School of Computing and Information Systems
The University of Melbourne
COMP30027, Machine Learning, 2025

Project 2: Traffic Sign Prediction

Task:	Build a classifier to predict the Traffic Signs based on image features.
Due:	Group Registration: Friday May 9, 11:59 pm Project: Friday 23 May, 7:00 pm
Submission:	Report (PDF) and code to Canvas; test outputs to Kaggle in-class competition
Marks:	The Project will be marked out of 20, and will contribute 20% of your total mark.
Groups:	Groups of 1 or 2, with commensurate expectations for each (see Sections 2 and 5).

1 Overview

The goal of this project is to build and critically analyze supervised Machine Learning methods for classifying German traffic signs. We have provided a subset of the GTSRB (German Traffic Sign Recognition Benchmark) dataset, which consists of traffic sign images and some basic extracted features, your task is to predict different traffic sign classes.

This assignment aims to reinforce theoretical concepts surrounding data representation, classifier construction, evaluation, and error analysis by applying them to an open-ended problem. You will also practice general problem-solving, programming, and analytical thinking skills.

2 Deliverables

The deliverables of the project are listed as follows. Details about deliverables are given in the Submission: 5

1. **Report**: a written report, of 1,300-1,800 words (for a group of one person) or 2,000-2,500 words (for a group of two people).
2. **Output**: the output of your classifiers, comprising the label predictions for test instances, submitted to the Kaggle[1] in-class competition described below.
3. **Code**: one or more programs, written in Python, which implement feature selection and machine learning models to make predictions and evaluate the results.

3 Data

The dataset is a subset of the German Traffic Sign Recognition Benchmark (GTSRB)[2]. The dataset comprises images from 43 classes representing various German traffic signs, including regulatory signs (e.g., speed limits), warning signs (e.g., curves, slippery roads), and mandatory instructions (e.g., turn left, go straight). The classification task involves correctly identifying the sign type from an image under real-world conditions such as varying lighting, angles, and occlusions. We have split the data further into two sets:

- **Training set**: Contains traffic sign images, extracted features along with the class labels.
- **Test set**: Contains traffic sign images and extracted features **without labels**, which you will need to predict.

In the dataset, in addition to the raw images the additional basic extracted features provided include:

[1] `https://www.kaggle.com/`
[2] `https://www.kaggle.com/datasets/meowmeowmeowmeowmeow/gtsrb-german-traffic-sign`

- *Color histograms*: Binned color information.
- *Histogram of Oriented Gradients*: Histogram of Oriented Gradients extracted from the grayscale images, then reduced via Principle Component Analysis (PCA).
- *Additional features*: Edge density, texture variance, average color channels.

You will be provided with a training set and a test set. The training set (consisting of 5488 rows) contains images and the `ClassId`, which is the "class label" of our task. The test set (consists of 2353 rows) only images without labels.

Students are provided with a training set and a test set:

- *train_metadata.csv*: the training image paths, and their labels (numeric class IDs).
- *test_metadata.csv*: the test image paths without labels.
- *color_histogram.csv, hog_pca.csv, additional_features.csv*: extracted features for training and test sets.

Important

Note: The students are expected to go beyond the provided features and get additional feature representations; **models previously trained or fine-tuned on the GTSRB dataset are not acceptable feature extractors**.

4 Task

You are expected to develop machine learning models that **classify traffic signs** based on the image-based and/or numeric features. You will **explore effective features, engineer additional features, implement and compare different machine learning models, and conduct error analysis**.

Various machine learning techniques have been (or will be) discussed in this subject (0R, Naive Bayes, Decision Trees, kNN, SVM, neural network, etc.); many more exist. You may use any machine learning method you consider suitable for this problem. *You are strongly encouraged to make use of machine learning software and/or existing libraries (such as `sklearn`) in your attempts at this project.*

In addition to different learning algorithms, consider different ways to preprocess and utilize the predictor variables. The additional files `color_histogram.csv`, `hog_pca.csv`, `additional_features.csv` are some possible representations fo the images, we have provided. For instance, feature engineering techniques like `normalization`, `standardization`, and `handling missing values` can significantly impact model performance. Feel free to apply any preprocessing steps or feature engineering techniques that you think could improve your model's ability to generalize from the training set to the test set. The ideal classification pipeline for trafic sign prediction is presented in Figure 1.

