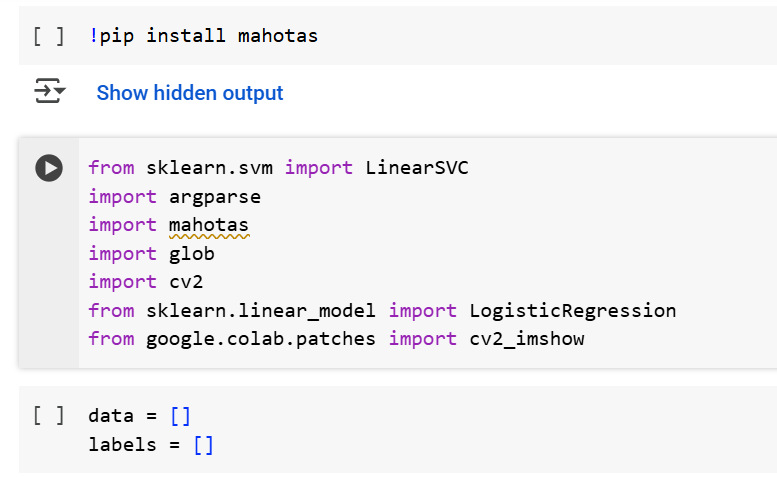
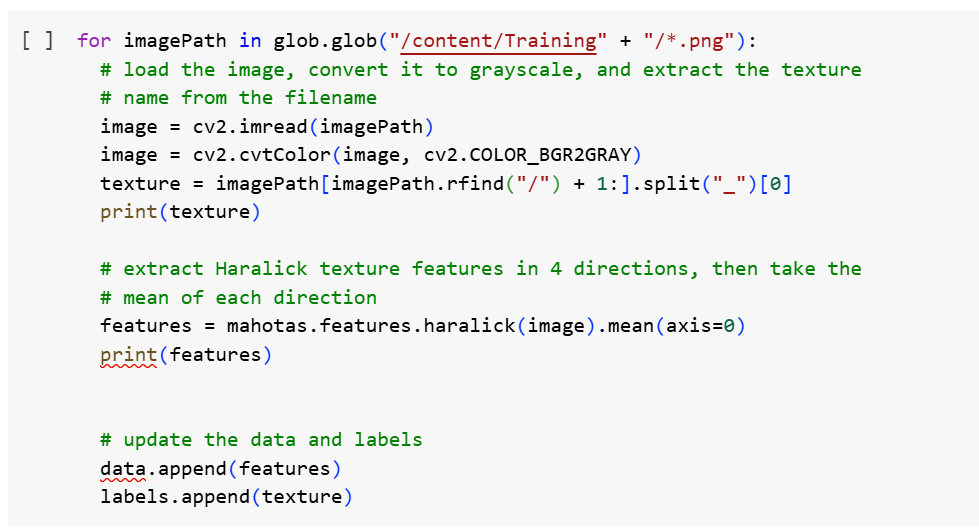
**Image Processing basis Texture**

**Code:**

****

****

****

**Details:**

**Glob Library**

Glob library in python is used to pick files with pathnames that match a specified pattern,

**It's a part of Python's standard library, so no additional installation is needed.**

It is used for pattern matching with wildcard characters to match filenames and paths.

* + \*: Matches zero or more characters.
  + ?: Matches any single character.
  + []: Matches any single character within the brackets.
  + [!...]: Matches any single character not within the brackets.



**CV2 Library**

CV2 library in python is used for image processing. It is used to load the image and then perform action on the image data.

