Image Classification Based on Textural Features using Artificial Neural Network (ANN)

Prof S K Shah, Fellow
V Gandhi, Non-member

Image classification plays an important part in the fields of Remote sensing, Image analysis and Pattern recognition. Digital image classification is the process of sorting all the pixels in an image into a finite number of individual classes. The conventional statistical approaches for land cover classification use only the gray values. However, they lead to misclassification due to strictly convex boundaries. Textural features can be included for better classification but are inconvenient for conventional methods. Artificial neural networks can handle non-convex decisions. The uses of textural features help to resolve misclassification. This paper describes the design and development of a hierarchical network by incorporating textural features. The effect of inclusion of textural features on classification is also studied.

Keywords: Gray values; Neural classifier; Supervised classifier; Textural features

IMAGE CLASSIFICATION
Digital image consists of discrete picture elements called pixels which are associated with a digital number represented as DN that depicts the average radiance of relatively small area within a scene. The range of DN values is normally 0 to 255. Digital image processing is a collection of techniques for the manipulation of digital images by computers.

Classification generally comprises four steps:
- Pre-processing — eg, atmospheric correction, noise suppression, and finding the band ratio, principal component analysis, etc.
- Training — selection of the particular feature which best describes the pattern.
- Decision — choice of suitable method for comparing the image patterns with the target patterns.
- Assessing the accuracy of the classification.

The informational data are classified into supervised and unsupervised systems.

ARTIFICIAL NEURAL NETWORK (ANN)
ANN according to Haykin is a massively parallel distributed processor that has a natural propensity for storing experiential knowledge and making it available for use. ANNs can provide suitable solutions for problems, which are generally characterized by non-linearities, high dimensionality noisy, complex, imprecise, imperfect or error prone sensor data, and lack of a clearly stated mathematical solution or algorithm. A key benefit of neural networks is that a model of the system or subject can be built just from the data.

Supervised learning is a process of training a neural network with examples of the task to learn, ie, learning with a teacher.

Unsupervised learning is a process when the network is able to discover statistical regularities in its input space and automatically develops different modes of behaviour to represent different classes of inputs.

NETWORK ARCHITECTURE
Texture is characterized by the spatial distribution of gray levels in a neighbourhood. Since, texture shows its characteristics by both pixel co-ordinates and pixel values, there are many approaches used for texture classification. The gray-level co-occurrence matrix seems to be a well-known statistical technique for feature extraction.

In texture classification the goal is to assign an unknown sample image to one of a set of known texture classes.

Textural features can be either scalar numbers, discrete histograms or empirical distributions. They characterize the textural properties of the images, such as spatial structure, contrast, roughness, orientation, etc and have some correlation with the desired output.

There are fourteen textural features. The design considers four features namely angular second moment (ASM), contrast, correlation, variance. However, it can be extended by inclusion of all features. The system architecture of the combined approach using both gray values and textural features is shown in Figure 1.

Architecture using Gray Values Only
The authors have considered a four layer ANN comprising three inputs, seven first layer hidden nodes, eleven second layer hidden nodes and five output nodes. Figure 2 shows the architecture of this network. Training of the network is done using standard BKP.
Using Gray Values and Textural Features

Combining both gray values and textural features form a four layer artificial neural network comprising seven inputs, eleven first layer hidden nodes, fifteen second layer hidden nodes and five output nodes. Figure 3 shows the architecture of this network.

Levels in a neighborhood. In order to capture the spatial dependence of gray-level values, which contribute to the perception of texture, a two dimensional dependence, texture analysis matrix is considered. Since, texture shows its characteristics by both pixel and pixel values, there are many approaches used for texture classification. The gray-tone co-occurrence matrix is used for feature extraction. It is a two dimensional matrix of joint probabilities $P_{d, r}(i, j)$ between pairs of pixels, separated by a distance, $d$, in a given direction $r$.

- For finding textural features for every pixel in the image every pixel is considered as a centre and followed by a $5 \times 5$ window about that centre pixel.
- The gray-one matrix for that particular window is calculated and normalized.
- The gray level co-occurrence matrix namely, $P_{h}, P_{v}, P_{rd}$ and $P_{ld}$ for each pixel is then obtained. Here, $P_{h}, P_{v}, P_{rd}$ and $P_{ld}$ are respectively the $0^\circ, 90^\circ, 45^\circ$ and $135^\circ$ nearest neighbours to a particular resolution cell.
- Standard deviation and mean are now obtained for each of these matrices and are used later to calculate the textural features.
- Now the particular entry in a normalized gray tone spatial dependence matrix is calculated for further reference, i.e., $P_{x+y}(k)$ and $P_{x-y}(k)$.
- Using the formulas of the textural features, the angular second moment, contrast, correlation and variance are calculated (Appendix A).

