	Faculty of Engineering and Natural Sciences **Computer/Software Engineering Department**	**Syllabus**

COURSE INFORMATION

Academic Year / Semester		2024 – 2025 / Spring
Course Code, Name, Credit (T-A-C)		3000004003, 3017002043 Deep Learning (3-0-3)
Lecture and Lab Schedules	**Theoretical**	3 hours
	Lab	Some Flip-Learning class activities
Course Type		Elective
Prerequisite		• Machine Learning (Normally) (Introduction to Neural Networks) • Basic Linear Algebra and Calculus. (vector/matrix manipulations, partial derivatives, …) • Basic Probability and Statistics. (common distributions, bayes rule, mean/median/model, …) • Python and good programming skills • Basic Data Structures and Algorithms
Day/Time and Class		**Monday:** 14:00 – 16:50 / **C2-03**

ACADEMIC PERSONAL

Title / First and Last Name	Prof. Dr. Haluk Gümüşkaya
Room	A7-12 (on Mondays)
E-Mail Address / Web Page	**E-mail:** haluk.gumuskaya@atlas.edu.tr and haluk@gumuskaya.com **Course Web Page:** Google Classroom (be announced in class)
Telephone	0850 450 3439-1316 / 444 34 39 - 1316
Office Hours	TBA
Assistant	-

COURSE DESCRIPTION

This is an introductory course in Deep Learning. It makes a broad introduction to many concepts and algorithms, theory ve practical work in Deep Learning. The following selected topics are presented in the course:

Introduction to AI, Machine Learning, Deep Learning, and Generative AI, Neural Networks and Deep Learning, Improving Deep Neural Networks: Hyperparameter Tuning, Regularization and Optimization, Structuring DL Projects, DL Hardware and Software, Computational Graphs, TensorFlow, Introduction to Computer Vision, Convolutional Neural Networks (CNNs), CNN Components and Computation Analysis, CNN Architectures and Transfer Learning, Computer Vision Applications: Object Detection Algorithms, Face Recognition, Neural Style Transfer, Sequence Models: Recurrent Neural Networks (RNNs), Natural Language Processing with RNNs and Attention, Introduction to Generative AI Models, LLMs and ChatGPT and Others, Variational Autoencoders (VAEs) and Generative Adversarial Networks (GANs), Transformers and Diffusion Models, Deep Reinforcement Learning (DRL) Term Project.

	Faculty of Engineering and Natural Sciences **Computer/Software Engineering Department**	**Syllabus**

COURSE OBJECTIVES

Upon successful completion of the course, the student is expected to learn and apply:
(to be updated **in this semester**)

- Basic techniques, terminology, and mathematics of Neural Networks and Deep Neural Networks.
- Neural Network Programming using Python, Tensorflow and other libraries.
- Fundamental neural network architectures, feedforward networks, Convolutional Neural Networks (CNNs), Recurrent Neural Networks (RNNs).
- Applications of CNNs and RNNs: Introduction to Computer Vision and Natural Language Processing.
- CNN architectures: LeNet, AlexNet, ZFNet, VGGNet, Inception (GoogLeNet), ResNet, U-Net
- Object Detection (YOLO/SSD), Semantic Segmentation with U-Net, Face Recognition, Neural Style Transfer
- How to use and fine-tune pre-trained models for best results (*Transfer Learning*)
- Generative AI models: Variational Autoencoders (VAEs), Generative Adversarial Networks (GANs), Transformers and Diffusion Models
- Various neural network applications.
- How to appropriately build and train different models for AI applications.
- ….

