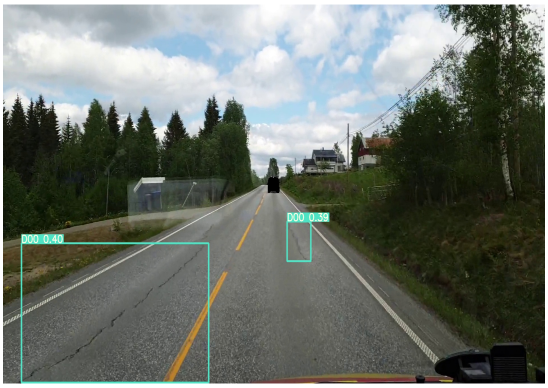
Final Project
Computer Vision and Deep Learning
Mamoona Birkhez Shami
mamoona.b.shami@ntnu.no

Delivery Deadline:
Friday, April 28th, 23:59PM.
This project counts **40%** of your final grade.
You can work in groups of up to **2 persons**.

1 Introduction

About. Roads are a vital infrastructure of a country. They require periodic checks and maintenance to ensure unhindered travel between cities and communities which is imperative for a functioning society. Given the limited resources, the decision of when and where to delegate them is very important. Traditionally road maintenance is done by manual inspection or through specialized cars which is costly. Given the modern technological advances in the field of computer vision and artificial intelligence, an intelligent solution can be built that can detect and classify the road defects. Cameras are cheap and a lot of surveillance data is already available. This solution will help the road maintenance authorities to make decisions and save time and cost on manual inspection.

In this project, our goal is to train a network that can detect different types of road damages from videos. This can be solved as an object detection task, where we require labed ground truth images (similar to MNIST in Assignment 4). Here, we will detect 4 common types of road defects, where their labels are given in Table 1.



Damage Type	Class Label
Longitudinal Crack	D00
Lateral Crack	D10
Alligator Crack	D20
Pothole	D40

Table 1: Road Damage Types and their Class labels [Maeda et al., 2018].

To solve this task, we will use a public road damage detection dataset from [Arya et al., 2022b] (called *RDD2022*) that is collected from six different countries including Norway. The article for this dataset can be found here [Arya et al., 2022a]. **Our test set will only consist of images from Norway.** However, you can use images from any country to augment your training set.

Your task will be to build upon these datasets and solve the following subtasks:

1. **Annotate data collected from Norway.** You have to annotate images containing instances of the four classes mentioned in Table 1. We have built a custom annotation server for this project. See section 2.
2. **Developing and building your model.** You will develop a baseline model and improve upon it. We will have two baselines for you to compete with. See section 3.

3. **Evaluating your model.** The last step is to evaluate the model on our test data. You will report the mean Average Precision on the test dataset and score your model. As a motivating factor, we have set up a public leaderboard to compare yourself against your classmates. See section 4.

4. **Documenting your approach**. In a real-world scenario documenting your approach is important. Either you're going to hand-off the project to someone else, or reporting about the amazing work you have done to your boss. For this project, you will deliver a short report and deliver a video about the project. We will conduct viva for each group for final evaluation. See section 5.

Project Rules

To achieve a fair project (and grading..), we ask you to follow these rules for all submssions to the leaderboard on tdt4265-annotering.idi.ntnu.no/leaderboard [1].

1. The code should be developed by yourself. 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. Annotating the test data by yourself and using it to train/validate your model is NOT allowed.
3. It is allowed to use any pre-trained model. However, you cannot use models pre-trained on RDD2022 from the CRDDC2022 challenge.
4. It is not allowed to train your model on any other data except the data provided by us. However, it is allowed to use backbone networks pre-trained on the ImageNet dataset, but you are not allowed to perform this training yourself. This is to prevent the winning solution be the one who uses the most amount of data/compute.
5. The project coincides with Easter break from 3rd to 10th April. The teaching assistants will not answer any questions during that break unless it is an issue with the annotation server.

