованием Keras.
5	Алгоритмы DeepDream и нейронная передача стиля	Реализация алгоритма DeepDream в Keras. Изучение нейронной передачи стиля для переноса стиля изображения-образца на целевое изображение. Практика: реализация DeepDream с моделью Inception V3 и изучение блокнота "Style Transfer" на Kaggle.
6	Детектирование и сегментация объектов	Изучение двухэтапных и одноэтапных методов детектирования объектов (Selective Search, anchor boxes). Обзор архитектур: R-CNN, Fast R-CNN, Faster R-CNN, YOLO, RetinaNet, Mask R-CNN. Методы сегментации: кластеризация, выделение краев. Практика: детектирование объектов с использованием TensorFlow Object Detection API, YOLO, RetinaNet.
7	Основы работы с библиотекой OpenCV	Основы OpenCV: загрузка изображений, работа с пикселями, преобразование в градации серого, устранение шума, обнаружение краев. Практика: создание мини-проекта по поиску фигур в изображениях с использованием Python и OpenCV.
8	Распознавание лиц с использованием OpenCV	Изучение методов распознавания лиц и ключевых точек с использованием OpenCV. Практика: реализация распознавания лиц средствами OpenCV.
9	Распознавание лиц с использованием сверточных нейронных сетей	Применение сверточных нейронных сетей для распознавания лиц. Практика: реализация распознавания лиц с использованием CNN и методов, таких как Triplet Loss или Contrastive Loss.
10	Отслеживание объектов в видеопотоке	Изучение методов отслеживания объектов в видеопотоке с использованием компьютерного зрения. Практика: реализация отслеживания

		объектов в видео с использованием Python и OpenCV.
11	Кластеризация изображений	Изучение методов кластеризации (К-средних, иерархическая кластеризация) для анализа изображений. Практика: применение метода К-средних к изображениям букв в разных шрифтах и кластеризация по цвету.
12	Генерация изображений с использованием вариационных автокодировщиков	Изучение принципов работы вариационных автокодировщиков для генерации изображений. Практика: реализация генерации изображений с использованием Python и Keras.
13	Поиск изображений по содержанию	Изучение методов поиска изображений по содержанию (content-based image retrieval). Практика: реализация поиска изображений на основе их содержимого.
14	Курсовой проект	Разработка и реализация проекта, включающего применение технологий глубокого обучения для решения задачи компьютерного зрения. Оформление отчета с анализом результатов и документацией.

4.3 Перечень лабораторных работ

Лабораторных работ не предусмотрено

4.4 Перечень практических занятий

Семестр № 5

№	Темы практических (семинарских) занятий	Кол-во академических часов
1	Классификация изображений с использованием CNN	2
2	Применение предварительно обученных моделей (VGG-16, ResNet)	6
3	Визуализация активаций и тепловых карт	4
4	Реализация DeepDream и нейронной передачи стиля	6
5	Детектирование объектов с TensorFlow Object Detection API, YOLO, RetinaNet	8
6	Обработка изображений с OpenCV: поиск фигур	4
7	Распознавание лиц с использованием OpenCV	4
8	Распознавание лиц с использованием CNN	4
9	Отслеживание объектов в видеопотоке	3
10	Кластеризация изображений методом К-средних	2
11	Генерация изображений с вариационными автокодировщиками	2

4.5 Самостоятельная работа

Семестр № 5

№	Вид СРС	Кол-во академических часов
1	Написание курсового проекта (работы)	14
2	Подготовка к экзамену	26
3	Решение специальных задач	29

В ходе проведения занятий по дисциплине используются следующие интерактивные методы обучения: 1. Проблемная лекция: Создание проблемных ситуаций в процессе изложения учебного материала, вовлечение студентов в анализ противоречий, стимулирование самостоятельных выводов о необходимости новых знаний. 2. Задания-провокации: Выполнение заданий, направленных на поиск и исправление ошибок (содержательных, в программном коде и других аспектах), способствующих развитию критического мышления. 3. Лекция-визуализация: Использование схем, опорных конспектов, видеофрагментов и других визуальных материалов для наглядного представления учебного содержания. 4. Лекция-диалог, дискуссия: Освещение содержания лекции через диалог со студентами, организация дискуссий для углубленного обсуждения тем. 5. Кейс-технологии: Решение практических кейсов, моделирующих реальные задачи компьютерного зрения, для применения теоретических знаний в практических сценариях.