You are expected to complete the following two phases for this task:

- **Training-evaluation phase**: the holdout or cross-validation approaches can be applied on the training data provided.
- **Test phase**: the trained classifiers will be evaluated on the unlabeled test data. The predicted labels of test cases should be submitted as part of deliverable.

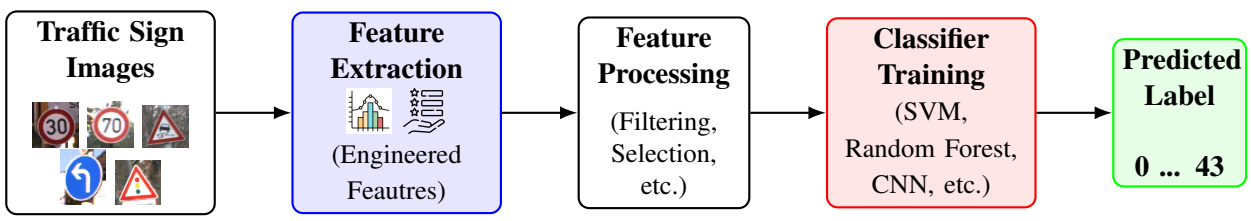

Figure 1: Traffic sign classification pipeline: input images must undergo feature extraction which are then used for classifier training.

5 Submission

The report and code should be submitted via Canvas; the predictions on test data should be submitted to Kaggle.

5.1 Individual vs. Team Participation

You have the option of participating individually, or in a group of two. In the case that you opt to participate individually, you will be required to implement **at least 2 and up to 4** distinct Machine Learning models. Groups of two will be required to implement **at least 4 and up to 5** distinct Machine Learning models, of which *one is to be an ensemble model – stacking based on the other models*. The report length requirement also differs, as detailed below:

Group size	Distinct models required	Report length
1	2–4	1,300–1,800 words
2	4–5	2,000–2,500 words

Group Registration

If you wish to form a group of 2, **only one** of the members needs to register by **Friday May 9, 11:59pm**, via the form "**Project 2 Group Registration**" on Canvas. For a group of 2, **only one** of the members needs to submit deliverables.

Note that after the registration deadline, you will not be allowed to change groups. If you do not register before the deadline above, we will assume that you will be completing the assignment as an individual.

5.2 Report

You report is expected to demonstrate the knowledge that you have gained and the critical analysis you have conducted in a manner that is accessible to a reasonably informed reader.

The report should be 1,300-1,800 words (individual) or 2,000-2,500 words (groups of two people) in length excluding reference list, figure captions and tables. The report should include the following sections:

1. Introduction: a basic description of the task and a short summary of your report.
2. Methodology: what you have done, including any learners that you have used, and features that you have engineered. *This should be at a conceptual level; a detailed description of the code is not appropriate for the report. The description should be similar to what you would see in a machine learning conference paper.*
3. Results: performance of your classifiers, in terms of evaluation metric(s) and, ideally include figures and tables.
4. Discussion and Critical Analysis: this section should include a more *detailed discussion* which contextualises the behaviour of the method(s), in terms of the theoretical properties we have identified in the lectures and error analysis of the method(s). This is the most important section of the report.
5. Conclusion: demonstrates your identified knowledge about the problem.
6. Reference: reference of related work.

Note that we are more interested in seeing evidence that you have thought about the task and investigated the reasons for the relative performance of different methods, rather than in the raw accuracy/scores of different methods. This is not to say that you should ignore the relative performance of different runs over the data, but rather that you should think beyond simple numbers to the reasons that underlie them, and connect these to the theory that we have discussed in this subject.

We provide LaTeXand Word style files that we would prefer that you use in writing the report. Reports must be submitted in the form of **a single PDF file**. If a report is submitted in any format other than PDF, we reserve the right to return the report with a mark of 0.

