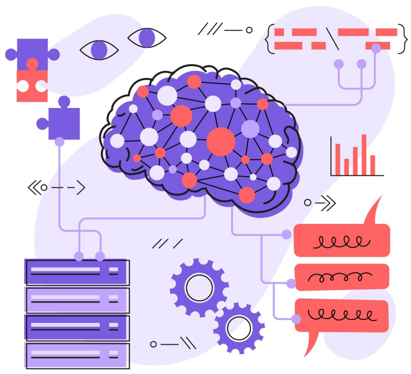
Final Project
Computer Vision and Deep Learning

Delivery Deadline: **Friday, May 3rd, 23:59PM.**
This project counts for **35%** of the final grade
You are allowed to work in groups of 1 or 2 people.

1 Introduction

This final project challenges you to apply your computer vision knowledge to solve a real-world problem. Choose from four exciting tracks: tracking for analyzing sports performance, 3D medical image segmentation to aid in diagnosis, exploring object detection for autonomous driving or getting creative with generative models in compute vision. Note that this is the first revision of the project description, so minor changes might occur. All changes will be announced on Blackboard.



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 and write a notice in your report.
2. You are allowed to use open-source architectures from PyTorch or MONAI etc.
3. Annotating the test data by yourself and using it to train/validate your model is NOT allowed.
4. It is allowed to use any pre-trained model. However, you cannot use models pre-trained on any of the respective datasets for each track.
5. Training your **final model** should not take more than **12 hours** on the IDUN cluster using 1 GPU or on the computers at the Cybele lab. Additionally, you may use up to **12 hours** for pre-training your model.

If you are unsure if something is allowed, please post on piazza or contact the teaching assistant through email: michael.s.larsen@ntnu.no. If we believe that you've broken any of these rules, we will train your model following the steps in your report, and validate that we achieve a similar performance. Breaking rules will be considered cheating on the project.

1.1 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 Tracks

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

2.1 Track 1: 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 as well as tracking.

Figure 1: 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
- **Keypoint detection 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.

2.2 Track 2: Medical Image Segmentation

Dive into the critical field of medical image segmentation by tackling the ASOCA challenge. Your mission will be to develop a model capable of precise 3D segmentation of coronary arteries in Computed Tomography Coronary Angiography (CTCA) scans. Accurate coronary artery segmentation can have a profound impact on the diagnosis and treatment of cardiovascular diseases. By precisely delineating these critical blood vessels in medical images, doctors can identify blockages (atherosclerosis), assess the severity of heart disease, and plan optimal interventions like stent placement or bypass surgery.

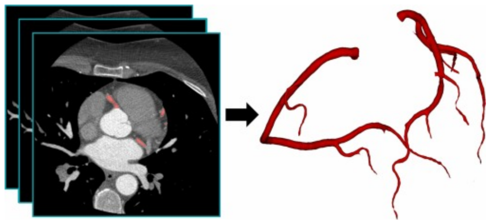


Figure 2: Sample with labels from the ASOCA grand challenge dataset

- **Dataset:** You'll work with the ASOCA Challenge dataset [1], a specialized collection of medical images designed to benchmark coronary artery segmentation algorithms. The dataset can be found on IDUN and on Cybele:
  - IDUN: */cluster/projects/vc/data/mic/open/Heart/ASOCA*
  - Cybele: */datasets/tdt4265/mic/asoca*

  Please note that redistribution of the dataset is prohibited!
- **Performance Metric:** The DICE coefficient and Hausdorff Distance (HD95) are 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, while HD95 assesses the distance between the boundaries of the predicted and true segmentation areas.
- **Recommended Loss Functions:** Specialized loss functions like the combination of Dice Loss and Focal Loss are often employed in medical image segmentation to address class imbalances and focus on the boundaries of complex anatomical structures. You may use other loss functions if you wish.
- **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: Implement at least one architecture.
  - Groups of 2: Implement at least two architectures.
- **Additional information:**
  - Asoca grand challenge - leaderboard

2.3 Track 3: Object Detection with LiDAR Data from Trondheim

Navigate the complexities of real-time perception for autonomous vehicles. In this track you will work on a LiDAR dataset collected by the NAPLab at NTNU. Your task here is to perform object detection for 8 classes.

