Table of Contents

**KHmelnYTSKYI NATIONAL UNIVERSITY**

**APPROVED BY**

|  |  |
| --- | --- |
| Dean of the Faculty of Information Technologies |  |
| \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ | **Tetyana HOVORUSHCHENKO** |
| Signature | Name, SURNAME |

2025

### WORK PROGRAM OF THE ACADEMIC DISCIPLINE

Conducting Experiments and Processing the Results of Scientific Research

Discipline name

| **Purpose of the Work Program** | For the educational program “COMPUTER SCIENCE” of the specialty F7 Computer Science |
| --- | --- |
| **Level of Higher Education** | Second (Master’s) |
| **Language of Instruction** | English |
| **Discipline Volume, ECTS Credits** | 4 |
| **Discipline Status** | Elective |
| **Faculty** | Information Technologies |
| **Department** | Computer Science |

| **Form of Education** | **Discipline Volume** |  | **Number of Hours** |  |  |  |  |  | **Form of Semester Control** |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
|  | **ECTS Credits** | **Hours** | **Classroom Sessions** |  |  |  |  | **Independent Study** |  |
|  |  |  | **Total** | **Lectures** | **Laboratory Work** | **Practical Sessions** | **Seminar Sessions** |  |  |
| Full-time | 4 | 120 | 48 | 16 | 32 |  |  | 72 | Credit |

The work program is based on the Master’s educational program “COMPUTER SCIENCE” and the higher education standard for the specialty F3 Computer Science.

|  |  |
| --- | --- |
| Work program compiled by \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ | **PhD Pavlo RADIUK** |
|  | Academic degree, academic title, Name, SURNAME |
| Signature |  |

Approved at the meeting of the Department of Computer Science

Protocol from **19.03** **2025** № **11**.

|  |  |
| --- | --- |
| Head of the Department of Computer Science | \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ **Oleksandr BARMAK** |
| Title | Signature            Name, SURNAME |

Khmelnytskyi 2025

### 1. Discipline Description

The academic course “Conducting experiments and processing the results of scientific research” is taught to students of the specialty F7 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.

### 2. Aim, Subject, and Objectives of the Discipline

**Aim of the discipline.** The aim of the course is to equip students with comprehensive practical skills for conducting scientific research in the field of computer science, covering the full cycle of experimental work: from fundamental concepts and the PyTorch workflow to building, training, modularizing code, replicating research papers, and deploying machine learning models.

**Subject of the discipline.** The subject of study is the methodology and toolkit for conducting experiments in machine learning, including the fundamentals of PyTorch, standard workflows, neural network architectures for classification and computer vision, working with custom datasets, principles of modular code, approaches to replicating research papers, and methods of model deployment.

**Objectives of the discipline.**

* Master the fundamental concepts of PyTorch, including tensor operations.
* Learn and practically apply the standard PyTorch workflow: from data preparation to model evaluation.
* Acquire skills in building, training, and validating models for classification and computer vision tasks.
* Learn to work with custom datasets, including their loading and augmentation.
* Master the principles of modular programming to structure and optimize research code.
* Develop skills in replicating architectures and results from research papers, particularly the Vision Transformer.
* Gain practical experience in deploying machine learning models as interactive web applications.

**Learning outcomes.** Upon completion of the course, the student must: **know** the full cycle of conducting experiments in machine learning, key components of the PyTorch framework, modern neural network architectures, model evaluation and validation methods, principles of modular code, and the basics of model deployment; **be proficient** with the PyTorch toolkit, libraries for visualization and deployment (Matplotlib, Gradio), and version control systems; **be able to** independently implement, train, and evaluate deep learning models to solve applied problems; refactor research code from notebooks into modular scripts; analyze research papers, replicate proposed architectures, and deploy trained models as web services.

