TECHNISCHE HOCHSCHULE DEGGENDORF

Electives I and II Guide
Artificial Intelligence and Data Science
Summer Semester

Faculty Computer Science
Date: 22.11.2022

Technische Hochschule Deggendorf
Dieter-Görlitz-Platz 1
94469 Deggendorf

Tel.: +49 991 3615-0
Fax: +49 991 3615-297

www.th-deg.de
info@th-deg.de

Table of Contents

AIN-B-11 Computational Logic

Module code	AIN-B-11
Module coordination	Prof. Dr. Cezar Ionescu
Course number and name	AIN-B-11 Computational Logic
Original study program	Bachelor Artificial Intelligence
Lecturers	Prof. Dr. Cezar Ionescu Prof. Dr. Josef Schneeberger
Duration of the module	1 semester
Module frequency	Annually, summer semester
Semester periods per week (SWS)	4
ECTS	5
Workload	Time of attendance: 60 hours self-study: 90 hours Total: 150 hours
Type of Examination	Written ex. 90 min. (specialized exam for M-AID students)
Duration of Examination	90 min.
Language of Instruction	English

Module Objective

Students acquire understanding and hands-on experience of various logical systems and their usage in artificial intelligence applications. Specifically, students will have achieved the following outcomes upon completion of the module:

Subject competency

Students understand the significance of logic for intelligent problem-solving.

Methodological competency

Students select the most appropriate logical system for solving a concrete practical problem, and use it to implement software-based solutions.

Personal competency

Students understand complex theoretical concepts and apply them to problems arising in practice.

Social competency

Students communicate clearly, argue and criticize logically and constructively, contribute to reasoned, team-oriented problem solving processes in the group.

Applicability in this and other Programs

Logic is foundational for all computer science courses and programmes. This module is a pre-requisite for the more advanced artificial intelligence lectures that build upon it.

Entrance Requirements

Recommended:

- Mathematics 1

- Foundations of Computer Science

Learning Content

Formal Logic: Syntax and Semantics

- Introduction to logical languages
- Basic concepts of logic
- Propositional logic
- Predicate (first-order) logic
- Formal proofs
- Set theory
- Classical semantics for first-order logic
- Resolution for propositional and first-order logic
- Semantics of logic programming

Logical Programming

- Prolog
- Answer Set Programming

Teaching Methods

- Interactive lectures
- Practical exercises using automatic proof checkers and theorem provers
- Software implementation of application-oriented examples

Recommended Literature

- Barwise, J und Etchemendy, J: Language, Proof and Logic, CSLI 2003 (or newer)
- Lifschitz, V.: Answer Set Programming, Springer Verlag 2019
- Gebser, M., Kaminski, R., Kaufmann, B., Schaub, T.: Answer Set Solving in Practice, Morgan & Claypool Publishers, 2013

TECHNISCHE HOCHSCHULE DEGGENDORF

AIN-B-22 Computer Vision

Module code	AIN-B-22
Module coordination	Prof. Dr. Patrick Glauner
Course number and name	AIN-B-22 Computer Vision
Original study program	Bachelor Artificial Intelligence
Lecturers	Prof. Dr. Patrick Glauner
Duration of the module	1 semester
Module frequency	Annually, summer semester
Semester periods per week (SWS)	4
ECTS	5
Workload	Time of attendance: 60 hours self-study: 90 hours Total: 150 hours
Type of Examination	Project work (specialized exam for M-AID students)
Language of Instruction	English

Module Objective

The goal of this module is to learn computer vision (CV), which allows computers to process visual input. We deal with CV dozens of times every day, e.g., face recognition, real-time translation of camera input, or automatic tagging of friends in photos. Modern CV algorithms rely heavily on machine learning methods, especially deep neural networks. Students acquire knowledge in CV and can deepen it in the future, e.g., in projects or further studies.

