Transformers and Attention-Based Deep Networks
Final Project

Vision-Language Models for Image Captioning

Middle East Technical University
Graduate School of Informatics

31.03.2025

1 Introduction

This is the documentation of the final project of DI 725: Transformers and Attention-Based Deep Networks. There will be three phases throughout this project. The first phase will include the preparation of a brief literature survey and a project proposal. The second phase will cover the preliminary results and benchmarking. The third phase will conclude the project with results and comparisons.

There will be a base vision-language foundation model (PaliGemma) that is suitable for various tasks including but not limited to visual question answering, image captioning and object detection. **Your task is to generate image captions**. You will structure your own research question, and propose a project to build on top of this foundation. ***It's highly recommended to begin your project as early as possible.***

2 Dataset

You will be using the RISC dataset throughout this project (Figure 1). Please follow the link to download the dataset from the repository (ODTUClass). The data consists of 44521 remote sensing images (satellite) with fixed 224x224 resolution and 222605 captions (5 per image) briefly summarizing the content of the image.

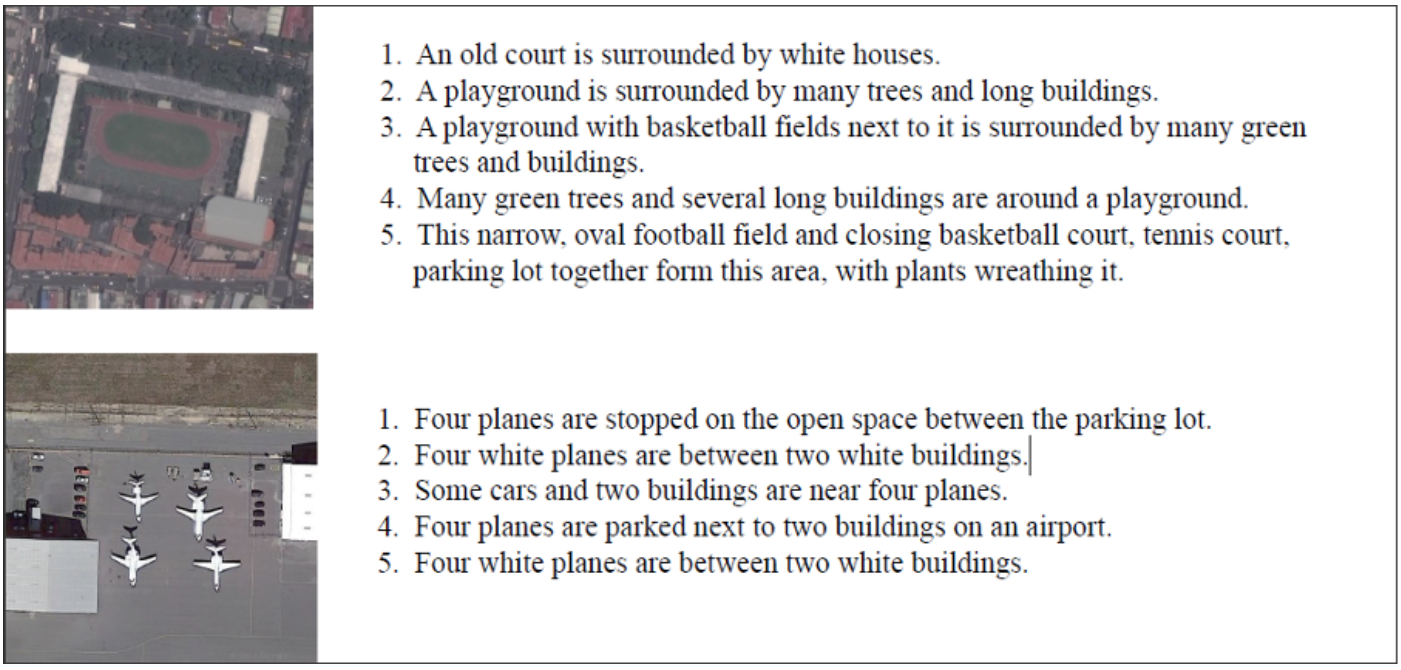

Figure 1: Remote Image Sensing and Captioning Dataset

It is very important that you use the data split as defined in the captions.csv file. Use the images and captions defined as **train** to train your models, **val** to validate your approaches and adjust your hyperparameters and tweaks and finally **test** to finalize your claim. You shall not overfit to the test set! **You must not leak test data in any way** to the train or validation splits. The test split is present just for the final conclusions. You shall start by understanding the data (images and captions) and finding the evaluation metrics suitable for image captioning. Hint: There are some discrepancies present in the captions, so think about how they can affect your evaluation and how you can fix these.