### 3. Explanatory Note

The discipline “Conducting experiments and processing the results of scientific research” is an elective component of the educational program designed to deepen students’ practical skills in conducting scientific experiments using modern machine learning tools. This course logically complements the theoretical knowledge gained from mandatory components such as “Models and Methods of Text Analytics” and “Large Computer Vision Models”, providing students with the opportunity to apply them to solve real-world problems. The main focus of the course is on the complete research cycle: from model development and training to its validation, optimization, and deployment. Throughout the course, students acquire important soft skills: the ability to independently study research papers, reproduce results, work with code, and present their findings in the form of interactive demonstrations.

**Aim of the discipline.** To master the complete cycle of conducting scientific experiments in computer science, from idea to a deployed model, 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.** After completing the course, the student must: **know** the theoretical foundations and practical aspects of the full research cycle in machine learning, modern neural network architectures, and methods for their evaluation and validation; **be proficient** with PyTorch, Gradio, Hugging Face, and relevant data visualization libraries; **be able to** independently develop, train, and deploy deep learning models, replicate results from research papers, effectively structure projects, and present their research findings.

### 4. Structure of ECTS Credits of the Discipline

| **Topic Name** | **Lectures** | **Laboratory Work** | **Independent Study** |
| --- | --- | --- | --- |
| Topic 1. PyTorch Fundamentals and Workflow | 4 | 8 | 18 |
| Topic 2. Building and Training Computer Vision Models | 4 | 8 | 18 |
| Topic 3. Code Modularization and Research Paper Replication | 4 | 8 | 18 |
| Topic 4. Model Deployment and Implementation | 4 | 8 | 18 |
| **Total for the semester:** | **16** | **32** | **72** |

### 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 Tensors. Core concepts of tensors, fundamental operations. Computational graphs and automatic differentiation.Source: Notebook 01. | 2 |
| 2 | The Standard PyTorch Workflow. Data preparation, model building, defining a loss function and optimizer, the training loop, and model evaluation.Source: Notebook 02. | 2 |
|  | **Topic 2. Building and Training Computer Vision Models** | **4** |
| 3 | Models for Classification. Architectures for classification neural networks, activation functions, non-linearity. Working with custom datasets.Source: Notebook 03, 05. | 2 |
| 4 | Computer Vision in PyTorch. The torchvision library, DataLoader, baseline models, and Convolutional Neural Networks (CNNs).Source: Notebook 04. | 2 |
|  | **Topic 3. Code Modularization and Research Paper Replication** | **4** |
| 5 | A Modular Approach to Code. Converting code from Jupyter Notebooks into Python scripts: data\_setup.py, engine.py, model\_builder.py, train.py.Source: Notebook 06. | 2 |
| 6 | Replicating Research Papers. Deconstructing a research paper (Vision Transformer). Step-by-step implementation of the model architecture based on its description and equations.Source: Notebook 07. | 2 |
|  | **Topic 4. Model Deployment and Implementation** | **4** |
| 7 | Introduction to Model Deployment. Deployment types: on-device vs. cloud. Tools for creating interactive demos: Gradio.Source: Notebook 08. | 2 |
| 8 | Deploying to Hugging Face Spaces. Preparing the file structure for a demo application (app.py, model.py, requirements.txt). Publishing the application.Source: Notebook 08. | 2 |
|  | **Total for the semester:** | **16** |