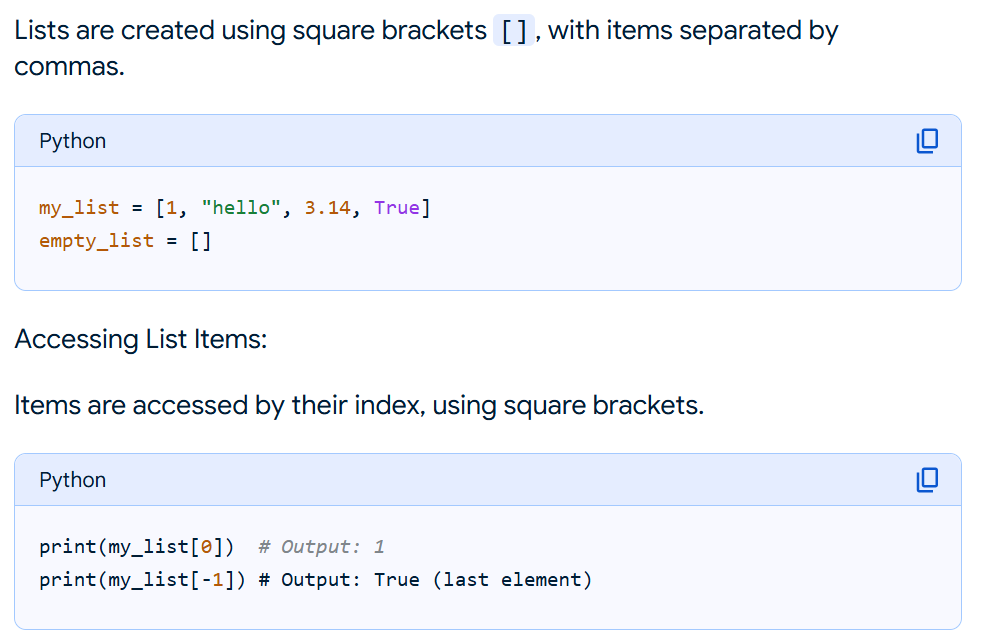


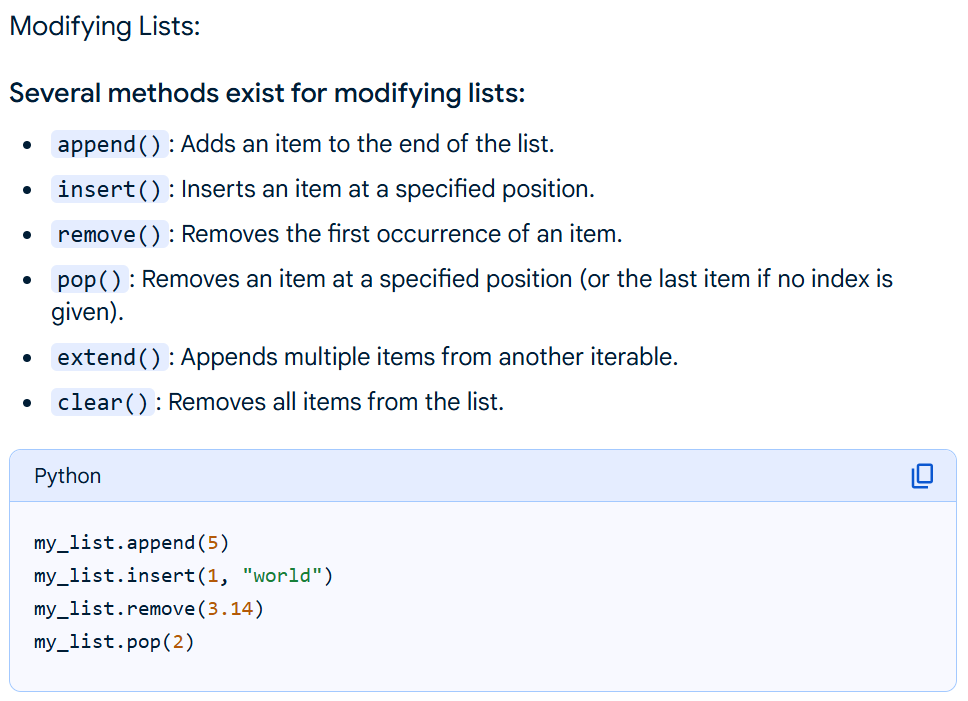
**Data Structures**

Data Structures in the world of Python is manages in 3 forms of collections or arrays of Data.

