Department of Computer Science and Software Engineering

COMP6721 Applied Artificial Intelligence Summer 2024
Course Project Guideline

Table of Contents

Resumé 2

Email Inquiries 2

Main Objective 3

Team Formation 4

Team Member Specialization. 4

Team Issues. 4

Training Data (Reference: Summer 2022 Project Assignment #1) 4

Writing a One-Page Proposal 4

  Proposal Submission 5

First Phase (Decision Trees) 5

Second (Final) Phase 6

  Reports Formatting 8

Final Demo 8

GitHub Submission 8

How to Submit Your Project Materials? 9

  Late Submissions 9

APPENDIX 1 Semi-Supervised Learning 9

APPENDIX 2 Frequently Asked Questions (FAQ) 11

Resumé

Image Dataset	No. of Observations	Image features	Goal
You choose	> =2500 images	Color images	Venue classification with your choice of 5 venues (classes)

Models	Libraries	Steps
Decision Tree supervised Decision Tree semi-supervised CNN supervised	PyTorch scikit-learn/Skorch	Image preprocessing Modeling x 3 Report

To compare	Performance metrics
Performance comparison of the 3 models Performance comparison modifying 3/2 hyperparameters	Accuracy, precision, recall, F_1-measure, confusion matrix

Section	Deadline (11:59PM)	Deliverable	Evaluation
Team Formation (3p)	Thursday, May 16th	Moodle	-
One-page proposal	Tuesday, May 21st	PDF file in Moodle (1+1 page)	5%
Phase I: Decision Tree Supervised and Semi-Supervised	Thursday, June 6th	PDF file in Moodle (2 pages), Code in GitHub In person demo with TA	40%
Phase II: Final Report (6 pages) README.txt Sample test dataset One-page contribution	Tuesday, June 18th	ZIP file in Moodle Code in GitHub In person demo with TA	55%

Email Inquiries

Lecturer	Arash Azarfar	arash.azarfar@concordia.ca
TA/Lab/PoD	Maryam Valipour	maryam.valipour@concordia.ca
TA/Lab/PoD	Rose Rostami	rose.rostami@mail.concordia.ca

All inquiries about the course project should be communicated via *email* using the addresses above. Your email subject line must follow a prefix topic.
[COMP6721: *{your subject}*].
For example, if you are inquiring about the team formation, the subject line would be
[COMP6721: Course Project-Team Formation].

Main Objective

The main goal of this project is to study a Machine + Deep Learning (ML+DL) task using both Decision Trees and Convolutional Neural Networks to address a machine learning problem, applying supervised and semi-supervised learning Classification. Each team is at liberty to choose any image dataset that includes venue labels, such as airport, beach, hotel, university, restaurant, street, etc and five venues (five classes), under the following circumstances:

1. Availability of datasets online: you will select the same dataset for all the steps of the project. With the same dataset, you will implement the following learning tasks (in addition to required data preparation and visualization steps) and compare the performance.

I. Supervised learning Classification with Decision Trees
II. Semi-supervised learning Classification with Decision Trees
III. Supervised learning Classification with a convolutional neural network (CNN)

2. The datasets should have at least 2500 observations. You have the freedom of eliminating parts of the datasets if you need to, given that you still meet the requirements. You can also combine datasets from different sources together.

3. Among different classes (venues) addressed in the dataset(s), you select five venues, and you will do a 5-class classification.

Once a problem is chosen, each team needs to train, optimize, and evaluate a Decision Tree and a CNN architecture to tackle the chosen classification problem. For the semi-supervised part, you will train multiple Decision Trees iteratively taking into account the labeled data and more confident predicted labels. All CNN methods must be implemented using the PyTorch library. Specifically, the project has the following 3 main components:

I. Descriptive image analysis (EDA)
II. Image Preprocessing: In this stage, the team needs to explore several data preprocessing methods.
III. Optimization: Each team attempts to optimize their model through hyperparameter tuning. You should choose at least three hyper parameters for the tree (e.g., depth, number of branches, pruning option, etc) and two for the CNN (remove/add pooling layer(s), change the number of convolutional layers.)
IV.

Please note the following

1. You should consider the computational complexity of the selected network models that need to be trained on a commodity GPU hardware from available resources (e.g., lab GPU, cloud, personal PC, Google Colab, etc).

2. You should use several evaluation metrics, in addition to plots, to compare the performance of the different models. This includes, but not necessarily limited to, accuracy, recall, precision, F-score, and confusion matrix.