3 Vision-Language Model

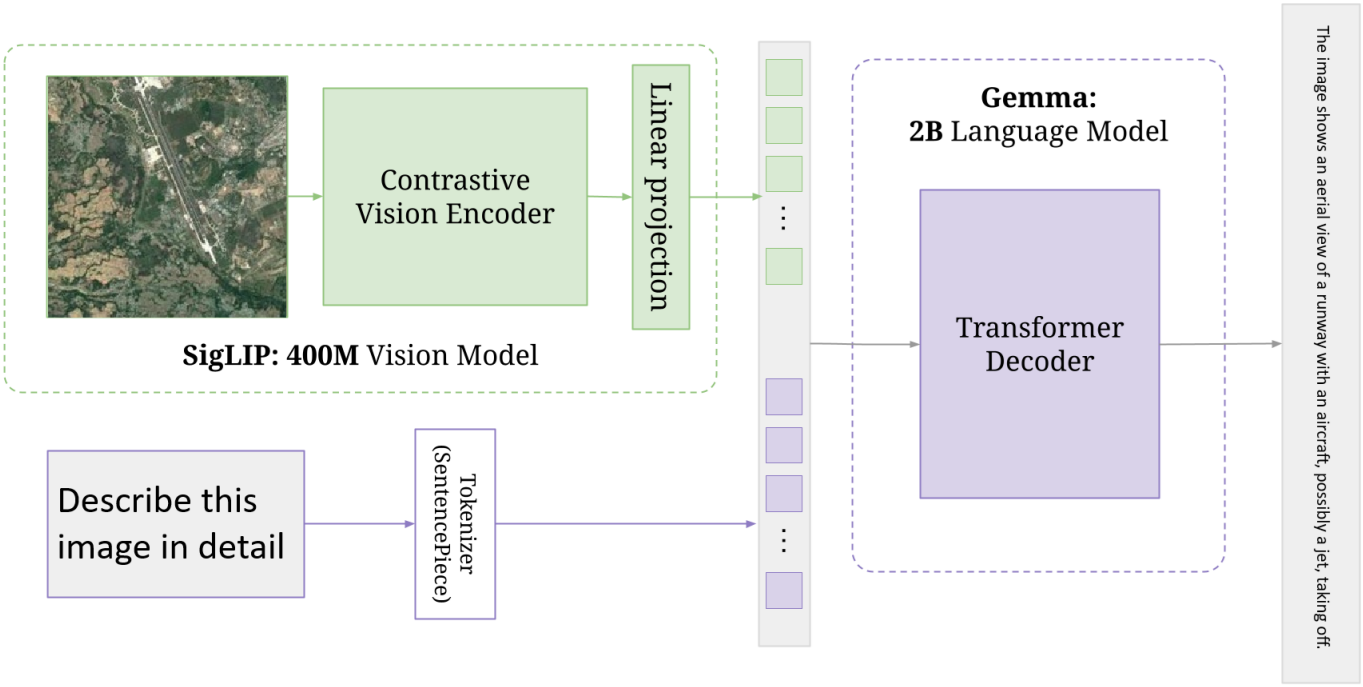

Figure 2: PaliGemma: A vision language model

We will be using PaliGemma, shown in Figure 2, model as our starting point in this project. You need to propose an improvement over this model to fine-tune the image captioning task. Your proposals on how to improve the model can be on various topics including, but not limited to:

- Architectural improvements
- Efficiency and lightening adaptations
- Improved inference speed or throughput
- Efficient fine-tuning processes

You shall prepare a summary of the literature and formulate research questions to improve the existing foundation. Please note that the base vision-language model (PaliGemma) model inherits limitations from the vision encoder and language decoder models. The following limitations come from the SigLip (image encoder) and Gemma (language decoder) models:

- Vision-Language Models (VLM) require clear prompts and instructions to work effectively.
- Creative and complex tasks are not (yet) suitable for VLMs.
- VLMs might not be able to understand nuances, sarcasm or figurative language, due to the inherent complexity of natural languages.
- VLMs are not knowledge bases, they can generate responses based on their training but this can be hallucination or incorrect.
- VLMs depend on statistical patterns of vision and language domains. They might not be able to apply common sense.