5 Перечень учебно-методического обеспечения дисциплины

5.1 Методические указания для обучающихся по освоению дисциплины

5.1.1 Методические указания для обучающихся по курсовому проектированию/работе:

а) Основная литература:

1) Воронцов К.В. Машинное обучение. Курс лекций. <http://www.machinelearning.ru>.

б) Дополнительная литература

2) Золотых Н.Ю. Машинное обучение. Курс лекций. Нижний Новгород: ННГУ, 2007. <http://www.uic.nnov.ru/~zny/ml>

г) Используемое программное обеспечение

Используется только открытое программное обеспечение, установленное на компьютерах обучающихся:

1) MS Windows установленная на компьютере обучающегося

2) MS Visual Studio Express 2015 или MS Visual Studio Express 2015 для Web (<https://www.microsoft.com/ru-ru/SoftMicrosoft/vs2015Web.aspx>) – бесплатная версия.

3) Установка языка Python [<http://www.python.org/>].

4) Библиотека автоматизации GUI тестирования pywinauto [<http://pywinauto.github.io/>]

5) ПО визуализации фильтров и выходов слоев в Caffe [<http://nbviewer.jupyter.org/github/BVLC/caffe/blob/master/examples/00-classification.ipynb>].

ПО визуализации фильтров и выходов слоев в Torch [<https://github.com/facebook/iTorch>].

5.1.2 Методические указания для обучающихся по практическим занятиям

1. Онлайн курс «Нейронные сети и компьютерное зрение» от экспертов Samsung AI Center. [Электронный ресурс]. – Режим доступа: <https://stepik.org/course/50352/info> . – (дата обращения 12.06.2025).

2. Тьюториалы (примеры кода на Python) на официальных сайтах изучаемых библиотек для глубокого обучения:

2.1. Keras <https://keras.io/examples/>

2.2. TensorFlow <https://www.tensorflow.org/tutorials>

2.3. PyTorch <https://pytorch.org/tutorials/>

2.4. Caffe <http://caffe.berkeleyvision.org/>

5.1.3 Методические указания для обучающихся по самостоятельной работе:

Обучающие курсы на платформе Kaggle <https://www.kaggle.com/learn/overview>

6 Фонд оценочных средств для контроля текущей успеваемости и проведения промежуточной аттестации по дисциплине

6.1 Оценочные средства для проведения текущего контроля

6.1.1 семестр 5 | Решение задач

Описание процедуры.

Описание процедуры: тестирование по материалам дисциплин «Программные средства для задач искусственного интеллекта» в рамках дисциплины «Технологии разработки программного обеспечения».

Задания базируются на авторском онлайн курсе «Введение в Python» <https://stepik.org/course/57179/syllabus>).

Пример задания по теме «Работа с файлами»:

Напишите программу, которая считывает из файла строку, соответствующую тексту, сжато с помощью кодирования повторов, и производит обратную операцию, получая исходный текст. Полученный текст разместите в окне ответа на это задание.

Тема (раздел): «Разработка моделей НС с использованием Keras: краткий обзор»

Описание процедуры: тестирование

Пример задания:

Количество нейронов на последнем слое зависит от:

А. Количества нейронов в нейронной сети

В. Можно подбирать произвольно

С. Количество категорий, которые нужно классифицировать

Д. Количество входных данных

Ответ: С.

Критерии оценивания.

Оценки «отлично» заслуживает обучающийся, обнаруживший всестороннее знание учебно-программного материала в сфере основ программирования и понятия об искусственном интеллекте. Понимает их роль для приобретаемой профессии.

Оценки «хорошо» заслуживает обучающийся, обнаруживший полное знание учебно-программного материала в сфере основ программирования и понятия об искусственном интеллекте. Осознает их роль для приобретаемой профессии.

Оценки «удовлетворительно» заслуживает обучающийся, обнаруживший знание учебно-программного материала в сфере основ программирования и понятия об искусственном интеллекте. Обладают необходимыми знаниями для устранения погрешностей в ответе под руководством преподавателя.