Team Formation

Students are required to form a team of *Three (3) members* for the course project. Please submit your team's details *on Moodle*. A *Q&A discussion forum* is created for the course on Moodle, and you can use the platform to open a discussion on team formation related topic. Students who cannot find a team will be randomly shuffled into incomplete teams. The team, once formed, will stay the same until the end of the semester.

Team Member Specialization. While all team members have to contribute equally to the project, you can consider the following roles in the team: A (I) Data Specialist, responsible for creating, pre-processing, loading & analyzing the datasets; a (II) Training Specialist, responsible for setting up and training the models; and a (III) Evaluation Specialist, responsible for analyzing, evaluating, and applying the generated model. Each specialist will write the corresponding part in the project report, detailed below. Note: this does not mean the designated person has to do all the work for the given task, but rather is mainly responsible for this work and can define and distribute sub-tasks to the other team members.

Team Issues. In hopefully rare cases where a team loses a member or there are conflicts, you should inform the TA(s) and the instructor as soon as possible. The evaluation will be adjusted to the situation.

Training Data (Reference: Summer 2022 Project Assignment #1)

Create datasets for training and testing your AI. You have to provide provenance information, i.e., where you obtained each image in your dataset. You should re-use existing datasets, but again please make sure you properly reference the source of the image datasets (name, author, source, license of the dataset). It is expected that you have a minimum of 2000 training images and (additionally) 500 testing images (across all classes). Note: This is before applying any data augmentation strategies. Make sure that both your training and testing data sets are balanced, i.e., have roughly the same number of images per class. Note that you will most likely have to perform suitable pre-processing (such as size-normalization) on your datasets. You must use real training data, i.e., using synthetic, generated data is not permitted.

Writing a One-Page Proposal

You should write a one-page proposal for the course project to cover the following topics:

- *Dataset Selection:* Explain your dataset(s) and provide the statistical details of your data (e.g., number of images, image size, etc). Specify where you have found the dataset and means of access to the data (e.g., published paper, downlink, etc). You may consult Kaggle, UCI, or Google Dataset Search to find possible datasets of interest. Note that venues should be diverse and general: They cannot be landmarks or different places of the house, for example.

- *Your five (5) selected classes (venues):* Mention the five venues that you have selected to do classification on, and the number of images available per class. Please make sure that your dataset for the selected five classes is, almost, balanced.

- *Possible Methodology:* Highlight the "possible methods" you could use to solve the problem. Specify how you will be handling/processing the data to train your deep learning pipeline using a CNN model, for instance. Furthermore, discuss the metrics that will be used to assess and evaluate the pipeline, and your expectations regarding the kind of results/performance to be achieved. You need to discuss the possible method(s) and how the obtained results will be compared and analyzed to each other.

- *Bibliography:* You can add an additional page (if needed) to extend your reference list cited in your proposal. The citations may include, but not limited to, published papers and domain links. (include a link to your dataset). Please note that failure to properly cite your references constitutes plagiarism and will be deemed for reporting.

Proposal Submission

Only the admin (one person) of your team needs to upload the proposal in *PDF* file in Moodle. For the report format, please consult "Reports Formatting" Section in the third page. You will be informed if the proposal can not be accepted (not a venue classification dataset, very trivial dataset with already available online codes, etc).

First Phase (Decision Trees)

Each team is required to submit a two (2)-page progress report highlighting the main steps taken after the proposal, and the results for the implementation of the decision trees (supervised and semi-supervised). The report should contain the following sections:

1. *Introduction and problem statement:* In addition to defining the problem and its applications, discuss the general strategy for tackling the issue at hand. Discuss the challenges faced in solving this problem and any possible solutions to address them.

2. *Proposed Methodologies:* Give updates regarding the methods used. Discuss the chosen dataset and the model in more detail than the proposal.

3. S*olving the problem:* Elaborate on failed and successful attempts at tackling the problem. Furthermore, discuss the results.

4. *Future Improvements:* Discuss briefly how and where you want to change to improve the accuracy of the model.

5. *References:* You can add an additional page (if needed) to extend your reference list cited in your progress report. The citations may include, but not limited to, published papers and domain

links (include a link to your dataset). Please note that failure to properly cite your references constitutes plagiarism and will be deemed for reporting.

6. *Supplementary Material* [this section is appended to the main report draft]:
You may include appendices to your report to support different sections of the main draft.

