Contents

1 Syllabus (Tentative)

Future updates and announcements will be posted on the course website https://summer.yzhu.io.

1. **Coding Preparation** (Day 1. Sec. 7)

   Learning Path:

   - To acquire basic Linux command-line skills, please complete a free course, or if you are already very familiar, work through the tutorial from Ubuntu (Sec. 7.4).
   - Learn to create an isolated Python virtual environment (Sec. 7.1).
   - Follow the git tutorial to practice git commands (Sec. 7.2, 7.4).
     (Optional) Install/configure git extension in the Python IDE and try out GitHub Desktop.

   Homework:

   - Work through the CS131 homework guideline to set up a Python virtual environment and try to use the jupyter notebook (locally in Anaconda or VS Code and online on Colab). Some python/numpy review notebooks in the syllabus are recommended for self-practice (optional). Besides, git clone the homework repository of CS131 (we will use that later), and install the required packages.

2. **Document Preparation** (Day 2. Sec. 7.5)

   Learning Path:

   - Prepare local LaTeX environment (VS Code w/ LaTeX, Texpad, etc.) on your PC/Mac.
   - Work through three parts of the free course on Overleaf.

   Homework:

   - Prepare your professional curriculum vitae (CV) using LaTeX and send it to your interested supervisor in an email. Please carefully follow the email etiquette and remember to check the grammar in the text (browser extension is handy!). You may find a researcher's CV full of research experiences and publications. Take it easy and replace these sections with your course project descriptions. There are many open-source CV templates on Overleaf and GitHub for references.

3. **Vision and Learning** (Day 3-4. Sec. 2. Instructors: Siyuan Huang, He Wang, Yang Liu)

   Learning Path:

   - Study the course materials of the first three CS131 lectures.
   - Watch some videos/demos of vision projects from our research group.

   Homework:

   - Complete the CS131 HW1 on your own (modification: TBA).

4. **Cognition** (Day 5-6. Instructor: Yixin Zhu)

   Learning Path:

   - Read P266–290 (top) of textbook *Psychology (3rd Edition)*, by Schacter, Gilbert, Wegner & Nock [ISBN: 9781464106033]. This will give you an intuitive example of the cognitive basis of modern machine learning algorithms. If you have trouble buying/accessing this book, you can find it here. (DO NOT distribute!)
   - Read one (or more) of the following (old) papers. You will find that modern, popular machine learning algorithms are first devised and published with a cognitive mind. This also illustrates that AI is (originally) a science, not alchemy.
     - Rumelhart, David E., Geoffrey E. Hinton, and Ronald J. Williams. "Learning representations by back-propagating errors." Nature 323.6088 (1986): 533-536.
     - Rosenblatt, Frank. "The perceptron: a probabilistic model for information storage and organization in the brain." Psychological Review 65.6 (1958): 386.
     - Sutton, Richard S., and Andrew G. Barto. "Toward a modern theory of adaptive networks: expectation and prediction." Psychological Review 88.2 (1981): 135.
     - Ackley, David H., Geoffrey E. Hinton, and Terrence J. Sejnowski. "A learning algorithm for Boltzmann machines." Cognitive Science 9.1 (1985): 147-169.
     - Elman, Jeffrey L. "Finding structure in time." Cognitive Science 14.2 (1990): 179-211.

- Read one (or both) of the following modern papers.
  - Lake, Brenden M., et al. "Building machines that learn and think like people." Behavioral and Brain Sciences 40 (2017).
  - Stahl, Aimee E., and Lisa Feigenson. "Observing the unexpected enhances infants' learning and exploration." Science 348.6230 (2015): 91-94. (Note that videos are in the supplementary material.)

Homework: Write up a report in English using LaTeX, including

- A brief summary of what you have learned from the textbook;
- What is the modern technique/algorithm that corresponds to the theory devised a few decades ago (one of the old papers you read)? What are the differences between the modern interpretations and the original ones?
- Based on one or both modern papers you have read, summarize authors' arguments or hypothesis, their methods to validate, and the conclusion. Of note, pay attention to the consistency among the hypothesis, the experimental design, and the conclusion.
- Write a short thesis on whether the next generation (or the modern) AI should have a biology basis. In particular, what do you think should be the principles of devising AI algorithms. There is no right or wrong answer to this question, but you should elaborate your ideas/thoughts with rigorous logic.
- Anything you would like to add.

