KHMELNYTSKYI NATIONAL UNIVERSITY

APPROVED BY

Dean of the Faculty of Information Technologies

________ Tetiana HOVORUSHCHENKO
Signature Name, SURNAME

1 09 2025

WORK PROGRAM OF THE ACADEMIC DISCIPLINE

Conducting experiments and processing the results of scientific research
Discipline title

Purpose of the Work Program	For educational programs of various specialties
Level of Higher Education	Second (Master's)
Language of Instruction	English
Discipline Volume, ECTS Credits	8
Discipline Status	Selective
Faculty	Information Technologies
Department	Computer Science

Form of Education	Discipline Volume		Number of Hours							Form of semester control
			Classroom Sessions						Independent work (including IT)	
	ECTS Credits	Hours	Total	Lectures	Laboratory Session	Practical Sessions	Seminar Sessions			Credit
Full-time	8	240	68	32	36				172	+

The work program is based on the educational programs of training and the standard of higher education of the specialty

Work program compiled by ________ Ph.D., Pavlo RADIUK
Signature Academic degree, academic title, Name, SURNAME

Approved at the meeting of the Department of Computer Science

Protocol from 29.08.2025 № 1.

Head of the Department of Computer Science ________ Oleksander BARMAK
Title Signature Name, SURNAME

Khmelnytskyi 2025

3. Discipline Description

The academic course "Conducting experiments and processing the results of scientific research" is taught to students of the specialty F3 Computer Science in English, who are studying at Khmelnytskyi National University under the educational program "COMPUTER SCIENCE". This course is an elective component within the "COMPUTER SCIENCE" educational program.

Aim of the discipline. To master the complete cycle of conducting scientific experiments in computer science, from idea to deployed models, using the PyTorch framework.

Subject of the discipline. The methodology and toolkit for conducting experiments in machine learning, which includes the PyTorch framework, neural network architectures for classification and computer vision, data handling, principles of modular programming, replication of research papers, and model deployment methods.

Tasks of the discipline. To teach students to apply PyTorch for creating and training deep learning models; to implement a complete experimental workflow; to work with various data types, including custom datasets; to structure code into modular scripts; to analyze and reproduce architectures from scientific publications; to deploy trained models as interactive web applications for demonstrating results.

Learning outcomes. Upon successful completion of this course, a student will be able to: *know* the complete lifecycle of a machine learning experiment, the core components of the PyTorch ecosystem, the architectures of modern neural networks, advanced model evaluation and validation techniques, principles of modular software design, and fundamentals of model deployment; *be proficient* with the PyTorch framework, data visualization and deployment libraries (e.g., Matplotlib, Gradio), and version control systems for effective code management; *be able* to independently design, implement, train, and evaluate deep learning models for solving real-world problems, refactor experimental code from notebooks into modular scripts, critically analyze and replicate architectures from scientific literature, and deploy trained models as accessible web services.

4. Structure of ECTS Credits of the Discipline

Section Name (topic)	Number of hours given to:		
	Lectures	**Laboratory Sessions**	**Independent Study**
Topic 1. PyTorch fundamentals and workflow.	4	4	17
Topic 2. Architectures for neural network classification.	4	4	17
Topic 3. End-to-end computer vision with torchvision.	4	4	17
Topic 4. Advanced data handling with custom datasets.	4	6	20
Topic 5. Modularizing code for reproducible experiments.	4	4	27
Topic 6. Paper replicating: implementing the vision transformer.	4	4	25
Topic 7. Creating an interactive machine learning demo with Gradio.	4	4	25
Topic 8. Machine learning model deployment and real-world applications.	4	6	24
Total for the semester:	**32**	**36**	**172**

5. Program of the Academic Discipline

5.1 Content of the Lecture Course

Lecture Number	List of Lecture Topics and Their Annotations	Number of Hours
	Topic 1. PyTorch fundamentals and workflow.	**4**
1	Introduction to PyTorch and tensor fundamentals. Core concepts of tensors, fundamental operations. Lit.: [1] p. 1–28; [8] p. 30–52; [11] p. 3–18; [20] p. 12–20.	2
2	Advanced tensor operations and GPU acceleration. Manipulating tensor shapes, running computations on different devices. Lit.: [1] p. 45–70; [8] p. 60–68; [11] p. 40–70; [22] p. 90–110.	2
	Topic 2. Architectures for neural network classification.	**4**
3	The standard PyTorch workflow. Data preparation, model building, defining loss functions and optimizers, the training loop. Lit.: [1] p. 80–110; [7] p. 120–150; [8] p. 90–110; [11] p. 85–110; [22] p. 120–140.	2
4	Building a complete linear regression model. Implementing an end-to-end workflow for a regression task, including model saving and loading.	2