Оценка «неудовлетворительно» выставляется обучающемуся, обнаружившему пробелы в знаниях, требуемых для изучения основного учебно-программного материала, допустившему принципиальные ошибки в выполнении заданий.

6.1.2 семестр 5 | Отчет по лабораторной работе

Описание процедуры.

Требования к оформлению доклада:

- Объем доклада – 5 страниц (без титульного листа и списка источников).
- Титульный лист должен быть оформлен по образцу (имеется файл с образцом).
- Основной текст работы оформлен в соответствии с требованиями, указанными ниже.
- В случае использования в тексте таблиц и/или рисунков на каждый объект должна быть ссылка в тексте работы. Например, «... основные виды программных средств представлены ниже (см. Таблица 1)» или «... схему передачи информации можно увидеть на рис. 1».
- Количество источников должно быть не менее трех, на все должны быть ссылки внутри текста.
- Список используемых источников должен быть оформлен в соответствии с требованиями, указанными ниже.

Для оформления основного текста работы:

- Шрифт – TimesNewRoman, размер – 14 пт.
- Абзац: междустрочный интервал – 1,5; выравнивание – «по ширине»; абзацный отступ – 1,25 см.
- Оформление рисунков (при необходимости): выравнивание рисунка – «по центру», подпись рисунка – «Рис. №. Название рисунка»; шрифт для подписи рисунка – TimesNewRoman, размер – 12 пт.
- Оформление таблиц (при необходимости): выравнивание таблицы – «по центру»; шрифт внутри таблицы – TimesNewRoman, размер – 11-12 пт.; выравнивание текста внутри таблицы – на усмотрение пользователя; подпись таблицы располагается над таблицей и состоит из двух частей: «Таблица №» – выравнивание по правому краю и «Название таблицы» – выравнивание по правому краю или по центру.

Для оформления источников (в соответствии с ГОСТ 2008):

- Источники должны быть расположены в алфавитном порядке и пронумерованы.
- В тексте доклада ссылка на источник выполняется в виде: [№], где № – номер источника в общем списке.
- Если в тексте используется дословная цитата, то она должна быть взята в кавычки, а в ссылке на источник указана страница: [5, с.15].

Критерии оценивания.

Отлично (90–100%):

Полное выполнение задачи компьютерного зрения (например, классификация, детектирование объектов) с использованием подходящей модели (например, ResNet, YOLO).

Код (Python, Keras/TensorFlow/OpenCV) корректен, хорошо прокомментирован, воспроизводим, демонстрирует творческий подход.

Отчет (5 страниц) полностью соответствует требованиям: структура (введение, теоретическая часть, практическая часть, заключение), оформление (Times New Roman, 14 пт, интервал 1.5, отступ 1.25 см), таблицы/рисунки с подписями («Таблица №», «Рис. №», шрифт 11–12 пт) и ссылками в тексте (например, «см. Таблица 1»).

Минимум 3 источника оформлены по ГОСТ 7.1-2008, с корректными ссылками ([№] или [№, с. №] для цитат).

Результаты включают метрики (например, accuracy, IoU) и визуализации (например, тепловые карты), с глубоким анализом и обоснованными выводами.

Защита: студент уверенно объясняет выбор модели, данные, результаты и ограничения, отвечает на все вопросы.

Хорошо (75–89%):

Задача выполнена с незначительными недочетами (например, субоптимальная модель или гиперпараметры).

Код функционален, прокомментирован, но может содержать мелкие ошибки, не влияющие на результат.

Отчет соответствует требованиям по структуре и оформлению, но допускает незначительные отклонения (например, некорректное выравнивание таблиц или рисунков).

Источники (минимум 3) оформлены по ГОСТ 7.1-2008, но возможны мелкие ошибки в ссылках.

Результаты включают метрики и визуализации, анализ достаточен, но менее глубокий.

Защита: студент объясняет проект, отвечает на большинство вопросов, но может затрудняться с углубленными аспектами.

Удовлетворительно (60–74%):

Задача выполнена частично (например, модель работает, но результаты низкого качества).

Код содержит ошибки, требует доработки, комментарии минимальны.

