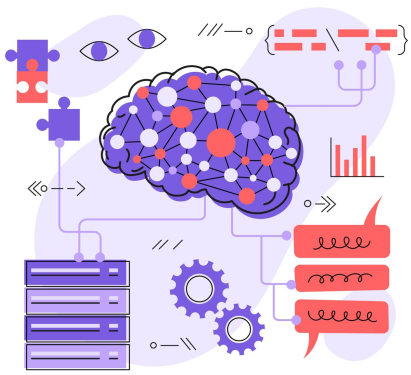
Mini-Project
Computer Vision and Deep Learning

Delivery Deadline:
Friday, May 2nd, 23:59PM
This project counts for **28%** of the final grade
You are allowed to work in groups of 1 or 2 people.

1 Introduction

This mini project is supposed to simulate a potential real-world scenario in some months, i.e., your first assignment, in your new ICT-consulting job, helping an external customer with some visual intelligence (VI) related tasks. The information provided is therefore limited to what you can find below. But you are free to solve the task the way you want, using whatever tools you find useful, just remember to report what you have done. Also, here you will be able to choose between the projects proposed below.



Group-sizes can be one or two participants.

It's possible to choose between three projects in the topics: AD (autonomous driving), MIC (medical image computing), football tracking / analysis. However, all projects deal with core visual intelligence (VI) tasks that can be applied in other domains as well.

1.1 Presentation Structure

The projects should be presented using the following structure: Background / Motivation, Approach / Strategy, Data-analysis, Methods/Models, Results, Discussion, Key learning points, References and who did what, if the group contains two participants.

1.2 Sustainability

Sustainability is supposed to be part of all courses now. The way we will do that in this module is as follows; you will make a note of the combined compute time needed to achieve your results (approximately, training mostly, but also inference). The compute-time can be converted to energy, and the needed energy could have alternatives uses, for example, how far, from Trondheim to Oslo, can you get in your brand new fully electric car, e.g., a Tesla Model Y (most sold car in Norway), using the same amount of energy.

1.3 Project Rules

To achieve a fair project (and grading..), we ask you to follow these rules:

1. The code should be developed by yourself, but you are free to use open-source repositories and libraries. If you take code from anywhere else, please attribute the original authors in your source code.

2. You are allowed to use any open-source model architecture from PyTorch or MONAI etc. You may also use pre-trained model weights, given that you fine-tune the model for your project.

1.4 Computing Resources

These projects will typically require a lot of computing power in the form of GPU's. If you don't have access to a modern Nvidia GPU, you are welcome to use one of the computing resources available to you via NTNU.

Option 1: Cybele Lab

You may use the Cybele Lab computers physically at the lab or via SSH. This option is a bit easier than option 2, but accessing the computers remotely is **not** allowed during school time (08:00 - 18:00 on weekdays) as other students in the course and other courses may use the computers in this time.

See: Cybele Guide

Option 2: IDUN Cluster

This cluster has a lot of computing power but requires a bit of learning in order to utilize it. We recommend the following guide to get you started: IDUN Guide

More information about the cluster can be found at their home page: https://www.hpc.ntnu.no/idun/

Please note that this cluster uses a scheduling system called Slurm, this means you request computing resources and are put into a queue. Typically, at the end of the semester there are many students / researchers using the cluster, so the waiting time can be quite long at times. If you choose to use this option, you should dedicate some time into learning how to use it, and conduct your experiments in good time before the project deadline to account for long queues.

2 Project options

In this section, the project options are described, which allows you to tailor your learning experience based on your interests. Each option will challenge you to develop specialized computer vision and deep learning expertise. **You should only select one of the following options!**

2.1 Option 1: Object Detection with LiDAR Data from Trondheim

Navigate the complexities of real-time perception for autonomous vehicles. In this option, you will work on a LiDAR and / or a natural image dataset collected by the NAPLab at NTNU.

There has been major advancements in Autonomous driving in recent years, attributed to computer vision (DL). However, such models are mostly focused on driving in ideal conditions, and thus struggles with challenging conditions like snowy roads. One way to localize the road in winter time is by relying on the location of snow poles, which are typically erected on either side of the road in areas prone to snow in the winter. Your task here is to perform real time object detection of snow poles, in order to further develop AD capabilities in winter conditions.