Lecture Number	List of Lecture Topics and Their Annotations	Number of Hours
	Lit.: [1] p. 112–130; [8] p. 180–210; [11] p. 110–130.	
	Topic 3. End-to-end computer vision with torchvision.	**4**
5	Architecting neural networks for binary classification. Understanding non-linearity and activation functions for binary problems. Lit.: [1] p. 140–158; [8] p. 220–240; [11] p. 140–170.	2
6	Scaling up: Multi-class classification. Adapting architectures for multi-class problems and evaluating performance. Lit.: [1] p. 158–182; [8] p. 240–260; [10] p. 7684–7690; [11] p. 170–210.	2
	Topic 4. Advanced data handling with custom datasets.	**4**
7	Introduction to computer vision with torchvision. Exploring torchvision.datasets and torchvision.transforms. Lit.: [1] p. 200–230; [3] p. 50–70; [11] p. 210–250; [21] p. 1–8.	2
8	Building and evaluating a baseline convolutional neural network (CNN). Creating a simple CNN and establishing a performance baseline. Lit.: [1] p. 230–270; [11] p. 250–310; [13] p. 1–8; [21] p. 1–10.	2
	Topic 5. Modularizing code for reproducible experiments.	**4**
9	Loading Data with ImageFolder and custom dataset classes. Techniques for handling standard and non-standard data structures. Lit.: [1] p. 272–290; [5] p. 1–3; [8] p. 310–330; [11] p. 312–340; [22] p. 160–180.	2
10	Data augmentation techniques for improved generalization. Using TrivialAugmentWide and other transforms to prevent overfitting. Lit.: [1] p. 292–310; [6] p. 1–10; [10] p. 7690–7698; [11] p. 340–380.	2
	Topic 6. Paper replicating: implementing the vision transformer.	**4**
11	Principles of modular code. The rationale and techniques for converting notebooks into reusable Python scripts. Lit.: [4] p. 1–8; [5] p. 1–4; [9] p. 1–12; [14] p. 15851–15857; [22] p. 200–230; [23] p. 31866–31875; [24] p. 20–40; [25] p. 6–18.	2
12	Creating reusable components. Building data_setup.py, engine.py, and model_builder.py for scalable projects. Lit.: [5] p. 2–4; [17] p. 1–12; [22] p. 230–260; [23] p. 31870–31879.	2
	Topic 7. Creating an interactive machine learning demo with Gradio.	**4**
13	Deconstructing a research paper: The vision transformer (ViT). Analyzing the paper's architecture, equations, and key contributions. Lit.: [6] p. 1–12; [10] p. 7682–7688; [12] p. 1–8; [14] p. 15850–15854.	2
14	Implementing the core components of ViT. Coding the patch embedding, attention mechanism, and encoder blocks. Lit.: [1] p. 410–440; [6] p. 8–12; [10] p. 7688–7698; [11] p. 550–590.	2
	Topic 8. Machine learning model deployment and real-world applications.	**4**
15	Introduction to model deployment with Gradio. Understanding the input-function-output paradigm and creating simple interfaces. Lit.: [3] p. 120–145; [17] p. 8–12; [20] p. 30–38; [22] p. 260–300; [23] p. 31870–31879.	2
16	Building and deploying a full-stack ML app. Structuring a Gradio project and deploying it to Hugging Face Spaces. Lit.: [2] p. 1–10; [4] p. 6–10; [9] p. 10–20; [13] p. 8–12; [22] p. 300–340; [23] p. 31870–31879.	2
	Total for the semester:	**32**

5.2 Content of Laboratory Sessions

#	Topic of the Laboratory Session	Number of Hours
1	Tensor creation, indexing, and manipulation in PyTorch. Lit.: [1] p. 1–28; [8] p. 30–52; [11] p. 3–18.	4
2	Building a complete end-to-end PyTorch workflow for a regression task. Lit.: [1] p. 112–130; [8] p. 180–210; [11] p. 110–130.	4
3	Developing a binary classification model for non-linear data. Lit.: [1] p. 140–158; [8] p. 220–240; [11] p. 140–170.	4
4	Training a baseline computer vision model on a standard dataset. Lit.: [1] p. 230–270; [11] p. 250–310; [13] p. 1–8; [21] p. 1–10.	6
5	Creating a custom dataset for a unique image classification task.	4

#	Topic of the Laboratory Session	Number of Hours
	Lit.: [1] p. 272–290; [5] p. 1–3; [8] p. 310–330; [11] p. 312–340.	
6	Refactoring a Jupyter Notebook into a set of modular Python scripts. Lit.: [4] p. 1–8; [5] p. 1–4; [14] p. 15851–15856; [22] p. 200–230; [23] p. 31866–31875; [24] p. 20–40; [25] p. 6–18.	4
7	Implementing the patch embedding and transformer encoder blocks of a ViT. Lit.: [1] p. 410–440; [6] p. 1–12; [10] p. 7684–7695; [11] p. 550–590.	4
8	Building an interactive web demo for a trained model using Gradio. Lit.: [3] p. 120–145; [17] p. 1–10; [22] p. 260–300; [23] p. 31870–31879.	6
	Total for the semester:	**36**

5.3 Content of the Student's Independent Study

The student's independent work consists of systematically studying the course material from the provided Jupyter Notebooks, preparing for laboratory sessions, completing individual assignments (IA), and preparing for the final assessment. Students have access to the department's page in the Modular Learning Environment, which hosts the Work Program and all necessary educational materials.