Отчет следует структуре, но нарушает требования оформления (например, шрифт, интервал, подписи таблиц/рисунков) или содержит недостатки в содержании.

Источники (3 или менее) оформлены с ошибками, ссылки в тексте неполные.

Результаты включают базовые метрики, анализ поверхностный, визуализации ограничены.

Защита: студент объясняет проект с трудом, отвечает только на основные вопросы, допускает ошибки.

Неудовлетворительно (0–59%):

Задача не выполнена или модель неработоспособна.

Код отсутствует, неработоспособен или не прокомментирован.

Отчет не соответствует требованиям (объем менее 5 страниц, нарушена структура, оформление, отсутствуют таблицы/рисунки или ссылки на них).

Источники отсутствуют, менее 3, или оформлены некорректно, ссылки в тексте отсутствуют.

Результаты отсутствуют или не проанализированы.

Защита: студент не может объяснить проект, не отвечает на вопросы.

6.1.3 семестр 5 | Тест

Описание процедуры.

Подготовка к тесту:

Изучите лекционные материалы по теме «Обзор задач и архитектур компьютерного зрения», включая типы задач (классификация, семантическая сегментация, сегментация экземпляра, обнаружение объектов), метрики, наборы данных (ImageNet, COCO) и архитектуры сетей (MobileNet, ResNet, EfficientNet).

Пройдите рекомендованные разделы онлайн-курса «Нейронные сети и компьютерное зрение» (Samsung AI Center, Stepik) и изучите tutorиалы на keras.io и tensorflow.org.

Доступ к тесту:

Тест проводится на платформе электронного образования ИРНИТУ (el.istu.edu).

Авторизуйтесь в системе, используя учетные данные студента, и перейдите к разделу курса «Глубокое обучение в технологиях компьютерного зрения».

Выполнение теста:

Тест состоит из вопросов с множественным выбором, открытых вопросов и задач на

анализ (например, выбор подходящей метрики для задачи сегментации).

Время выполнения: 60 минут.

Вопросы охватывают:

Типы задач компьютерного зрения и их особенности.

Характеристики наборов данных (например, ImageNet).

Архитектуры сверточных нейронных сетей (например, ResNet, MobileNet) и их применение.

Метрики оценки (например, accuracy, IoU).

Подача результатов:

После завершения теста нажмите «Отправить» на платформе.

Результаты автоматически сохраняются и передаются преподавателю для проверки.

Обратная связь:

Результаты теста и комментарии преподавателя будут доступны на платформе ИРНИТУ в течение 7 рабочих дней.

При необходимости обратитесь к преподавателю для разъяснения ошибок.

Критерии оценивания.

Отлично (90–100%):

Студент демонстрирует глубокое понимание задач компьютерного зрения, наборов данных, метрик и архитектур сетей.

Правильно отвечает на 90–100% вопросов, включая сложные аналитические задачи (например, выбор архитектуры для задачи с ограниченными данными).

Ответы на открытые вопросы полные, с точными примерами и обоснованиями.

Хорошо (75–89%):

Студент демонстрирует хорошее знание темы, но допускает незначительные ошибки в 10–25% вопросов.

Корректно отвечает на большинство вопросов с множественным выбором и аналитических задач, но может испытывать затруднения с углубленными аспектами (например, сравнение ResNet и EfficientNet).

Ответы на открытые вопросы достаточные, но менее детализированные.

Удовлетворительно (60–74%):

Студент демонстрирует базовое понимание темы, но допускает ошибки в 25–40% вопросов.

Корректно отвечает на простые вопросы (например, определение задач компьютерного зрения), но затрудняется с аналитическими задачами или характеристиками архитектур.

Ответы на открытые вопросы минимальны, содержат неточности.

Неудовлетворительно (0–59%):

Студент демонстрирует значительные пробелы в знаниях, ошибается в более чем 40% вопросов.

Не может корректно ответить на базовые вопросы (например, типы задач или метрики).

Ответы на открытые вопросы отсутствуют, некорректны или неполные.