If you are unsure if something is allowed, please post on piazza or contact the teaching assistant through email: mamoona.b.shami@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 mAP. Breaking rules will be considered cheating on the project.

ΩΩ

Grading (40 points)

1. [5*pts*] **Annotating Data**: Each group is required to annotate a minimum of *two tasks per person in the group*. Therefore, if you are a group of two, you are required to annotate a minimum of 4 tasks to receive a full score.
2. [5*pts*] **Data Analysis**: You are required to analyse the dataset before designing your model. You must use the insights from the dataset to develop your model.
3. [5*pts*] **Model Development Requirements:** 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.
4. [4*pts*] **Model Performance**:
   - [1*pts*] mAP higher than baseline 1.
   - [3*pts*] mAP higher than baseline 2.
5. [14*pts*] **Submission**:

[1] Password for login is given in section 2

- [1*pts*] Leaderboard Submission.
- [8*pts*] Video Submission. See section 5 for more information.
- [5*pts*] Report Submission. See section 5 for more information.

6. [2*pts*] **Additional Analysis**: You are also required to report your final inference time and the carbon footprint of training.

7. [5*pts*] **Going Above Beyond**: You will be awarded extra points if we see you have put in extra effort in the project.

2 Annotating Data

Annotating data precisely is an important part of any deep learning project. To get you started rapdily, we've built an annotation server on top of opencv/cvat, at tdt4265-annotering.idi.ntnu.no.

You can login on the server using

- username: group[blackboard group number] (for example: group1)
- password: SR5cF8RFqQF9A3aF

Before you start to annotate, make sure to:

1. Login to the annotation system, and validate that the group number in the right corner matches the project group number on blackboard.
2. Before you start to annotate a task, set yourself as the "asignee" of the task before starting from top right corner. This will register that your group is the one that annotated this task.
3. Make sure to mark the task as finished before assigning yourself to a new video. It is possible that several of the persons in the group annotate several videos in parallel.

We will validate that you have annotated the data by looking into the server database.

Before you start to annotate the data, make sure to read the following tutorial:
`https://github.com/TDT4265-tutorial/TDT4265_StarterCode/blob/master/project/tutorials/annotation_tutorial.md`

The deadline for Annotation task is **12th April.**

3 Building the Model

For this project you will use what you've learned from previous assignments to improve the model from assignment 4. There are no strict requirements or guidelines on the development, and you're free to use whatever means necessary to achieve a high mAP (as long as you follow the competition rules).

Here are a couple of recommendations to get you started:

- Try other single-shot models like YOLOv5 or YOLOv7 which perform better than SSD.
- Pre-train your model on the RDD2022 dataset for other countries before fine-tuning it for the Norwegian dataset.
- Look at various types of data augmentation to improve training (especially on the RDD2022 dataset). The SSD paper mentions several augmentations which they achieved good results with.

- Customize your model to use another image ratio. The Norwegian Dataset images have higher resolution and in the assignment 4 starter code we did a naive approach of directly resizing the image to 300×300. This causes artifacts in the image (such as a crack would be much shorter in the resized image) and might hamper training and final mean average precision. Maybe you could change the model to process a non-square image?

4 Model Evaluation

To evaluate your model and compare your model to the rest of the class, we have created a leaderboard reporting the mean Average Precision for every group. The leaderboard and submission details will be released later.

Furthermore, we've created a couple of benchmarks for you to beat: Baseline 1 and Baseline 2.

The baselines will be released later during the project.

Qualitative Evaluation

To evaluate your model on real scenarios, test the model on two video files. You can download these from: video-1 and video-2

Show one of the videos in the presentation.

5 Documenting your Approach

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

Video

Students working alone will have 6 minutes to present, and groups of two will have 8 minutes.