Training the Combined Network using BKP

The following assumes the sigmoid function $f(x)$

$$f(x) = \frac{1}{1 + e^{-x}}$$

The popular BKP algorithm$^{1,2}$ is implemented using following steps:

Step 1: Initialize weights to small random values.

Step 2: Feed input vectors $X_0, X_1, \ldots, X_6$ through the network and compute the weighting sum coming into the unit and then apply the sigmoid function. Also, set all desired outputs $d_0, d_1, \ldots, d_5$ typically to zero except for that corresponding to the class the input is from.

Step 3: Calculate error term for each output unit as

$$\delta_j = y_j (1 - y_j) (d_j - y_j)$$

where $d_j$ is the desired output of node $j$; and $y_j$ is the actual output.

Figure 1 System architecture of combined approach

Figure 2 Architecture of NN with only gray values

The number of bands and classes determine the number of input and the output nodes, respectively.

ALGORITHM

Feature Extraction

Texture and tone bear an inextricable relationship to one another. Tone and texture are always present in an image, although at times one property can dominate the other. For example, when a small area patch of an image has little variation of features of discrete gray tone, then tone is the dominant property. Important property of tone texture is the spatial pattern of resolution cells composing each discrete tonal feature. When there is no spatial pattern and the gray tone variation between features is wide, a fine textural image results.

Texture is one of the most important defining characteristics of an image. It is characterized by the spatial distribution of gray levels in a neighborhood. In order to capture the spatial dependence of gray-level values, which contribute to the perception of texture, a two dimensional dependence, texture analysis matrix is considered. Since, texture shows its characteristics by both pixel and pixel values, there are many approaches used for texture classification. The gray-tone co-occurrence matrix is used for feature extraction. It is a two dimensional matrix of joint probabilities $P_{d, r}(i, j)$ between pairs of pixels, separated by a distance, $d$, in a given direction $r$.

- For finding textural features for every pixel in the image every pixel is considered as a centre and followed by a $5 \times 5$ window about that centre pixel.
- The gray-one matrix for that particular window is calculated and normalized.
- The gray level co-occurrence matrix namely, $P_{h}, P_{v}, P_{rd}$ and $P_{ld}$ for each pixel is then obtained. Here, $P_{h}, P_{v}, P_{rd}$ and $P_{ld}$ are respectively the $0^\circ, 90^\circ, 45^\circ$ and $135^\circ$ nearest neighbours to a particular resolution cell.
- Standard deviation and mean are now obtained for each of these matrices and are used later to calculate the textural features.
- Now the particular entry in a normalized gray tone spatial dependence matrix is calculated for further reference, i.e., $P_{x+y}(k)$ and $P_{x-y}(k)$.
- Using the formulas of the textural features, the angular second moment, contrast, correlation and variance are calculated (Appendix A).

Training the Combined Network using BKP

The following assumes the sigmoid function $f(x)$

$$f(x) = \frac{1}{1 + e^{-x}}$$

The popular BKP algorithm$^{1,2}$ is implemented using following steps:

Step 1: Initialize weights to small random values.

Step 2: Feed input vectors $X_0, X_1, \ldots, X_6$ through the network and compute the weighting sum coming into the unit and then apply the sigmoid function. Also, set all desired outputs $d_0, d_1, \ldots, d_5$ typically to zero except for that corresponding to the class the input is from.

Step 3: Calculate error term for each output unit as

$$\delta_j = y_j (1 - y_j) (d_j - y_j)$$

where $d_j$ is the desired output of node $j$; and $y_j$ is the actual output.
Step 4: Calculate the error term of each of the hidden units as

$$\delta_j = x_j (1 - x_j) \sum_k \delta_k w_{jk}$$

where $k$ is over all nodes in the layers above node $j$; and $j$ is an internal hidden node.

Step 5: Add the weight deltas to each of

$$W_j(t+1) = W_j(t) + \eta \delta_j x_j$$

All the steps excepting step 1 are repeated till the error is within reasonable limits and then the adjusted weights are stored for reference to the Recognition Algorithm.