6.2 Оценочные средства для проведения промежуточной аттестации

6.2.1 Критерии и средства (методы) оценивания индикаторов достижения компетенции в рамках промежуточной аттестации

Индикатор достижения компетенции	Критерии оценивания	Средства (методы) оценивания промежуточной аттестации
---	----------------------------	--

ПКС-3.6	Отлично: Обучающийся демонстрирует глубокое и систематическое знание теоретических основ и практических методов глубокого обучения в компьютерном зрении. Способен самостоятельно проводить исследования, выбирать подходящие модели, интерпретировать результаты и обосновывать выводы. Успешно выполняет все задания, включая курсовой проект, с проявлением творческого подхода. Точно отвечает на вопросы, связанные с темой исследования. Хорошо: Обучающийся демонстрирует полное знание учебного материала, успешно выполняет практические задания и курсовой проект. Способен анализировать результаты и устранять ошибки самостоятельно. Допускает незначительные погрешности, которые не влияют на общий результат. Уверенно отвечает на большинство вопросов, связанных с темой исследования. Удовлетворительно: Обучающийся демонстрирует базовое знание учебного материала, выполняет задания с незначительными ошибками, которые могут быть устранены под руководством преподавателя. Способен проводить исследования с использованием готовых решений, но допускает погрешности в интерпретации результатов или обосновании выводов. Отвечает на основные вопросы, но может испытывать затруднения с углубленными аспектами. Неудовлетворительно: Обучающийся демонстрирует значительные пробелы в знаниях, допускает принципиальные ошибки в выполнении заданий или курсового проекта. Не способен самостоятельно проводить исследования или интерпретировать результаты. Ответы на вопросы некорректны или отсутствуют.	Опрос: Устные вопросы по теоретическим основам и практическим методам глубокого обучения в компьютерном зрении. Собеседование: Индивидуальная беседа с преподавателем для оценки понимания студентом темы исследования и его способности обосновывать выводы. Компьютерные тесты: Тестирование на платформе электронного образования ИРНИТУ, включающее вопросы по теоретическим аспектам и практическим задачам (например, выбор числа нейронов в последнем слое, анализ метрик). Курсовой проект: Оценка выполнения курсового проекта, включающего разработку модели глубокого обучения, анализ результатов и оформление отчета (5 страниц, с соблюдением
---------	--	---

		требований к структуре и оформлению). Практические задания: Выполнение лабораторных работ, таких как реализация алгоритмов классификации, детектирования объектов, визуализации активаций и кластеризации изображений.
ПКС-5.5	Отлично: Обучающийся демонстрирует глубокое понимание технологий глубокого обучения, их применимости и ограничений. Способен обосновать выбор модели для конкретной задачи, анализировать ограничения на основе метрик и визуализаций, а также предлагать пути их преодоления. Точно отвечает на вопросы, связанные с применимостью и ограничениями технологий. Хорошо: Обучающийся демонстрирует полное знание технологий глубокого обучения и их ограничений. Способен выбрать подходящую модель и проанализировать ее ограничения, но допускает незначительные неточности в обосновании или интерпретации. Уверенно отвечает на большинство вопросов. Удовлетворительно: Обучающийся демонстрирует базовое понимание технологий и их ограничений. Может выбрать модель и описать ее применимость, но допускает ошибки в анализе ограничений или интерпретации результатов. Отвечает на основные вопросы, но затрудняется с углубленным анализом. Неудовлетворительно: Обучающийся демонстрирует пробелы в понимании технологий глубокого обучения, не может обосновать выбор модели или	Опрос: Устные вопросы по технологиям глубокого обучения, их применимости и ограничениям в задачах обработки изображений. Компьютерные тесты: Тестирование на платформе электронного образования ИРНИТУ, включающее вопросы по выбору моделей, анализу метрик и ограничений (например, выбор архитектуры для задачи с ограниченными данными). Практические задания: Выполнение лабораторных работ, связанных с анализом применимости моделей