Figure 3: Sample with labels from the NAPLab-LiDAR dataset

- **Dataset:** You'll utilize local LiDAR data from captured in Trondheim around Gløshaugen campus. This data is extracted as grayscale images in png format. There are 8 bounding box classes in the accompanying labels, in the YOLO v1.1 format.
  - IDUN: */cluster/projects/vc/data/ad/open/NAPLab-LiDAR*
  - Cybele: */datasets/tdt4265/ad/NAPLab-LiDAR*

  Please note that redistribution of the dataset is prohibited!
- **Performance Metrics:** We expect you to at least measure the COCO mAP@0.5:0.95 metric in addition to inference speed. You may also include other metrics you find relevant.
- **Recommended Loss Function:** IoU loss function variants like GIoU, DIoU and CIoU may serve as a good loss function for object detection. Read more here. You may use other loss functions if you wish.
- **Expectations based on group size**:
  - Groups of 1: Implement at least one architecture.
  - Groups of 2: Implement at least two architectures.

2.4 Track 4: Explorations in Generative Image and Video Creation

This track encourages you to delve into the exciting world of generative models, empowering you to craft images and videos using cutting-edge deep learning techniques. You'll have the flexibility to experiment with diverse approaches and discover creative applications.

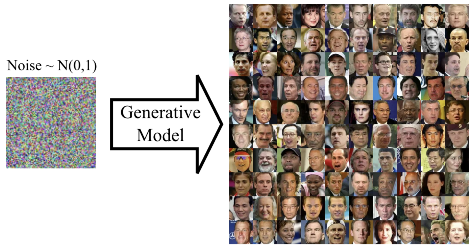


Figure 4: Illustration of VAE for the CelebA dataset

Possible Approaches:

1. **Train Your Own Generative Model:** Select a generative architecture: GANs (StyleGAN, BigGAN, etc.), VAEs, Diffusion Models, or explore emerging techniques. Choose a compelling dataset: Consider domains like faces (CelebA), natural scenes (LSUN), objects (ImageNet), or specialized datasets that match your interests. Train your model, carefully tuning hyperparameters and observing its image / video generation capabilities.

2. **Utilize Pre-trained Networks:**

   We recognize that many of the cutting-edge generative networks are trained with far more data than what it is feasible for you to do in this course. Therefore, you may also utilize a pre-trained network if you are more interested in any of these.

   Analyze the underlying architecture (GAN variants, diffusion model types, etc.) and understand how the model was trained. Experiment with the model to generate images or videos, potentially manipulating input parameters or applying techniques like style transfer.

Evaluation (Regardless of Approach)

1. **Realism and/or Quality:** Critically assess the visual quality, realism, and detail of your generated images or videos.

2. **Architecture and Training:** Whether you train your own model or utilize a pre-trained one, provide a clear explanation of: The chosen generative architecture and its key components. The training data and any specific training procedures employed. We will expect a deeper dive into the architecture for groups that utilize pre-trained networks.

3. **Potential Applications:** Articulate how your work with generative models could be applied or adapted for a real-world scenario. Get creative and have fun exploring the amazing potential of generative models!

Expectations based on group size:

- Groups of 1: Explore at least one architecture.
- Groups of 2: Explore at least two architectures.

3 Code

The code you write should be documented with enough comments such that it is possible to understand it. A typical setup should consist of the following components:

1. **Data preparation and analysis:** You are required to analyze the dataset before designing your model. You must use the insights from the dataset to develop your model.
   (a) Exploratory Data Analysis
   (b) Data Augmentations
2. **Training setup:** Writing the training setup is a major part of training a deep learning model. You should take into account the recommended loss functions and the required performance metrics for the selected track in section 2.
   - Loss function
   - Performance metrics
   - Optimizer
   - Regularization
   - Early stopping
3. **Model Architecture:** Model development should be systematic. You are expected to create a baseline model and then improve upon it with incremental additions. Compare the improvement after each addition with the baseline model.
   - Train a baseline model
   - Improve upon the baseline model
4. **Pre-training:** The datasets for track 1-3 are quite small, so experimenting with pre-training on similar datasets is highly relevant.

4 Documenting your Approach

Documenting and reporting your approach is an important part of any deep learning project. This will consist of a presentation and a short and concise report.

Video presentation

Students working alone will have maximum 12 minutes to present, groups of two will have 14.

Topics you should cover in the presentation are:

1. **Development:** The approach you decided on and what steps you did to improve your model. It should clearly describe the reasons for the changes you did, and what kind of improvements you noticed from these changes. Remember from previous assignments, your reasoning should be supported by either theoretical arguments, previous experiences made from reading the curriculum, or empirical experiments. An example of this could be

    > ResNet is known to improve gradient flow and diminish the problem of the degradation problem, therefore, we decided to use ResNet as the backbone. By replacing the backbone we notice a significant improvement that is shown in the loss curve as you see here on our powerpoint slide.