Week Number	Type of Independent Work	Number of Hours
1–2	Review of lecture materials (Topic 1). Preparation for Laboratory Work #1.	14
3–4	Review of lecture materials (Topic 2). Preparation for Laboratory Work #2. Completion of IA #1 "Comparative analysis of classification models on a custom dataset".	20
5–6	Review of lecture materials (Topic 3). Preparation for Laboratory Work #3.	14
7–8	Review of lecture materials (Topic 4). Preparation for Laboratory Work #4. Completion of IA #2 "Modularizing a research project: from notebook to a structured Python application".	23
9–10	Review of lecture materials (Topic 5). Preparation for Laboratory Work #5.	24
11–12	Review of lecture materials (Topic 6). Preparation for Laboratory Work #6. Completion of IA #3 "Replicating and extending a research paper's architecture (ViT)".	28
13–14	Review of lecture materials (Topic 7). Preparation for Laboratory Work #7.	24
15–16	Review of lecture materials (Topic 8). Preparation for Laboratory Work #8. Completion of IA #4 "Developing and deploying a full-stack FoodVision application with Gradio and Hugging Face". Defense of IA.	25
	Total for the semester:	**172**

Notes: LW – laboratory work, IA – individual assignment.

For independent study, students are assigned individual homework assignments on relevant topics formed in the modular environment on the MOODLE platform. Guidance on independent work and monitoring of the completion of individual homework assignments are provided by the instructor during scheduled consultation hours outside of class time.

The requirements for completing the individual homework assignments are outlined in the Modular Learning Environment on the discipline's page.

6. Teaching Technologies and Methods

The learning process for the discipline is based on the use of modern teaching technologies and methods, specifically: *lectures* (utilizing multimedia presentations, live coding demonstrations, problem-based and interactive learning); *laboratory sessions* (with a focus on the hands-on implementation, training, and analysis of machine learning models, solving applied problems, and case studies of research replication); *independent study* (systematic work through the provided Jupyter Notebooks, preparation for the completion and defense of laboratory work, preparation for final assessments, and completion and defense of individual homework assignments), utilizing modern information and computer technologies such as Jupyter Notebooks, PyTorch, Gradio, and distance learning platforms.

7. Methods of Control

Ongoing assessment is conducted during laboratory sessions and through the evaluation of scheduled control measures as defined by the work program and academic schedule, including the use of the Modular Learning Environment. The following methods of ongoing control are utilized:

– evaluation of the defense of laboratory work;

– defense and evaluation of the completion of individual assignments.

The final semester grade is determined by the results of ongoing control. A student who scores less than 60 percent of the maximum possible points for any type of academic work is considered to have an academic deficiency. The process for resolving academic deficiencies from semester control is conducted during the examination session or according to a schedule established by the dean’s office, in accordance with the «Regulations on the Control and Evaluation of Learning Outcomes of Higher Education Students at Khmelnytskyi National University».

8. Discipline Policy

The discipline’s policy is governed by the system of requirements for higher education students, as stipulated by the University’s current regulations on the organization and methodological support of the educational process. This includes undergoing safety training and mandatory attendance of all classes. For valid reasons (confirmed by documentation), theoretical learning may be conducted online with the lecturer’s approval. Successful mastery of the discipline and the formation of professional competencies and learning outcomes require thorough preparation for laboratory sessions (studying theoretical material on the topic, preparing a work report, and preparing for an oral quiz for admission to the session), active participation in class, and involvement in discussions regarding decisions made during the execution of laboratory work and individual assignments.

Students must adhere to the established deadlines for all types of academic work, as specified in the work program. The defense of a laboratory work is considered timely if the student defends it during the session immediately following its completion. A missed laboratory session must be completed by the student within a period set by the instructor, but no later than two weeks before the end of theoretical classes in the semester.

The student’s mastery of the theoretical material is assessed through quizzes during the defense of individual assignments and the final written exam.

When completing independent work for the discipline, students must adhere to the policy of academic integrity (cheating and plagiarism, including the use of mobile devices, are prohibited). If a violation of the academic integrity policy is detected in any form of academic work, the student will receive an unsatisfactory grade and must re-do the assignment for the corresponding topic (type of work) as provided for in the work program. Any form of academic dishonesty is ***not permitted***.