COURSE TEXTBOOK AND OTHER MATERIALS

Main Textbooks and Resources	No single textbook and other resource are strictly followed. We will excessively use many references, online materials (video lectures, blog, posts, surveys, papers, etc.). I prepare my own lecture slides and other materials from several resources. The following books' some parts and Python Jupyter Notebook examples will be used to learn and develop practical Deep/Machine Learning applications: 1. *Dive into Deep Learning*, http://d2l.ai/index.html 2. *Understanding Deep Learning*, Simon J.D. Prince, MIT Press, Dec 5th 2023. https://udlbook.github.io/udlbook/ 3. *Hands-On Machine Learning with Scikit-Learn, Keras and TensorFlow* (2019), 5th edition, Aurélien Geron (with Python Jupyter Notebook examples) https://github.com/ageron/handson-ml 4. *Applications of Deep Neural Networks*, Jeff Heaton, https://arxiv.org/abs/2009.05673 5. *Deep Learning*, Ian Goodfellow, Yoshua Bengio, and Aaron Courville, MIT Press, 2016, http://www.deeplearningbook.org/ 6. ….
Tools	Python and PyCharm (Professional Edition) (Python IDE) Rapid Miner Studio

WEEKLY LECTURE SCHEDULE (tentative)

W/D	Topics Covered	Activities

	Faculty of Engineering and Natural Sciences **Computer/Software Engineering Department**	**Syllabus**

1 **10/Feb**	**01-Course Overview and Introduction** **01.1-Course Overview** Getting to Know Each Other, Why this Course? Course Description and Objectives, Requirements and Assumptions, Course Activities and Grading, Lectures and Activities Schedule, Course Project, Course Web Site, Course Materials and References, Policies and Regulations **01.2-Introduction to Deep Learning and Applications** History of Deep Learning, A Quick Tour of AI Applications, 2012-Now: Some Recent Successes, Convolutional Neural Networks (CNNs), Recurrent Neural Networks (RNNs), Restricted Boltzmann Machines (RBMs), Deep Belief Networks (DBNs), Autoencoders, Getting Started with Deep Learning, Deep Learning Frameworks (Weekly Activity: Reading Assignment)	**Getting Started: (Time 2-3 weeks):** 1. Reviewing / Mastering or *Learning* Python and its data science libraries and the course IDE: PyCharm 2. Go over and refresh your knowledge on: Basic Calculus, Linear Algebra and Statistics with some notebook examples: 1. Data Science-Notebooks-3d (Hands-on Machine Learning-5 jupyter notebooks) 2. *Dive into Deep Learning*: Read Chap 1, Chap 2 and go over Basic Math http://d2l.ai/chapter_preliminaries/index.html 3. *Mathematics for Machine Learning and Data Science Specialization*, Coursera **Review Fundamentals of Neural Networks: Representation and Training:** Training, Validation and Testing using TensorFlow and Keras, Learning Curves, Gradient Descent, … **Neural Networks-4 Videos!** **01.3-Introduction to AL, ML, DL and GenAI** AI in the News, What is Artificial Intelligence? What is Machine Learning? Machine Learning Knowledge Areas, Types of Learning, What is Data Mining? What is Deep Learning? What is Generative AI?
2 **17/Feb**	**02-Neural Networks and Deep Learning** **02.1-Neural Network Programming** Modeling with Neurons, Logistic Regression as a Neural Network, Python and Vectorization, Vectorizing Logistic Regression, Broadcasting in Python, A Note on Python numpy Vectors **02.2-Shallow Neural Network-One Hidden Layer Neural Networks** Neural Network Representation, Vectorizing Across Multiple Examples, Activation Functions, Gradient Descent for Neural Network	*HW1: Implementing a Neural Network from Scratch*
3 **24/Feb**	**02.3-Deep Neural Networks** Deep L-Layer Neural Network, Forward Propagation in a Deep Network, Getting your Matrix Dimensions Right, Why Deep Representations, Building Blocks of Deep Neural Networks, Forward and Backward Propagation, Parameters vs Hyperparameters, What does this have to do with the Brain	**02.4-Neural Networks-Examples** What is a Neural Network, Gradient Descent, Backpropagation , Mastering Multi-Layer Perceptrons (MLPs), Backpropagation: The Backbone of Neural Network Training, Computing Gradients using AutoGrad in Pytorch, Computing Gradients using AutoDiff in Tensorflow, Backpropagation using Computation Graphs,