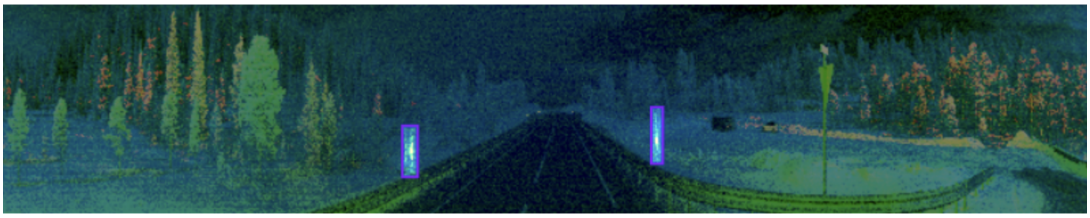


Figure 1: Sample with labels from the NAPLab-LiDAR Pole dataset

Figure 2: Sample with labels from the NAPLab Natural Image Pole dataset

- **Dataset:**

  There are two separate datasets for this task. The first dataset is a selection of natural images (RGB), and the second dataset consists of LiDAR images. The LiDAR images are combined as RGB images by combining Near-IR, Signal, and Reflectivity channels. Near-IR maps to blue, Signal to green, and Reflectivity to red.

  The labels for both datasets consists of 1 bounding box class (snow poles) in the YOLO format. The test set labels will be hidden, and we will create a leaderboard for you to compare your model to your peers. The leaderboard will be announced later.

  Both datasets can be found on IDUN and Cybele at these locations:

  - IDUN: */cluster/projects/vc/data/ad/open/Poles*
  - Cybele: */datasets/tdt4265/ad/Poles*

Please note that redistribution of the dataset is prohibited!

- **Performance Metrics:** We expect you to calculate the following metrics on the test set:
  - Precision
  - Recall
  - map@50
  - mAP@0.5:0.95
- **Model size:** As the model is ultimately supposed to run on an edge device for real time object detection, you should select a model that is suited for this task. Tiny or Small versions of YOLO could be a good fit in this regard.
- **Additional information:**
  - A Pole Detection and Geospatial Localization Framework using LiDAR-GNSS Data Fusion
  - SnowPole Detection: A comprehensive dataset for detection and localization using LiDAR imaging in Nordic winter conditions
- **Expectations based on group size**:
  - Groups of 1: Implement at least one architecture. Use at least one of the two datasets.
  - Groups of 2: Implement at least two architectures. Use both LiDAR and natural images, as a combined model or as separate models for each data type.

2.2 Option 2a: Medical Image Segmentation

Dive into the critical field of medical image segmentation by tackling the Head and Neck Tumor Segmentation for MR-Guided Applications (HNTS-MRG) Challenge.

Radiation therapy (RT) is essential in treating various cancers, especially head and neck cancer (HNC). MRI-guided RT planning, increasingly preferred over CT, offers better soft tissue contrast, functional imaging, and adaptive RT through MRI-Linac devices, improving tumor targeting while reducing side effects. However, the time-intensive process of manually segmenting HNC tumors in MRI data is challenging and impractical, prompting the exploration of AI-driven auto-segmentation methods.

The HNTS-MRG challenge addresses automated HNC tumor segmentation for MRI-guided adaptive RT. The challenge consists of two subtasks:

1. Tumor segmentation on preRT MRI images.
2. Tumor segmentation on midRT MRI images. Note: co-registered version of the preRT MRI images are also provided for this task, it is up to you to decide if you want to use these images.

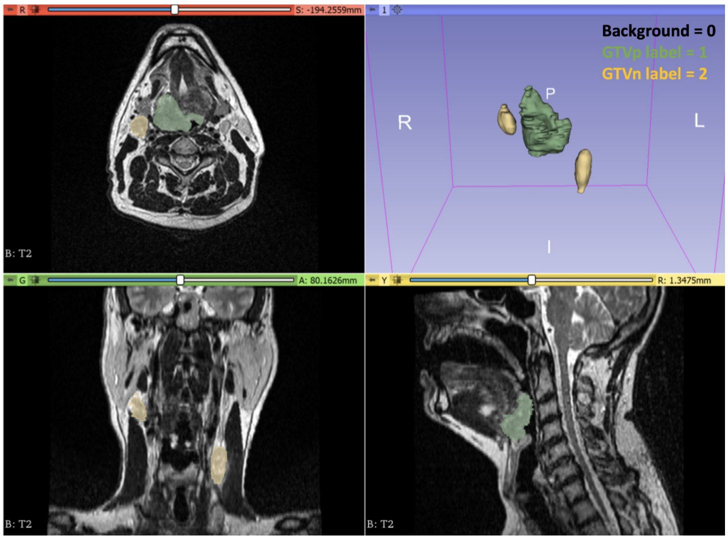


Figure 3: Illustration with labels from the HNTS-MRG grand challenge dataset

- **Dataset:** The dataset can be found on IDUN:
  - IDUN: */cluster/projects/vc/data/mic/open/HNTS-MRG*
  - Cybele: */datasets/tdt4265/mic/HNTS-MRG*

