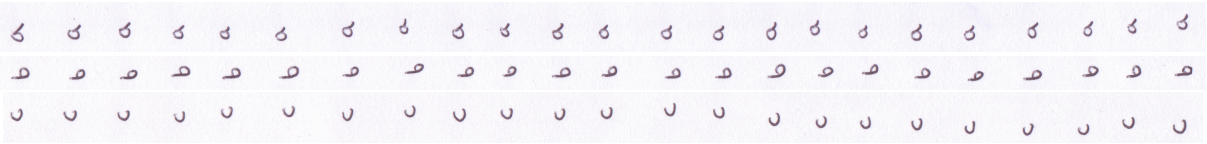
Instructions:
In this assignment, you will train a Support Vector Classifier to read your own handwriting. Submit your code (a `.zip` of your script `hw8.py` and handwriting data image files) on CCLE.

1. Before you begin, you might want to see the example Recognizing Hand-Written Digits from the scikit-learn documentation. You need to conceptually understand what the data and target arrays represent before you continue. In the example, the data and target arrays are provided. Training and testing the SVC is trivial once you have these arrays. Therefore, much of the time of your time on this assignment will be spent generating data and target arrays for your own handwriting.

2. In order to train the SVC, you'll need to provide it with several handwritten examples of each symbol you want it to learn, and you'll need to format this information like the data and target arrays in the example. For example, if we want our SVC to learn to recognize handwritten letters "a","b", and "c" , we need to write letters in a grid like:



3. Then we need to manually separate the image into separate files for each letter and write an algorithm to automatically separate the image into an array of images, each tightly cropped around a single symbol like:

We then convert these into the data array required by NumPy, where the number 0 is used to represent "a", 1 to represent "b", and 2 to represent "c" in the target array.

Feel free to use the startup code in `hw8_separate.py` for converting a column (rotated above to fit on page) of single characters into a list of separate images using a very simple algorithm. The code also resizes each of the letters to a 10×10 square.

4. Generate your own handwriting data: using a dark pen on a white sheet of paper, write in a column several examples of a single symbol. Do this for at least three different symbols. Scan or photograph (cell phone cameras can work) your data and separate it into separate images for each symbol. Generate data and target arrays for your handwriting data. Despite the fact that I provide 23 examples of each letter, five examples of each symbol should be sufficient if your handwriting is consistent.

5. Once you have arrays of all your own handwriting data and target values, you need to separate these into test and training sets. Write a function `partition` that accepts your data and target arrays and a third parameter p representing the percentage of the data that is to be used for training; the remainder will be for testing. It should return `train_data, train_target, test_data,` and `test_target`.

6. Train and test a Support Vector Classifier (SVC). You can use either LinearSVC or a regular SVC. Print results like:

```
Predicted: [ 2. 0. 1. 1. 2. 2.]
Truth: [2. 0. 1. 1. 2. 2.]
Accuracy:  100.0 %
```