5.3 Predictions of test data

To give you the possibility of evaluating your models on the test set, we will be setting up a Kaggle in-class competition for this project. You can submit results on the test set there, and get immediate feedback on your model's performance. There is a Leaderboard, that will allow you to see how well you are doing as compared to other classmates participating online. The Kaggle in-class competition URL and instructions are provided on the LMS.

You will receive marks for submitting at least one set of predictions for the unlabelled test set into the competition; and get good accuracy (higher than 50%). The focus of this assignment is on the quality of your critical analysis and your report, rather than the performance of your Machine Learning models.

6 Assessment Criteria

The Project will be marked out of 20, and is worth 20% of your overall mark for the subject. The mark breakdown will be:

Report	18 marks
Performance of classifier / Kaggle	2 marks
TOTAL	20 marks

The report will be marked according to the rubric, which is published on the Canvas. You have to submit your code that supports the results presented in your report. If you do not submit an executable code that supports your findings, you will receive a mark of 0 for the "Report".

The performance of classifier is for submitting at least one set of model predictions to the Kaggle competition (1 mark); and getting better accuracy (higher than 50%) (1 mark).

Any submitted method that involves training on the test set (or simply looking up the ground truth labels for the Kaggle test set) will be considered cheating and will receive 0 marks.

7 Using Kaggle

The Kaggle in-class competition URL will be announced on Canvas shortly. To participate in the competition:

- Each student will have to use their student email to access the invite-only competition.
- You may make up to 8 submissions per day. An example submission file can be found on the Kaggle site.
- Submissions will be evaluated by Kaggle for accuracy, against just 50% of the test data, forming the public leaderboard.
- Prior to competition close, you may select a final submission out of the ones submitted previously – by default the submission with highest public leaderboard score is selected by Kaggle.
- After competition close, public test scores will be replaced with the private leaderboard test scores (100% test data).

8 Assignment Policies

8.1 Terms of Use

Please note that the dataset is a sample of actual data posted to the World Wide Web. As such, it may contain information that is in poor taste, or that could be considered offensive. We would ask you, as much as possible, to look beyond this to the task at hand.

If you object to these terms, please contact Basim Azam (basim.azam@unimelb.edu.au) as soon as possible.

8.2 Changes/Updates to the Assignment Specifications

We will use Canvas to advertise any (hopefully small-scale) changes or clarifications in the assignment specifications. Any addenda made to the assignment specifications via Canvas will supersede information contained in this version of the specifications.

8.3 Late Submissions

The submission mechanism will remain open for one week after the submission deadline. Late submissions will be penalised at 10% per 24-hour period after the original deadline.

To request an extension on this assignment, please see the **FEIT Extension Policy** and follow the steps below:

- **To request an extension of 1-3 business days (without AAP)**, complete the FEIT Extension Declaration Form at the website above and upload it to Canvas under *Assignment 2 extension request*. **We do not accept these forms by email in COMP30027**; the form must be uploaded to Canvas.
- **To request a longer extension (without AAP)**, please apply for Special Consideration.
- **If you have an AAP**, please request an extension by completing the *Assignment 2 extension request form* on Canvas and uploading your AAP.

Please note that we can only accept extension requests via Canvas up until the assignment deadline. Late extension requests can only be granted through Special Consideration. The longest extension granted on this assignment is 10 business days.

8.4 Academic Honesty

While it is acceptable to discuss the assignment with others in general terms, excessive collaboration with students outside of your group is considered collusion. Your submissions will be examined for originality and will invoke the University's Academic Misconduct policy (`https://academicintegrity.unimelb.edu.au/`) where either inappropriate levels of collaboration or plagiarism are deemed to have taken place.

Since Kaggle competition performance contributes to the final mark on this assignment, your submissions to Kaggle must be your own original work and must abide by the rules of the Kaggle competition. Submitting predictions to Kaggle which are not the results of your own models (or allowing another student to submit your model's predictions under their Kaggle account) will be considered a breach of the university's Academic Misconduct policy, as will any attempts to circumvent the rules of the Kaggle competition (for example, exceeding the competition's daily submission limit).