Within the scope of this discipline, the recognition and crediting of learning outcomes acquired through non-formal education are provided. These may include courses on accessible platforms that contribute to the formation of competencies and the deepening of learning outcomes defined by the Work Program, or that cover a relevant topic and/or type of work from the discipline’s program (for more details, see the «Regulations on the Procedure for Recognizing and Crediting the Learning Outcomes of Higher Education Students at Khmelnytskyi National University»). Thus, individual learning outcomes from the discipline may be credited if the student has obtained results from non-formal education confirmed by a relevant document (certificate, diploma, educational program, etc.):

– As a result of completing LW #1–2, the open online course is credited: https://shorturl.at/m0TC3

– As a result of completing LW #3–4, the online course is credited: https://course.fast.ai/

– As a result of completing LW #5–6, the online course is credited: https://shorturl.at/iCJ1H

– As a result of completing LW #7–8, the online course is credited: https://shorturl.at/mST8S

9. Assessment of Student Learning Outcomes in the Semester

The assessment of a student’s academic achievements is conducted in accordance with the «Regulations on the Control and Evaluation of Learning Outcomes of Higher Education Students at Khmelnytskyi National University». During the ongoing assessment of the work performed by the student for each structural unit and the results obtained, the instructor assigns a c ertain number of points from the range established by the Work Program for that type of work. Each structural unit of academic work can be credited if the student scores at least 60 percent (the minimum level for a positive grade) of the maximum possible points for that unit.

When assessing the learning outcomes of students for any type of academic work (structural unit), it is recommended to use the generalized criteria provided below:

Table – Criteria for Assessing the Academic Achievements of a Higher Education Student

Grade and Level of Achievement of Planned Learning Outcomes and Formed Competencies	Generalized Content of the Assessment Criterion
Excellent (high)	The student has deeply and fully mastered the content of the educational material, navigates it easily, and skillfully uses the conceptual apparatus; can link theory with practice, solve practical tasks, and

	confidently express and justify their judgments. An excellent grade presupposes a logical presentation of the answer in the language of instruction (in oral or written form), demonstrates high-quality work formatting, and proficiency with special instruments, tools, and application programs. The student does not hesitate when the question is modified, can draw detailed and generalizing conclusions, and demonstrates practical skills in solving professional tasks. The student made two or three ***minor errors*** in their answer.
Good (intermediate)	The student has demonstrated a complete understanding of the educational material, is proficient with the conceptual apparatus, and is oriented in the studied material; consciously uses theoretical knowledge to solve practical tasks; the presentation of the answer is competent, but there may be some inaccuracies in the content and form of the answer, unclear formulations of rules, principles, etc. The student's answer is based on independent thinking. The student made two or three ***errors*** in their answer.
Satisfactory (sufficient)	The student has demonstrated knowledge of the main program material to the extent necessary for further study and professional activity, and can handle the practical tasks provided by the program. As a rule, the student's answer is based on reproductive thinking, the student has a weak knowledge of the discipline's structure, makes inaccuracies and ***major errors*** in the answer, and hesitates when answering a modified question. However, the student has acquired the skills necessary to perform simple practical tasks that meet the minimum assessment criteria and possesses knowledge that allows them to correct inaccuracies in the answer under the guidance of the instructor.
Unsatisfactory (insufficient)	The student has demonstrated fragmented, unsystematic knowledge, cannot distinguish between primary and secondary information, makes mistakes in defining concepts, distorts their meaning, presents the material chaotically and uncertainly, and cannot use knowledge to solve practical tasks. As a rule, an «unsatisfactory» grade is given to a student who cannot continue their studies without additional work on the discipline.

Structuring the Discipline by Types of Academic Work and Assessing Student Learning Outcomes

Classroom Work								**Control Measures**	**Independent Work**	**Semester Control**
Laboratory Works #:								Final Control Work (FCW)	Individual Assignment (IA)	Credit
1*	2	3	4	5	6	7	8			
Points for each type of academic work (min-max)										
3-5	3-5	3-5	3-5	3-5	3-5	3-5	3-5	12-20	24-40	Based on rating
24-40								**12-20**	**24-40**	**60-100**

Notes: *****If a student scores below the established minimum for any type of academic work, they receive an unsatisfactory grade and must retake it within the period set by the instructor (dean). The institutional grade is determined according to the «**Correspondence between the institutional grading scale and the ECTS grading scale**» table.

Assessment of Laboratory Work Defense

A laboratory work, completed and formatted by the student according to the requirements set by the instructor at the beginning of the semester, is comprehensively evaluated by the instructor during its defense, taking into account the following criteria: independence and correctness of the program implementation of key concepts and architectures; completeness of the answer and knowledge of theoretical foundations (e.g., principles of deep learning, model training workflows, PyTorch API, deployment strategies); quality of the program code, its structure, and adherence to principles of modularity and reproducibility.

The result of each laboratory work's completion and defense is evaluated according to the «**Criteria for Assessing the Academic Achievements of a Higher Education Student**» table (minimum positive score – 3 points, maximum – 5 points).

