Syllabus for the Course "Computer Vision"

1. Abstract of the Discipline

The Computer Vision course aims to develop an engineering understanding of methods for analyzing images and video data in modern artificial intelligence systems.

Modern digital systems increasingly interact with the physical world through sensors, cameras, and other surveillance devices. In such systems, computer vision methods play a key role in converting visual information into data suitable for analysis, interpretation, and decision-making.

Within the framework of this discipline, the following topics are considered:

- the physical nature of image formation;
- methods of processing images and video data;
- algorithms for extracting features and objects;
- deep learning methods for computer vision tasks;
- architectures of modern computer vision systems;
- integration of computer vision into applied AI systems.

Particular attention is paid to the engineering aspects of the development and application of computer vision technologies, including:

- limitations and errors of computer vision systems;
- influence of lighting conditions, geometry, and sensors;
- checking the correct operation of models;
- integration of computer vision into human-machine decision-making systems.

The practical portion of the course is based on laboratory experiments using real cameras, objects, and physical setups. This allows students to directly observe the impact of shooting parameters, lighting, and scene geometry on algorithm performance.

Large language models (LLMs) are used in the discipline as a tool for engineering analysis, hypothesis generation, and design of computer vision system architectures.

2. General Characteristics of the Discipline

The course belongs to the professional module of the Master’s degree program «AI-Augmented Digital Systems Engineering» in the field of study 09.04.02 «Information Systems and Technologies».

The total volume of the course is 2 ECTS credits (72 academic hours). The discipline is implemented in the 4th semester.

Labor intensity structure:

- theoretical classes — 16 academic hours;
- practical classes — 16 academic hours;

- SDS of students — 40 hours.

The final assessment form is a credit with a grade.

The course is implemented in the AI-Augmented Engineering Learning format, which involves the active use of modern artificial intelligence tools in the educational process.

3. The Place of the Discipline in the Structure of the Educational Program

The discipline is included in the educational track «Artificial Intelligence Technologies (AI Technologies)» and is aimed at developing engineering competencies for the development of visual data analysis systems.

The course builds on the learning outcomes of the following courses:
- Fundamentals of Artificial Intelligence and Large Language Models;
- Neural Networks, Machine Learning, and Deep Learning;
- Natural Language Processing and Large Language Models;
- Testing, Verification, and Validation of AI Systems.

While the previous disciplines develop an understanding of machine learning methods and intelligent systems architectures, the Computer Vision discipline is dedicated to the application of these methods to the analysis of visual information.

This course is designed to develop an understanding of how artificial intelligence systems interact with the physical world through sensors and surveillance systems.

The acquired knowledge is used in studying the following disciplines:
- Edge AI;
- Architecture of Intelligent Digital Systems;
- Project Activities of the Master's Degree Program.

4. Objectives of Mastering the Discipline

The aim of the course is to develop an engineering understanding of methods for analyzing visual information and developing computer vision systems as part of modern AI solutions.

Key educational outcomes of the course:
- understanding of the physical and algorithmic principles of image formation;
- mastering image processing methods;
- study of algorithms for extracting features and objects;
- mastering deep learning methods for computer vision tasks;
- understanding the architecture of computer vision systems;
- analysis of limitations and errors of computer vision systems;
- development of skills in designing applied image analysis systems.

5. Objectives of the Discipline

The main objectives of the discipline are:

- study of the physical nature of image formation;
- mastering methods of image processing and analysis;
- study of feature extraction algorithms;
- mastering methods of object recognition;
- study of modern neural network architectures for computer vision tasks;
- analysis of errors and limitations of computer vision systems;
- mastering methods of experimental study of algorithm behavior;
- developing skills for integrating computer vision into applied AI systems.

6. Planned Learning Outcomes

Know:

- main tasks of computer vision;
- image processing methods;
- feature extraction methods;
- neural network architectures for image analysis;
- limitations of computer vision technologies;
- typical sources of errors in computer vision systems.

Be able to:

- analyze images and video data;
- apply image processing algorithms;
- train computer vision models;
- analyze model errors;
- design computer vision system architectures.

To own:

- image analysis methods;
- tools for developing computer vision models;
- methods of experimental study of the behavior of models;
- skills in integrating computer vision into AI systems.

7. Methodological Concept of the Discipline

The discipline is built around the idea that computer vision systems are part of broader human-machine decision-making systems.

Methodological axis of the course:
Physical world → sensors → image → algorithms → interpretation → solution

This course examines three levels of analysis of computer vision systems:

7.1. Physical Layer

The physical nature of image formation is studied:

- shooting geometry;
- lighting;
- optical properties of the camera;
- influence of observation conditions.

Understanding the physical layer is necessary to explain algorithmic errors.