	проанализировать ее ограничения. Ответы на вопросы некорректны или отсутствуют.	(например, сравнение производительности ResNet и MobileNet на небольшом наборе данных). Курсовой проект: Оценка раздела проекта, посвященного анализу применимости выбранной технологии и ее ограничений, с использованием метрик и визуализаций. Собеседование: Индивидуальная беседа для оценки способности студента объяснить выбор технологии и ее ограничения в контексте конкретной задачи.
ПКС-6.1	Отлично: Обучающийся демонстрирует глубокое знание сред программирования и их использования для интеграции модулей. Способен самостоятельно разрабатывать, тестировать и документировать сложные интегрированные решения. Успешно реализует пайплайны обработки данных и моделей, не допуская ошибок. Точно отвечает на вопросы, связанные с интеграцией модулей. Хорошо: Обучающийся демонстрирует полное знание сред программирования и успешно выполняет интеграцию модулей. Может разрабатывать и тестировать пайплайны, но допускает незначительные ошибки, которые устраняются самостоятельно. Уверенно отвечает на большинство	Опрос: Устные вопросы по принципам интеграции программных модулей и использованию сред программирования. Компьютерные тесты: Тестирование на платформе ИРНИТУ, включающее вопросы по реализации модулей в Python и работе с библиотеками

	вопросов. Удовлетворительно: Обучающийся демонстрирует базовое знание сред программирования и способен выполнять интеграцию модулей с использованием готовых решений. Допускает ошибки в реализации или тестировании, которые устраняются под руководством преподавателя. Отвечает на основные вопросы, но затрудняется с углубленными аспектами. Неудовлетворительно: Обучающийся демонстрирует пробелы в знаниях, не способен реализовать или протестировать интегрированные модули. Допускает принципиальные ошибки, которые препятствуют корректной работе системы. Ответы на вопросы некорректны или отсутствуют.	(например, настройка ImageDataGenerator). Практические задания: Выполнение лабораторных работ, таких как интеграция модели YOLO с OpenCV или настройка пайплайна для классификации изображений. Курсовой проект: Оценка раздела проекта, посвященного разработке и интеграции программных модулей, включая код, документацию и результаты тестирования. Собеседование: Индивидуальная беседа для оценки способности студента объяснить процесс интеграции модулей и их взаимодействие в рамках задачи компьютерного зрения.
--	---	--

6.2.2 Типовые оценочные средства промежуточной аттестации

6.2.2.1 Семестр 5, Типовые оценочные средства для проведения экзамена по дисциплине

6.2.2.1.1 Описание процедуры

Подготовка к экзамену:

Изучите лекционные и практические материалы по всем темам курса (1–13), включая

задачи компьютерного зрения (классификация, сегментация, детектирование), архитектуры сетей (ResNet, YOLO, Mask R-CNN), визуализацию, DeepDream, нейронную передачу стиля, OpenCV, кластеризацию, и генерацию изображений.

Повторите практические задания (например, классификация с Keras, детектирование с TensorFlow Object Detection API) и материалы онлайн-курса «Нейронные сети и компьютерное зрение» (Samsung AI Center, Stepik).

Ознакомьтесь с туториалами на keras.io, tensorflow.org, pytorch.org и [Kaggle](https://kaggle.com).

Доступ к экзамену:

Экзамен проводится на платформе электронного образования ИРНИТУ (el.istu.edu).

Авторизуйтесь в системе с учетными данными студента и перейдите к разделу курса.

Выполнение экзамена:

Экзамен включает тест с вопросами с множественным выбором, открытыми вопросами, и практическими задачами (например, анализ кода или выбор модели).

Время выполнения: 90 минут.

Примерные вопросы к билету:

Какие задачи компьютерного зрения существуют, и в чем их различия? Приведите примеры метрик для каждой задачи.

Опишите архитектуру ResNet и ее преимущества по сравнению с VGG.

Как работает ImageDataGenerator в Keras, и для чего используется аугментация данных?

В чем разница между одноэтапными и двухэтапными методами детектирования объектов?

Сравните YOLO и Faster R-CNN.

Объясните процесс визуализации тепловых карт активации в сверточных нейронных сетях.

Как реализовать распознавание лиц с использованием OpenCV? Опишите ключевые шаги.

Что такое нейронная передача стиля, и как она реализована в Keras?

Как вариационные автокодировщики используются для генерации изображений?

Подача результатов:

После завершения теста нажмите «Отправить» на платформе.

Ответы на открытые вопросы и практические задачи проверяются преподавателем.