If a student demonstrates a level of knowledge below 60 percent of the maximum score for any structural unit, the laboratory work is not credited. In this case, the student must study the material more thoroughly, correct any major errors, and re-defend the work at a time designated by the instructor.

Assessment of the Final Control Work

The FCW involves completing three tasks (two theoretical questions and one practical task that requires solving a problem by developing a program in a functional style to process a dataset). When assessing the FCW, the following are considered: the completeness and correctness of the answer, and the quality of the practical task's solution. Each theoretical question is worth 6 points, and the practical task is worth 8 points; the total score for a positive grade ranges from 12 to 20 points.

Number of correct answers	1	2	3
Percentage of correct answers	**0-59**	**60**	**100**
Points received	6	12	20

If a student receives a negative grade, the FCW should be retaken before the end of the credit week.

Assessment of Individual Assignment Completion

The IHA is completed and formatted by the student according to the requirements set by the instructor at the beginning of the semester and is comprehensively evaluated by the instructor based on the following criteria: independence of completion; correctness of the solution to the given tasks; justification of the choice of methods (e.g., selection of model architecture, justification of the experimental setup, and evaluation metrics); completeness of explanations and argumentation of conclusions; quality of the report and program code.

The result of each IA is evaluated according to the «**Criteria for Assessing the Academic Achievements of a Higher Education Student**» table (minimum positive score – 24 points, maximum – 40 points). Based on the defense, a corresponding score is assigned.

If a student demonstrates a level of knowledge and completion of the IA that is below 60 percent of the maximum score, the assignment is not credited. In this case, the student must re-work the assignment, correct errors, and submit the revised IA for review within the deadlines agreed upon with the instructor.

Final Semester Grade

The final semester grade on the institutional scale and the ECTS scale is determined automatically after the instructor enters the assessment results in points for all types of academic work into the electronic journal. The correspondence between the institutional grading scale and the ECTS grading scale is shown in the table below.

Table – Correspondence between the Institutional Grading Scale and the ECTS Grading Scale

ECTS Grade	**Rating Scale (Points)**	**Institutional Grade** (Level of achievement of planned learning outcomes)	
		Credit	**Exam/Differentiated Credit**
A	90-100	Credited	***Excellent*** – A high level of achievement of the planned learning outcomes, indicating the student's unconditional readiness for further study and/or professional activity.
B	83-89		***Good*** – An intermediate (maximally sufficient) level of achievement of the planned learning outcomes and readiness for further study and/or professional activity.
C	73-82		
D	66-72		***Satisfactory*** – The minimum sufficient level of achievement of the planned learning outcomes for further study and/or professional activity.
E	60-65		
FX	40-59	Not Credited	***Fail*** – A number of the planned learning outcomes are not met. The level of acquired learning outcomes is insufficient for further study and/or professional activity.
F	0-39		***Fail*** – The learning outcomes are absent.

The semester credit is awarded at the last session, provided that the total score accumulated by the student in the discipline from the ***ongoing*** assessment is between 60 and 100 points. In this case, the institutional grade is «***credited***,» and the ECTS grade is assigned a letter designation corresponding to the student's total score, according to the «**Correspondence**» table. The student's presence is not mandatory in this case.

10. Questions for Self-Assessment of Learning Outcomes

1. The concept of a tensor in PyTorch and its fundamental operations.
2. The main stages of the standard machine learning workflow in PyTorch.
3. The roles of a loss function and an optimizer in model training.
4. The architecture of a basic neural network for classification tasks.
5. The definition of a Convolutional Neural Network (CNN) and its primary applications.
6. The function of nn.Conv2d and nn.MaxPool2d layers in a CNN.
7. The process of creating and utilizing a custom Dataset in PyTorch.
8. The concept of data augmentation and its importance in model training.
9. The rationale for converting code from Jupyter Notebooks to modular Python scripts.
10. The core components of the Vision Transformer (ViT) architecture.
11. The concept of "patch embedding" as used in the Vision Transformer.
12. The definition and purpose of the self-attention mechanism.
13. Different approaches to deploying machine learning models, such as on-device and cloud-based.
14. The functionality and use case of the Gradio library for creating ML demos.
15. The typical file structure for a Gradio application deployed on Hugging Face Spaces.
16. The process of splitting a dataset into training, validation, and testing sets.
17. The concept of a learning rate and its impact on model training.