1. List – [ ] (Square brackets)
2. Tumple
3. Dictionary

**List**

****

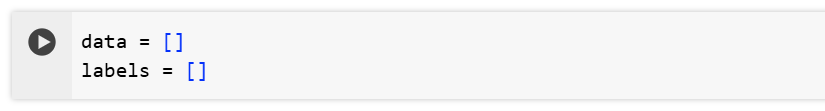
****

**HARALICK TEXTURE – Training**

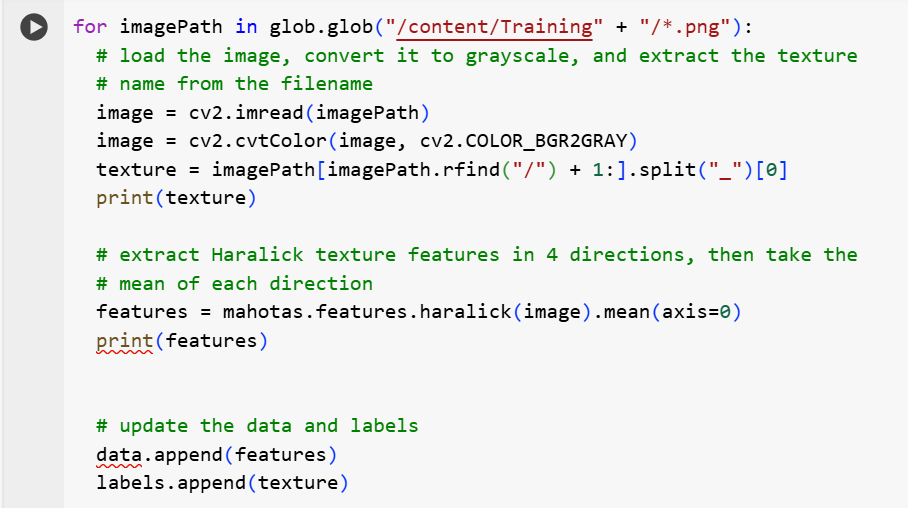
****

Step 1

Two List data structures are created:



Step 2



Each ***imagepath*** in the folder Training is pick one by one using Glob.

Image is picked from path and added to the variable ***image***

Same image is then coverted to GREY-SCALE and restored into the same variable ***image***

The portion of the imagename of the image is filtered from complete pathname sorted in variable ***imagepath*** and stored in variable ***texture*** as ID for identification and later appended into the **List variable *Labels[ ]***

**Now extraction of the Feature**

**# extract Haralick texture features in 4 directions,**

**then take the mean of each direction**

**features = mahotas.features.haralick(image).mean(axis=0)**

**Understanding Haralick**

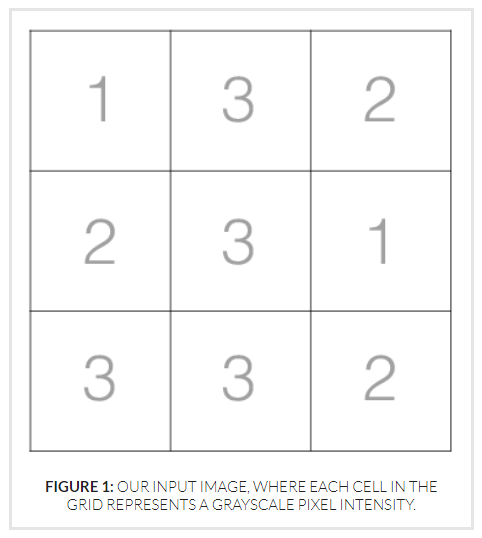
**What are Haralick texture features?**

Haralick features are used to describe the texture of an image. Texture refers to the appearance, consistency, or “feeling” of a surface. Examples of textures include “rough” vs. “soft.” Potential applications of Haralick features include determining if a road is paved vs. gravel.

