

An AI Course in Computational Engineering at UT Austin

Joe Stubbs, PhD
Sept, 2025

About the Speaker

- Lead, Cloud and Interactive Computing Group, TACC, UT Austin
 - 15 full-time staff plus students
- Faculty appointment, Computational Engineering
- Co-PI of NSF-funded ICICLE AI Institute
- Interests: Cloud computing and distributed systems, formal methods and machine learning



COE 379L: Software Design For Responsible Intelligent Systems

Course Overview

- Objective: Introduce students to scalable data analysis and machine learning.
- Focus on techniques for designing, implementing, validating and operating responsible intelligent systems – how do they know if their system will be reliable?
- Projects-based course with “real” software development at the core.
- All materials available online, e.g., <https://coe379l-sp25.readthedocs.io/en/latest/>

History

- Initially taught in Spring of 2024 with about 20 students
- In Spring 2025 we gave it a second time and it had 30 students plus ~110 on the waitlist.
- Currently teaching the third iteration (Fall 2025) with about 35 students.

Motivation for the Course

1. COE 332 (Software Design for Cloud and Distributed Systems) very popular, and many students often requested “more”.
2. With COE 332, the goal was partially self-fulfilling: what skills did students need to be successful in an undergraduate research experience project (the kinds that my grants had funds for).
3. With the ICICLE AI Institute we were doing more AI/ML work and had funding for students to work on those projects.
4. We wanted an excuse to learn more about certain topics ourselves! :)

COE 379L: Overview and Topics

Material

- Unit 1: Data Analysis and Data Preparation
- Unit 2: The Machine Learning Application Development Life Cycle and Classic Machine Learning Algorithms
- Unit 3: Neural Networks and Deep Learning
- Unit 4: Advanced Topics (Transformers, Benchmarks, RAG, TinyML, etc)

Software

- Unit 1: Numpy, Pandas, Matplotlib, Seaborn
- Unit 2: Scikit-learn
- Unit 3: Tensorflow Keras, Docker and HTTP servers (e.g., Flask)
- Unit 4: Transformers, LangChain

Projects

- Project 1: Breast Cancer classification (solo)
- Project 2: Housing Prices (solo)
- Project 3: Predicting Damage to Buildings in Images pairs)
- Project 4: Open-ended (pairs)

Interactive, Hands-on Lectures

- Every student has their own VM running in our cloud with access to required software
- Lectures consist of a series of presentations followed by a hands-on portion that reinforces the concept
- Students are given time throughout the lecture to work on these hands-on portions
- We aim to give students the opportunity to explore the topics with open-ended questions and gain confidence in using the tools

Unit 3: Advanced Techniques for Machine Learning

- Introduction to Deep Learning
- Hands-on Neural Networks
 - Step 1: Importing required libraries and data
 - Step 2: Image Pre-processing
 - Step 3: Data pre-processing on output column.
 - Step 4: Building a sequential model neural network
 - Step 5: Evaluate the Model's Performance on Test
- Project 2 Summary
 - Convolutional Neural Networks (CNNs)
 - Hands-on CNN
 - MLOps
- Unit 4: Recent Advancements

PROJECTS:

- Project 01 - 20 Points
- Project 02 - 20 Points
- Project 03 - 25 Points
- Project 04 - 35 Points

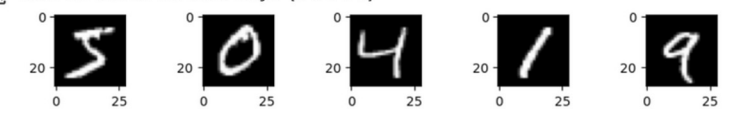
necessary, grayscale images can provide sufficient information for analysis.

Lets look at few sample images from the dataset along with their labels.

```
import matplotlib.pyplot as plt
plt.figure(figsize=(10, 1))
for i in range(5):
    # Set the (i+1)st subplot in a plot with 5 images in 1 row.
    plt.subplot(1, 5, i+1)
    plt.imshow(X_train[i], cmap="gray")
    print('label for each of the above image: %s' % (y_train[0:5]))
```

The first parameter of `subplot` represents the number of rows, the second represents the number of columns and the third represents the subplot index. Subplot indices start from 1, so `i+1` ensures that the subplot position starts from 1 and increases by 1 in each iteration.