5. **Language** (Day 7. Sec. 5. Instructors: Zilong Zheng, Wenjuan Han)

   Learning Path:

   - Study word vector chapter (the first two lectures) of CS224N. Study the rest on your own interest.
   - Read books and papers listed in Sec. 5.2.

   Homework:

   - Play with Jupyter notebook (Python 3) and finish Assignment 1 (you can finish this with your own laptop).

6. **Robotics** (Day 8-9. Sec. 4. Instructor: Hangxin Liu)

   Learning Path:

   - Install and setup ROS environment following the tutorial.
   - Understand URDF for a robot description.
   - Play with robot navigation using the Husky simulation.
   - (Optional) Play with robot manipulation using Kinova arm.

   Homework:

   - Complete ROS setup and launch ROS packages.
   - Build your own robot model using URDF and visualize that in RViz.
   - Demonstrate GMapping and exploration results.

   Submit a report containing screenshots that demonstrate the above tasks.

7. **Multi-Agent Systems** (Day 10. Sec. 6. Instructor: Xue Feng)

   Learning Path:

   - Study the first two SOE-YCS0002 lectures. Study the rest on your own interest.
   - Study the first COMPGI13 lecture. Study the rest on your own interest.
   - Read books and papers listed in Sec. 6.2 and Sec. 6.3, according to your interest.

   Homework:

   - Finish the tests of the first two SOE-YCS0002 lectures.

2 Computer Vision (CV)

2.1 Courses

1. CS131: Computer Vision: Foundations and Applications [Introductory]
   An introduction to fundamental principles and important applications of computer vision. Provided by Stanford.

2. CS231A: Computer Vision, From 3D Reconstruction to Recognition [Intermediate]
   Major topics: geometry and 3D understanding. Provided by Stanford.

3. 16-385: Computer Vision [Advanced]
   Major topics: image processing, detection and recognition, geometry-based and physics-based vision and video analysis. Provided by CMU.

4. CSE291-J00: Computer Vision [Advanced]
   A guide for paper reading, presentation preparation, project planning in the field of computer vision. Click the 'Schedule' link for details. Provided by UCSD.

2.2 Papers

Major Conferences: CVPR, ICCV, ECCV [CVF Open Access]. Major Journals: TPAMI, IJCV.
(One could find the official website of a specific conference by searching [CVPR/ICCV/ECCV + year].
e.g., `http://cvpr2021.thecvf.com/`. There are many interesting co-located workshops in various areas.
e.g., 3D Scene Understanding for Vision, Graphics, and Robotics, Vision Meets Cognition, etc.)
Github resources: a curated list of awesome vision (scene understanding) resources.

1. A Stochastic Grammar of Images
   Proposes a stochastic and context sensitive grammar of images, which is embodied in the And–Or graph representation.

2. *Dark, Beyond Deep: A Paradigm Shift to Cognitive AI with Humanlike Common Sense*
   A journal article published in Engineering 2020 (IF: 6.61). A "small data for big tasks" paradigm: a blueprint for the next generation of AI.

3. *Visual Commonsense Reasoning: Functionality, Physics, Causality, and Utility*
   Yixin's Ph.D. dissertation (2018). Empower a machine with capabilities to reason about the unobservable knowledge: functionality, physics, causality and utility.

4. *Human-like Holistic 3D Scene Understanding*
   Siyuan's Ph.D. dissertation (2021). Build an intelligent machine to imitate the human's capability in perception, interaction, learning, and reasoning for solving holistic tasks.

2.3 Books

1. *Vision: A Computational Investigation into the Human Representation and Processing of Visual Information* (David Marr)
   The Marr Prize, one of the most prestigious awards in computer vision, is named in honor of David Marr. David Marr–in this landmark book–proposes a general framework for understanding visual perception and considers fundamental questions about the brain and its functions.

