CP 318: Data Science for Smart City Applications

August-Nov Semester 2022, Project 1 Spec

Due: Report due Monday 11:59PM IST, October 10th 2022
Competition closes 11:00am IST (5:30 AM UTC) same day

Competition Link: `https://www.kaggle.com/t/e0e180c77a91459fbd3955a6aa2d47e9`

Weight: 20% of final mark

Introduction

Traffic sign detection and recognition have received an increasing interest in the last years. It is a core component of traffic sign assist system for providing timely instructions and warnings to the drivers regarding traffic restrictions and information. Also, it can provide the required contextual awareness for the self-driving vehicle.

In this project, you will develop a machine learning model/framework for classifying traffic signs most of which are specific to India. **Note that** unlike developed countries, India has many different types of traffic signs out there, each with different colours, shapes and sizes. Sometimes, there are two signs may have a similar colour, shape and size, but have two totally different meanings. Your job is to identify for a given image (as feature vectors) of a traffic sign the identity of that sign.

You may use any existing ML models which are already publicly available for similar datasets, as long as it is properly justified or acknowledged. You are also encouraged to introduce some innovative techniques/methods. There will be 1 bonus mark for using such innovative approach (which was not taught in the classes and workshops, and its use is properly justified in the report).

To make the project fun, we will run it as a Kaggle in-class competition. Kaggle is one of the most popular online platforms for predictive modelling and analytics tasks. You will be competing with other students in the class. The following sections give more details on data format, the use of Kaggle, and marking scheme. Your assessment will be based on your final ranking in the competition, the absolute score that you achieve, and your report. The marking scheme is designed so that you will pass if you put in effort. So fear not and embrace the power of machine learning!

Data Format

You will have access to three types of files, primarily train_data.csv, test_data.csv, and sample_submission.csv. These files are available on the Class Teams under "Project" Channel. File train_data.csv contains about 932 images along with the class (label) column. Each row corresponds to a traffic sign image and a small number of other features, and the columns have a meaning as shown below:

Column(s)	Name	Meaning
0	ID	An integer identifier which is unique within a file for each image.
1	Class	An integer identifier the traffic sign type (See Table 2 for their meanings)
2 - 2501	C1 - C2500	Pixels of the 50 by 50 image stored in greyscale, with each pixel an integer value in range $[0, 255]$. The image matrix has flattened into a vector; for viewing, reshape these elements to the original image size (e.g., using np.reshape and imshow in matplotlib.) Note that images have been preprocessed by re- moving noise, mapping to greyscale, and rescaled to a uniform size

.

Next, file test_data.csv contains 400 records (images) with the same fields as above, but with all "Class" values set to a place- holder value of 0, because this is what you need to predict in this project. Train and test data comprise two non-overlapping sets of IDs, even though in both files there might be records from the same classes.

Finally, sample_submission.csv shows the beginning of an example submission file. The test_data.csv and sample_submission.csv comprise same IDs in their first column in the same order. Once you have done predictions for each ID in the test_data.csv file, you should create a submission file in CSV format with the following structure.

```
Id,Class
1869, 26
2221, 27
3971, 4
...
```

The first line should be a header, exactly as shown in sample_submission.csv. There should be 400 rows (excluding header row) in total, each with a unique ID. The IDs of predictions should match the IDs of entries in the test file, in the same order.

Note that you **should not** inspect the test data closely or hand label the data in submission file by just inspecting the test data, as this is cheating and compromises the point of the project (though please inspect your submissions to ensure your files are in the right format, of the right size, etc.). Note that you will have to provide your code in your submission (discussed later) and we will run it at our end to ensure you have not done any such cheating.

As we provide no explicit validation set, you may want to reserve part of the training partition for this purpose during model development. Your job is to develop an algorithm that can automatically capture the nuances of the problem, in order to generalise well to unseen data (estimated here over the test set.)

Kaggle In-class Competition

Link: `https://www.kaggle.com/t/e0e180c77a91459fbd3955a6aa2d47e9`

Please do the following by the end of the first week after receiving this assignment:

- Setup an account on Kaggle with username and email being your IISc student email.
- Form your team of student peers (Note that: Some or all teams may be formed by the Course Instructor to make sure each team has student(s) with some prior experience with programming)
- Connect with your team mates on Kaggle as a Kaggle team, using a team name in `CP-318-[team-name]` format. You can choose any team name e.g., Shaktimaan, Spyderman etc. Only submit via the team; and

- Register your team using the Google forms (will be shared soon)

You should only make submissions using the team name, individual submissions are not allowed and may attract penalties. Note that teams will be limited to 5 submissions per day.

