Artificial Intelligence and Expert Systems

Faculty of mechanical engineering

Instructors:
Dr.Esmaeil Najafi
Reza Behbahani Nejad

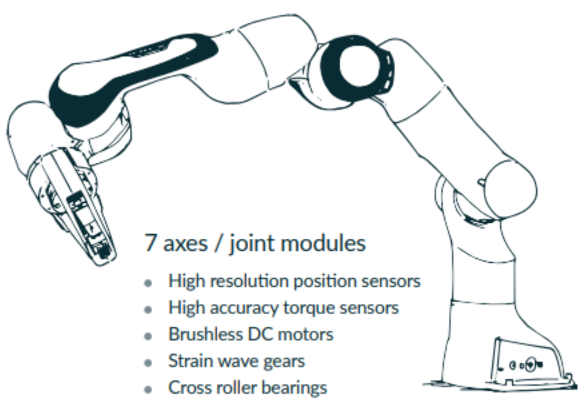
Artificial Neural Networks Project

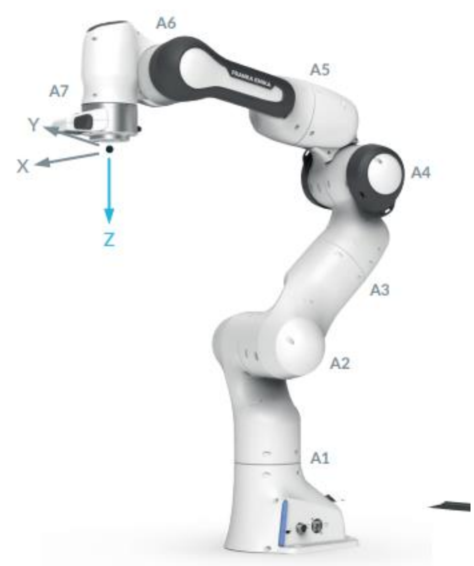
Due date: 1402/03/26

PHASE I: MLP and RNN

Determining the kinematic properties of a robotic manipulator is a crucial aspect of using such a device. It involves solving direct and inverse kinematic equations to enable further calculations. Direct kinematic equations facilitate the conversion from joint variable space to tool configuration space. They allow us to calculate the position of the tool in the workspace based on predefined joint rotation values. Conversely, inverse kinematic equations enable the transformation from the tool configuration space to the joint variable space. By knowing the desired position in the workspace, we can determine the joint values necessary to position the tool accordingly.

While the determination of the robotic manipulator direct kinematics is relatively straightforward, and there are methods such as Denavit–Hartenberg (D-H) that allow for the simple determination, determining inverse kinematics is a more complex process.2 Determining the inverse kinematic equations for complex robots has high algebraic complexity.3 Determining the solution numerically is an option, however, it takes a comparatively long time compared to using a direct solution, such as an equation.



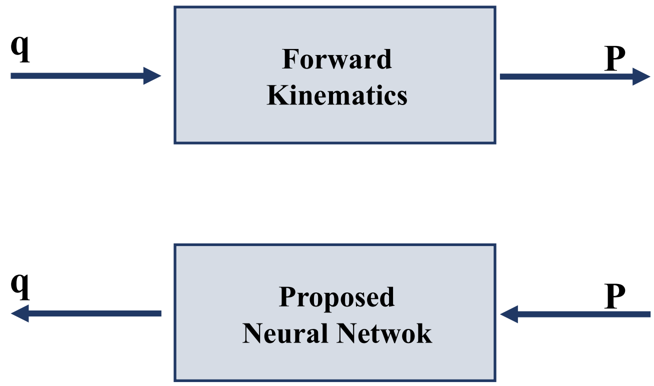


Panda UR5 Franka manipulator

The objective of this project phase is to estimate the inverse kinematics of a Panda UR5 Franka manipulator using the provided dataset. The dataset has been generated through system identification and the utilization of direct kinematic equations. Initially, input joint values (q) are provided to the direct kinematic equations, resulting in the calculation of the pose (p) of the robotic manipulator's end effector.

q represents $\boldsymbol{\theta_1, \theta_2, \theta_3, \ldots\ \theta_7}$

p represents x, y, z, $\boldsymbol{\varphi, \psi, \gamma}$



In this phase, you are tasked with developing two types of neural networks: a multilayer perceptron (MLP) network and a recurrent neural network (RNN) network. The architecture for both networks can be arbitrary, drawing from the knowledge gained during the course. These networks will be utilized to estimate the inverse kinematics of the robot.