Illustration

As an application the image Figure 4(a) is considered from which a portion Figure 4(b) has been extracted which needs to be classified into five different classes considering both gray values and textural features. The pixels selected for training have been taken from the whole image except the portion, which is extracted for classification. For evaluating the accuracy of the classification, pixels of known class have been used.

| TRAINING PIXELS: |  
| --- | --- | --- | --- | --- | --- |
| Land Cover Type | No of Pixels |
| Barren (B) | Class 1 | 20 |
| Sand (S) | Class 2 | 20 |
| Urban (U) | Class 3 | 20 |
| Vegetation (V) | Class 4 | 20 |
| Water (W) | Class 5 | 20 |
| Total | 100 |

RESULTS AND DISCUSSION

With Combined Features

| Land Cover Type |  
| --- | --- | --- | --- | --- | --- |
| Classes | B | S | U | V | W |
| Class 1 | 19 | 0 | 1 | 0 | 0 | 95% | 91% |
| Class 2 | 0 | 20 | 0 | 0 | 0 | 100% |
| Class 3 | 3 | 0 | 17 | 0 | 0 | 85% | 91% |
| Class 4 | 0 | 0 | 0 | 20 | 0 | 100% |
| Class 5 | 1 | 0 | 0 | 4 | 15 | 75% |
| Total | 23 | 20 | 18 | 24 | 15 |  

Figure 4(a) Original image

Figure 4(b) Segmented image

Figure 5(a) [Left] with only gray values

Figure 5(b) [Right] : including textural features
With Gray Values Only

<table>
<thead>
<tr>
<th>Classes</th>
<th>B</th>
<th>S</th>
<th>U</th>
<th>V</th>
<th>W</th>
<th>Overall</th>
<th>Average</th>
</tr>
</thead>
<tbody>
<tr>
<td>Class 1</td>
<td>20</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>100%</td>
<td></td>
</tr>
<tr>
<td>Class 2</td>
<td>0</td>
<td>16</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>80%</td>
<td></td>
</tr>
<tr>
<td>Class 3</td>
<td>0</td>
<td>10</td>
<td>0</td>
<td>0</td>
<td>10</td>
<td>50%</td>
<td>46%</td>
</tr>
<tr>
<td>Class 4</td>
<td>20</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0%</td>
<td></td>
</tr>
<tr>
<td>Class 5</td>
<td>20</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0%</td>
<td></td>
</tr>
<tr>
<td>Total</td>
<td>60</td>
<td>26</td>
<td>0</td>
<td>0</td>
<td>14</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Figure 5(a) shows the classified image using only gray values and Figure 5(b) shows the classified image using both gray values and textural features. Figures 6(a)-6(f) show the three individual classes for both approaches thereby allowing for comparison.

CONCLUSION

It is clear from Figures 6(a)-6(f) that adoption of combined approach increases the classification accuracy. As a result the misclassification is reduced. The finer information from the
image can be extracted which is visible in Figures 5(a)-5(b) when compared with the original image in Figure 4(b).

ACKNOWLEDGEMENT

Authors are thankful to Dr B Kartikeyan, Scientist ‘SF’, ISRO, SAC, Ahmedabad for his guidance and technical help for successful completion of the work.

REFERENCES

4. Balaguruswamy. ‘Programming in ANSI C.’
5. E Hall. ‘Computer Image Processing and Recognition.’
7. J Anderson. ‘An Introduction to Neural Network.’
13. L Thomas. ‘Remote Sensing and Image Interpretation.’

APPENDIX

Flowchart to Obtain Textural Features

Computation of Co-occurrence Matrix

\[ p_{x+y}(k) = \sum_{j=1}^{Ng} \sum_{i=1}^{Ng} p(i,j) \]

where \( x \) and \( y \) are the spatial coordinates, \( Ng \) is the number of distinct gray levels in the quantized image.
Computation of Textural Features

From the co-occurrence matrix the textural features are defined as follows:

Angular Second Moment (ASM)

\[ f_1 = \sum_i \sum_j p(i, j)^2 \]

Angular second moment gives a strong measurement of uniformity. Higher non-uniformity values provide evidence of higher structural variations.

Contrast

\[ f_2 = \frac{N_{bg} - 1}{n^2} \left\{ \frac{N_{bg}}{i = 1} \sum_{j = 1}^{N_{bg}} p(i, j) \right\} \]

The contrast feature is a difference moment of the P matrix and is a standard measurement of the amount of local variations presented in an image. The higher the values of contrast are, the sharper the structural variations in the image are.

Correlation

\[ f_3 = \left\{ \frac{\sum_i \sum_j (i,j) \cdot p(i,j)}{\mu_x \mu_y} \right\} / \sigma_x \cdot \sigma_x \]

Variance

\[ f_4 = \left\{ \sum_i \sum_j (i - \mu_x)^2 \cdot p(i,j) \right\} \]

Variance is a measure of how spread out a distribution is. Numerically, variance equals the average of several squared deviations from the mean.