7.2. Algorithmic Level

The following image processing algorithms are considered:

- filtering;
- edge detection (highlighting boundaries);
- segmentation;
- feature extraction;
- object recognition.

7.3. Architectural Level

The architectures of modern computer vision systems are studied:

- convolutional neural networks (CNN);
- detection architectures;
- segmentation architectures;
- multimodal AI systems.

8. Experimental Learning Model

The course is based on the principle of experimental research of the behavior of algorithms. Students complete laboratory work, including:

- working with cameras and sensors;
- experiments with lighting;
- analysis of the influence of shooting parameters;
- training computer vision models;
- analysis of algorithm errors.

This learning model allows for the development of a practical understanding of how computer vision technologies work.

9. The Role of LLM in the Educational Process

Large language models are used in the discipline as a tool for engineering activities.

9.1. LLM as an Analytical Tool

LLMs are used for:

- analysis of algorithm architectures;
- hypothesis generation;
- explanations of experimental results.

9.2. LLM as a Development Tool

LLMs are used for:
- generation of program code;
- development of system architectures;
- analysis of algorithm errors.

9.3. LLM as a Tool for Critical Analysis

Students use the LLM to:
- verify experimental results;
- search for alternative solutions;
- analyze model limitations.

10. Educational Technologies

The discipline uses:

- lecture classes;
- seminars and discussions;
- digital laboratory work;
- experiments with physical stands;
- project activities;
- using LLM as an engineering analysis tool.

11. Differentiated Assessment Model

The assessment is based on three levels:

Basic level:
- understanding of image processing methods;
- correct execution of laboratory work.

Advanced level:
- analysis of algorithm errors;
- design of computer vision system architecture.

Research level:
- conducting experimental research;
- development of prototypes of computer vision systems.

12. Final Certification

The final assessment is carried out in the form of a credit with a grade (not a test). The student presents the results of laboratory work and defends a mini-project of a computer vision system.

The project includes:

- task description;
- system architecture;
- image processing methods;
- model error analysis;
- suggestions for improving the quality of the system.

Curriculum schedule for the course

Key:

- **Lecture (Lect.)** — theoretical material delivery.
- **Practical / Lab.** — hands-on activities, simulations and discussions.
- **Self-directed work (SDS)** — independent study on project tasks and analysis.
- **LLM use** — applying Large Language Models for analysis, generation and decision support (integrated into independent study tasks).

Week	Content	Lectures (hours)	Practice / lab (hours)	SDS (h)
1	**Lecture:** The course begins with a discussion of real-world problems in which digital systems must analyze visual information, including quality control in manufacturing, video surveillance systems, autonomous robot navigation, and medical diagnostics. Students discuss the customer's expectations for such a system and the types of information it must extract from an image. The course then introduces the fundamental notion of an image as a numerical structure: for a computer, an image is a two-dimensional matrix of numbers (pixels), where each value represents brightness or color. This demonstrates that computer "vision" is not tied to human perception but is based on data processing. **Seminar / Demonstration:** The practical part begins with a demonstration of a modern object detection system (e.g., the YOLO (You Only Look Once) object detection algorithm) working with a webcam in real time. Students launch the object detection system and observe how the algorithm recognizes people and objects in the classroom. **Lab:** During the lab, students conduct their first engineering experiment: they vary the lighting, the distance to the	2	2	4

Week	Content	Lectures (hours)	Practice / lab (hours)	SDS (h)
	camera, and the position of objects, observing how the recognition quality changes. This experiment immediately reveals the limitations of the technology and creates the first “presence effect” of computer vision. **SDS:** Prepare a brief description of a real-world computer vision system and analyze the tasks it solves and the data it uses.			
2	**Lecture:** The lecture examines a typical industrial task: detecting defects on the surface of a component. The customer does not require “image understanding,” but expects the system to identify specific visual features — scratches, cracks, or irregularities. The scientific section introduces the concept of image feature extraction. The computer can highlight object boundaries, contours, textures, and contrast transitions, which become the basis for further image analysis. **Lab:** In the laboratory, students apply classical image processing methods, such as the Sobel operator and the Canny edge detector. This allows us to see which image structures can be detected without the use of neural networks. Engineering Experiment: Students vary the noise level, lighting, and image scale and observe how the robustness of edge detection changes. This helps them understand how algorithms depend on the quality of input data. **SDS:** Prepare a short report on the impact of image processing parameters on edge detection quality.	2	2	4
3	**Lecture:** The lecture examines the practical problem of automatically sorting objects in a warehouse or logistics center. Neural networks capable of automatically extracting image features are used to solve such problems. Students are introduced to the architecture of a convolutional neural network (CNN). The idea of convolution, feature maps, and pooling operations is explained. These reduce the size of the data and highlight stable visual structures. **Lab:** In this lab, students train a simple convolutional neural network to classify images from a small dataset. This allows them to see the full cycle of the model’s operation: data preparation, training, and accuracy testing. Engineering Experiment: Students vary the amount of training data and model parameters, observing the phenomenon of overfitting. This demonstrates the dependence of model quality on the data and the network architecture. **SDS:** Prepare an explanation of the operating principle of a convolutional neural network using one network layer as an example.	2	2	4