To obtain the required data for this task, please access the dataset through the following link: Robot Data

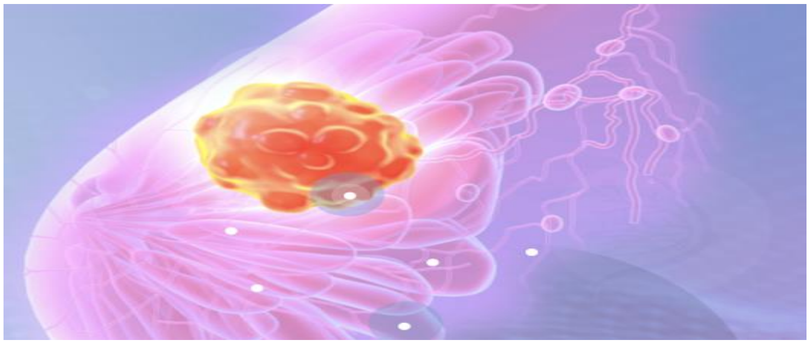
PHASE II: Image Classification

ext info:

According to the WHO, breast cancer is the most commonly occurring cancer worldwide. In 2020 alone, there were 2.3 million new breast cancer diagnoses and 685,000 deaths. Yet breast cancer mortality in high-income countries has dropped by 40% since the 1980s when health authorities implemented regular mammography screening in age groups considered at risk. Early detection and treatment are critical to reducing cancer fatalities, and machine learning algorithms could help streamline the process radiologists use to evaluate screening mammograms.

Currently, early detection of breast cancer requires the expertise of highly-trained human observers, making screening mammography programs expensive to conduct. A looming shortage of radiologists in several countries will likely worsen this problem. Mammography screening also leads to a high incidence of false positive results. This can result in unnecessary anxiety, inconvenient follow-up care, extra imaging tests, and sometimes a need for tissue sampling (often a needle biopsy).

The goal of this phase of the project is to identify cases of breast cancer in mammograms from screening exams. It is important to identify cases of cancer for obvious reasons, but false positives also have downsides for patients. As millions of women get mammograms each year, a useful machine-learning tool could help a great many people.



Breast cancer detection

In this part of the project, you should use your skills in image classification to solve this problem. You should train your image classifier on the dataset. All the design parameters including the type of the network, architecture, train parameters, evaluation metrics, batch size, optimizer, input resolution, etc. are arbitrarily and should be chosen based on your knowledge, so be careful about your design. The minimum requirement of this part is:

1. Training a CNN network with optimized parameters (layers, kernel size, …)

2. Train three networks with state-of-the-art architectures (VGG16, VGG19, ResNet50, Inceptionv3, MobileNetv2, Xception, …) as the feature extractors. Freeze all the layers in the networks and train your own classifier on top of them.

3. Train the best network from section 3 again but unfreeze last 2 layers.

4. Train the best network from section 3 again but unfreeze last 6 layers.

5. Train the best-performing network from sections 1-4 again, with and without data augmentation.

Don't forget that the discussion on choosing parameters and the output results (train and validation accuracy, train and validation loss, metrics, confusion matrix, …) is important to us!

RSNA Screening Mammography breast cancer detection dataset:

The dataset contains radiographic breast images of female subjects. The dataset has been published by the Radiological Society of North America (RSNA).

[train/test].csv Metadata for each patient and image. Only the first few rows of the test set are available for download.

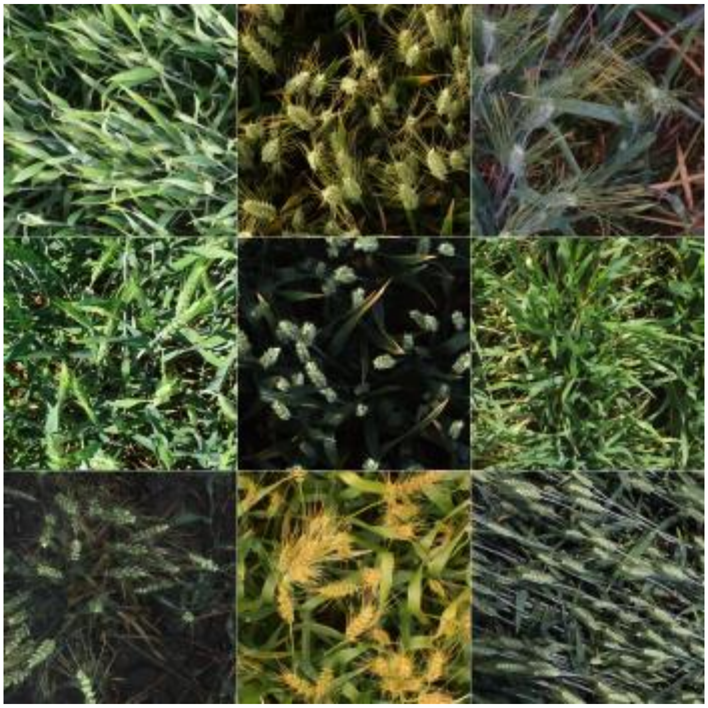
- site_id - ID code for the source hospital.
- patient_id - ID code for the patient.
- image_id - ID code for the image.
- laterality - Whether the image is of the left or right breast.
- view - The orientation of the image. The default for a screening exam is to capture two views per breast.
- age - The patient's age in years.
- implant - Whether or not the patient had breast implants. Site 1 only provides breast implant information at the patient level, not at the breast level.
- density - A rating for how dense the breast tissue is, with A being the least dense and D being the most dense. Extremely dense tissue can make diagnosis more difficult. Only provided for train.
- machine_id - An ID code for the imaging device.
- cancer - Whether or not the breast was positive for malignant cancer. The target value. Only provided for train.