    Of course, doing this for every improvement will take a lot of time, but we expect analysis like this for the major improvements you made.

2. **Final model:** Describe your final model. This should be short since we should have a clear picture of your final model from the development process. Things you might include:
    - A brief overview of the final architecture.
    - How you trained it, for example what kind of loss function did you use? What kind of data augmentation did you use? What kind of data pre-processing did you do?
    - Amount of time required to train it.
3. **Results:** Show results of your model. Include quantitative and qualitative results. Quantitative results are for example loss curves, performance metrics. Qualitative analysis could be testing your model on different images in the dataset and see what kind of classes it's good at segmenting, and what it struggles with.
4. **Runtime Analysis:** Perform a runtime analysis of your model. What is the inference time for your model?
    - Did you make any effort to improve the runtime of your model?
    - If not, what could you potentially do to improve the runtime?
5. **Carbon Footprint:** Calculate the carbon footprint of your model training. Record the power usage from all the different experiments you do, and compare it to something like how many kilometers you can drive with a Tesla Model 3.
6. **Discussion** Discuss your approach for solving this task and the final results you achieved. Examples of questions you might want to answer is:
    - Did you test something that did not work?
    - Was there any unexpected results?
    - Looking back at the task, is there anything you would want to do differently?
    - If you did not have the limitation of the rules in this assignment, what would you do?
    - Is there any further work you would like to do? (for example, things you didn't get time for)
7. **Group member contribution:** In the end of the presentation, shortly describe every group members contribution to the project and what they were responsible for.

The final presentation will be a significant part of your grade. I recommend you to start early and prepare yourself well for the actual presentation. For general guidelines for presenting a project, Professor Charles Elkan has some good advice.

With these guidelines, we are trying to help you to show your knowledge about the curriculum in the course, and prevent you from waste time on nitty, gritty details. Often students struggle with managing the time during the presentation and spend all their time on describing the initial model, architecture etc. This leads to a very rushed discussion and result section, making it hard for the evaluators to understand the work gone into the project and the student's understanding of the underlying concepts. Also, we are not interested in what learning rate you used, what batch size you used or any kind of hyperparameters you chose; we can read up on this in the report ourselves if we are interested.

Documentation of Details

The report should be short and concise. What we expect you to include in this is anything "boring" and **note that we will not read this document except** if we are interested in technical details of your model, or we want to re-run/validate your experiments. Note that anything included in the presentation should not be included in the report. What we expect is that the report includes anything required to re-produce your results. Such as:

- Hyperparameters. This can be referred to as "We used the config file `our_amazing_model.yml` and all hyperparameters are there". Nothing else is required.
- How to train your model. Assume that we want to re-run all your experiments. Document clearly how we should be able to do this. An example of this could be

  > To setup your environment, install the additional packages "some-package" (Not required if you used the default environment used in the assignments). Then, you can train the model on cityscapes by running the file "some_train.py". Finally, run the evaluation script.

- Specific details of your model architecture. Examples of this can be the tables with models given in previous assignments.
- Any additional results that you did not have place for in the report. However, we do not want any discussion of this result in the report.

The reason we want such a short report is that we do not have enough staff resources to read through everything. Even though we truly enjoy reading your assignments and reports, it would take us way too much time getting through all of your reports!

5 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. Has followed the guidelines
7. The presentation is clear and easy to follow

6 Deliverables

We expect the following three deliverables for this project. These files should be uploaded to the assignment submission in blackboard, within the deadline!

– Source code as a zip file (Do not include datasets!)
– A PDF report
– A video presentation (maximum 12 min for groups of 1 and 14 for groups of 2)

References

[1] R. Gharleghi, D. Adikari, K. Ellenberger, S.-Y. Ooi, C. Ellis, C.-M. Chen, R. Gao, Y. He, R. Hussain, C.-Y. Lee, J. Li, J. Ma, Z. Nie, B. Oliveira, Y. Qi, Y. Skandarani, J. L. Vilaça, X. Wang, S. Yang, A. Sowmya, and S. Beier, "Automated segmentation of normal and diseased coronary arteries – the asoca challenge," *Computerized Medical Imaging and Graphics*, vol. 97, p. 102049, 2022. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S0895611122000222