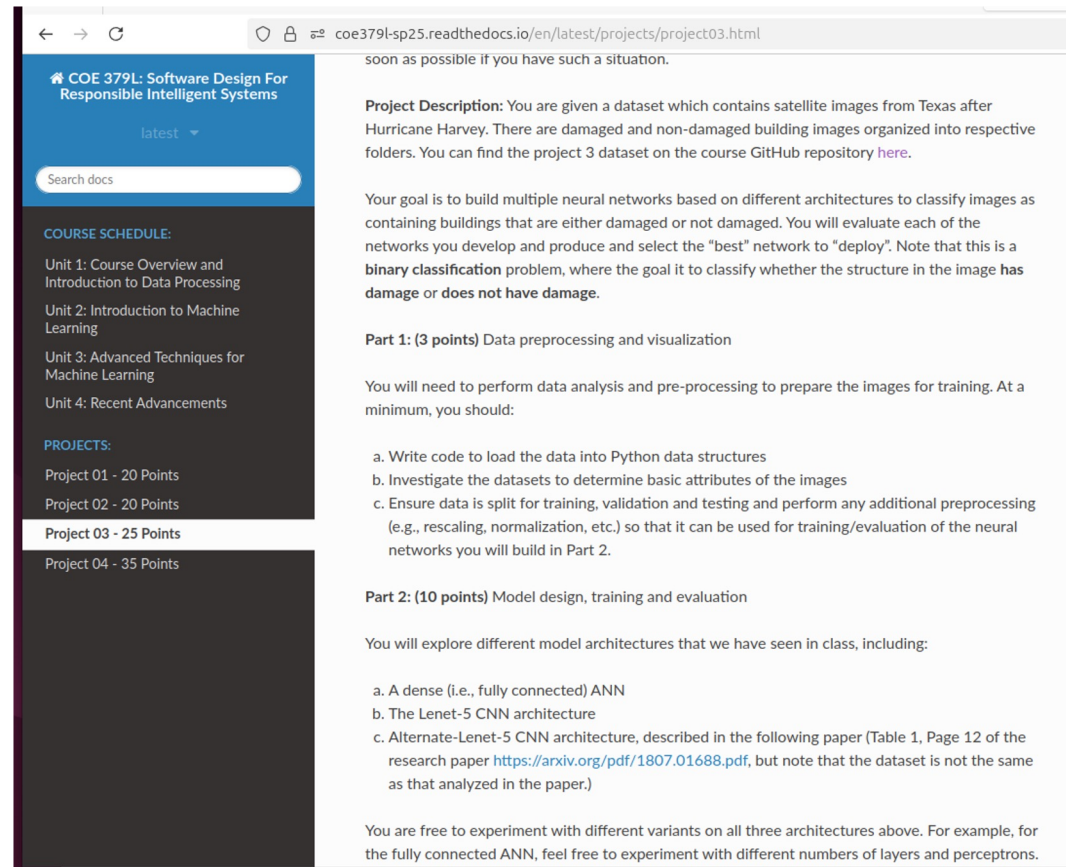
label for each of the above image: [5 0 4 1 9]



Each image has a total of $28 * 28 = 784$ pixels representing intensities between 0-255. Each of these pixel values is treated as an independent feature of the images. So the total number of input dimensions/features of the images is equal to 784. But the image provided to us is 2D array of size 28x28. We will have to reshape/flatten it to generate a 1D vector of size 784 so it can be fed to the very first dense layer of the neural network. We will use the `reshape` method to transform the array to desired dimension.

Project Assignments

- Analyze a real-world dataset using one or more machine learning methods.
- Do the necessary pre-processing, model design, training, and validation.
- In some cases, we have them read papers from the literature and ask them to incorporate ideas from the paper into their work.
- In some cases, we ask them to develop deployment artifacts (e.g., containers) so that code can be used
- Write a report describing the analysis they did, the choices they made, the results they obtained, etc.



The screenshot shows a web browser window with the address bar displaying `coe379l-sp25.readthedocs.io/en/latest/projects/project03.html`. The page is titled "COE 379L: Software Design For Responsible Intelligent Systems" and features a sidebar with a "latest" dropdown and a "Search docs" input field. The sidebar also contains a "COURSE SCHEDULE:" section with four units and a "PROJECTS:" section with four projects. The main content area is titled "Project 03 - 25 Points" and contains the following text:

soon as possible if you have such a situation.

Project Description: You are given a dataset which contains satellite images from Texas after Hurricane Harvey. There are damaged and non-damaged building images organized into respective folders. You can find the project 3 dataset on the course GitHub repository [here](#).

Your goal is to build multiple neural networks based on different architectures to classify images as containing buildings that are either damaged or not damaged. You will evaluate each of the networks you develop and produce and select the "best" network to "deploy". Note that this is a **binary classification** problem, where the goal is to classify whether the structure in the image **has damage** or **does not have damage**.

Part 1: (3 points) Data preprocessing and visualization

You will need to perform data analysis and pre-processing to prepare the images for training. At a minimum, you should:

- Write code to load the data into Python data structures
- Investigate the datasets to determine basic attributes of the images
- Ensure data is split for training, validation and testing and perform any additional preprocessing (e.g., rescaling, normalization, etc.) so that it can be used for training/evaluation of the neural networks you will build in Part 2.

Part 2: (10 points) Model design, training and evaluation

You will explore different model architectures that we have seen in class, including:

- A dense (i.e., fully connected) ANN
- The Lenet-5 CNN architecture
- Alternate-Lenet-5 CNN architecture, described in the following paper (Table 1, Page 12 of the research paper <https://arxiv.org/pdf/1807.01688.pdf>, but note that the dataset is not the same as that analyzed in the paper.)

You are free to experiment with different variants on all three architectures above. For example, for the fully connected ANN, feel free to experiment with different numbers of layers and perceptrons.

Use of AI

Goal

- Want students to learn to use AI tools **and** to learn the actual AI/ML topics.
- AI can be a great tool for *learning* when used correctly (e.g., AI tutors)
- AI can also be a great tool for *doing* development – but users must be able to recognize when the AI gives incorrect, misleading or otherwise suboptimal solutions.

In Practice

- Students will use the AI to solve an entire project – it's just too tempting.
- Students don't understand when the AI makes a “mistake” (or when the student tries to use an AI-generated solution out of context)
- Some students use of AI prevents them from learning the material.

AI Policy and Written Exam

Use of AI Policy:

- Every use of AI that leads to new code must be documented in a “Use of AI” file – like a bibliography that contains the tool, prompt used, and code generated.
- Within the code, must add (in a comment) a reference to the Use of AI document anywhere that code was generated.
- Do not use AI to generate the reports.

Exam:

- This semester we will have a written exam covering conceptual ideas from the course.
- The goal of the exam is to make students aware that we want them to learn the material, not just have AI do everything.