
DSC 140B - Homework 08

Due: Wednesday, March 4

Instructions:

- Write your solutions to the following problems **by hand**, either on another piece of paper that you scan or using a tablet. Typed solutions will not be accepted for credit!
 - Code listings are an exception. You do not need to handwrite code, and you can instead include the code as a screenshot.
- Unless otherwise noted by the problem's instructions, show your work or provide some justification for your answer.
- Homework problems are graded pass/fail on completeness and effort, not correctness.
- Homeworks are due via Gradescope at 12:00 AM.

Problem 1. (3 credits)

As a data scientist you will have the opportunity to work on problems that are of great importance to society. This is not one of those problems.

The *menu-match* dataset consists of 646 images of food from three different restaurants: an Asian restaurant, an Italian restaurant, and a soup restaurant. The data set was constructed by employees of Microsoft Research¹.

The data set is available at the following link:

<https://f000.backblazeb2.com/file/dsc-data/menu-match.npz>

The file is in compressed numpy format; it can be loaded with 'np.load'. Once loaded, it behaves like a dictionary with four keys: `X_train`, `X_test`, `y_train`, and `y_test`, corresponding to the training data, test data, training labels, and test labels, respectively. For example, to get the training data:

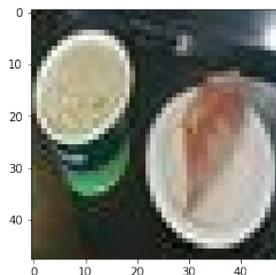
```
>>> data = np.load("menu-match.npz")
>>> data['X_train']
```

The training data is a 480 x 48 x 48 x 3 tensor, since it consists of 480 images, each 48 x 48 pixels with 3 colors.

Note that the given labels are strings: "a" for Asian restaurant, "i" for Italian, and "s" for soup.

You may display an image with `matplotlib` using `plt.imshow`. For example:

```
>>> # plot test example #30
>>> plt.imshow(data['X_test'][30])
```



¹<http://neelj.com/projects/enumatch/>

- a) Using PyTorch, train a convolutional neural network to predict whether a given image is from the Asian restaurant or not (thus turning the problem into a binary classification problem). You will need to determine the network architecture, but the example code given in lecture is a good starting point. Report the test accuracy – your model should be able to get above 70% of the test set correct (preferably higher!). Show your code.

In this section, you should not use any pre-trained models or transfer learning; you should train the network from scratch.

- b) Now use a pre-trained model (e.g. ResNet) and transfer learning to solve the same problem. The model you use is up to you; ResNet-18 is a good starting choice. Report your code and your results.
- c) Perform a qualitative error analysis of your best model by looking at the test examples that it got wrong. Do you notice any patterns in the errors? For example, are there certain types of images that the model struggles with? Are there any common features among the misclassified images?