		Understanding Forward and Backward Propagation with a Simple NN Example, How Deep Neural Networks Work? *HW2: Implementing a Neural Network using Vectorized Implementation and AutoGrad/AutoDiff Techniques*
4 **3/Mar**	**03-Improving Deep Neural Networks: Hyperparameter Tuning, Regularization and Optimization** **03.1-Setting up and Regularizing your Neural Network** Setting up your Machine Learning Application, Regularizing your Neural Network, Other Regularization Methods, Setting up Your Optimization Problem **03.2-Optimization Algorithms** **03.3-Hyperparameter Tuning, Batch Normalization, Multi-Class Classification, and Frameworks** Hyperparameter Tuning, Batch Normalization, Multi-Class Classification, Introduction to Programming Frameworks	
5 **10/Mar**	**04-Structuring Deep Learning Projects** Introduction to DL Strategy, Setting up Your Goal, Comparing to Human-Level Performance, Error Analysis, Mismatched Training and Devtest Set, Learning from Multiple Tasks, End-to-End Deep Learning, A Case Study: Flight Simulator **05-DL Hardware and Software, Computational Graphs, TensorFlow** Deep Learning Hardware, Deep Learning Software, Tensors, Introduction to TensorFlow Graphs, Static vs Dynamic Graphs, Dive Deeper into TensorFlow, TensorFlow Graphs, Keras: High-Level API, Saving and Restoring a Model and Using Callbacks, Using TensorBoard for Visualization, How to Use GPUs for DL Apps?, A Comparison of TensorFlow and PyTorch	
6 **17/Mar**	**06-Introduction to Computer Vision and Convolutional Neural Networks** **06.1-Introduction to Computer Vision** Computer Vision is Deep Learning, Deep Learning is Hard, Image Classification, Nearest-Neighbor	**06.3-Fundamentals of Image Processing: Pixels and Filters** Image Sampling and Quantization, Image Histograms, Images as Functions, Linear Systems (Filters), Convolution and Correlation

	Classifiers, Image Classification using K-NN Classifier, Linear Classifier **06.2-Convolutional Neural Networks** What Computers "See", Learning Visual Features, Feature Extraction and Convolution: A Case Study, Convolutional Neural Networks (CNNs), CNNs for Classification: ImageNet, An Architecture for Many Applications, Deep Learning for Computer Vision: Impact and Summary	
7 **24/Mar**	Midterm Exam Week	
8 **31/Mar**	Ramazan Bayramı (30-03 – 1-04-2025)	
9 **7 Apr**	**07-CNN Components and Computation Analysis** **07.1-CNN Components and Computation Analysis** Convolution Operation and Vertical Edge Detection, Padding, Strided Convolution, Convolution over Volumes, One Layer of a Convolutional Network, Pooling Layers, Fully Connected (FC) Layer, CNN Example, Why Convolutions	**07.2-Convolutional Neural Networks: Examples** Recap: Start with Simple Neural Network Examples, Convolution: Edge Detection Filter, Handwriting Numbers Recognition-MNIST, CNN-Low Level Implementation-MNIST, ConvNetJS CIFAR-10 Demo, How to Use ChatGPT to Generate CIFAR-100 Code, CNN-Tensorflow Tutorial-CIFAR-10, Classifying Images with CIFAR-10 Dataset, Classifying Flowers: Overfitting, Data Augmentation and Dropout
10 **14 Apr**	**08-CNN Architectures and Transfer Learning** **08.1-CNN Architectures** LeNet ImageNet, ILSVRC Competitions and CNN Architectures, AlexNet, ZFNet, VGGNet, Networks in Networks, Inception (GoogLeNet), ResNet (Residual Network), Comparing Complexities of CNN Architectures, New CNN Architectures and Summary **08.2-Transfer Learning** Transfer Learning, Using Pretrained Models from Keras, Pretrained Models for Transfer Learning	**08.3-CNN Architectures and Transfer Learning: Examples** LeNet, AlexNet, VGGNet, Network in Network, Inception (GoogLeNet), ResNet (Residual Network), Transfer Learning, Using Pretrained Models from Keras, Pretrained Models for Transfer Learning