The progress report should be in PDF format and uploaded in Moodle. For the report format, please consult "Reports Formatting" Section in the third page. Please note only the admin (one person) of your team needs to upload the progress report in PDF file in Moodle.

Second (Final) Phase

Create a suitable Convolutional Neural Network (CNN) architecture, implement it in PyTorch, and train it using your dataset. You must implement the complete workflow in your CNN, that is, you cannot use any external libraries, except for the ones mentioned below. Your code must save the created model after training and be able to run a saved model on the test dataset, as well as an individual image (application mode).

Provide a comparison of the performance for the same model when you change hyper-parameters, and among the three models. You must automate the training/test-split in your project (e.g., do not use two separate, manually split datasets). You can use scikit-learn (including Skorch) for your evaluation process, but you must create your model using standard PyTorch. General Python modules like pandas, NumPy, etc., also are ok to use.

The final report will have the same sections but include all the solutions and results.

1. *Abstract.* Articulate the abstract presentation of the project and what to expect by reading your report in full detail. Briefly discuss the problem, proposed methods and used data, and the achieved results. [maximum of 150 words]

2. *Introduction* [the abstract & introduction should be no longer than 1.5 pages].
a) Write a section to cover the problem statement. What are the associated challenges with respect to the problem? How these challenges have been addressed in literature? What are the pros/cons of the existing solutions? How is this report trying to solve the problem and a challenge in mind? Elaborate on the high-level abstract explanation of your methodology and what kind of implementations you have done. What kind of results you are obtaining?
b) Related works. Write a short subsection to cover literature review and related work descriptions.

3. *Methodology* [this section should be no longer than 2 pages].

The methodology section should cover several subsections as follows:

a) *Datasets*. A comprehensive description of the datasets, including where and (how) there were collected, a complete statistical details, distribution and analysis of the datasets and

any preprocessing and filtering steps you have taken to make it ready to be fed to your models. Explain your train/validation/test breakdown, cross-fold validations, resolution level for training, etc

b) *Decision Tree Model*. Describe the architecture of the decision tree.

c) *CNN Model*. Describe the architecture of the selected CNN model. Elaborate on why you think the selected model is suitable for your practice.

c) *Optimization Algorithm*. Discuss how you validated and optimized your models (hyperparameters tuning). What optimization algorithm(s) you are choosing to train the CNN model? What metric evaluations are considered for reporting the performance of the optimization algorithm. Describe the properties of the algorithm and its associated hyper-parameters for training.

4. *Results* [this section should be no longer than 2.5 pages].

This section describes and analyzes the experimental design and obtained results in detail. More specifically

a) *Experiment Setup.* you need to describe how you setup your experiments, optimized and validated your models, the performance of your models using appropriate metrics (precision, recall, F1-measure, …). Explain the ranges of hyper-parameters and rational behind selecting as such in relation to your data and models.

b) *Main Results.* Demonstrate the main results in figure/table formatting and analyze the performance of your trained models, as well as comparison with other available results.

c) *Ablative Study*. Demonstrate the ablation results from tweaking different hyper-parameters such as number of classes for training, number of images per class training, different range of learning rates, different range of batch-size, tree depth, branching, pruning, etc, and explain your observations.

5. *References* [this section lists all references on your report]:
Cite any references you used in the projects, including any source code and dataset you have used in the project. Please note that failure to properly cite your references constitutes plagiarism and will be deemed for reporting.

7. *Supplementary Material* [this section is appended to the main report draft]:
You may include appendices to your final report to support different sections of the main draft.

Submission. You must submit your code electronically on Moodle by the due date (late submission will incur a penalty, see Moodle for details). Include a single Expectation of originality form (see https://www.concordia.ca/ginacody/students/academic-services/expectation-of-originality.html), (electronically) signed by all team members.

Reports Formatting

The proposal (1 page + 1 page bibliography), the progress report (2 pages+1 page bibliography), as well as the final report (6 pages + possible appendices) should **all** be written in IEEE 2-column conference format and submitted as PDF (You may use Word or LaTeX). Note to use the *reviewing style* for LaTeX compilation.

https://template-selector.ieee.org/secure/templateSelector/publicationType

https://cvpr2022.thecvf.com/sites/default/files/2021-10/cvpr2022-author_kit-v1_1-1.zip

Final Demo

There will be a live in-person demo session with a TA after each submission (Phase I and II). You will book a timeslot on Moodle. All three members of the team should attend this session.

