# TensorFlow skill testing project: Image Recognition in TensorFlow

**Objective:** Tackle an open problem in image recognition: the Google Street View Housing Numbers dataset.

**Architecture:** Your code will consist of three modules. This document contains the specification for the third module.

---

## Module 3: `learn.py`

This module will run the actual training loops. It should allow you to specify training batch sizes, the algorithm's learning rate, and the number of training epochs. A **training epoch** is one run through all your training data, so two training epochs would mean that you effectively train your algorithm on each example in your dataset twice.

This module should contain a function with the following signature:

**`train_model(model, num_epochs, batch_size, learning_rate) => (no output)`**

| | |
|---|---|
| **`model`** | a string giving the name of the model to be trained (possible values: `'logistic'`, `'2-layer'`, or `'knn'`) |
| **`num_epochs`** | an `int` giving the number of training epochs |
| **`batch_size`** | an `int` giving the number of examples in each training batch |
| **`learning_rate`** | the learning rate to be used in the training step |
| **`train_model`** | the function that trains the model **`model`**, using the parameters **`num_epochs`**, **`batch_size`**, and **`learning_rate`**. This function should use the `matplotlib` library to plot training and testing batch accuracy as a function of batch number. **Note:** you shouldn't be checking test set performance for every batch, as this will slow down training unnecessarily. Instead, feed in some testing data every few batches, such that you end up with at least 10 test set accuracy values by the end of your training. |

Your module can contain additional functions, but you must have **`train_model()`**somewhere in your script.

Once you've completed this exercise, send it to **@yazabi** and we'll give you feedback on your code!