Feature Engineering

Papangkorn Inkeaw, Ph.D.

Feature Extraction for Image

Lab 6

Read Image File

Dataset: Hand Sign Images Dataset (https://www.kaggle.com/datasets/ash2703/handsignimages)

The data set includes 27,455 gray-scale images of size 28*28 pixels.

- Import libraries

```
import cv2
import numpy as np
from skimage import feature
import glob
```

- Load an image file

```
imgGray = cv2.imread('Your image file’)
#imgGray = cv2.cvtColor(imgRBG, cv2.COLOR_BGR2GRAY)
r,c = imgGray.shape
```

From Image to Feature Vector

- Flatten the image

```
raw_vector = imgResized.flatten()
print(raw_vector.shape)
```

- Calculate the HOG of the image

```
hog_vector = feature.hog(imgResized, orientations=9, pixels_per_cell=(8, 8), cells_per_block=(2, 2),
    block_norm="L1")
print(hog_vector.shape)
```

- Perform Canny edge detection

```
edges = cv2.Canny(image=imgResized, threshold1=100, threshold2=200)
```

- Calculate Moments

```
moments = cv2.moments(edges)
```

- Calculate Hu Moments

```
moments_hu = cv2.HuMoments(moments)
moments_vector = moments.flatten()
print(moments_vector.shape)
```

From Image to Feature Vector

- Create a function for construct histogram of features

```
def hist_feature(im, block_size = 8):
    r,c = im.shape
    n_bins = int(im.max() + 1)
    feature = []
    for i in range(0, block_size, r ):
             for j in range(0, block_size, c):
                     hist,_ = np.histogram(im[i:np.min(i+block_size-1,r-1),
                                      j:np.min(j+block_size-1,c-1)], bins= bins)
                      feature.append(hist)

     return np.array(feature)
```

- Calculate LBP array

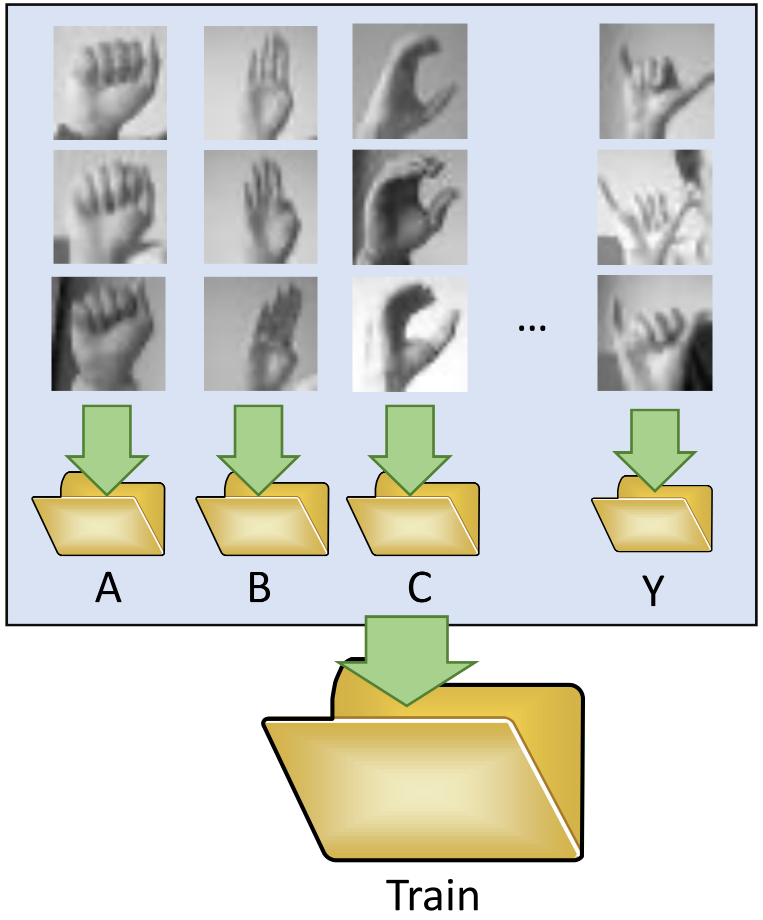
```
radius = 3
n_points = 8 * radius
lbp = feature.local_binary_pattern(imgGray, n_points, radius, ‘default’)
```

From Image to Feature Vector

- Extract LBP feature vector

```
lbp_vector = hist_feature(lbp, block_size = 8)
```

Dealing with Many Image Files



Dealing with Many Image Files

- List all JPEG files in all subfolder in the corpus

```
filenames = []
y_train = []  #class labels list

for dirName in glob.glob(”Train/*/"):          #List all subfolders in the folder Train
  for imgFile in glob.glob(dirName+"*.jpg"):
    filenames.append(imgFile)
    y.append(..). #append class name to list y_train
```

- Retrieve each image and process it

```
x_train = np.empty((0,feature_len), dtype=float)
for imgFile in filenames:
  img = cv2.imread(imgFile)
  # extract feature vector here
 feature_vector =
  # append the vector to x_train
 x_train = np.append(x_train, feature_vector, axis=0)
```

Your work!

1. Load the Hand Sign Images Dataset from https://www.kaggle.com/datasets/ash2703/handsignimages
2. Extract feature vectors of samples on both training and test sets
3. Construct a classifier using the training samples
4. Evaluate performance of the classifier on the test set
5. Submit your program to the assignment submission system (http://hw.cs.science.cmu.ac.th/).

Note:

- Put your name and student ID in the first cell using comment tag.
- Name your python notebook file with the pattern Lab_06_XXXXXXXXX.py (XXXXXXXXX is your student ID)

References & Study Resources

- https://www.kaggle.com/datasets/ash2703/handsignimages
- https://www.learnopencv.com/histogram-of-oriented-gradients/
- https://www.pyimagesearch.com/2015/12/07/local-binary-patterns-with-python-opencv/
- https://scikit-image.org/docs/dev/api/skimage.feature.html
- https://www.learnopencv.com/shape-matching-using-hu-moments-c-python/