ENEL4AI H2 - Artificial Intelligence - 2018

Classification using Neural Networks
Assigned on: 06 August 2018
Due date : Monday 10 September 2018

1 Goal

Artificial Neural Networks(ANN) is a popular and effective machine learning solution to many classification problems. In this project, You will learn the different steps that are needed to design and use ANN. It is then used to classify manufactured items. Data sets are provided.

2 Data preparation

In this project we will be comparing two features extraction methods, Local Directional Pattern(LDP) and Gray Level Co-occurrence Matrix (GLCM), to characterise surface and identify defectuous items .

2.1 Local Directional Patterns

Given a grayscale image, write a module in C++ and OpenCV that extracts Local Direction Pattern (LDP) features[3] of this image. Local Direction Pattern (LDP) computes the edge response values in different directions and uses these to encode the image texture. To calculate the eight directional edge response values of a particular pixel, Kirsch masks, shown in Fig 1, are used.

$$\underset{M_0(\text{East})}{\begin{bmatrix} -3 & -3 & 5 \\ -3 & 0 & 5 \\ -3 & -3 & 5 \end{bmatrix}} \quad \underset{M_1(\text{North East})}{\begin{bmatrix} -3 & -3 & 5 \\ -3 & 0 & 5 \\ -3 & -3 & 5 \end{bmatrix}} \quad \underset{M_2(\text{North})}{\begin{bmatrix} -3 & 5 & 5 \\ -3 & 0 & 5 \\ -3 & -3 & -3 \end{bmatrix}} \quad \underset{M_3(\text{North West})}{\begin{bmatrix} 5 & 5 & 5 \\ -3 & 0 & -3 \\ -3 & -3 & -3 \end{bmatrix}}$$

$$\underset{M_4\ (\text{West})}{\begin{bmatrix} 5 & 5 & -3 \\ 5 & 0 & -3 \\ -3 & -3 & -3 \end{bmatrix}} \quad \underset{M_5(\text{South West})}{\begin{bmatrix} 5 & -3 & -3 \\ 5 & 0 & -3 \\ 5 & -3 & -3 \end{bmatrix}} \quad \underset{M_6(\text{South})}{\begin{bmatrix} -3 & -3 & -3 \\ 5 & 0 & -3 \\ 5 & 5 & -3 \end{bmatrix}} \quad \underset{M_6(\text{South Est})}{\begin{bmatrix} -3 & -3 & -3 \\ -3 & 0 & 5 \\ -3 & 5 & 5 \end{bmatrix}}$$

Figure 1: Kirsch masks

Given a pixel (x,y) of an image, $\mathbf{I}$. For each direction i and using the corresponding mask M_i, the i^{th} directional response $m_i(x,y)$ can be computed as

$$m_i(x,y) \quad = \quad \sum_{k=-1}^{1}\sum_{l=-1}^{1} M_i(k+1,l+1) \times I(x+k,y+l) \tag{1}$$

A vector $(m_0,\ldots,m_7)$ is derived from each of the eight directions, where for each pixel (x,y) m_i represents $m_i(x,y)$. The k most significant responses are selected from the directional response vector. Thus, placing the corresponding positions to 1 bit code, leaving the remaining $(8-k)$ to 0 bit code. The LDP code, $LDP_{x,y}$, of the pixel (x,y) base on the directional response $(m_0,\ldots,m_7)$, is derived using Equation 2.

$$LDP_{x,y}(m_0, m_1, ..., m_7) = \sum_{i=0}^{7} s(m_i - m_k) \times 2^i \tag{2}$$

where m_k is the k^{th} most significant response and $s(x)$ is defined as

$$s(x) = \begin{cases} 1 & x \geq 0 \\ 0 & otherwise \end{cases} \tag{3}$$

From the LDP transformed image, an histogram, H, is extracted. H is defined as

$$H_i = \sum_{x=0}^{M-1} \sum_{y=0}^{N-1} p(LDP_{x,y}, C_i) \tag{4}$$

Where C_i is the i^{th} LDP pattern value, $i = 1, ..., {}^8C_3$ and p is given as

$$p(x,a) = \begin{cases} 1 & \text{if } x = a \\ 0 & \text{otherwise} \end{cases}$$

Given a gray level image, I, and the number of significant bits k, a feature vector $ldb_{k,I}$ is generated and represented as

$$ldp_{k,T} = (H_1, H_2, \ldots, H_{56}) \tag{5}$$

The generated vector is then saved in a text file. It means, if you are given 100 images you will create a text file of 100 lines.

2.2 Gray Level Co-occurrence Matrix

Given a grayscale image, write a module in C++ and OpenCV that extracts Haralick features[2] of this image. Haralick features are extracted from the GLCM (Gray Level Co-occurrence Matrix); they are used for modeling textural characteristics. The GLCM, $GLCM(i, j|d, \theta)$, encodes the spatial dependencies of tonal intensities for a given distance and orientation, providing a basis for extraction of second-order statistical features. Given an image I with spatial dimensions $M \times N$ and L grey levels, the GLCM is defined as

$$GLCM(i, j|d, \theta) = \sum_{x,y} G\{I(x,y) = i \text{ and } I(x + d\theta_0, y + d\theta_1) = j\} \tag{6}$$

where,
$0 \leq x \leq M-1$, $0 \leq y \leq N-1$, $0 \leq i, j \leq L-1$ and

$$G(x) = \begin{cases} 1, & \text{if x is true} \\ 0, & \text{otherwise} \end{cases} \tag{7}$$