2. *Computer Vision: Statistical Models for Marr's Paradigm* (Song-Chun Zhu and Ying Nian Wu)
   A bridge between Marr's theory of vision and the modern treatment of computer vision with statistical models. Covering the low-entropy (texton) regime and the high-entropy regime (textures). Both authors are Marr Prize winners.

3. *Computer Vision: Stochastic Grammars for Parsing Objects, Scenes, and Events* (Song-Chun Zhu)
   It presents stochastic grammars for parsing objects, scenes, and events using spatial, temporal, and causal and-or graphs, posing computer vision as a joint parsing problem. Covering the middle entropy regime.

4. *Computer Vision: A Modern Approach* (David Forsyth and Jean Ponce)
   Textbook for CS131, 231A recommended above.

5. *Computer Vision: Algorithms and Applications* (Richard Szeliski)
   Another updating popular textbook in computer vision. Related course slides could be found on the website. For senior-level undergraduates.

3 Machine Learning (ML)

3.1 Courses

1. Machine Learning Crash Course [Introductory]
   A 20-hour practical self-study guide in machine learning with video lectures, notes and exercises. Available in both English and Chinese. Provided by Google.

2. M231B: Machine Learning Methods [Intermediate]
   Introduction of mathematical tools for analysis of learning with neural networks and graphical models with latent variables. Click the link to access slides. Related course notes are attached in the notes/books section below. Provided by UCLA.

3. CS229: Machine Learning [Intermediate]
   A broad introduction to machine learning and statistical pattern recognition. Click the 'syllabus' link to obtain course notes. Provided by Stanford.

4. CS231n: Convolutional Neural Networks for Visual Recognition [Intermediate]
   A 10-week course in convolutional neural networks, with applications in vision. Provided by Stanford.

3.2 Papers

Major Conferences: NeurIPS, AAAI, ICML, ICLR, etc. (Some are listed in the Section 2.2: Vision Papers.)
Publications of Center for Vision, Cognition, Learning, and Autonomy (VCLA):
`http://vcla.stat.ucla.edu/publications.html`

1. Auto-Encoding Variational Bayes
   VAE: Variational Autoencoder.

2. Mastering the game of Go with deep neural networks and tree search
   AlphaGo Lee is a modification of Monte Carlo tree search (MCTS), adapted for playing the game of Go. Related introductions could be found in the Chapter 2 of Monte Carlo Methods (Adrian Barbu and Song-Chun Zhu).

3. Learning FRAME models using CNN filters
   The DeepFRAME model is the generalization of the FRAME model, inspired by the successes of deep convolutional neural network.

3.3 Notes/Books

1. A Note on Machine Learning Methods (Ying Nian Wu)
   Notes for M231B recommended above.

2. Statistical Learning Method (Hang Li)
   An intro-level textbook in machine learning. Related code implementation could be found on Github. Available in Chinese.

3. Pattern Recognition and Machine Learning (Christopher Bishop)
   A comprehensive introduction to the fields of pattern recognition and machine learning. For advanced undergraduates or first-year Ph.D. students.

4. Machine Learning (Zhihua Zhou)
   An intro-level textbook. Coverage is great. Explanation is intuitive.

4 Robotics

4.1 Setting Up ROS (Robot Operating System)

1. ROS Tutorial 1.1 and 1.2: http://wiki.ros.org/ROS/Tutorials
2. ROS Melodic Morenia for Ubuntu 18.04 or ROS Noetic Ninjemys for Ubuntu 20.04

4.2 Simulating A Robot

http://wiki.ros.org/urdf/Tutorials

1. Prerequisite
2. Learning URDF Step by Step

4.3 Simulating Navigation

http://www.clearpathrobotics.com/assets/guides/melodic/husky/SimulatingHusky.html

1. AMCL Demo
2. GMapping Demo
3. Frontier Exploration Demo

4.4 Simulating Manipulation

https://github.com/Kinovarobotics/ros_kortex

1. kortex_examples: https://github.com/Kinovarobotics/ros_kortex/blob/kinetic-devel/kortex_examples/readme.md
2. kortex_gazebo: https://github.com/Kinovarobotics/ros_kortex/blob/kinetic-devel/kortex_gazebo/readme.md