Topics you should cover in the video are:

1. **Dataset Analysis:** Perform an exhaustive dataset analysis to explore the attributes of the Norwegian dataset. The analysis should highlight commonalities, limitations and perhaps interesting samples in the dataset (both images and labels). You are free to choose how to present the analysis, which can be both statistical and qualitative. To get started, you can try to analyze the size of the objects in the dataset. You can analyse the class imbalance between different countries. This step is (arguably) the most important part of any computer vision project. The insights you bring from the dataset analysis will guide your modelling decisions and help you build an efficient model tailored to the needs of the task.

2. **Development:** The approach you decided on and what steps you did to improve your model. It should clearly describe the reasons to 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.

3. **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 how did you pre-train it? What kind of data augmentation did you use? What kind of data pre-processing did you do?
   - Amount of time required to train it.
4. **Results:** Show results of your model. Include quantitative and qualitative results. Quantitative results are for example loss curves, mean average precision, and average precision for specific classes. Qualitative analysis could be testing your model on different images in the dataset and see what kind of objects it's good at detecting, and what it struggles with.
5. **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?
6. **Carbon Footprint:** Calculate the carbon footprint of your model training. You can see what is the power consumption of your GPU and use that to find your carbon footprint.
7. **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)
8. **Group member contribution:** In the end of the video, shortly describe every group members contribution to the project and what they were responsible for.

With these guidelines, we are trying to help you to show your knowledge about the curriculum in the course and prevent you from wasting time on nitty, gritty details. Often students struggle with managing their time during the presentation and spend all their time 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 ourself 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 video should not be included in the report. What we expect is that the report includes anything required to reproduce your final model. Such as:

- Hyperparameters. This can be referred to as "We used the config file and all hyperparameters are there". Nothing else is required.
- How to train your model. Assume that we want to re-run your experiment. 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”. Furthermore, fine-tune the model on the Norwegian dataset by running ”some_train2.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 a place for in the video. 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 to get through all of your reports! **Delivery** We ask you to follow these guidelines:

- **Report:** Deliver your answers as a **single PDF file**. Include all tasks in the report, and mark it clearly with the task you are answering (Task 1.a, Task1.b, Task 2.c etc). There is no need to include your code in the report.
- **Plots in report:** For the plots in the report, ensure that they are large and easily readable. You might want to use the ”ylim” function in the matplotlib package to ”zoom” in on your plots. Label the different graphs such that it is easy for us to see which graphs correspond to the train, validation and test set.
- **Source code:** Upload your code as a zip file. In the assignment starter code, we have included a script () to create your delivery zip. **Please use this**, as this will structure the zipfile as we expect. (Run this from the same folder as all the python files).

  To use the script, simply run:
- **Upload to blackboard:** Upload the ZIP file with your source code and the report to blackboard before the delivery deadline.
- The delivered code is taken into account for the evaluation. Ensure your code is well documented and as readable as possible.

Any group who does not follow these guidelines or delivers late will be subtracted in points.

References

[Arya et al., 2022a] Arya, D., Maeda, H., Ghosh, S., Toshniwal, D., and Sekimoto, Y. (2022a). Rdd2022: A multi-national image dataset for automatic road damage detection.

[Arya et al., 2022b] Arya, D., Maeda, H., Sekimoto, Y., Omata, H., Ghosh, S. K., Toshniwal, D., Sharma, M., Pham, V. V., Zhong, J., Al-Hammadi, M., Shami, M. B., Nguyen, D., Cheng, H., Zhang, J., Klein-Paste, A., Mork, H., Lindseth, F., Seto, T., Mraz, A., and Kashiyama, T. (2022b). RDD2022 - The multi-national Road Damage Dataset released through CRDDC'2022.

[Maeda et al., 2018] Maeda, H., Sekimoto, Y., Seto, T., Kashiyama, T., and Omata, H. (2018). Road damage detection and classification using deep neural networks with smartphone images. *Computer-Aided Civil and Infrastructure Engineering*, 33(12):1127–1141.