GitHub Submission

Whether you use Git to organize your coding throughout the project or not, each team should create a new GitHub page for the project from the beginning. The GitHub page should be created in "private" mode and each member should be given access to commit their updates on a regular basis during the course of project. Furthermore, the assigned TA for the project team as well as the lecturer should be given access to the GitHub page for monitoring the progress of the team. Note that git commits from each team member will be monitored for the engagement of individuals and considered as one of the means of marking to contribute to their final project.

The final GitHub page should contain the following:

- High level description/presentation of the project
- Requirements to run your Python code (libraries, etc)
- Instruction on how to train/validate your model
- Instructions on how to run the pre-trained model on the provided sample test dataset
- Your source code package in Scikit-learn and PyTorch
- Description on how to obtain the Dataset from an available download link

Please note that if the instructions to run your code are incomplete or not explicit enough, you might lose marks for that part of the project. You should add the professor and the TAs as contributors to your project.

How to Submit Your Project Materials?

Submit all the files in one zip file including
- PDF file of the final report
- README.txt containing the following two links
  - A link to your GitHub page.
- A sample small test dataset
- **One page that includes a table listing the contribution of each team member to the project**. The table format should be in Three (3) columns pertinent to individual members of the team. The pertinent information will be considered to grade individual contribution to the project.

The zip file should be uploaded by the admin of the team in Moodle by the final submission deadline.

Late Submissions

If you submit any part of the project later than the specified deadline on Moodle, your submission will be accepted until the cut-off date. However, you will lose 20% of the mark for each day you submit late. The cut-off date is maxed up to two (2) days and submission after the cut-off date will not be accepted. Further, please note that resubmitting your files will result in erasing all the previously submitted versions and their respective dates. The date of the last attempt at submission will be counted as the final submission date.

APPENDIX 1 Semi-Supervised Learning

In an ideal word, data are labelled, meaning we have a large dataset with the value of target variables (e.g., image subject, object, text sentiment, etc) already determined for each observation. However, in the real world, it is usually very costly to label the data, so perfectly labelled datasets are rare! Consider the example of image recognition or image context prediction. It is easy to do a web search and find thousands of images (our features), but the data will be unlabeled. Companies may hire employees whose tasks are just browsing the images and assigning the right label.

Semi-supervised learning is a machine learning method (we may say a sub-method of supervised learning) which targets these situations where we have a large dataset, and the majority of the

observations are unlabeled. However, a low percentage of the observations are labelled (usually less than 20%).

The main idea in semi-supervised learning is as follows:

- We do supervised learning with the labelled data only (let's say 20%).
- The resulting model is applied to the unlabeled data, and unlabeled observations are pseudo-labelled (the rest 80%).
- Among the observations with pseudo-labels, we take the ones with a high confidence (e.g., the predicted probability instead of labels in Decision Tree, Logistic regression, or Naïve Bayes models) (Let's say top 10% predictions)
- These high-confidence observations are mixed with the originally labelled data to form the new labelled subset (now 20%+10% = 30%). Ignore the predicted pseudo labels for the rest.
- We re-run the steps above (We have now 30% "labelled" and 70% unlabeled).
- Usually, 5-10 iterations are required to finalize the labels (predictions) for all unlabeled data.

So, for your dataset,

- Put aside the same final test set (10-15%) that you used for the supervised learning (for a faire comparison).
- Among the others, randomly select 20% of data as labelled and ignore the labels of the 80% (consider them unlabeled).
- Make your supervised learning model and predict (pseudo-label) these 80% observations.
- Check the probabilities and select the ones with high confidence (e.g., =<0.15 and >=0.85 for a binary classification). Mix them with the labelled data and reiterate.

APPENDIX 2 Frequently Asked Questions (FAQ)

1) Should only one model or all three models mentioned in the description be implemented?
   You should build all these three models: supervised decision tree, semi-supervised decision tree which is the same as supervised with few additional iterative steps, and CNN.

2) Features for decision tree models with images
   For training a decision tree with the image, you can select your own features, so it depends on your dataset and your creativity. You may use pixels as features (similar to handwritten digits classification), a combination of pixels, or decide to include other features. For example, for image data, you may use the overall color or brightness of the image as features.

3) Using pretrained models (such as ImageNet, ResNet)
   The objective of the project is to build your CNN model from scratch, so pretrained models are not acceptable at this phase.

4) May we combine datasets?
   Yes, you can. You may naturally need some preprocessing to make images from several sources compatible.
5) May we use Tensorflow instead of PyTorch?
   No, the library to be used in this course is PyTorch.