Week	Content	Lectures (hours)	Practice / lab (hours)	SDS (h)
4	**Lecture:** The lecture examines the automatic classification of objects as a practical task — for example, recognizing different types of products or parts. This task is widely used in retail, logistics, and automated sorting systems. Students explore the principles of learning classification models: the loss function, accuracy, and the process of optimizing the model parameters. It explains how the model gradually improves its predictions during the training process. **Lab:** Students train an image classification model on a small dataset. Particular attention is paid to analyzing the model's errors. Engineering Experiment: Students vary the ratio of classes in the dataset and observe the emergence of systematic error (model bias). **SDS:** Prepare a brief analysis of the classification model's errors and possible causes.	2	2	4
5	**Lecture:** The problems of object detection are considered (object detection), such as people counting indoors, traffic analysis, or access control. Unlike classification, the system must determine not only the type of object but also its position in the image. The principle of object detection using bounding boxes and modern architectures of detection models are explained: YOLO and Faster R-CNN (Faster Region-based Convolutional Neural Network) families. **Lab:** Students use a ready-made object detection model to analyze a video stream. This allows them to see how the algorithm simultaneously detects multiple objects. Engineering Experiment: Students test the system's robustness to challenging conditions: object occlusion, changing camera distance, and poor lighting. **SDS:** Prepare a brief description of the architecture of a modern object detection model.	2	2	4
6	**Lecture:** The problem of identifying the exact shape of an object in an image is considered — image segmentation. Such tasks are used, for example, in medical diagnostics, agriculture, and robotics. The difference between semantic segmentation (where the system identifies classes of objects) and instance segmentation (where it identifies individual instances of objects) is explained. **Lab:** Students use an image segmentation model (e.g., U-Net or Segment Anything Model). This allows us to see how the algorithm identifies objects at the individual pixel level. Engineering Experiment: Analysis of segmentation errors when changing object size, background, and image quality. **SDS:** Prepare a comparative analysis of two image segmentation approaches.	2	2	4

Week	Content	Lectures (hours)	Practice / lab (hours)	SDS (h)
7	**Lecture:** The lecture examines video data analysis systems, such as intelligent video surveillance or motion analysis in industrial processes. Such systems must consider not only individual frames, but also image changes over time. Students are introduced to object tracking methods (object tracking) and motion analysis. It explains how the system associates the same object in different video frames. **Lab:** Implementing object tracking in a video stream. Students observe how the algorithm identifies the movement of people or objects. Engineering Experiment: Analyzing complex situations: rapidly moving objects, a large number of people, or intersecting trajectories. **SDS:** Prepare an analysis of video stream analysis application scenarios.	1	1	4
8	**Lecture:** This lecture examines systems that combine image and text analysis. Such systems are used, for example, to analyze documents, create intelligent assistants, or robotic interfaces. The operating principle of vision-language models (multimodal models of vision and language) is explained. An example is the CLIP (Contrastive Language–Image Pretraining) model. **Lab:** Students use a model that can automatically describe an image with text (image captioning — the generation of a textual description of an image). Engineering Experiment: Analysis of errors in such models: ambiguous images, complex scenes, and unusual objects. **SDS:** Prepare examples of tasks that require joint processing of images and text.	1	1	4
9	**Lecture:** This lecture examines the architecture of a full-fledged computer vision system. The sequence of data processing steps is discussed: image acquisition, data preparation, model training, inference, and integration of the results into a digital system. Students are introduced to the concepts of a dataset, image annotation, and model quality assessment metrics. **Lab:** Students create a small dataset and train a model to solve a specific problem. Engineering Experiment: This demonstrates how changing the quality of the data markup or structure affects the quality of the model. **SDS:** Prepare a diagram of the computer vision system architecture for a selected task.	1	1	4
10	**Lecture:** The final week is devoted to the integration of computer vision technologies into real-world digital systems: smart cameras, robotic systems, quality control systems, and analytical platforms. The limitations of these	0	2	6

Week	Content	Lectures (hours)	Practice / lab (hours)	SDS (h)
	technologies are discussed: computing resources, processing latency, data dependence, and potential model errors. Lab/Mini-Project: Students implement a small computer vision project — for example, an object counting system, defect detection,			