The real labels for the test data are hidden from you, but were made available to Kaggle. Each time a submission is made, half of the predictions (50% of the test data) will be used to compute your public score and determine your rank in public leaderboard. This information will become available from the competition page almost immediately. At the same time, the other half of predictions is used to compute a private accuracy and rank in private leaderboard, and this information will be hidden from you. At the end of the competition, only private scores and private ranks will be used for assessment, and will be revealed publicly. This type of scoring is a common practice and was introduced to discourage overfitting to public leaderboard. A good model should generalize and work well on new data, which in this case is represented by the portion of data with the hidden accuracy.

The evaluation score used in this competition is the accuracy over all classes, defined as the number of instances labelled correctly as a fraction of the total number of instances. Before the end of the competition each team will need to choose 3 best submissions for scoring. These do not have to be the latest submissions. Kaggle will compute a private accuracy for the chosen submissions only. The best out of the 3 will then be automatically selected and this private score and the corresponding private leaderboard ranking will be used for marking.

Each participant can do maximum 3 submissions everyday. Before the end of the competition, each of you will need to choose your 2 best submissions for final scoring. These do not have to be the latest submissions. Kaggle will compute a private accuracy for the chosen submissions only. The best out of the 2 will then be automatically selected and this private score and the corresponding private leaderboard ranking will be used for marking. If you don't choose any submission, Kaggle will by default consider your best submission performance on public leaderboard for computing the private accuracy.

Report

Each team will submit a report with the description, analysis, and comparative assessment (where applicable) of the method or methods used. There is no fixed template for the report, but it should start with a very brief introduction of the problem and notation used. Then the report should describe the approaches that you have attempted along with the motivation for trying them. Reflect on why the method(s) performed or didn't perform well. If you tried different models or different hyperparameters, compare the methods to each other in the context of this competition. Your reasoning can be in the form of empirical evaluation, but it must be to support your reasoning (examples like "method A with X features and Y value of parameter for accuracy 0.60 and method B, got accuracy 0.7, hence we use method B", with no further explanation, will be marked down).

If you used any feature transformations, selected only some useful features, or generated new features, you should also describe them in the report along with the expected effect from using such features and effect observed after implementation and evaluation. In comparing methods, you may want to use an evaluation besides measuring accuracy, in order to better understand the kinds of mistakes being made (e.g., with rare (minor) classes.)

Your description of the algorithms should be clear and concise. You should write it at a level that a postgraduate student can read and understand without difficulty. If you use any existing algorithms, you do not have to rewrite the complete description, but must provide a summary that shows your understanding and references to the relevant literature. In the report, we will be very interested in seeing evidence of your thought processes and reasoning for choosing one algorithm over another. Dedicate space to describing the features you considered and tried, class distributions, validation, sampling techniques (if used), any interesting details about software setup or your experimental pipeline, and any problems you encountered and what you learned. In many cases these issues are at least as important as the learning algorithm, if not more important.

The report should be submitted as a PDF, and be no more than three A4 pages of content, including all plots, tables and references[1] (single column, font size of 11 or more and margins at least 1 cm, much like this document). You do not need to include a cover page. **If a report is longer than three pages in length, we will only read and assess the report up to page three and ignore further pages.**

Submission and Assessment

In summary, each student is required to make the following submissions for this project:

- One or more submission files with predictions for test data (at Kaggle). This submission must be of the expected format as described above, and produce a place somewhere on the leaderboard. Invalid submissions do not attract marks for the competition portion of grading;
- Report in PDF format (via "Assignment" Section for this project in Class Teams);
- Source code used in this project as a single ZIP archive (via "Homework" Section in "Class Notebook" in Class Teams). Your code can be in any of the following languages C, C++, Python, Jupyter Notebook, R or MATLAB. If there is another language you like to use, please contact us first. If the language requires compiling, a makefile or script must be provide to build the executables. We may or may not run your code, but we will definitely read. You should not include the training or test data file in the ZIP file.

The project will be marked out of 30. No late submission of Kaggle portion will be accepted; late submissions of reports will incur a deduction of 3 marks per day, or part thereof. Based on our experimentation with the project task and the design of the marking scheme below, we expect that all reasonable efforts at the project will achieve a passing grade or higher. So relax and have fun!