PaliGemma is:

- A general pre-trained model suitable for fine-tuning tasks.
- A solid starting point for adapting a VLM model.

PaliGemma is not:

- A zero-shot model that can be used as-is, there are better models for this task.
- A chatbot with multiple question-and-answer sessions.

4 Deliverables

Your submission is expected to be uploaded to ODTUCLASS in a single zip file with the naming convention "DI725_project_phase_X_S
Do not upload the dataset or any large file.

5 Grading

Grading in this project will be done according to the following:

- Phase 1: 20 pts
- Phase 2: 20 pts
- Phase 3: 60 pts
- **Report**: Prepare a report that clearly explains the steps in this project, changes and tweaks over the starter code and how these affected the results. The report shall have the following sections: abstract, introduction, dataset, modeling, evaluation, results, discussion, and conclusion.
- **Code**: The code that is to be shared must be working and reproducible, submit your code to Odtuclass, and also upload it to your public GitHub repo.
- **Version Control**: Your implementation shall be traceable from your GitHub repository. Your repository shall be public so that we can examine your code.
- **Experiment Tracking**: Experiments in this project are to be recorded to WANDB. Your WANDB experiments page or WANDB report page shall be public so that we can examine your experiments.

6 Project Phases

There will be three phases for the term project. It is recommended that you start your project work as soon as possible. All of the project phases will be graded according to the guidelines in the grading section and you can find the summary table of deliverables and due dates in the Table 1.

Phase	Deliverable	Due Date
1	Literature and Project Proposal	**20th of April**
2	Preliminary Results and Benchmarking	**4th of May**
3	Final Submission	**1st of June**

Table 1: A summary of deliverables and due dates.

6.1 Phase 1 (Due 20th of April)

Your deliverables in this phase are a report (2-page maximum) and your preliminary version control (GitHub repo links in your report), experiment tracking (WANDB links in your report) and codebase (if you have started with your experiments).

6.1.1 Literature

Conduct a comprehensive survey of existing research on your chosen proposal. You are expected to provide a concise yet comprehensive review of the relevant literature on your chosen topic. Your review should be well-supported with appropriate sources, demonstrating a clear understanding of the field. By analyzing the existing research, you should identify potential gaps and opportunities for further study.

6.1.2 Project Proposal

Formulate a research question and a detailed proposal outlining your approach's importance, and performance metrics. At this stage, you should define the objectives, potential benefits, and expected outcomes of your project. It is essential to outline an approach related to the concept of transformers and structure your planned experiments accordingly.

6.1.3 Phase 1 Guidelines

Please carefully read the guidelines for grading. Each of these points is crucial for earning complete points from this project.

- **Version Control (2 pts)**: In the first phase, you will not be doing any coding. However, you shall set up a git account and initialize a public GitHub repository (e.g. with a markdown readme).
- **Experiment Tracking (2 pts)**: In the first phase, you will not be doing any experimentations. However, you shall set up a public WANDB account and initialize a basic public WANDB experiments or report page.
- **Literature (8 pts)**: Find at least four appropriate research papers about the project of your choice. These papers have to be from reputable journals and conferences and be indexed by reputable databases such as Web of Science and Scopus. Read, understand and prepare a short literature summary about these related works. Finally, discuss research opportunities you find in the literature. You can reuse the literature part in the final report's literature section.
- **Project proposal (8 pts)**: Propose your methodology, explain your approach, novelty and conceptual merits of your proposal. In this section you will also prepare an explanatory data analysis part where you will explore and explain the data you are going to use throughout the project. It is imperative that you thoroughly examine the data before committing more of your time and resources. The project proposal, if prepared appropriately, can be reused in the final report's methodology and dataset section!

6.2 Phase 2 (Due 4th of May)

Your deliverables in this phase is an intermediate report (4-page maximum) and your version control links, experiment tracking links and code base.

6.2.1 Preliminary Results

Execute your proposed method and refine your code framework. This stage focuses on refining your project concept, addressing inconsistencies in your training process, and obtaining initial results. By this point, you should have a solid grasp of the literature and a well-defined experimentation plan.

6.2.2 Benchmarking

Conduct comparative benchmarking tests and analyze results on top of the baselines. Once you obtain your initial results, you will proceed with benchmarking tests. Your goal is to surpass the baseline performance through an ablation study. Proper preparation for benchmarking is critical to achieving meaningful improvements.