Specifically, students will have achieved the following learning outcomes upon completion of the module:

Professional Competence
Students will understand the concepts of the most common approaches to image understanding. (2 - Understanding)

Methodological competence
Students will have the ability to create high quality programs using image understanding technologies. (3 - Apply)

Personal competence
Students will be able to implement their own procedures and defend them against competing approaches. (6 - Create)

Social Competence
Programming exercises take place as part of the course. Students are thus able to understand, criticize and complement programs of other students. (5 - Evaluate)

Applicability in this and other Programs

Among others:

- AI Project
- Deep Learning/Big Data

Entrance Requirements

AIN-B-1 Mathematics 1
AIN-B-8 Programming 2
AIN-B-9 Algorithms and Data Structures

Learning Content

- Introduction: applications, computational models for vision, perception and prior knowledge, levels of vision, how people see
- Pixels and filters: digital cameras, image representations, noise, filters, edge detection
- Image regions: Segmentation, perceptual grouping, Gestalt theory, segmentation approaches, image compression
- Feature recognition: RANSAC, Hough transform, Harris corner detector
- Object recognition: challenges, template matching, histograms, machine learning
- Convolutional neural networks: neural networks, error functions and optimization, backpropagation, convolutions and pooling, hyperparameters, AutoML, efficient training, selected architectures
- Image sequence processing: motion, image sequence tracking, Kalman filters, correspondence problem, optical flow
- Fundamentals of mobile robotics: robot motion, sensors, probabilistic robotics, particle filters, SLAM
- Outlook: 3D vision, generative adversarial networks, self-supervised learning

Teaching Methods

- Lectures
- Projects

Recommended Literature

- R. C. Gonzalez and R. Woods, "Digital Image Processing", Pearson, 4th edition, 2018.
- I. Goodfellow, Y. Bengio and A. Courville, "Deep Learning", MIT Press, 2016.

TECHNISCHE HOCHSCHULE DEGGENDORF

AIN-B-19 Natural Language Processing

Module code	AIN-B-19
Module coordination	Prof. Dr. Patrick Glauner
Course number and name	AIN-B-19 Natural Language Processing
Original study program	Bachelor Artificial Intelligence
Lecturers	Prof. Dr. Udo Garmann Prof. Dr. Patrick Glauner
Duration of the module	1 semester
Module frequency	Annually, summer semester
Semester periods per week (SWS)	4
ECTS	5
Workload	Time of attendance: 60 hours self-study: 90 hours Total: 150 hours
Type of Examination	Exercise Performance, Written ex. 90 min. (specialized exam for M-AID students)
Duration of Examination	90 min.
Language of Instruction	English

Module Objective

The goal of this module is to learn Natural Language Processing (NLP), which allows computers to process human speech. We deal with NLP dozens of times every day, such as performing a Google search, spelling correction on a smartphone, classifying emails as spam, or recognizing handwriting. Modern NLP algorithms rely heavily on machine learning methods. Students acquire knowledge in NLP and can deepen this knowledge in the future, e.g. in projects or further studies.
Students know terms from linguistics such as syntax, semantics, etc. They understand the different structures of language. They can understand and use regular expressions (analysis and application) in Python. The students know the Natural Language Toolkit (NLTK). They will be able to apply the NLTK to various forms of language processing.

Specifically, students will have achieved the following learning outcomes upon completion of the module:

Subject Competency
Students will understand the concepts of the most common approaches to language processing. (2 - Understanding)

Methodological competence
Students will have the ability to create high quality programs using speech understanding technologies. (3 - Apply)

Personal Competence
Students will be able to implement their own procedures and defend them against competing approaches. (6 - Create)

Social Competence
Programming exercises take place as part of the course. Students are thus able to understand, criticize and complement programs of other students. (5 - Evaluate)

Applicability in this and other Programs

Among others:
- AI Project
- Deep Learning/Big Data

Entrance Requirements

AIN-B-1 Mathematics 1
AIN-B-8 Programming 2
AIN-B-9 Algorithms and Data Structures

Learning Content

- Basics: Stemming, stopwords, n-grams
- Text classification: Naive Bayes, spam filtering, speech recognition, logistic regression
- Spelling correction
- Search engines: ranking, vector space model, PageRank
- Basics of formal languages
- Regular expressions and finite automata
- Context-free grammars
- Analysis of speech signals
- Outlook: Embeddings, recent advances in NLP