  Please note that redistribution of the dataset is prohibited!
- **Performance Metric:** The DICE coefficient is a common metrics in medical image segmentation and should be used to measure the performance of your model. DICE measures overlap between predicted and ground-truth segmentations.
- **Recommended tools:**
  - MONAI is a great python framework built on top of PyTorch with functionality tailored to working with medical images.
  - For visualizing the 3D images, and the predictions, as well as the two combined (e.g. the prediction as overlay to the raw data) we recommend 3D Slicer.

- **Expectations based on group size**:
  - Groups of 1: Train a DL model for subtask 1 (preRT).
  - Groups of 2: Train a DL model for both subtasks (preRT and postRT).
- **Additional information:**
  - HNTS-MRG 24 - Grand challenge

2.2.1 Option 2b: Medical Image Segmentation (Simple)

Note that this option can maximally lead to 80% of the max score.

We also provide a "backup" project this year (i.e., the project is a bit easier, and parts of the solution is already given).

Both a 2D dataset and a 3D dataset must be addressed. And you have to implement another model than what is given in the starter code.

link to github repo: MedMNIST

2.3 Option 3: Football analysis

Harness the power of deep learning to analyze football match footage, extracting valuable insights for RBK (or their opponents). The main tasks here will be object detection of the players / referee and ball, as well as tracking.

Figure 4: Sample with labels from the RBK_TDT17 dataset

- **Dataset:** The dataset consists of three 1-minute video sequences (1802 frames each) meticulously annotated with 2D bounding boxes for players, referees, and the ball. Each player and referee has a consistent tracking ID throughout the sequence, even when temporarily out of camera view (marked with a ”not visible” property). Two sequences are designated for training and validation (along with any pre-training you wish to employ), while the remaining sequence is strictly for final testing and evaluation. It's crucial to avoid using the testing data during training to ensure a fair performance assessment.
  - IDUN: */cluster/projects/vc/data/other/open/RBK_TDT17*
  - Cybele: */datasets/tdt4265/other/rbk*

  Please note that re-distribution of the dataset is prohibited!
- **Main Tasks:**
  - Object detection
  - Tracking
- **Bounding box detection / tracking and analysis:** Accurate player tracking lays the foundation for analyzing performance metrics like distance, velocity, and acceleration. However, to translate player positions on screen to real-world field measurements, you must first calibrate the camera view against the field's physical dimensions. This calibration hinges on detecting and tracking well-defined field keypoints (like line intersections) – a computer vision task known as keypoint estimation. To be useful, at least three keypoints must be visible in each frame. Your task involves designing a keypoint detection strategy, which includes labeling each frame's keypoints (as ”visible” or ”not visible”). Consider creating a small dataset to train a basic keypoint detection model for this purpose. You may use https://app.cvat.ai/ for the labeling.
- **Expectations based on group size**:
  - Groups of 1: May choose to only tackle the main tasks.
  - Groups of 2: Are expected to try the keypoint detection and analysis task.

3 Deliverables

Documenting and reporting your approach is an important part of any deep learning project. This will consist of a video presentation. Students working alone will have maximum 12 minutes to present, groups of two will have 14. Please make sure to follow the presentation structure guidelines in section 1.1, and include a part about sustainability as mentioned in section 1.2.

In addition to the video presentation, we may need to review your source code for clarification. You can either provide a link to your GitHub repository or keep the code locally on your computer, ready to share if requested during grading.

Upload your presentation in blackboard under "Course work → Work 2: Mini-Project".

Please note that we may request a teams meeting for additional clarifications.

4 Evaluation

As your submissions will likely be very diverse, we will evaluate the projects based on the overall impression of your work based on your video presentation.

Some of the things we look for:

1. Exploratory data analysis was used to guide the model development
2. Understanding of the chosen method
3. Complexity of the method
4. Performance of the chosen model(s)
5. Thoroughness in the work
6. The presentation is clear and easy to follow