- `biopsy` - Whether or not a follow-up biopsy was performed on the breast. Only provided for train.
- `invasive` - If the breast is positive for cancer, whether or not the cancer proved to be invasive. Only provided for train.
- `BIRADS` - 0 if the breast required follow-up, 1 if the breast was rated as negative for cancer, and 2 if the breast was rated as normal. Only provided for train.
- `prediction_id` - The ID for the matching submission row. Multiple images will share the same prediction ID. Test only.
- `difficult_negative_case` - True if the case was unusually difficult. Only provided for train.

The dataset can be obtained via following link: [RSNA_Breast_Cancer]

PHASE III: Object detection

In this phase, you will use your skills in object detection to train a network on the GWHD dataset. All the design parameters including type of the object detection network, backbone (feature extractor), input resolution, batch size, number of steps for training, etc. are arbitrarily and should be chosen based on your knowledge, so be careful about choosing each parameter. The minimum requirement of this part is training one object detection network and testing and visualizing the network on the test images. Don't forget that the discussion on choosing parameters and the output result is important to us!



Wheat heads

Global Wheat Head Detection dataset:

The dataset is composed of more than 6000 images of 1024x1024 pixels containing 300k+ unique wheat heads, with the corresponding bounding boxes. The images come from 11 countries. The task is to localize the wheat head contained in each image. The goal is to obtain a model which is robust to variation in shape, illumination, sensor and locations. A set of boxes coordinates is provided for each image.

Files:

images: the folder contains all images

train.csv, val.csv, test.csv: contains the splits and labels you can use during training

Metadata.csv: contains additional metadatas for each domain (Not relevant to training the network)

Labels: All boxes are contained in a csv with three columns image_name, BoxesString and domain.

image_name is the name of the image, without the suffix. All images have a .jpg extension.

BoxesString is a string containing all predicted boxes with the format [x_min,y_min, x_max,y_max].

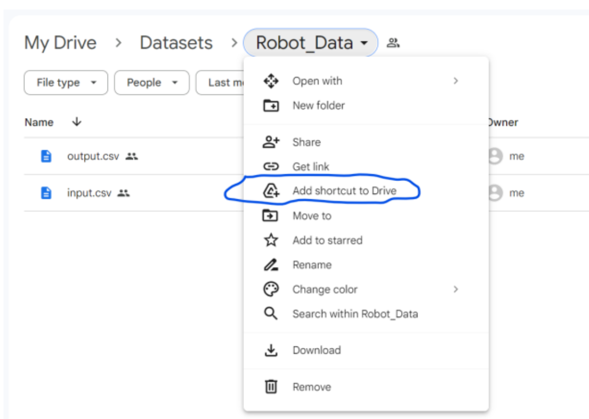
Domain give the domain for each image (The region in which the image was taken)

Important note: This dataset has only 1 class (wheat head!). The domain should not be regarded as class labels. All image bounding boxes are coordinates of wheat heads in the corresponding images. You are attempting to predict bounding boxes around each wheat head in images that have them. If there are no wheat heads, you must predict no bounding boxes.

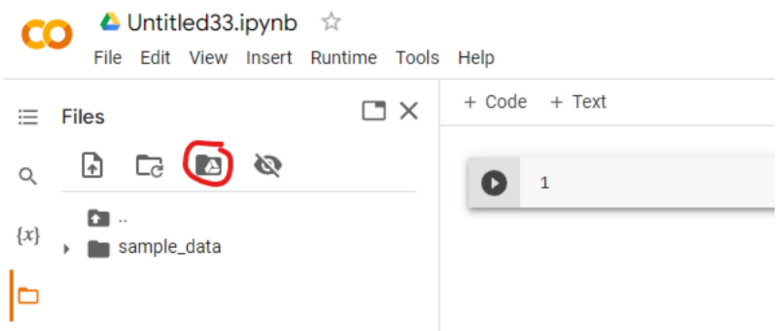
The dataset can be obtained via the following link: [Global Wheat Head]

Guide on importing the datasets to google Colab:

1. Open the dataset folder link in your browser (make sure to login in to your gmail account and have at least 7GB free storage space in your google drive beforehand):
2. Right-click on the Robot_Data tab and press "Add a shortcut to Drive"



3. Add a shortcut to the "My drive" folder
4. Open Google Colab and mount your google drive.



5. You should find the datasets in "/content/drive/MyDrive/". If the data contains zip files you should first unzip them using the command: "!unzip / (path of the file)"

Good Luck!