The orientation θ is quantized to four values , which are represented as shown in Eq. (8)

$$\theta = \begin{cases} 0^\circ, & \text{if } \theta_0 = 0 \text{ and } \theta_1 = 1; \\ 45^\circ, & \text{if } \theta_0 = -1 \text{ and } \theta_1 = -1; \\ 90^\circ, & \text{if } \theta_0 = 1 \text{ and } \theta_1 = 0; \\ 135^\circ, & \text{if } \theta_0 = 1 \text{ and } \theta_1 = -1; \end{cases} \tag{8}$$

This work uses all four orientations shown in Eq. (8) and two distances, $d \in [1,2]$. Eqs. 9-13 compute six Haralick features extracted from the GLCM matrix. We will assume for the rest of the text that, as soon as the orientation θ and the distance d are chosen, $GLCM(i, j|d, \theta)$will just be represented by $GLCM(i, j)$ and we can compute $p_{i,j} = \frac{GLCM(i,j)}{\sum_{i=0}^{L-1}\sum_{j=0}^{L-1} GLCM(i,j)}$.

$$\text{Energy} = \sum_{i,j=0}^{L-1} p_{i,j}^2 \tag{9}$$

$$\text{Homogeneity} = \sum_{i,j=0}^{L-1} \frac{p_{i,j}}{1+|i-j|} \tag{10}$$

$$\text{Contrast} = \sum_{i,j=0}^{L-1} p_{i,j}|i-j|^2 \tag{11}$$

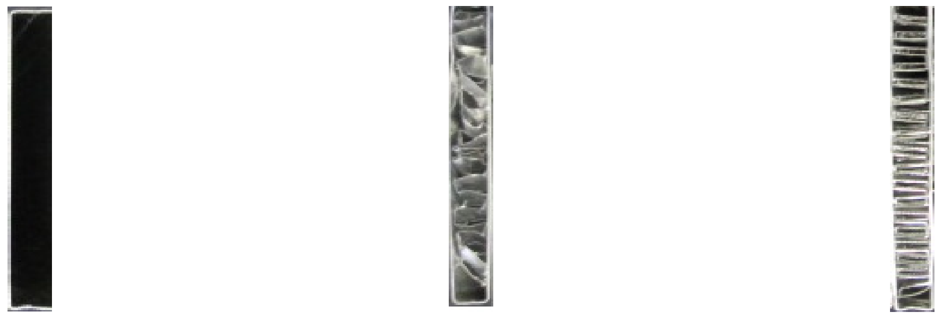
$$\text{Correlation} = \sum_{i,j=0}^{L-1} \frac{p_{i,j}(i-\bar{\mu}_i)(j-\bar{\mu}_j)}{\sigma_i \sigma_j} \tag{12}$$

$$\text{Entropy} = \sum_{i,j=0}^{L-1} p_{i,j}(-\ln p_{i,j}) \tag{13}$$

Given a gray level image, one can generate a vector of 40 (2 distances x 4 orientations x 5 features) features, and save it in a text file. It means, if you are given 100 images you will create a text file of 100 lines.

3 Data set

The data set used consists of images of manufactured items to be classified in one of the three following classes: empty, bad, good [see Figure 2]. This data set is split into a **training set** (made $\frac{2}{3}$ of the entire data set), and a **test set** (made $\frac{1}{3}$ of the entire data set). The training set files are saved in a folder named **train**, and the test set files are saved in a folder named **test**. File names in each of the files start with img_ followed by the category of the image (empty, good, bad), followed by a number (example: img_bad1.jpg, means the image is the **bad** image number 1, and its format is jpeg).



(a) Empty (b) Bad (c) Good

Figure 2: Examples Different categories of items to be classified

4 Design and Implement a Multi-Layer Perceptron to classify items

The task in this project is to build a system to classify images using Neural Networks based on Haralick and LDP features extracted from them.

1. Read Lecture 8 and related materials on Neural networks.

2. Study the OpenCv Neural Networks library to understand how it can be used to learn and classify objects using MLP.

3. Design and implement a Multilayer perceptron (MLP),MLP1, using OpenCV and C++, that uses Haralick features extracted in section 2.2 to classify a given item as empty, bad or good.

4. Design and implement a Multilayer perceptron (MLP), MLP2, using OpenCV and C++, that uses Local Directional Patterns features extracted in section 2.1 to classify a given item as empty, bad or good.

5. Perform experiments to test and compare performances of MLP1 and MLP2, implemented above, using the following steps:

   (a) Read image files from the source.

   (b) Compute Haralick and LDP features using the modules implemented in sections 2.2, and 2.1 respectively.

   (c) Train MLP1 an MLP2 with the training set prepared in section 3.

   (d) Test with test set prepared in section 3.

   (e) Compute how many items are correctly classified and also how many are wrongly classified for each class, when MLP1 and MLP2 are used. Build the confusion matrix, and calculate True Positive and the False Positive rates [1], for MPL1 and MLP2 and compare the results obtained and discuss.

SUBMISSION INFORMATION

Programming Language : C++ and OpenCV

Your submission must include the following:

(a) Your source code and executable.

(b) A report using the template provided.

Submitting the project

Submit your soft copy at : **http://learn.ukzn.ac.za** and a hard copy at office 5:03 (Electrical Engineering Building North) Make sure you include your details in all documents related to your submission.

References

[1] Fawcett, T., ROC graphs: notes and practical considerations for data mining researchers. *Technical Report* HPL-2003-4, Palo Alto, CA: HP Laboratories, 2003.

[2] Haralick, R.M. and Shanmugam, K., 1973. Textural features for image classification. *IEEE Transactions on systems, man, and cybernetics*, (6), pp.610-621.

[3] Jabid, T., Kabir, M. H., and Chae, O., 2010. Robust facial expression recognition based on local directional pattern. *ETRI journal*, 32(5), pp. 784-794.