Univ.-Prof. Dr.-Ing. Matthias Boehm
Technische Universität Berlin
Faculty IV - Electrical Engineering and Computer Science
Berlin Institute for the Foundations of Learning and Data (BIFOLD)
Big Data Engineering (DAMS Lab) Group

1 AMLS SoSe2023: Exercise – Star/Galaxy Classification

Published: Apr 26, 2023 (last update: Apr 26)
Deadline: Jul 04, 11.59pm

This exercise is an alternative to the AMLS programming projects, and aims to provide practical experience in the exploratory development of machine learning (ML) pipelines. The task is to classify coordinates in image patches of the Sloan Digital Sky Survey as stars or galaxies. You may use any programming language(s) of your choosing, and utilize existing open source ML systems and libraries. The expected result is a zip archive named `AMLS_Exercise_<student_ID>.zip` (replace `<student_ID>` by your student ID) of max 5 MB, containing:

- The source code used to solve the individual sub-tasks
- A PDF report of up to 8 pages (10pt), including the names of all team members, a brief summary how to run your code, and a description of the solutions to the individual sub-tasks.

Data: We will be using data from the Sloan Digital Sky Survey (SDSS). A description of the data and how to download it can be found at https://github.com/damslab/datasets/tree/master/stars.

Grading: This exercise can be pursued in teams of 1 to 3 persons (one submission). The overall grading is a *pass/fail* for the entire team. Exercises with $\geq 50/100$ points are a pass, and the quality expectations increase with the team size.

1.1 Data Acquisition and Alignment (15/100 points)

Obtain the dataset mentioned above, extract the following input data for your ML pipeline, and materialize these inputs as files. Read the description of the dataset carefully, and see the SDSS website for additional details. The different spectral bands require alignment. You may use gray-scale encoding of each spectral band (combined with the IRG images), and work with small images or patches to simplify their handling. Subsequently, compute meaningful summary statistics and visualization (e.g., a comparison of aligned and unaligned images).

- A tensor with a channel for each of the aligned spectral bands.

  Dimensions [N, H, W, C] or [N, C, H, W] – N: Number of Images, H: Image Height, W: Image Width, and C: # Channels

- Two tensors that contain the pixel coordinates of each star and galaxy in the individual images.

  [N, G, 2] – G: Number of Galaxies, and 2: x and y coordinate in picture

  [N, S, 2] – S: Number of Stars, and 2: x and y coordinate in picture

Expected Results: Code for data acquisition and alignment, as well as statistics and plots describing the data.

Figure 1: 301/8162/6/frame-irg-008162-6-0080.jpg

1.2 Data Preparation (25/100 points)

As a prerequisite for model training and testing, split the data into train/validation/test sets of similar data distributions (e.g., label distribution and potentially other properties such as visual similarity or data provenance). For the sake of comparison, everyone is required to use the spectral bands associated with 301/8162/6/frame-irg-008162-6-0080.jpg (see Figure 1) as test data (not in raw form but using the same data acquisition and alignment steps from Task 1.1). Make sure that the test data is not leaking into the train and validation sets. Therefore, your code should the able to exclude data from specific sky-coordinates when creating the train and validation sets. Again, consider working with smaller patches of the input (i.e., reduced H and W) in order to reduce memory requirements and ensure reasonable training times.

Expected Results: Code for data preparation of disjoint train, validation, and test data with similar distributions. The report should justify the chosen selection procedure and how it preserves the similarity of specific properties.

1.3 Modeling and Tuning (35/100 points)

Construct an ML pipeline—using the prepared train and validation sets—for classifying specified coordinates as stars or galaxies. Choose an appropriate loss function and evaluation metric, and evaluate this metric on both the train and validation data. Example models to use include (but are not limited to) U-Net, SegNet, FastFCN, Gated-SCNN. Subsequently, tune the hyper-parameters of your model (e.g., regularization parameters). You may use different sizes of your model for faster exploration. To further simplify training, coordinates may also be mapped to instances of small patches (e.g., 5x5 pixel). Parallelize the hyper-parameter tuning and model training (e.g., by appropriate configurations). Finally, report the evaluation metric on the train, validation, and test data (again without leaking the test data into model training or hyper-parameter tuning).

Expected results: Code for model training and runing, as well as descriptions of the used model architecture, ML pipeline, and its evaluation. The report must include the statistics or plots of the quality of your model with and without tuning.

1.4 Data Augmentation (25/100 points)

Further improve your model quality via data augmentation techniques such as rotations, reflections, modulated noise, zoom levels, shearing/distortions, or synthetic stars and galaxies. Only data preparation steps should be added, whereas the model training and hyper-parameter tuning should remain unchanged. Conduct systematic experiments on the train and validation sets, and properly document the intuition and impact on model quality of the applied techniques. Avoid changing too many techniques and their parameters at once. In order to reduce the runtime of these experiments you may use proxy models that are not trained to full convergence. Finally, evaluate the quality of your model with and without data augmentation on the train, validation, and test sets.

Expected Results: Code for applying data augmentation techniques and a summary of their impact on model quality.

1.5 Extra Credit

As an opportunity for extra credit, we provide a list of optional bonus points that can be given.

- Provide scripts to setup your code and reproduce your results. A suggestion is to use Docker for simplifying installation and execution. [10 points]
- Describe and document how hard it is to distinguish stars and galaxies based on your experience with this task. [2 points]
- Use explainable AI techniques to create explanations of your models predictions (for instance, using Integrated Gradients or Grad-CAM). [8 points]