4.5 Papers

Major Conferences: ICRA, IROS, RSS, etc.
Some good review papers for different fields in Robotics:

1. Embodiment in Socially Interactive Robots
2. Integrated Task and Motion Planning
3. Recent Advances in Robot Learning from Demonstration

5 Natural Language Processing (NLP)

5.1 Courses

1. CS 124: From Languages to Information [Introduction]
   An introduction to basic machine learning algorithms on natural language processing. Provided by Stanford.

2. CS224n: Natural Language Processing with Deep Learning [Intermediate]
   An introduction to deep learning techniques on natural language processing. Provided by Stanford.

3. COMS E6998: Machine Learning for Natural Language Processing [Advanced]
   An advanced course in machine learning for natural language processing. The methods we will cover will be relevant to many NLP applications, for example machine translation, dialog systems, natural language parsing, and information extraction. Provided by Columbia University.

5.2 Reading Materials

5.2.1 Books

1. Speech and Language Processing (3rd ed. draft) (Dan Jurafsky and James H. Martin)
   An introduction to basic machine learning algorithms on natural language processing.

2. Artificial Intelligence: A Modern Approach (Russell, Stuart, and Peter Norvig)
   "The publication of this textbook was a major step forward, not only for the teaching of AI, but for the unified view of the field that this book introduces. Even for experts in the field, there are important insights in almost every chapter." — Prof. Thomas Dietterich, Oregon State

3. The Oxford Handbook of Computational Linguistics 2nd edition (Editor: Ruslan Mitkov)
   A perspective of computational linguistics on natural language processing.

5.2.2 Papers

Major Conferences: ACL, EMNLP, NAACL, EACL, etc. [Association for Computational Linguistics].
Major Journals: TACL, Computational Linguistics.
(One could easily find the official website of a specific conference by searching [ACL/EMNLP/NAACL + year].
Github resources: a list of NLP tasks and (resources) to track the progress in Natural Language Processing (NLP), including the datasets and the current state-of-the-art for the most common NLP tasks.

1. Kewei Tu, Maria Pavlovskaia, and Song-Chun Zhu, "Unsupervised Structure Learning of Stochastic And-Or Grammars". In *Advances in Neural Information Processing Systems 26 (NIPS 2013)*.

2. Wenjuan Han, Yong Jiang, and Kewei Tu, "Enhancing unsupervised generative dependency parser with contextual information". In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics (ACL 2019)*.

3. Zhang, Songyang, et al., "Video-aided Unsupervised Grammar Induction". In *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies (ACL 2021)*.

4. Yining Hong, Qing Li, Song-Chun Zhu, Siyuan Huang "VLGrammar: Grounded Grammar Induction of Vision and Language".

5.2.3 Tutorials & Slides

1. Language Modeling

2. Unsupervised Parsing Tutorial (EACL 2021)
   An introduction on what unsupervised parsing does and how it can be useful for and beyond syntactic parsing.

6 Multi-agent Systems (MAS)

6.1 Courses

1. COMPGI13: UCL Course on RL
   An introduction to the field of reinforcement learning, including the core challenges and approaches. Provided by University College London.

2. SOE-YCS0002: Game Theory
   An introduction to game theory and strategic thinking. Provided by Stanford.

6.2 Papers

Major Conferences: AAMAS, ACC, NIPS, ICML, ICLR, AAAI, etc.
Major Journals: Automatica, IEEE Transactions on Automatic Control, PLOS Computational Biology, Proceedings of the Royal Society B: Biological Sciences, Journal of The Royal Society Interface.
Github resources: a paper collection of Multi-Agent Reinforcement Learning

1. An Overview of Multi-Agent Reinforcement Learning from Game Theoretical Perspective
   This paper elaborates the game theoretical foundations of modern multi-agent reinforcement learning methods and summarises the recent advances

2. Multi-Agent Actor-Critic for Mixed Cooperative-Competitive Environments *This paper introduce the multiagent DDPG learning algorithm.*

3. Five Rules for the Evolution of Cooperation
   Cooperation, a decisive organizing principle of human society, cannot emerge naturally. This paper provides five mechanisms for the evolution of cooperation: kin selection, direct reciprocity, indirect reciprocity, network reciprocity, and group selection.