6.2.3 Phase 2 Guidelines

Please carefully read the guidelines for grading. Each of these points are crucial for earning complete points from this project.

- **Version Control (2 pts)**: The second phase of the project requires you to have at least 3 meaningful commits (including initial code push) on different days. Please use the issues in the GitHub to communicate the bugs, limitations and issues you encounter, and try to close any open issues with commits. Use the issues board to write down notes.
- **Experiment Tracking (2 pts)**: All of your experiments, including the preliminary results and the current benchmarking tests you have completed so far shall be visible from the WANDB experiment tracking tool. Your reports or experiments page shall be public, else you will not get any points from this section.
- **Preliminary Results (8 pts)**: Your preliminary results, which are solid evidence of your effort to find an adequate research opportunity and implement it, will be graded in this section. Your preliminary results shall surpass the pre-defined baseline of the task, and shall be accompanied by a code framework that is reproducible and documentation to explain your intention. If you keep your documentation neat and your code clear, you can reuse the documentation in your final report and reuse the code for benchmarking.
- **Benchmarking (8 pts)**: Benchmarking, hyper-parameter tuning and ablation study are crucial aspects of any research. In the second phase, your benchmarking shall start right after you obtain preliminary results. You shall start tweaking hyperparameters or adapting your modeling architecture to formulate ablation studies. By tweaking and adapting your approach you will obtain many different combinations that can be promising to offer even better results than your preliminary result. To obtain full grades from this section, you shall propose at least two different hyper-parameters to tweak or architectural changes to make to your model. If you start benchmarking as early as possible and try to finish before the deadline of phase 3, you can use many of these results to formulate even better competitive advantages!

6.3 Phase 3 (Due 1st of June)

Your deliverables in this phase are a final report (6-page maximum) and all of your version control, experiment tracking and code base.

6.3.1 Submission

You will upload the report and all necessary documentation at this final stage. Your experimental setup, ablation studies and results shall be briefly mentioned and discussed in your report. Be sure to follow the project guidelines carefully. Before the project concludes, you have to submit your work and reports via Odtuclass, and share your git repository and WANDB experiments/report page.

6.3.2 Bonus

A special novelty award may be granted to those with the most innovative and original approaches.

6.3.3 Phase 3 Guidelines

Please carefully read the guidelines for grading. Each of these points are crucial for earning complete points from this project.

- **Version Control (6 pts)**: Your code shall be publicly available in your GitHub repository. You shall have at least 5 commits throughout this project on different days. You shall use issues to document any bug or limitation you face and close these issues appropriately if you can. Your repository shall be clean, with no redundant or unnecessary files (use .gitignore). Your repo shall be publicly available, if it is not you will not get any points from version control. If you are aware of an issue and were unable to solve it, leaving it open in your GitHub will be favorable for you.
- **Experiment Tracking (6 pts)**: Your experiments shall be reported via WANDB tool. Your WANDB experiments page or reports page shall be publicly available, if it is not you will not get any points from experiment tracking.
- **Clean Code Practices (6 pts)**: Your code shall be clear to be traced easily, contain necessary comments and shall not contain any redundant parts. If you have followed clean code practices in the previous phases and maintained them you will earn full grades from this section, if not now is the time to clean your code!
- **Demonstration (12 pts)**: You shall prepare a short video presentation where you will record your screen and your voice (or prepare an annotation subtitle if you don't want your voice to be recorded) and explain briefly every part of your implementation (code), method, results and limitations. This video presentation can be about 5-10 minutes long.
- **Documentation (12 pts)**: Your documentation is your only way of communication for your project. Proper documentation shall explain clearly all necessary steps you have taken. Your final report shall have the following parts: abstract, introduction, dataset, modeling, evaluation, results, discussion, and conclusion.
- **Approach (18 pts)**: Your approach will be graded according to the benchmarking setup, experiment variety, results, and depth of implementation. If you have successfully adopted a transformers-based approach, for the task, and achieved better results than both the baseline and preliminary results, you will earn full grades from your approach section.
- **Bonus (10 pts)**: Your approach's novelty will be awarded accordingly. You need to demonstrate the novelty with a marginal improvement to the literature.