11 **21 Apr**	**09-Computer Vision Applications** **09.1-Computer Vision Applications: Object Detection Algorithms** Data Augmentation, Object Localization, Object Detection, Bounding Box Predictions, Intersection Over Union, Non-max Suppression, Anchor Boxes, Region Proposals: R-CNN, YOLO Algorithm, Semantic Segmentation with U-Net, Transpose Convolutions, U-Net Architecture Intuition, U-Net Architecture **09.2-Computer Vision Applications: Semantic Segmentation** Introduction to Semantic Segmentation, Semantic Segmentation with U-Net, Transpose Convolutions, U-Net Architecture Intuition, U-Net Overview, U-Net Architecture, Example: Building a U-Net with TensorFlow and Keras	**09.3-Computer Vision Applications: Face Recognition, Neural Style Transfer** What is Face Recognition? One Shot Learning, Siamese Network, Triplet Loss, Face Verification and Binary Classification, What is Neural Style Transfer? What are Deep ConvNets Learning, Cost Function, Content Cost Function, Style Cost Function, 1D and 3D Generalizations **09.4-Computer Vision Applications: Neural Style Transfer** What is Neural Style Transfer?, What are Deep ConvNets Learning, Cost Function, Content Cost Function, Style Cost Function, 1D and 3D Generalizations
	10-Sequence Models: Recurrent Neural Networks (RNNs) **10.1-Recurrent Neural Networks** Sequence Modeling, Notation, Why not a Standard Network or other Solutions? A Recurrent Neural Network (RNN), Different Types of RNNs, Language Model and Sequence Generation, Sampling Novel Sequences, Backpropagation Through Time (BPTT), Gated Recurrent Unit (GRU), Long Short-Term Memory (LSTM) Networks, Bidirectional RNN and Deep RNNs, RNN Applications	**10.2-Recurrent Neural Networks (RNNs): Examples** Recurrent Neurons and Layers, Training RNNs, Forecasting a Time Series, Handling Long Sequences
	11-Natural Language Processing with RNNs and Attention **11.1-Introduction to Natural Language Processing** Some NLP Terms, Machine Learning with Text Data, Text Pre-processing, Text Vectorization, Word Vectors **11.2-NLP with RNNs and Attention Mechanisms** Generating Shakespearean Text Using a Character RNN, Sentimental Analysis, An Encoder-Decoder Network for Neural Machine Translation, Attention Mechanisms, Recent Innovations in Language Models	
	12-Transformers Transformer Network Intuition, Self-Attention, Multi-Head Attention, Transformer Network, Applications	

20 May **31 May**	Final Exams Week	
	Kurban Bayramı (6-6 – 9-6-2025)	
16 June	Final Makeup Exams Week	
	Introduction to Generative Adversarial Networks (GANs) and LLMs Introduction to GANs, Supervised vs Unsupervised Learning, Generative Modeling, Autoencoders, Variational Autoencoders (VAEs), Generative Adversarial Networks (GANs), Recent Advances: Generative AI and LLMs, ChatGPT, DeepSeek, …	
	Deep Reinforcement Learning (DRL) Introduction to Reinforcement Learning, Reinforcement Learning (RL): Key Concepts, Examples of Reinforcement Learning, Deep Reinforcement Learning Algorithms, Policy Gradient Methods, The Game of Go	

GRADING (tentative)

Practical Part (mostly team work): ~ %70 in Grading

- **%30: Homework (Ödev)**
  %20: HW Assignments and FL Activities
  %10: ADC (Attendance/Discussion/Contribution)
- **%30: Project (Proje)**

Theoretical Part: ~ %30 in Grading

- **%40: Final Exam (Yıl Sonu Sınavı)**

Final Exam: %75 Written Final Exam Grade + %25 Project Grade