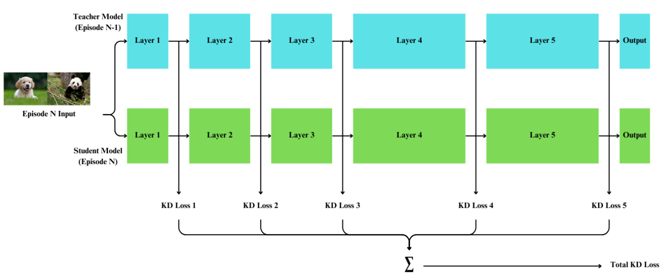
Seonghwan Lim

Hello, I’m Seonghwan Lim, a Master of Science in Computer Science student at the Illinois Institute of Technology. I am passionate about computer vision and AI, focusing on deep learning optimization and real-world problem-solving using efficient algorithms and clean, maintainable code. This portfolio showcases key projects such as the Logistic Smart Inspection System, Marionette Motion Control System, and Incremental Learning Enhancement, highlighting my experience in object detection, motion control, and model optimization. With a commitment to continuous learning and a results-driven approach, I aim to deliver innovative solutions that create real impact.

E-mail: slim24@hawk.iit.edu
LinkedIn: http://www.linkedin.com/in/slim-97-kr
GitHub: https://github.com/seonghwan97/portfolio

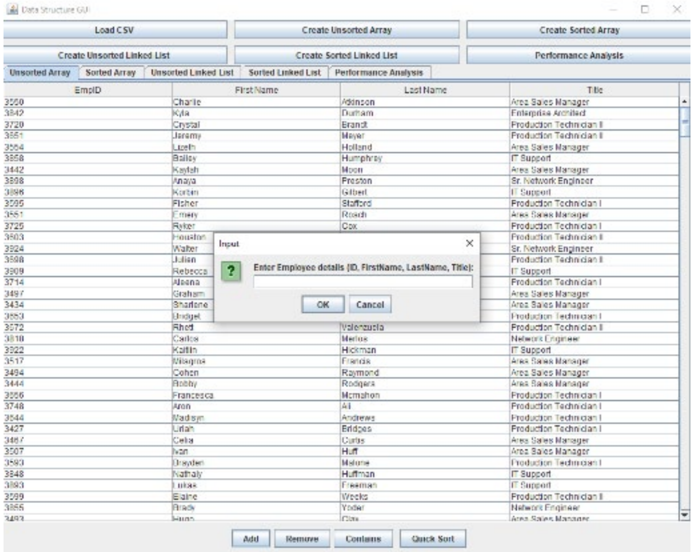
Projects



NICE with Knowledge Distillation

Improving Incremental Learning with Knowledge Distillation

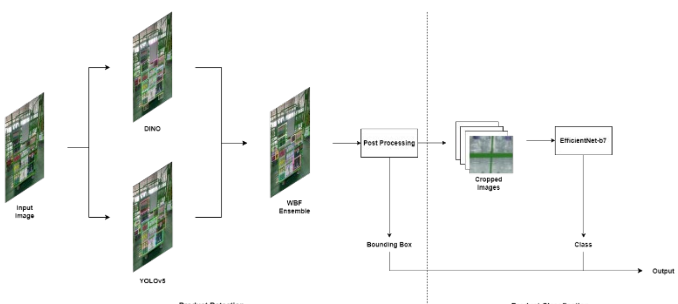
This project enhances NICE (Neurogenesis-Inspired Contextual Encoding) for class incremental learning by integrating Knowledge Distillation (KD). Using a teacher-student learning approach, the model learns from past episodes to reduce catastrophic forgetting. The teacher model from the previous episode provides "soft" predictions, which guide the student model for better learning of new tasks. This approach improves accuracy from 44% to 59% on MNIST and 39% to 49% on CIFAR-10, while also reducing misclassifications. Neuron aging is used to manage model capacity, with young neurons used for learning and mature neurons frozen for memory retention. Future enhancements include dynamic neuron allocation and advanced context detection for better adaptability in incremental learning tasks.



Data Structures Implementation Project

Building and Analyzing Data Structures for Efficient Data Management

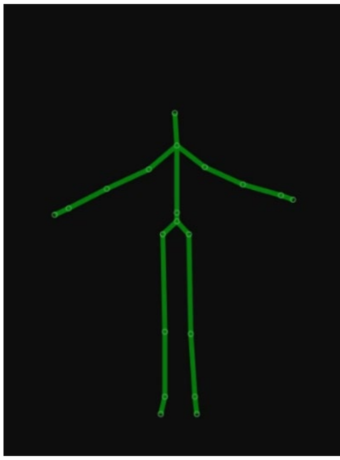
This project implements and analyzes five key data structures: Unsorted Array, Sorted Array, Unsorted Linked List, Sorted Linked List, and Binary Search Tree (BST). Each structure supports essential operations like add, remove, contains, and get. The project includes a Graphical User Interface (GUI) that allows users to load employee data from CSV files and visualize how different data structures handle operations. The system also provides performance analysis, showing the number of comparisons required to locate specific elements. Results highlight the efficiency of Sorted Arrays and BSTs, which perform binary search with O(log N) complexity, unlike linked lists that rely on linear search. The project emphasizes practical application of data structures in employee record management and showcases skills in Java programming, data visualization, and algorithm optimization.



CJ Logistics Future Technology Challenge

Delivery Box Object Detection

This project developed an object detection model for CJ Logistics, designed to automate the inspection of outgoing products in logistics centers. The system utilizes YOLOv5 and DINO models for object detection and EfficientNet for classification, enabling the recognition of items in a cart and counting their quantity. Key steps include bounding box detection, image augmentation, and classification. To improve detection accuracy, methods like TTA (Test Time Augmentation) and WBF (Weighted Boxes Fusion) were applied to merge bounding box predictions from YOLOv5 and DINO. Classification was enhanced with semi-supervised learning and weighted cross-entropy to handle class imbalance. The final system can identify and classify items from cart images, significantly reducing manual inspection time. The project demonstrates expertise in computer vision, deep learning, and Python development.



Marionette Motion Control System

Controlling Puppet Movements Using Kinect and Motor Coordination

This project develops a Marionette Motion Control System using Kinect v1 for motion tracking and motor-based string control to mimic human movements. The system extracts joint coordinates from the user's body via skeleton tracking and maps them to the corresponding puppet's movement. Key steps include coordinate extraction, data storage, and motor control. The Kinect tracks the shoulders, arms, legs, and spine to obtain X, Y, and Z position data, which is stored in a 2D array and exported as a .txt file. This data is transformed into motor control commands, adjusting string lengths to control the puppet’s joints. The result is a puppet that mimics human movements in real time. The project showcases skills in motion tracking, C/C++ programming, and robotics control systems. Future improvements include enhancing tracking accuracy and refining motor movement precision to achieve smoother puppet motions.