7 Appendices

7.1 References

You should not choose a random paper returned from your query in Google Scholar. The paper should have been published in a respectable journal or conference proceeding. You can refer to the Google Scholar's Journal & Conferences Rank list to find high-impact publications in Artificial Intelligence category. There are many ways to check the quality of the paper. Some of them are:

1. It might be indexed by Web of Science or Scopus.
2. It might have been published by respectable publishers such as IEEE, ACM, Elsevier, or AAAI.
3. Check the paper or the conference/journal name from ScienceDirect or Scimago Journal & Country Rank

You are also expected to reference them properly in your report in IEEE format. All the fields should be complete and proper. Do not rely on what returns from Google Scholar as they are often missing. Check the origin of the paper. Although some appear to be published in Arxiv, they might have been published in a respectable journal or conference afterward. Hence you need to update the details accordingly. If you cite your paper incorrectly or don't use a paper from a good resource (such as one from a predatory journal), you will lose grades accordingly.

7.2 Academic Conduct and Generative AI

Students are expected to uphold the highest standards of academic integrity in all aspects of this project. The use of generative AI tools (e.g., ChatGPT, DALL·E, Copilot, etc.) is permitted under the following conditions:

- Ensure that code generation does not introduce security vulnerabilities and licensing conflicts.
- AI-generated code must be thoroughly reviewed, tested and understood by the author.
- Any code snippet, that is used as-is, should be referenced by either the original contributor or the AI-generator.
- AI-generated visualization shall be reviewed and checked with the original data.
- Use of AI tools shall be transparent. All instances of AI tools shall be explicitly disclosed in the documentation (project report).

You can use AI generation tools for the following:

- **Debugging and optimization**: Errors are common when working with state-of-the-art and cutting-edge development. Fixes and solutions for different environments can be present in the open-source domain and AI tools are suitable for finding a solution (akin to searching through stack-overflow).
- **Spell and grammar checking**: AI tools can be used to check for spelling and grammar errors.
- **Code suggestions and completions**: AI tools such as ChatGPT and Copilot are suitable for code completion as long as the author knows what and how to implement them.
- **Visualization creation**: Creating visuals from existing data with AI tools can be productive and labor-saving.
- **Generative data augmentation**: AI tools can be employed for generating synthetic data.
- **Textual drafts**: You can generate drafts with AI to adapt to a template.

You should not use AI generation tools for the following:

- **Creative content generation**: AI tools are infamous for plagiarism and copyright violations.
- **Finishing entire projects solely with AI tools**: Using AI to complete entire implementation tasks without understanding undermines the purpose of the projects and assignments. The projects and assignments are designed to be instructive, real world tasks are often much harder, complex and time-consuming.
- **Review literature completely with AI**: AI tools can be used for summarizing or finding relevant content, however, using only AI for literature review can introduce inherent biases to your work. More importantly, AI tools can hallucinate with the literature content, often leading to irrelevant or sometimes non-existent content.
- **Substitute yourself with AI tools**: Human creativity, consistency and character are irreplaceable with any AI tool, you shall not diminish your effort and contribution by replacing yourself with any AI tool. However, it is beneficial to use AI tools as productivity tools to augment your work.

7.3 General Guidelines for Grading

- The projects should be prepared on an individual basis. You are not allowed to work together. You are expected to follow academic integrity.
- Your whole analyses should be connected/related to each other. You should not do something for the sake of doing it. For instance, if you do apply a feature engineering technique and do not use the results in somewhere else or do not relate it to your research questions, your mark will be degraded accordingly
- Storytelling: You are expected to conduct your analysis by referring to your research questions. You can pose sub-research questions. Check for any spelling or grammar mistakes before the submission of your report. You can use LLMs, Grammarly or Outwrite for this.
- You can refer to ChatGPT ONLY for the purpose of revising and re-writing your paragraphs to enhance the flow. We have tools available to check if you let ChatGPT write nonsense about the topic.
- Creativity / Novelty: We appreciate the hard and creative work that you put into your project. If you demonstrate the work is novel/creative (based on literature etc.), you will be awarded a bonus.
- Your Jupyter notebooks should be self-explanatory and easy to run. We should see all the steps you followed for your analyses in the notebook. You cannot use any other platform (such as Excel or R, even for partial runs) because it will be difficult for us to check their validity.