18. The difference between binary, multi-class, and multi-label classification.
19. The purpose of non-linear activation functions like ReLU and Sigmoid.
20. The functionality of torchvision.transforms for image preprocessing.
21. The relationship between torch.utils.data.Dataset and torch.utils.data.DataLoader.
22. The definition of a “baseline model” and its role in machine learning experiments.
23. The performance-speed tradeoff in selecting a model for deployment.
24. The concept of transfer learning and how to implement it using a feature extractor.
25. The process of freezing layers in a pretrained model.
26. The steps involved in replicating a model architecture from a research paper.
27. The role of the forward() method in a PyTorch nn.Module.
28. The concept of a residual or skip connection in neural networks.
29. How to use torchinfo.summary() to inspect a model’s architecture.
30. The distinction between model logits, prediction probabilities, and prediction labels.
31. The function of the softmax activation in multi-class classification.
32. The process of creating a requirements.txt file for a Python application.
33. The concept of “production code” in the context of machine learning.
34. The pros and cons of using notebooks versus Python scripts for ML projects.
35. The purpose of setting a random seed for reproducibility in experiments.
36. The definition of hyperparameters and examples from the ViT paper (e.g., MLP size, number of heads).
37. The difference between a “layer” and a “block” in a neural network architecture.
38. The role of the learnable [class] token in the Vision Transformer.
39. The purpose of position embeddings in the Vision Transformer.
40. How a convolutional layer can be used to create image patches.
41. The process of flattening a tensor and its application in neural networks.
42. The definition of “underfitting” and “overfitting” and how to identify them using loss curves.
43. Techniques for mitigating overfitting, such as data augmentation and dropout.
44. Techniques for mitigating underfitting, such as increasing model complexity.
45. The function of model.train() and model.eval() modes in PyTorch.
46. The purpose of the torch.inference_mode() context manager.
47. How to save and load a trained model’s state_dict in PyTorch.
48. The definition of an API and its role in model deployment.
49. The benefits of deploying a model as an interactive web application.
50. The workflow for deploying a Gradio application from a local machine to Hugging Face Spaces.

11. Educational and Methodological Support

Given the specifics of the academic discipline «Conducting Experiments and Processing the Results of Scientific Research,» namely the study of modern approaches to machine learning experimentation, the implementation of deep learning models, the application of software tools (PyTorch, torchvision, Gradio, Hugging Face), and the methods for paper replication and model deployment, and their rapid development, the primary educational resources used are the latest scientific articles from leading academic journals, materials from top scientific conferences, and official documentation where the latest advancements for the discipline are presented. The list of these sources is provided in the «Recommended Literature» section.

12. Material, Technical, and Software Support

For conducting lectures and laboratory sessions, specialized classrooms and computer labs of the department are used, equipped with modern computer technology and multimedia equipment. The software includes operating systems (Windows), interactive development environments (e.g., Jupyter Notebook, Google Colab), the Python programming language, the PyTorch deep learning framework, key libraries such as `torchvision`, `torchinfo`, `pandas`, `matplotlib`, and `Gradio`, as well as versio n control systems (Git). All software used within the scope of this discipline is either licensed or freely distributed.

13. Recommended Literature

1. Deep learning with PyTorch / H. Huang et al. 2nd ed. Shelter Island, NY, USA : Manning Publications, 2025. 600 p. URL: https://www.manning.com/books/deep-learning-with-pytorch-second-edition (date of access: 28.08.2025).
2. Kovalchuk O., Barmak O., Radiuk P., Klymenko L., Krak I. Towards transparent AI in medicine: ECG-based arrhythmia detection with explainable deep learning. *Technologies*. 2025. Vol. 13, no. 1. P. 34. URL: https://doi.org/10.3390/technologies13010034 (date of access: 28.08.2025).
3. Mitchell J., Deckard T. Leveraging data visualization to communicate effectively. Indianapolis, IN, USA : PALNI Open Press, 2025. 482 p. URL: https://doi.org/10.59319/ADKD6575 (date of access: 28.08.2025).