4. Simulating Dynamical Features of Escape Panic
   A model of pedestrian behaviour to investigate the mechanisms of (and preconditions for) panic and jamming by uncoordinated motion in crowds.

6.3 Books

1. Multiagent systems: Algorithmic, game-theoretic, and logical foundations (Yoav Shoham & Kevin Leyton-Brown)
   Multiagent systems combine multiple autonomous entities, each having diverging interests or different information. This overview of the field offers a computer science perspective, but also draws on ideas from game theory, economics, operations research, logic, philosophy and linguistics.

2. Reinforcement Learning: An Introduction (Richard S. Sutton & Andrew G. Barto)
   A really detailed book about the fundamentals of Reinforcement learning

3. Agent-Based Models (Nigel Gilbert)
   This short book explains what agent-based modeling is. It also warns of some dangers and describes typical ways of doing agent-based modeling. Finally, it offers a range of examples from many of the social sciences.

4. Games, Strategies, and Decision Making (Joseph E. Harrington, Jr.)
   This book introduces core concepts with a minimum of mathematics in order to give you insights into human behavior.

7 Coding and Document Preparation

7.1 Python Environment

1. Python tutorials: A Byte of Python (English), Novice Tutorial (Chinese)
2. Create isolated Python virtual environments (virtualenv): Why and How?, Brief tutorial
3. Install Python packages: Tutorial (Pip)

7.2 Integrated Development Environment (IDE)

1. Google Colaboratory (Colab)
   Free GPUs (V100, P100). A web-based deep learning environment with basic Python packages and TensorFlow already set up. A hosted Jupyter notebook service.
2. Visual Studio Code (VS Code)
   Free. A lightweight code editor with support for almost every programming language. Built-in Git. Reliable official support for Jupyter notebooks.
3. PyCharm
   Free community edition. Free pro edition with educational license. A complete Python IDE. With Git support.

7.3 Machine Learning Libraries

1. PyTorch
   - Installation: `https://pytorch.org/get-started/locally`
   - Quickstart: `https://pytorch.org/tutorials/beginner/basics/quickstart_tutorial.html`
   - Documentation: `https://pytorch.org/docs/stable/index.html`
   - Free course: `https://www.udacity.com/course/deep-learning-pytorch--ud188`
2. TensorFlow
   - Installation: `https://www.tensorflow.org/install`
   - Quickstart: `https://www.tensorflow.org/tutorials/quickstart/beginner`
   - Documentation: `https://www.tensorflow.org/api_docs/python/tf`
   - Keras: `https://keras.io/getting_started/intro_to_keras_for_researchers`
   - Free course: `https://www.udacity.com/course/intro-to-tensorflow-for-deep-learning--ud187`

7.4 Linux Basics

1. Linux command line basics: tutorial, free courses (Udacity, Coursera).
2. Useful tools in Linux: The Missing Semester of Your CS Education,
3. Git and GitHub: Tutorial, GitHub Desktop, GitHub Student Pack
4. Remote development (SSH): basic SSH commands, VS Code using SSH, SCP command (file transfer)

7.5 Document Preparation

1. Online LaTeX editor: Overleaf
   - LaTeX documentation, quickstart/guide and Overleaf guides: `https://www.overleaf.com/learn`
   - LaTeX templates: `https://www.overleaf.com/latex/templates`
   - LaTeX free course: `https://www.overleaf.com/learn/latex/Free_online_introduction_to_LaTeX_(part_1)`
2. Local LaTeX editors:
   - VS Code with LaTeX Workshop extension: Installation Guide (Almost real-time compilation with "Auto Save" turned on)

- A live typesetter (mainly for MacOS): Texpad

3. Email etiquette: Basics (slide), Format, Tips, and Structure (w/ video)
4. Grammar check: Grammarly