Final Project Option 2: Image Classification

Data 100/200A: Principles and Techniques of Data Science

Fall 2019

The purpose of this project is to put into practice what you have learned in this course through the design and implementation of an image classification workflow, including data manipulation, visualization, exploratory data analysis, feature selection, class prediction, and predictor performance assessment.

To make the project more self-contained, you are provided with a learning set of "real-world" images of 20 different types (e.g., "dog", "goat") and your task is to train and evaluate several predictors for the class of an image. You will then evaluate your proposed final classifier using a test set for which we have withheld the class labels.

Dataset

The learning set consists of a total of 2,921 images, of 20 different types and of possibly different sizes (i.e., numbers of pixels). Each image is represented as 3-d array with the first two dimensions corresponding to the row and column pixels and third dimension to the color. The third dimension size is always 3, and each value corresponds to a red, green, or blue (RGB) color intensity between 0 and $2^8 - 1$.

The learning set can be found in the 20_categories_training folder in the project starter directory.

The test set can be found in the 20_validation folder in the project starter directory.

Project Guidelines

The project involves carrying through the following steps.

1. **Data input.**
   - Read in all the provided learning and test set images.
   - Store the images in two data frames, one for the learning set (with class labels) and one for the test set (without class labels).
2. **Exploratory data analysis and feature extraction.** (Learning set only.)
   - Display three of the learning set images.
   - Provide graphical summaries of the sizes of the images, pixel intensities, and class frequencies.
   - Provide functions that summarize pixel intensity data (e.g., `https://docs.opencv.org/3.0-beta/doc/py_tutorials/py_feature2d/py_table_of_contents_feature2d/py_table_of_contents_feature2d.html#py-table-of-content-feature2d`). Compute at least 15 such image features (a method for each), including the following (NOTE: At least 10 of these must be scalar features and 2 matrix-based features): (i) image size, (ii) average of the red-channel intensity, (iii) aspect ratio.
   - Examine how these image features vary between classes.

3. **Classifier training.** (Learning set only.) Using all or a selected subset of the features from 2) and all or a selected subset of individual pixel intensities, build the following classifiers using only the learning set.

   - Logistic regression: `http://scikit-learn.org/stable/modules/generated/sklearn.linear_model.LogisticRegression.html`.
   - k-nearest neighbors (kNN): `http://scikit-learn.org/stable/modules/generated/sklearn.neighbors.KNeighborsClassifier.html`.
   - Classification tree: `https://scikit-learn.org/stable/modules/tree.html`.
   - Random Forests: `http://scikit-learn.org/stable/modules/generated/sklearn.ensemble.RandomForestClassifier.html`.
   - Support vector machines (SVM): `http://scikit-learn.org/stable/modules/svm.html`.

   Feel free to generate a combination of these models for your final predictions.

4. **Classifier performance assessment.** (Learning set only.) Evaluate the performance of the classifiers in 3) using five-fold cross-validation of the learning set to estimate the misclassification error rate, i.e., perform all the tasks in 3) (including feature selection) on the training sets and compute misclassification rates on the validation sets.[1]

5. **Neural networks.** (Optional. Extra credit: 5% of project score, but your total project score cannot exceed 100%.) Build a neural network classifier using an architecture of your choosing. This application of deep learning can be done in PyTorch, TensorFlow, or a framework of your choice. This is the industry standard for image classification. Describe your network and assess its performance. To receive extra credit, your neural network classifier must outperform your other methods.

Report Format and Submission

The project submission should include the following three components, to be submitted on Gradescope as a zip file: Jupyter Notebooks, narrative, and CSV file of test set predictions.

1. **Jupyter Notebooks.** Use the provided starter notebooks to complete the following aspects of the project.

   (a) Data input.

   (b) Exploratory data analysis and feature extraction.

   (c) Classifier training and performance assessment.

   (d) Neural networks (Extra credit).

   Note: We will run the notebooks in that order when grading, so please account for that. Additionally, transferring data between notebooks is part of the project, and thus copying and pasting final data frames will result in deductions.

2. **Project narrative.** This typed PDF document should summarize your workflow and what you have learned. It should be structured as a research paper and include a title, list authors, abstract, introduction, description of data, description of methods, summary of results, and discussion. Make sure to number figures and tables and include informative captions. Specifically, you should address the following in the narrative.

   Note: There is a page limit of 6 pages, excluding figures and tables.

[1] Terminology: In cross-validation, the learning set is divided into training and validation sets. The test set is never used.

- Frame the question.
- Describe the data.
- Perform exploratory data analysis (EDA) and provide data visualizations.
- Describe any data cleaning or transformations that you perform and why they are motivated by your EDA.
- Carefully describe the methods you are using and why they are appropriate for the question to be answered.
- Summarize and interpret your results (including visualization). Address the following three specific questions. (i) What were two or three of the most interesting features you came across? Describe the process of finding those feature. (ii) Describe one feature you thought would be useful, but turned out to be ineffective. (iii) Describe the differences in the classifiers that you used. Why did some work better than others? Which turned out to be the most effective?
- Provide an evaluation of your approach and discuss any limitations of the methods you used.
- Describe any surprising discoveries that you made and future work.

3. **Image class predictions for test set.** Run the classifier of your choice on the test data of 716 images and generate a CSV file of the classification for each image. Submit this CSV file on Gradescope. It is your responsibility to follow the order of the files when creating the CSV (predict validation_1 before validation_2...).

   Note 1: You are not to use the test data in any way for training or creating your classifer.

   Note 2: You are not allowed to use a neural network for your final classifier.

Grading. You will be graded based on: The code (Jupyter Notebooks), the narrative, and the accuracy of your final classifier on the test set.

Team work. You must complete the project together with one other classmate. You will both be graded equally.