Teaching Methods

- Lectures
- Discussion of scientific articles and current news

Exercises including computer exercises

Recommended Literature

- S. Bird, E. Klein and E. Loper, "**Natural Language Processing with Python – Analyzing Text with the Natural Language Toolkit**", Online at [NLTK website](https://www.nltk.org/book), visited 20/03/31.
- C. Bishop, "**Pattern Recognition and Machine Learning**", Springer, 2006.
- D. Jurafsky, "**Speech and Language Processing, An Introduction to Natural Language Processing**", Computational Linguistics, and Speech Recognition, Third Edition draft, available online at [Jurafsky:Homepage] (https://web.stanford.edu/~jurafsky), visited 20/03/31.
- C. Manning, P. Raghavan and H. Sch•ütze, "**Introduction to Information Retrieval**", Cambridge University Press, 2008.
- B. Pfister und T. Kaufmann, "**Sprachverarbeitung, Grundlagen und Methoden der Sprachsynthese und Spracherkennung**", 2., aktualisierte und erweiterte Auflage, Springer-Verlag GmbH Deutschland 2017, ISBN 978-3-662-52837-2.

- S. Russel and P. Norvig, "**Artificial Intelligence: A Modern Approach**", Prentice Hall, third edition, 2009

AIN-B-20 Human Factors and Human-Machine Interaction

Module code	AIN-B-20
Module coordination	Prof. Dr. Armin Eichinger
Course number and name	AIN-B-20 Human Factors and Human-Machine Interaction
Original study program	Bachelor Artificial Intelligence
Lecturers	Prof. Dr. Armin Eichinger
Duration of the module	1 semester
Module frequency	Annually, summer semester
Semester periods per week (SWS)	4
ECTS	5
Workload	Time of attendance: 60 hours self-study: 90 hours Total: 150 hours
Type of Examination	Course assignment Written ex. 90 min. (specialized exam for M-AID students)
Duration of Examination	90 min.
Language of Instruction	English

Module Objective

Upon completion of the module, students will have achieved the following learning objectives:

Subject Competency:
- Application of human factors fundamentals to the content domain.
- Identification of diverse influences and determinants on the quality of work and interaction

Methodological competence
- Knowledge of diverse methodological approaches to the investigation and evaluation of human-machine interaction
- Systematic analysis and classification of situational influences
- Systematic analysis of error sources and types

Personal competence:
- Realistic assessment of systemic influences on the work situation in the medical environment
- Improvement of team skills through knowledge of group mechanisms

Applicability in this and other Programs

All courses of study in which the interaction of artifacts with human operators is the central subject.

Entrance Requirements
Sufficient statistical and methodological skills typically developed in two one-semester statistics/stochastics courses each.

Learning Content
Introduction to the field of human-machine interaction
- Design of everyday objects
- Cognitive basics
- Phenomena and mechanisms of attention

Information design
- Display of information
- Principles of display design

Usability, UX
- Concepts, models, process
- Analysis: Methods
- Evaluation: Methods

Decision ergonomics
- Phenomena and Mechanisms
- Applications and design

Teaching Methods
Lecture, seminar parts, exercises, group work

Recommended Literature

- Ariely, D. (2009), Predictably Irrational, Harper, New York
- DIN EN ISO 9241-11 (1998). Anforderungen an die Gebrauchstauglichkeit.
- DIN EN ISO 9241-210 (2010). Prozess zur Gestaltung gebrauchstauglicher interaktiver Systeme.
- Kahneman, D. (2012), Schnelles Denken, langsames Denken, Siedler, München
- Heinecke, A. M. (2011), Mensch-Computer-Interaktion, Springer Berlin, Berlin
- Krug, S. (2009), Rocket Surgery Made Easy: The Do-It-Yourself Guide t- Finding and Fixing Usability Problems, 1 edition, New Riders, Berkeley, CA
- Krug, S. (2013), Don't Make Me Think: A Common Sense Approach t- Web Usability, 3rd revised edition, New Riders
- Norman, D. A. (1993), Things that make us smart: defending human attributes in the age of the machine, Addison-Wesley Publishing Company, Basic Books, Massachusetts [etc.]; New York
- Norman, D. A. (2013), The design of everyday things, Basic Books, New York, NY
- Pruitt, J., & Adlin, T. (2006), The persona lifecycle keeping people in mind throughout product design, Elsevier: Morgan Kaufmann Publishers, an imprint of Elsevier, Amsterdam, Boston
- Pruitt, J., & Adlin, T. (2010), The essential persona lifecycle your guide t- building and using personas, Morgan Kaufmann, Elsevier Science [distributor], San Francisco, Calif, Oxford