- The deadline is strict. Hence, start working on your project and documentation as soon as possible. You can't finish it if you leave it to the last minute.
- Before submitting, make sure your notebook runs without errors from start to finish.
- If you do not submit your source code, or if the submitted code does not run successfully, your project documentation will not be evaluated by itself.
- You should share your experiments and code in a completely public manner. Your GitHub and WANDB pages shall be set as public, you can't and shouldn't expect to invite everyone and every stakeholder who will examine your experiments or reference your code to your private repository. Thus if you want to attempt experiment tracking and reproducible code grades your repositories shall be accessible via non-members of the respective services. You can check the accessibility by testing to reach your repositories in an incognito browser mode. If you are having problems with enabling public access to your WANDB experiments, you can use the WANDB reports (check this tutorial).

7.4 Report Format

The final report should be in IEEE format and have a maximum of 6 pages including the visualizations and references. Your report should be at most 2 pages in the first deliverable, at most 4 pages in the second deliverable and at most 6 pages in the final deliverable. You can find the template here. Please follow this path to select the correct format: Conferences, Original Research, Word or Latex.

Your report must include a minimum of one graph and table. The maximum number for both is four. Each figure and table must have a clear and descriptive caption. Make sure the graphs you provide are exported at 300 dpi and not copied and pasted. Failing to do so, or not staying within limits will result in grade reductions.

Your results must be shown in tables or other appropriate structures. You can follow the data-to-viz visualization tool to efficiently and beautifully represent your data and findings. Do not copy and paste code or outputs from your notebook for the paper. This will result in grade deductions. Make sure your graphs, tables, and other material are legible, i.e. readable. Make sure different parts of your graphs (like lines, boxes, and points) can be identified by their shape as well as color. (Like changing line types, markers, etc.) Treat this as if this paper would be printed out in black and white.

Your report will also be evaluated based on adherence to the required format, correctness and sufficient explanation of each section, creation of clear and accurate captions for figures and tables, and references being written completely, accurately, and in the correct format.

7.5 Research Question

You should write research questions considering the following factors: A good research question should be:

- **Clear and focused**. The question should clearly state what the writer needs to do.
- **Not too broad and not too narrow**. The question should have an appropriate scope. If the question is too broad, it will not be possible to answer it thoroughly within the word limit. If it is too narrow you will not have enough to write about and you will struggle to develop a strong argument.
- **Not too easy to answer**. The question should require more than a simple yes or no answer.
- **Not too difficult to answer**. You must be able to answer the question thoroughly within the given time frame and word limit.
- **Researchable**. You must have access to a suitable amount of quality research materials, such as academic books and refereed journal articles.
- **Analytical rather than descriptive**. Your research question should allow you to produce an analysis of an issue or problem rather than a simple description of it.

We urge you to check these links as they will guide your question-asking and literature search:

- Developing a research question [7:23] University of Melbourne, Academic Skills Youtube Series
- How to Create Research Questions & A Literature Review [54:57] METU Academic Writing Center by Zeynep Ünlüer
- How to Write a Research Question (George Mason University)
- Identifying a Research Problem and Question, and Searching Relevant Literature (Daniel J. Boudah, 2011)

7.6 Reproducible Code

Ensuring reproducibility in machine learning projects is crucial for validating results, enabling future improvements, and facilitating collaboration. Follow the guidelines below to make your project easily reproducible.

- The dataset should not be embedded within the submitted Jupyter notebook. Instead, use relative paths when accessing data files (e.g., data/dataset.csv).
- Record all dependencies in requirements.txt to ensure that others can install the necessary packages and replicate your environment.
- To learn more about code reproducibility, you can watch this presentation. Additionally, you can read the following documents: Organizing Your Projects and Reproducible Research: Goals, Guidelines and Git.
- If you are not familiar with Git and GitHub, you can take the following courses on Datacamp: Introduction to Git and Introduction to GitHub Concepts. Please reach out to me for access to these courses.
- Even though all the details of your project will be available in your GitHub repository, you must submit the final report, the final Jupyter notebook, and the link to your GitHub repository via ODTUClass.

You will use The Alan Turing Institute's Reproducible Project Template to structure and present your project on GitHub. Although the full template is more comprehensive, the following structure is sufficient for this project and you will follow this structure:

- data: Folder containing the dataset.
- notebooks: Final Jupyter notebook.
- source: Source code folder.
- reports: Final IEEE paper.
- figures: Generated figures used in reporting.
- requirements.txt: The list of required Python packages.
- README.md: An explanation of your project (optional).

7.7 Late Submission

Each phase may be submitted up to 48 hours late at most. After this 2-day period, the submission system will close. You will get 10-point penalty for every day of late submission.