4. John M. M., Holmstrom Olsson H., Bosch J. An empirical guide to MLOps adoption: Framework, maturity model and taxonomy. *Information and Software Technology*. 2025. Vol. 183. P. 107725. URL: https://doi.org/10.1016/j.infsof.2025.107725 (date of access: 28.08.2025).
5. Lighter: Configuration-driven deep learning / I. Hadzic et al. *Journal of Open Source Software*. 2025. Vol. 10, no. 111. P. 8101. URL: https://doi.org/10.21105/joss.08101 (date of access: 28.08.2025).
6. Vision transformers for image classification: A comparative survey / Y. Wang et al. *Technologies*. 2025. Vol. 13, no. 1. P. 32. URL: https://doi.org/10.3390/technologies13010032 (date of access: 28.08.2025).
7. Aguinis H. Research methodology: Best practices for rigorous, credible, and impactful research. Thousand Oaks, CA, USA : SAGE Publications, Inc, 2024. 616 p. URL: https://us.sagepub.com/en-us/nam/research-methodology/book280088 (date of access: 28.08.2025).
8. Heumann C., Schomaker M., Shalabh. Introduction to statistics and data analysis: Eith exercises, solutions and applications in R. 2nd ed. Cham, Switzerland : Springer, 2024. 584 p. URL: https://doi.org/10.1007/978-3-031-11833-3 (date of access: 28.08.2025).
9. A joint study of the challenges, opportunities, and roadmap of MLOps and AIOps: A systematic survey / J. Diaz-De-Arcaya et al. *ACM Computing Surveys*. 2024. Vol. 56, no. 4. P. 1–30. URL: https://doi.org/10.1145/3625289 (date of access: 28.08.2025).
10. A survey on efficient vision transformers: algorithms, techniques, and performance benchmarking / L. Papa et al. *IEEE Transactions on Pattern Analysis and Machine Intelligence*. 2024. Vol. 46, no. 12. P. 7682–7700. URL: https://doi.org/10.1109/TPAMI.2024.3392941 (date of access: 28.08.2025).
11. Ayyadevara V. K., Reddy Y. Modern computer vision with PyTorch. 2nd ed. Birmingham, UK : Packt Publishing, Limited, 2024. 746 p. URL: https://www.packtpub.com/product/modern-computer-vision-with-pytorch/9781803231334 (date of access: 28.08.2025).
12. Radiuk P., Barmak O., Manziuk E., Krak I. Explainable deep learning: A visual analytics approach with transition matrices. *Mathematics*. 2024. Vol. 12, no. 7. P. 1024. URL: https://doi.org/10.3390/math12071024 (date of access: 28.08.2025).
13. Melnychenko O., Scislo L., Savenko O., Sachenko A., Radiuk P., Intelligent integrated system for fruit detection using multi-UAV imaging and deep learning. *Sensors*. 2024. Vol. 24, no. 6. P. 1913. URL: https://doi.org/10.3390/s24061913 (date of access: 28.08.2025).
14. Reproduce, replicate, reevaluate: the long but safe way to extend machine learning methods / L. Werner et al. *Proceedings of the 38th AAAI Conference on Artificial Intelligence (AAAI)*. 2024. Vol. 38, no. 14. P. 15850–15858. URL: https://doi.org/10.1609/aaai.v38i14.29515 (date of access: 28.08.2025).
15. Barmak O., Krak I., Yakovlev S., Manziuk E., Radiuk P., Kuznetsov V. Toward explainable deep learning in healthcare through transition matrix and user-friendly features. *Frontiers in Artificial Intelligence*. 2024. Vol. 7. P. 1482141. URL: https://doi.org/10.3389/frai.2024.1482141 (date of access: 28.08.2025).
16. Forero D. A., Curioso W. H., Wang W. Ten simple rules for successfully carrying out funded research projects. *PLoS Comput. Biol.* 2024. Vol. 20, no. 9. P. e1012431. URL: https://doi.org/10.1371/journal.pcbi.1012431 (date of access: 28.08.2025).
17. Nazir R., Bucaioni A., Pelliccione P. Architecting ML-enabled systems: Challenges, best practices, and design decisions. *Journal of Systems and Software*. 2024. Vol. 207. P. 111860. URL: https://doi.org/10.1016/j.jss.2023.111860 (date of access: 28.08.2025).
18. Lai B. S. The grant writing guide: A road map for scholars. Princeton, NJ, USA : Princeton University Press, 2023. 240 p. URL: https://doi.org/10.1515/9780691231891 (date of access: 28.08.2025).
19. Ormrod J. E., Leedy P. D. Practical research: Design and process. 13th ed. Harlow, UK : Pearson, 2023. 525 p. URL: https://eric.ed.gov/?q=process&ff1=pubBooks&id=ED650785 (date of access: 28.08.2025).
20. Armstrong J. S., Green K. C. The scientific method: A guide to finding useful knowledge. *Community ecology*. 2023. Vol. 24. P. 143. URL: https://doi.org/10.1017/9781108866807 (date of access: 28.08.2025).
21. Kalyta O., Barmak O., Radiuk P., Krak I. Facial emotion recognition for photo and video surveillance based on machine learning and visual analytics. *Applied Sciences*. 2023. Vol. 13, no. 17. P. 9890. URL: https://doi.org/10.3390/app13179890 (date of access: 28.08.2025).
22. Haviv Y., Gift N. Implementing MLOps in the enterprise: A production-first approach. Sebastopol, CA, USA : O'Reilly Media, Inc., 2023. 377 p. URL: https://www.oreilly.com/library/view/implementing-mlops-in/9781098136574/ (date of access: 28.08.2025).
23. Kreuzberger D., Kuhl N., Hirschl S. Machine learning operations (MLOps): Overview, definition, and architecture. *IEEE Access*. 2023. Vol. 11. P. 31866–31879. URL: https://doi.org/10.1109/ACCESS.2023.3262138 (date of access: 28.08.2025).
24. Shamoo A. E., Resnik D. B. Responsible conduct of research. 4th ed. Oxford, UK : Oxford University Press, 2022. 344 p. URL: https://doi.org/10.1093/oso/9780197547090.001.0001 (date of access: 28.08.2025).
25. Ніколаєв Є. Академічна доброчесність: інформаційний бюлетень (зведене видання всіх випусків). Аналіт. центр "ОсвітАналітика", 2022. 105 с. URL: https://www.skeptic.in.ua/wp-content/uploads/Integrity-bulletin-all-issues.pdf (дата звернення: 28.08.2025).

