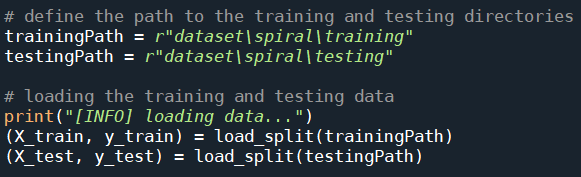
**Loading Train Data And Test Data**

After importing the necessary libraries, the next step is to define the training path and testing path.

Our dataset contains both hand-drawn spiral and wave patterns. Here we are taking spiral patterns into consideration and training the model.

We split the data into train and test. Using the training dataset we train the model and the testing dataset is used to predict the results.



**Note:** Can try the same with wave pattern dataset

**load\_split()**

The load\_split  function accepts a dataset path and returns all feature data and associated class labels.

* From there we grab input image paths by making use of imutils .
* Initialising data and labels lists.
* loop over all image paths which we have grabbed in the previous step.
* Each label is extracted from the os.path.split() method in Python which is used to split the pathname into a pair head and tail. Here, the tail is the last pathname component and the head is everything leading up to that.
* Each image is loaded and preprocessed.
* Read the input image
  + **Imread()** a pre-defined method is used to read the input image
* **Convert image to grayscale**
  + cv2.cvtColor(frame, flag): is used for color conversion.
  + cv2.COLOR\_BGR2GRAY: The flag is used to convert the coloured image to grayscale.
  + we convert the image to grayscale to reduce the processing time
* **Resizing the image**
  + resize it to 200x200 pixels,
* **Threshold image**
  + The thresholding step segments the drawing from the input image, making the drawing appear as a white foreground on a black background.
  + Please refer to this [link](https://opencv-python-tutroals.readthedocs.io/en/latest/py_tutorials/py_imgproc/py_thresholding/py_thresholding.html)for more information on image thresholding concept.
  + Features are extracted via our quantify\_image  function.
  + The features and label are appended to the data and labels lists respectively.
  + Finally, data and labels are converted to NumPy arrays and returned conveniently in a tuple.