- Richter, M., & Flückiger, M. D. (2013), Usability Engineering kompakt benutzbare Produkte gezielt entwickeln, Springer Vieweg, Berlin
- Sarodnick, F., & Brau, H. (2010), Methoden der Usability Evaluation: Wissenschaftliche Grundlagen und praktische Anwendung. Verlag Hans Huber, Bern
- Shneiderman, B., & Plaisant, C. (2010), Designing the user interface: strategies for effective human-computer interaction, Addison-Wesley, Boston
- Stapelkamp, T. (2010a), Informationsvisualisierung: Web - Print - Signaletik. Erfolgreiches Informationsdesign: Leitsysteme, Wissensvermittlung und Informationsarchitektur, Springer Berlin, Berlin
- Stapelkamp, T. (2010b), Interaction- und Interfacedesign: Web-, Game-, Produkt- und Servicedesign; Usability und Interface als Corporate Identity, Springer, Heidelberg
- Thaler, R., Sunstein, C. (2009), Nudge. Improving Decisions About Health, Wealth, and Happiness, Penguin, New York, London
- Tufte, E. R. (2001), The Visual Display of Quantitative Information, 2nd edition, Graphics Pr.
- Tufte, E. R. (2006), Beautiful evidence, Graphics Press, Cheshire, Conn.
- Tufte, E. R. (2010), Visual explanations: images and quantities, evidence and narrative, Graphics Press, Cheshire, Conn.
- Tufte, E. R. (2011), Envisioning information, Graphics Press, Cheshire, Conn.
- Ware, C. (2008), Visual thinking for design. Burlington, Morgan Kaufmann, MA
- Ware, C. (2013). Information visualization: perception for design, 3rd revised edition, Morgan Kaufmann
- Wickens, C. D., Hollands, J. G., Parasuraman, R. (2013). Engineering Psychology and Human Performance, Pearson Education, Upper Saddle River

MAI-FWP Datacenter Network Programming

Module code	MAI-FWP
Module coordination	Prof. Dr. Andreas Kassler
Course number and name	MAI-FWP Datacenter Network Programming
Original study program	Master Applied Computer Science
Lecturers	Prof. Dr. Andreas Kassler
Duration of the module	1 semester
Module frequency	Annually, summer semester
Semester periods per week (SWS)	4
ECTS	5
Workload	Time of attendance: 60 hours self-study: 90 hours Total: 150 hours
Type of Examination	Written ex. 90 min. (specialized exam for M-AID students)
Duration of Examination	90 min.
Language of Instruction	English

Module Objective

Students acquire understanding and hands-on experience of how the data plane of modern datacenter networking equipment can be programmed using the high-level and popular programming language P4 (see http://p4.org). They learn the basic concepts of the P4 language and understand, how offloading simple computational tasks to the data plane of programmable networking devices (such as datacenter routers or network cards) can be used to speed up the performance of Deep Learning, Big Data Analytics use-cases within modern datacenters. They understand, how the data plane can be used to accelerate distributed high-performance computing (HPC) building blocks including distributed key-value stores, where load-balancing and network monitoring of the datacenter networking fabric is important for achieving high speed and low latency. They setup their own development environment in the network emulator Mininet and implement simple data plane programs in the P4 language. They know how to use P4 to parse packet headers, apply different actions and modify packets before forwarding them. They know basic P4 constructs, how to store stateful information (e.g. parts of a neural network) and how to perform simple computational tasks in the data plane. Based on this knowledge and understanding, students implement a small-scale project in a team. They use their acquired knowledge on P4 and programmable datacenter networking. They evaluate the results of other project groups and get evaluated by other groups. For this project work, they have used standard tools (Mininet, P4 toolchain, command line interface) for programming the data plane of an (emulated) datacenter router.
After finishing this module, students can design, implement and evaluate their own P4 programs using the network emulator Mininet.

Specifically, students will have achieved the following outcomes upon completion of the module:

Subject competency

Students understand the significance of datacenter network programming and how datacenter network programming can be used to improve the performance of distributed high-performance applications such as distributed training and inference of large-scale ML models.