Обратная связь:

Результаты экзамена будут доступны на платформе ИРНИТУ в течение 10 рабочих дней.

Обратитесь к преподавателю для разъяснения ошибок, если необходимо.

Пример задания:

Задание:

Вы разрабатываете модель для классификации изображений на наборе данных «Dogs vs. Cats» (Kaggle).

Какую архитектуру сверточной нейронной сети вы выберете (например, VGG-16, ResNet, MobileNet) и почему?

Опишите процесс аугментации данных с использованием ImageDataGenerator в Keras.

Укажите не менее трех методов аугментации.

Какие метрики вы будете использовать для оценки модели, и почему?

Напишите фрагмент кода на Python для настройки ImageDataGenerator с указанными методами аугментации.

Пример ответа (для ориентира):

Выберу ResNet-50, так как она эффективна для глубоких сетей благодаря остаточным связям, минимизирующим проблему затухания градиента [1].

Аугментация включает: горизонтальное отражение (`horizontal_flip`), вращение (`rotation_range=20`), масштабирование (`zoom_range=0.2`). Это увеличивает разнообразие данных и предотвращает переобучение.

Метрики: `accuracy` (общая точность), `precision` и `recall` (для оценки ошибок

классификации).

Код:

```
from tensorflow.keras.preprocessing.image import ImageDataGenerator  
datagen = ImageDataGenerator(horizontal_flip=True, rotation_range=20, zoom_range=0.2)
```

6.2.2.1.2 Критерии оценивания

Отлично	Хорошо	Удовлетворительн о	Неудовлетворительно
Студент демонстрирует глубокое и систематическое знание всех тем курса, правильно отвечает на 90–100% вопросов, включая сложные аналитические и практические задачи. Ответы на открытые вопросы полные, с примерами, обоснованиями и ссылками на изученные материалы. Практические задачи корректны, хорошо прокомментированы, демонстрируют творческий подход. Студент уверенно объясняет решения при необходимости.	Студент демонстрирует полное знание тем, но допускает незначительные ошибки в 10–25% вопросов. Корректно отвечает на большинство вопросов и задач, но может испытывать затруднения с углубленными аспектами. Ответы на открытые вопросы достаточные, но менее детализированные, с минимальными неточностями. Практические задачи выполнены с мелкими ошибками, код функционален.	Студент демонстрирует базовое знание тем, допускает ошибки в 25–40% вопросов. Корректно отвечает на простые вопросы (например, определение задач компьютерного зрения), но затрудняется с аналитическими или практическими задачами. Ответы на открытые вопросы минимальны, содержат неточности. Практические задачи содержат ошибки, требующие доработки.	Студент демонстрирует значительные пробелы в знаниях, ошибается в более чем 40% вопросов. Не может ответить на базовые вопросы или выполнить практические задачи. Ответы на открытые вопросы отсутствуют, некорректны или неполные. Код отсутствует или неработоспособен.

6.2.2.2 Семестр 5, Типовые оценочные средства для курсовой работы/курсового проектирования по дисциплине

6.2.2.2.1 Описание процедуры

1. Выбор темы:

Select a computer vision task (e.g., image classification, object detection, segmentation, face recognition, image generation).

Consult with the instructor to confirm the topic and dataset (e.g., Kaggle, ImageNet).

Sample project tasks/questions:

Develop a convolutional neural network for classifying images in the CIFAR-10 dataset using Keras.

Implement an object detection model with YOLOv5 for a custom dataset (e.g., traffic signs).

Create a face recognition system using OpenCV and a CNN with Triplet Loss.

Apply neural style transfer to generate artistic images from a given dataset.

2. Data Preparation:

Download and preprocess the dataset (e.g., normalization, augmentation with ImageDataGenerator in Keras).

Split data into training, validation, and test sets.

3. Model Implementation:

Implement a deep learning model using Python and libraries like Keras, TensorFlow, PyTorch, or OpenCV.

Tune hyperparameters and train the model, using cloud platforms (e.g., Google Colab) if local hardware (e.g., Dell Alienware 13) is insufficient.

4. Result Analysis:

Evaluate the model using metrics (e.g., accuracy, IoU, mAP) and generate visualizations (e.g., confusion matrices, bounding boxes).