#### 5.2 Content of Laboratory Sessions

| **№** | **Topic of the Laboratory Session** | **Number of Hours** |
| --- | --- | --- |
| 1 | Tensor Manipulation and Operations in PyTorch.Source: Notebook 01. | 2 |
| 2 | Advanced Tensor Operations and GPU Acceleration.Source: Notebook 01. | 2 |
| 3 | Implementing a Full Linear Regression Workflow.Source: Notebook 02. | 2 |
| 4 | Experimenting with Different Optimizers and Loss Functions.Source: Notebook 02. | 2 |
| 5 | Building a Binary Classification Model.Source: Notebook 03. | 2 |
| 6 | Implementing a Multi-Class Classification Model with Non-Linearities.Source: Notebook 03. | 2 |
| 7 | Building a Baseline CNN Model with torchvision.Source: Notebook 04. | 2 |
| 8 | Analyzing the Speed vs. Performance Trade-off in Computer Vision Models.Source: Notebook 04. | 2 |
| 9 | Creating a Custom Dataset Class for Image Loading.Source: Notebook 05. | 2 |
| 10 | Applying TrivialAugment and Other Advanced Data Augmentation Transforms.Source: Notebook 05. | 2 |
| 11 | Converting Notebook Cells into Reusable Python Functions.Source: Notebook 06. | 2 |
| 12 | Building a Trainable Script (train.py) from Modular Components.Source: Notebook 06. | 2 |
| 13 | Implementing the Patch Embedding Layer of a Vision Transformer (ViT).Source: Notebook 07. | 2 |
| 14 | Assembling and Summarizing a Complete ViT Model.Source: Notebook 07. | 2 |
| 15 | Creating an Interactive Machine Learning Demo with Gradio.Source: Notebook 08. | 2 |
| 16 | Final Project: Deploying a Trained ViT Model to Hugging Face Spaces.Source: Notebook 08. | 2 |
|  | **Total for the semester:** | **32** |

#### 5.3 Content of the Student’s Independent Study

The student’s independent work consists of systematically studying the course material from the provided sources, preparing for laboratory sessions, completing individual homework assignments (IHA), 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 necessary educational materials for the discipline.

| **Week Number** | **Type of Independent Work** | **Number of Hours** |
| --- | --- | --- |
| 1-4 | Review of lecture materials. Preparation for the completion and defense of Lab Works №1-2. | 18 |
| 5-8 | Review of lecture materials. Preparation for the completion and defense of Lab Works №3-4. Completion of IHA №1 “Comparative Analysis of Classification Models on a Custom Dataset.” | 18 |
| 9-12 | Review of lecture materials. Preparation for the completion and defense of Lab Works №5-6. Completion of IHA №2 “Modularizing a Research Project: From Notebook to a Structured Python Application.” | 18 |
| 13-16 | Review of lecture materials. Preparation for the completion and defense of Lab Works №7-8. Completion of IHA №3 “Replicating and Extending a Research Paper’s Architecture (ViT).” Completion of IHA №4 “Developing and Deploying a Full-Stack FoodVision Application with Gradio and Hugging Face.” | 18 |
|  | **Total for the semester:** | **72** |

**Notes:** LW – laboratory work, IHA – individual homework assignment.

### 6. Teaching Technologies and Methods

The learning process for this discipline is based on the use of modern teaching technologies and methods, including: **lectures** (using problem-based and interactive learning, code demonstrations, and information-communication technologies); **laboratory sessions** (with a focus on the practical implementation, training, and analysis of machine learning models); **independent study** (reviewing theoretical material, preparing for laboratory work, preparing for the final assessment, completing and defending individual homework assignments), using Jupyter Notebooks, PyTorch, Gradio, and other modern tools.

### 7. Methods of Control

Ongoing assessment is carried out during laboratory sessions and on days of control measures established by the work program. The following methods of ongoing assessment are used: \* Evaluation of the results of defending laboratory work. \* Defense and evaluation of the results of individual homework assignments.

The final semester grade is determined based on the results of the ongoing assessment.

### 8. Discipline Policy

The discipline’s policy is determined by the system of requirements for the student, as provided by the University’s current regulations on the organization of the educational process. Attendance is mandatory. Successful completion of the discipline requires active participation in classes, timely completion and defense of all laboratory work and individual assignments. All forms of academic dishonesty, including plagiarism and cheating, are **not permitted**. In case of violations, the student will receive an unsatisfactory grade and must redo the assignment.

### 9. Assessment of Student Learning Outcomes in the Semester

The assessment of the student’s academic achievements is carried out on a 100-point scale. Each laboratory work and individual assignment is graded according to a set number of points. To receive a positive grade for the discipline, a student must score at least 60 points.