Methodological competency

Students select the most appropriate P4 constructs for solving a concrete practical problem and use it to implement a concrete use-case of a data plane program.

Personal competency

Students understand complex theoretical concepts and apply them to problems arising in practice.

Social competency

Students communicate clearly, argue and criticize logically and constructively, contribute to reasoned, team-oriented problem-solving processes in the group.

Applicability in this and other Programs

This Module is suitable for the following programs:

- Master in Angewandte Informatik/Infotronik
- Master in Artificial Intelligence and Data Science
- Master in High Performance Computing/Quantum Computing

Entrance Requirements

Students should have basic understanding of Network Technologies and/or Communication Networks. Basic knowledge of Programming and basic knowledge in Python helps in the Project Part of the course.

Learning Content

The Module is decomposed into two parts:
Part I: “Introduction to Datacenter Network Programming” and Part II “Project in Datacenter Network Programming”

Content Part I:
(1) Introduction to Programming the Data Plane of a Datacenter networking device:

- Difference between Data and Control Plane
- Introduction to P4 language
- P4 programming model
- Compiling and deploying P4 programs
- P4 Targets: Behavioral Model (BMv2), Programmable Switching ASIC Intel Tofino, Mellanox Bluefield DPU, Netronome SmartNIC
- Basic P4 concepts: header parsing, applying tables and actions, header rewriting.
- Workshop: Setup Development environment with Mininet and Command Line Interface (CLI), implement, test and debug simple P4 language constructs and programs using the Mininet network emulator

(2) Datacenter Networking and Load Balancing:

- Datacenter networking fundamentals, routing and forwarding within the datacenter networking fabric
- Workshop: Advanced P4 concepts: stateful information, register arrays, counters and meters.
- Loadbalancing in Datacenter networks, Equal Cost Multipath Routing, Conga, Hula
- Workshop: Implementing ECMP in P4

(3) In Network support for Monitoring and Caching:
- Active and passive network monitoring
- Inband Network Telemetry (INT) for fine-granular network monitoring
- Accelerating Distributed Key-value stores in the data plane of the data center
- Using telemetry for fine-grained loadbalancing
- Workshop: Implementing Hula and INT in P4

(4) In Network support for Distributed Machine Learning:
- Role of the datacenter network for distributed training and inference
- In network support for Distributed Machine Learning Inference for in-switch traffic classification
- Mapping trained machine learning models (decision trees, SVMs, neural networks) to programmable data plane devices
- In network support for distributed training within a datacenter network

Content Part II:
Project: Implementation of your own small dataplane program in P4 and testing it in the Mininet network emulator.

Teaching Methods
- Interactive lectures
- Practical workshop style exercises using the network emulator Mininet
- Software implementation of your own P4 program

Remarks
The module is comprised of two parts. The second part (project work) can be done in groups of max. 3 students.

Recommended Literature
Recommended Literature will be provided at the start of the course by a set of research and practical oriented articles that are available online.

LSI-12 Data Visualisation

Module code	LSI-12
Module coordination	Prof. Dr. Phillipp Torkler
Course number and name	LSI-12 Data Visualization
Original study program	Master Life Science Informatics
Lecturers	Prof. Dr. Phillipp Torkler
Duration of the module	1 semester
Module frequency	Annually, summer semester
Semester periods per week (SWS)	4
ECTS	5
Workload	Time of attendance: 60 hours self-study: 45 hours virtual learning: 45 hours Total: 150 hours
Type of Examination	Written student research project
Language of Instruction	English

Module Objective

Data Visualization is the graphic representation of a data analysis to achieve clear and effective communication of results and insights. Complex ideas are presented in charts and graphs with the goal of quickly and easily disseminating key, actionable information. Data visualization is an essential part of data science and analytics, especially when working with large, complicated data sets like sequencing data. The visualization tells a story, whether as a stand-along graph or combined with other graphs, charts and design elements in an infographic or dashboard.
After completing the Data Visualization module, students will have obtained the following learning competencies:

Professional competence

After successfully completing the module, students will:
- know the data visualization principles.
- be familiar with file formats and their usage in the different analysis approaches.
- know about common data analysis workflows and be able to interpret and visualize the achieved results.

Methodological competence

After successfully completing the module, students will:
- know how to use ggplot2 in R to create custom plots.
- be able to explore and check alignments with alignment viewers.
- be familiar with genome browsers.