Analyze results and identify limitations.

5. Report Preparation:

Prepare a 5-page report (excluding title page and references):

Format: Times New Roman, 14 pt, 1.5 line spacing, justified, 1.25 cm indent.

Structure: Introduction (goals, objectives), Theoretical Section (model, methods), Practical Section (data, code, results), Conclusion (findings, limitations).

Tables/Figures: Centered, labeled as “Таблица №. Название” (above, 11–12 pt) or “Рис. №. Название” (below, 12 pt), referenced in text (e.g., “см. Таблица 1”).

References: Minimum 3 sources, formatted per ГОСТ 7.1-2008, cited as [№] or [№, с. №] for quotes.

6. Submission:

Submit the report (.docx/.pdf) and code (.py/.ipynb, with comments) via the ИРНИТУ e-learning platform (el.istu.edu) or as instructed.

Prepare for a defense, explaining the project and answering questions.

Пример задания:

Develop a model for semantic segmentation on the Cityscapes dataset using a U-Net architecture.

1. Describe the U-Net architecture and its suitability for semantic segmentation.

2. Implement the model in Python using TensorFlow/Keras, including data preprocessing (e.g., resizing, normalization).

3. Evaluate the model using IoU and visualize segmentation masks (e.g., overlay on images).

4. Prepare a 5-page report with an introduction, theoretical section (U-Net, segmentation metrics), practical section (data, code, results with tables/figures), and conclusion. Include at least 3 references per ГОСТ 7.1-2008.

6.2.2.2.2 Критерии оценивания

Отлично	Хорошо	Удовлетворительно	Неудовлетворительно
Task fully completed with an appropriate model (e.g., U-Net, YOLO), code is correct, well-commented, and reproducible. Report (5 pages) adheres to all formatting requirements (Times New Roman, 14 pt, ГОСТ 7.1-2008 references), includes clear tables/figures with proper captions and text references. Results feature robust metrics, detailed visualizations, and insightful analysis. Defense: Student confidently explains the project, justifies choices, and answers all questions.	Task completed with minor issues (e.g., suboptimal hyperparameters). Code is functional but may have small errors; report follows requirements but has slight formatting inconsistencies (e.g., table captions). References (≥ 3) have minor citation errors; analysis is adequate but less comprehensive. Defense: Student explains the project well, answers most questions, but struggles with deeper aspects.	Task partially completed (e.g., low-quality results). Code has errors, minimal comments; report deviates from requirements (e.g., formatting, missing figure references), analysis is basic. References (≥ 3) have significant errors or incomplete citations. Defense: Student struggles to explain the project, answers only basic questions.	Task not completed or model is non-functional. Code is missing or unusable; report fails requirements (e.g., 5 pages, no structure, improper references). Results lack analysis or are absent. Defense: Student cannot explain the project or answer questions.

7 Основная учебная литература

1. Селянкин В. В. Компьютерное зрение. Анализ и обработка изображений : учебное пособие / В. В. Селянкин, 2023. - 152.

[Сайт] – URL: <https://e.lanbook.com/book/276455>

8 Дополнительная учебная литература и справочная

1. Форсайт Д. А. Компьютерное зрение. Современный подход : [пер. с англ.] / Дэвид Форсайт, Жан Понс, 2004. - 926 с.

9 Ресурсы сети Интернет

1. <http://library.istu.edu/>
2. <https://e.lanbook.com/>

10 Профессиональные базы данных

1. <http://new.fips.ru/>
2. <http://www1.fips.ru/>

11 Перечень информационных технологий, лицензионных и свободно распространяемых специализированных программных средств, информационных справочных систем

1. Microsoft Windows Seven Professional [1x1000] RUS (проведен апгрейд с Microsoft Windows Seven Starter [5x200])-поставка 2010

12 Материально-техническое обеспечение дисциплины

1. Телевизор LG 49" 49LF540V
2. Экран настенный ScreenMedia SPM-1102 Economy-P 180x180cm Matte White 1:1
3. Проектор Acer X125H
4. Ноутбук Dell Alenware 13 Core i5 6200U 8Gb.1Tb,nVidia GeForce GTX 960M 2Gb,13,3,