Marking Scheme

Kaggle competition (15 marks) This mark takes into account both achieved accuracy, as well as your standing in the class. Assuming N is the number of students, and R is your rank in the class, the mark you get for the competition part is

$$12 \times \frac{\max\{\min(acc, 0.92) - 0.20, 0\}}{72} + 3 \times \frac{N - R}{N - 1}$$

The first term constitutes up to 12 marks, and rewards high accuracy systems with a maximum score for excellent systems with $\geq$ 92% accuracy, and zero score to those with scores $\leq$ 20% which are just little better than random guessing. The second term, worth 3 marks, is based on your rank and is designed to encourage competition and innovation. Ties are handled so that you are not penalised by the tie. All who are tied will get the same marks for score, but ranking will be decided based on total number of submission entries. The score with less entries will be ranked higher among tied ones.

External teams of unenrolled students (auditing the subject) may participate, but their entries will be removed before computing the final rankings and the above expression, and will not affect registered students' grades. Note that invalid submissions will come last and will attract a mark of 0 for the score, so please ensure your output conforms to the specified requirements, and have at least some kind of valid submission early on!

Report (15 marks) The report will be marked using the rubric in Table 1.

[1] Plots can be useful for model selection, assessing convergence, features importance, displaying results and model interpretation, among other things. For instance, plotting the parameters of your model with respect to the objective function can often give insights into what the model has learned.

Critical Analysis (8 marks)	Report Clarity and Structure (7 marks)
7–8 marks Final approach is well motivated and its advantages/disadvantages clearly discussed; thorough and insightful analysis of why the final approach works/not work for provided training data; insightful discussion and analysis of other approaches and why they were not used	*6–7 marks* Very clear and accessible description of all that has been done, a postgraduate student can pick up the report and read with no difficulty
5–6 marks Final approach is reasonably motivated and its advantages/disadvantages somewhat discussed; good analysis of why the final approach works/not work for provided training data; some discussion and analysis of other approaches and why they were not used	*4–5 marks* Clear description for the most part, with some minor deficiencies/loose ends (e.g., there are no- table gaps and/or unclear sections)
3–4 marks Advantages/disadvantages discussed; limited analysis of why the final approach works/not work for provided training data; limited discussion and analysis of other approaches and why they were not used	*2–3 marks* Generally clear description, but there are notable gaps and/or unclear sections.
1–2 marks Final approach is barely or not motivated and its advantages/disadvantages are not discussed; no analysis of why the final approach works/not work for provided training data; little or no discussion and analysis of other approaches and why they were not used	*1 mark* The report is unclear on the whole, omits all key reference,and the reader can barely discern what has been done

Table 1: Report marking rubric.

Bonus Mark (1 mark) you will get 1 bonus mark if you have used any ML model which was not taught in the classes/workshops before the submission deadline, or if you have used any innovative techniques in that improves your model performance on test data. You need to provide this information in your report with proper justification, to get this 1 bonus mark.

1	BULLOCK PROHIBITED	9	LEFT HAND CURVE	17	RIVER BANK	25	STEEP DESCENT
2	CATTLE	10	NARROW BRIDGE	18	RIGHT HAND CURVE	26	STRAIGHT PROHIBITED
3	CYCLE CROSSING	11	NARROW ROAD AHEAD	19	RIGHT TURN PROHIBITED	27	TRUCK PROHIBITED
4	CYCLE PROHIBITED	12	NO PARKING	20	ROAD WIDENS AHEAD	28	T INTERSECTION
5	DANGEROUS DIP	13	NO STOPPING OR STANDING	21	ROUNDABOUT	29	U-TURN PROHIBITED
6	FALLING ROCKS	14	OVERTAKING PROHIBITED	22	SCHOOL AHEAD	30	Y INTERSECTION
7	HEIGHT LIMIT	15	PEDESTRIAN PROHIBITED	23	SLIPPERY ROAD	-	-
8	HORN PROHIBITED	16	PRIORITY FOR ONCOMING VEHICLES	24	STEEP ASCENT	-	-

Table 2: Mapping between class numbers and traffic sign

Plagiarism policy

You are reminded that all submitted project work in this subject is to be your own individual work. Automated similarity checking software will be used to compare submissions. It is University policy that cheating by students in any form is not permitted, and that work submitted for assessment purposes must be the independent work of the student(s) concerned. For more details, please see the policy at `https://iisc.ac.in/about/student-corner/academic-integrity/`.