Visualizing the World: A Computer Vision Data Science Project

Description: This project will consist of a large dataset including many different images of different categories. Your task is to identify key distinguishing features of these images and develop models to classify them in Python. We will explore several machine learning techniques including Logistic Regression, KNN, SVM, Random Forest, and TensorFlow.

1. Data Cleaning
    a. Take care of all imports
    b. Read in all the files, and add them to a data-frame with the image, and the encoding for that feature
    c. Reference: Starter Jupyter Notebook
2. Feature Extraction
    a. Select between 15 and 20 image features to train your model on.
        i. This will require significant exploratory research and EDA
        ii. Suggested Reading: 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
    b. Produce at least three unique/ interesting graphs you encountered through EDA
    c. Implement the two pre-described features
        i. Image size is already implemented
        ii. The average of the red-channel pictures for the images
        iii. The aspect ratio of the image
        iv. Features should be implemented in the same way, a method for each
3. Model Training (Past-Experience)
    a. Split your data into training and test, ensure cross-validation
    b. For each of the following algorithms, train your model on the data, make sure given an image, you can automatically calculate all the features, and return a number that maps to a certain classification.
        i. Logistic Regression
            Documentation: http://scikit-learn.org/stable/modules/generated/sklearn.linear_model.LogisticRegression.html
        ii. KNN
            Documentation:http://scikit-learn.org/stable/modules/generated/sklearn.neighbors.KNeighborsClassifier.html
        iii. Random Forest
            1. Documentation:http://scikit-learn.org/stable/modules/generated/sklearn.ensemble.RandomForestClassifier.html
        iv. SVM
            1. Documentation: http://scikit-learn.org/stable/modules/svm.html

    c. Test your accuracy throughout the process, with and without different features, optimizing as much as possible.
4. Tensorflow
    a. Tensorflow (Optional)
        i. Use Tensorflow, and the methods in that package to categorize your images. The accuracy should be significantly higher than your features due to the usage of neural nets.
        ii. This is an industry standard, and if you want to do any work in the industry, we would highly recommend it.

Teams: For this project, you are required to work in teams of 2-3. If you need an exception, please reach out and we will consider it.

Submission: You will submit three parts, 3-4 Jupyter Notebooks, a typed report and a csv.

1. Jupyter Notebooks
    a. Data-Cleaning
    b. Feature Extraction
    c. Training Models
    d. TensorFlow (Optional)

    Note: We will be running them in that order when grading, so please account for that.
2. Report
    a. You will need to comprise a two page report, summarizing the process of the entire project and what you learned.
    b. Some key questions that must be answered
        i. What were two or three of the coolest/ unique features you came across? Describe the process of finding that feature.
        ii. Describe one feature you thought would be useful, but turned out to be ineffective.
        iii. Describe the differences in the modeling techniques that you used. Why did some work better than others? What turned out to be the most effective?
3. Gradescope
    a. Run the model of your choice on the test data providing, and generate a csv of classifications for each image.
    b. Submit on gradescope.
    c. NOTE: you are NOT to use the test data in any way for training/creating your model. Doing so may result in a 0 for this section.
4. Grading
    a. You will be provided with a rubric, but three main focuses
        i. Report
        ii. Code

iii. Model Accuracy

b. Test Accuracy
    i. When checking your results with our test data, we will run the following code-block