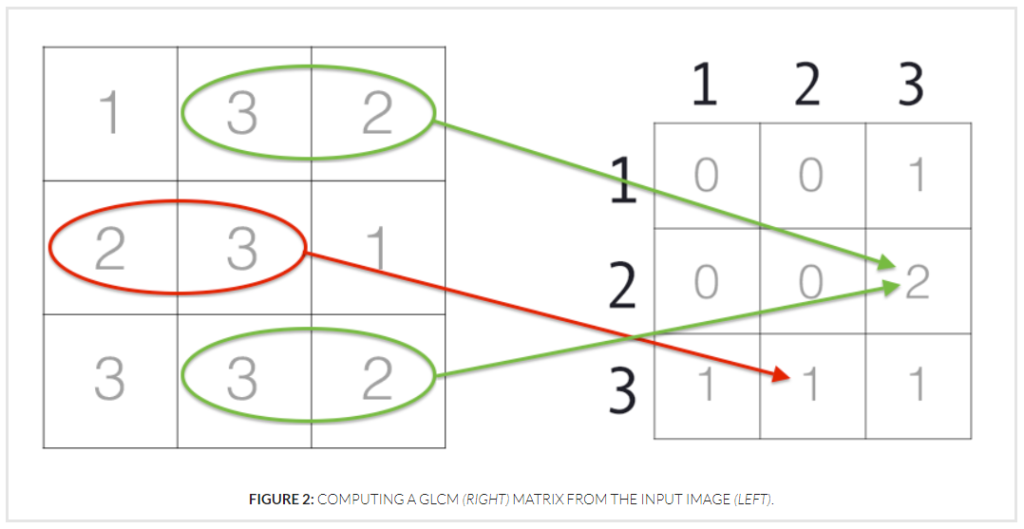
**How do Haralick texture features work?**

Haralick texture features are computed using the Gray-Level Co-occurrence Matrix (GLCM). This matrix characterizes texture by *recording how often pairs of adjacent pixels with specific values* occur in an image.

To understand how the GLCM works, take a look at the following figure, where the values in each block represent the pixel intensity of a grayscale (single channel) image:



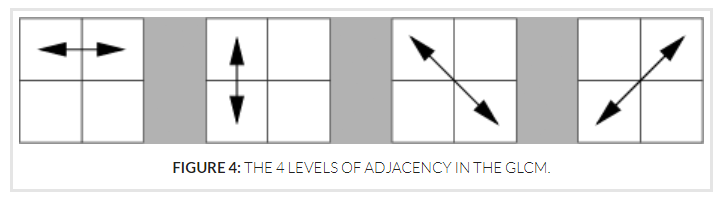
Then, to construct our GLCM, we *look at pairs of adjacent pixels* and record the number of times these two values appear next to each other:



Here we can see the pixel intensities *2 and 3* appear next to each other *once*, so they have an entry of *1* in the GLCM. The pixel intensities *3 and 2*; however, appear next to each other *twice*, so they have a corresponding entry of *2*in the GLCM.

However, we are not just limited to recording the number of times a pixel value appears to the *left or right*of a given pixel.

We *can actually specify four different****directions of adjacency***: *left to right*, *top to bottom*, *top-left to bottom-right*, and *top-right to bottom-left*:



This leaves us with a total of 4 GLCM matrices that we can use to compute Haralick features.

Now that we have these 4 GLCMs, we can *compute our Haralick features for each of the GLCMs*. Again, these values are simply statistics computed from the GLCM used to characterize and represent contrast, correlation, dissimilarity, entropy, homogeneity, and other desirable statistical properties.

After computing these statistics for each of these GLCMs we are left with 4 feature vectors (one row per direction), each of 13-dimensionality (or 14-dimensionality, depending on whether or not you computed the final 14th statistic). [ 4 x 13 matrix ]

Finally, we take the ***average of these directions*** to form a final feature vector of 13-dimensionality. This averaging is performed in an attempt to make the feature vector more robust in changes in rotation. We’ll demonstrate this averaging in the next section of this article. [ 1 x 13 matrix ]

**Where are Haralick texture features implemented?**

Haralick texture features are implemented in the ***mahotas***Python package. You can also compute the GLCM explicitly using the scikit-image library.

|  |  |
| --- | --- |
| 1  2  3 | import mahotas  gray = cv2.cvtColor(image, cv2.COLOR\_BGR2GRAY)  features = mahotas.features.haralick(gray).mean(axis=0) |