14. Information Resources

1. Modular learning environment of KhNU. URL: https://msn.khmnu.edu.ua/course/view.php?id=7932
2. Electronic library of KhNU. URL: http://library.khmnu.edu.ua/
3. Institutional repository of KhNU. URL : https://elar.khmnu.edu.ua/home

CONDUCTING EXPERIMENTS AND PROCESSING THE RESULTS OF SCIENTIFIC RESEARCH

Discipline Type	Elective
Level of Higher Education	Second (*Master's*)
Language of Instruction	English
Term	–
ECTS Credits	4.0
Forms of Education	Full-time (day)

Learning outcomes. *know* the complete lifecycle of a machine learning experiment, the core components of the PyTorch ecosystem, the architectures of modern neural networks, advanced model evaluation and validation techniques, principles of modular software design, and fundamentals of model deployment; *be proficient* with the PyTorch framework, data visualization and deployment libraries (e.g., Matplotlib, Gradio), and version control systems for effective code management; *be able* to independently design, implement, train, and evaluate deep learning models for solving real-world problems, refactor experimental code from notebooks into modular scripts, critically analyze and replicate architectures from scientific literature, and deploy trained models as accessible web services.

Content of academic discipline. Introduction to PyTorch and tensor fundamentals. Advanced tensor operations and GPU acceleration. The standard PyTorch workflow. Building a complete linear regression model. Architecting neural networks for binary classification. Scaling up: multi-class classification. Introduction to computer vision with torchvision. Building and evaluating a baseline convolutional neural network (CNN). Loading data with ImageFolder and custom dataset classes. Data augmentation techniques for improved generalization. Principles of modular code. Creating reusable components. Deconstructing a research paper: the vision transformer (ViT). Implementing the core components of ViT. Introduction to model deployment with Gradio. Building and deploying a full-stack ML app.

Planned learning activities. The minimum volume of study for one ECTS credit of the academic discipline for the ***second*** (Master's) level of higher education in the full-time form of education is 10 hours per 1 ECTS credit.

Forms (methods) of teaching: *lectures* (utilizing multimedia presentations, live coding demonstrations, problem-based and interactive learning); *laboratory sessions* (with a focus on the hands-on implementation, training, and analysis of machine learning models, solving applied problems, and case studies of research replication); *independent study* (systematic work through the provided Jupyter Notebooks, preparation for the completion and defense of laboratory work, preparation for final assessments, and completion and defense of individual homework assignments), utilizing modern information and computer technologies such as Jupyter Notebooks, PyTorch, Gradio, and distance learning platforms.

Forms of assessment of learning outcomes: assessment of the completion and defense of laboratory work; assessment of the completion and defense of individual homework assignments.

Type of semester control: credit.

Educational resources:

1. Deep learning with PyTorch / H. Huang et al. 2nd ed. Shelter Island, NY, USA : Manning Publications, 2025. 600 p. URL: https://www.manning.com/books/deep-learning-with-pytorch-second-edition (date of access: 28.08.2025).
2. Vision transformers for image classification: A comparative survey / Y. Wang et al. *Technologies*. 2025. Vol. 13, no. 1. P. 32. URL: https://doi.org/10.3390/technologies13010032 (date of access: 28.08.2025).
3. Aguinis H. Research methodology: Best practices for rigorous, credible, and impactful research. Thousand Oaks, CA, USA : SAGE Publications, Inc, 2024. 616 p. URL: https://us.sagepub.com/en-us/nam/research-methodology/book280088 (date of access: 28.08.2025).
4. Heumann C., Schomaker M., Shalabh. Introduction to statistics and data analysis: Eith exercises, solutions and applications in R. 2nd ed. Cham, Switzerland : Springer, 2024. 584 p. URL: https://doi.org/10.1007/978-3-031-11833-3 (date of access: 28.08.2025).
5. A survey on efficient vision transformers: algorithms, techniques, and performance benchmarking / L. Papa et al. *IEEE Transactions on Pattern Analysis and Machine Intelligence*. 2024. Vol. 46, no. 12. P. 7682–7700. URL: https://doi.org/10.1109/TPAMI.2024.3392941 (date of access: 28.08.2025).
6. Ormrod J. E., Leedy P. D. Practical research: Design and process. 13th ed. Harlow, UK : Pearson, 2023. 525 p. URL: https://eric.ed.gov/?q=process&ff1=pubBooks&id=ED650785 (date of access: 28.08.2025).
7. Modular learning environment of KhNU. URL: https://msn.khmnu.edu.ua/course/view.php?id=7932
8. Electronic library of KhNU. URL: http://library.khmnu.edu.ua/

Instructors: PhD, Senior Lecturer Pavlo RADIUK