Setting up Environment

* Devise Notebook organization
  + Imports
  + Global variables (adjustable parameters)
  + Helper Functions
  + Step 0: Loading data
    - Training data
    - Testing data
  + Step 1: Data visualization
    - Histogram of training data
  + Step 1a: Generating data
  + Step 1b: Visualize new dataset
  + Step 2: Design and Test a Model Architecture
  + Step 2a: LeNet
  + Step 2b: Modified LeNet
  + Step 3: Test a Model on New Images
* Determine library imports based on general pipeline
  + Load data
    - os
    - pickle
  + Examine data content and classification distribution
    - matplotlib
    - textwrap
  + Generate new data (as needed) based on distribution
    - cv2random
  + Preprocess data
    - numpy
    - skimage
    - sklearn
  + Devise model
    - tensorflow
  + Test on additional data samples
    - PIL

Creating Helper Functions­

-

Visualizing Dataset

-

Generating Fake Data

* Split raw data into Train/Valid sets
  + Pros: no train data spill into valid data set
  + Cons: diversity of training model is not guaranteed
* Split data into Train/Valid sets after generating fake data
  + Pros: diversity of image samples
  + Cons: train data spill into valid data set causes overfitting
* Split data into Train/Valid sets according to class shares based on raw data class distribution and generate fake data (based on Train data partition) for modeling
  + Pros: guarantees diverse data modeling and prevents train data spill into valid data set
  + Cons: data split threshold may determine accuracy results for classes of low training samples
    - E.g., class of raw sample size 1 will have 0 validation samples to assess validation accuracy