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

| **Laboratory Work №:** |  |  |  |  |  |  |  | **Individual Homework Assignment (IHA)** |  |  | **Credit** |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **1** | **2** | **3** | **4** | **5** | **6** | **7** | **8** | **1** | **2** | **3** | **4** |
| **Points for each type of academic work (min-max)** |  |  |  |  |  |  |  |  |  |  |  |
| 4-7 | 4-7 | 4-7 | 4-7 | 4-7 | 4-7 | 4-7 | 4-7 | 6-10 | 6-10 | 6-10 | 6-12 |
| **32-56** |  |  |  |  |  |  |  | **24-42** |  |  | **60-100** |

### 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

The educational process for the discipline is fully and sufficiently provided with necessary educational and methodological materials in the form of a series of Jupyter Notebooks, which contain theoretical explanations, code examples, and practical assignments.

### 12. Material, Technical, and Software Support

Specialized classrooms and computer labs of the department, equipped with modern computer technology and multimedia equipment, are used for lectures and laboratory sessions. The software includes operating systems (Windows, Linux), a Jupyter Notebook environment (or Google Colab), Python 3.x, and the main libraries: PyTorch, torchvision, torchinfo, pandas, matplotlib, Gradio.

### 13. Recommended Literature

1. PyTorch Fundamentals (Jupyter Notebook)
2. PyTorch Workflow Fundamentals (Jupyter Notebook)
3. PyTorch Neural Network Classification (Jupyter Notebook)
4. PyTorch Computer Vision (Jupyter Notebook)
5. PyTorch Custom Datasets (Jupyter Notebook)
6. PyTorch Going Modular (Markdown document)
7. PyTorch Paper Replicating (Jupyter Notebook)
8. PyTorch Model Deployment (Jupyter Notebook)
9. PyTorch Documentation. URL: https://pytorch.org/docs/stable/index.html

### 14. Information Resources

1. Modular Learning Environment of Khmelnytskyi National University.
2. Electronic Library of Khmelnytskyi National University. URL: http://library.khmnu.edu.ua/
3. Institutional Repository of Khmelnytskyi National University. URL: https://elar.khmnu.edu.ua/home

### ANNOTATION OF THE DISCIPLINE (FOR THE CATALOG)

CONDUCTING EXPERIMENTS AND PROCESSING THE RESULTS OF SCIENTIFIC RESEARCH

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

**Learning outcomes.** After completing the course, the student must: **know** the theoretical foundations and practical aspects of the full research cycle in machine learning, modern neural network architectures, and methods for their evaluation and validation; **be proficient** with PyTorch, Gradio, Hugging Face, and relevant data visualization libraries; **be able to** independently develop, train, and deploy deep learning models, replicate results from research papers, effectively structure projects, and present their research findings.

**Content of the academic discipline.** Fundamentals of PyTorch, tensors, operations, and automatic differentiation. The standard machine learning workflow: data preparation, model building, training, evaluation. Neural network architectures for classification and computer vision, Convolutional Neural Networks (CNNs). Working with custom datasets and data augmentation. Modularization of code and conversion of Jupyter Notebooks into Python scripts. Analysis and replication of architectures from research papers, focusing on the Vision Transformer (ViT). Deployment of models as interactive web applications using Gradio and Hugging Face Spaces.

**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 (using problem-based and interactive learning, code demonstrations); laboratory sessions (with a focus on the practical implementation, training, and analysis of models); independent study (reviewing theoretical material, completing and defending individual homework assignments) using Jupyter Notebooks and modern tools for deep learning.

**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. A series of educational materials in the Jupyter Notebook format.
2. Official PyTorch documentation.
3. Scientific articles from leading conferences and archives (e.g., arXiv).
4. Modular Learning Environment of Khmelnytskyi National University.
5. Electronic Library of Khmelnytskyi National University.

**Instructors:** PhD, Senior Lecturer Pavlo M. Radiuk