Social competence

- Interdisciplinary and interpersonal collaboration when working together in small groups on developing R scripts for data analysis and data visualization.

- Working together with fellow-students in small groups on designing and developing biostatistical validation of biomedical datasets within R.

Applicability in this and other Programs

Master seminar, master thesis

Entrance Requirements

Recommended or advantageous:
Basic Knowledge in R
Module: LSI-04 Biostatistics I

Learning Content

1 R Packages for data visualization
2 Alignment Viewers
3 Genome Browsers
4 Open access visualization tools

Teaching Methods

Tutorial, practical exercises, application examples
The module consists of an interactive theoretical part with blended learning components. Within the tutorial the students use example NGS datasets to perform the biomedical data visualization. In the practical part of the tutorial the students should learn to find various visualization tools, possibilities and methods and discuss their advantages and disadvantages to represent statisitical significance.

Remarks

The iLearn teaching and learning platform provides students with additional literature references and learning material to prepare for the lectures.

Recommended Literature

Detailed lecture notes are available online for preparation and follow-up work
- The Biostars Handbook: Bioinformatics Data Analysis Guide; 2019; https://www.biostarhandbook.com/

HPC-M-04 Software Engineering

Module code	HPC-M-04
Module coordination	Prof. Dr. Helena Liebelt
Course number and name	HPC-M-04 Software Engineering
Original study program	Master High Performance Computing / Quantum Computing
Lecturers	Prof. Dr. Marcus Barkowsky Prof. Bernhard Zeller
Duration of the module	1 semester
Module frequency	Annually, summer semester
Semester periods per week (SWS)	4
ECTS	5
Workload	Time of attendance: 60 hours self-study: 90 hours Total: 150 hours
Type of Examination	Written ex. 90 min.
Duration of Examination	90 min.
Language of Instruction	English

Module Objective
The students know and understand important design patterns. They are able to apply these design patterns and are able to argue about advantages and disadvantages of a specific design.
They also have acquired an understanding of the practical uses in software engineering and are able to follow a structured approach towards a software design.

Applicability in this and other Programs
Software design and programming lectures

Entrance Requirements
None

Learning Content
Advanced methods of software engineering:
- Design patterns
- Applications in software development
- System design

Teaching Methods
Lecture with exercises

HPC-M-07 High Performance Computing/Quantum Computing Technology

Module code	HPC-M-07
Module coordination	Prof. Dr. Helena Liebelt
Course number and name	HPC-M-07 HPC/QC Technology
Original study program	Master High Performance Computing / Quantum Computing
Lecturers	Prof. Dr. Peter Faber Prof. Dr. Helena Liebelt
Duration of the module	1 semester
Module frequency	Annually, summer semester
Semester periods per week (SWS)	4
ECTS	5
Workload	Time of attendance: 60 hours self-study: 90 hours Total: 150 hours
Type of Examination	Written student project
Language of Instruction	English

Module Objective
Student are introduced to technological issues particular to HPC and/or QC systems. They know hardware technologies relevant to the area and collected experience in building / setting up systems in hands-on sessions. They are able to set-up (part of) a modern HPC system and install and configure such a system on a small scale.

Applicability in this and other Programs
Hardware / system design for complex modern computing systems

Entrance Requirements
None

Learning Content
The aim of this course is to discover the technological particularities of HPC and QC systems.
The module is divided into two parts, both covering theoretical as well as practical aspects including hands-on sessions:

- Hardware
  - Setting up a compute node
  - Rack technologies
  - Cooling aspects
- Software
  - Setting up an operating system

- Middleware
- Access and scheduling

Teaching Methods

Lecture with lab sessions / exercises

Recommended Literature

- Andrew S. Tanenbaum; Herbert Bos. Modern Operating Systems. Prentice Hall, 4th ed. 2014
- Evi Nemeth, Garth Snyder, Trent R. Hein et al. Unix and Linux System Administration Handbook. Addison-Wesley, 5th ed. 2018
- Christine Bresnahan, Richard Blum. Mastering Linux system administration. Wiley. 2021. https://ebookcentral.proquest.com/lib/th-deggendorf/detail.action?docID=6658986
- Further literature as indicated in the lecture