Image Binarization Using Multi-Layer Perceptron: A Semi-Supervised Approach

Amlan Raychaudhuri
B. P. Poddar Institute of Management & Technology,
137, VIP Road, Kolkata, India
amlanrc@gmail.com

Jayanta Dutta
Cognizant Technology Solutions India Pvt. Ltd.,
Kolkata, India.
jayanta1989@gmail.com

Abstract—In this paper, we have discussed the Image Binarization technique using Multilayer Perceptron (MLP). The purpose of Image Binarization is to extract the lightness (brightness, density) as a feature amount from the Image. It converts a gray-scale image of up to 256 gray levels to a black and white image. We use Backpropagation algorithm for training MLP. It is a supervised learning technique. Here K-means clustering algorithm has been used for clustering a 256 x 256 gray-level image. The dataset obtained by this is fed to the MLP and processed in a Semi-Supervised way where some training samples are taken as Known patterns (for training) and others as Unknown patterns. Finally through this approach a Binarized image is produced.

Key Words — Backpropagation Algorithm, Image Binarization, K-means clustering, Semi-supervised learning

I. INTRODUCTION

The goal of Image Binarization is to convert a 256 x 256 gray level image into a binary image or more specifically a bi-level image. The image binarization finds it use in Change detection of Remote Sensing Images, extraction of visual information from images, fingerprint analysis, Character Recognition from images and badly illuminated texts etc. For doing the Image Binarization, a multilayer neural network which is also called Multilayer Perceptron is used. A Multilayer Perceptron (MLP) [10] is a feed forward artificial neural network model that maps sets of input data onto a set of appropriate output. A MLP consists of multiple layers of nodes in a directed graph, which is fully connected from one layer to the next. Except for the input nodes, each node is a neuron (or processing element) with a nonlinear activation function. The intensities of each pixel from a gray-level (256 x 256) image are extracted and using K-means algorithm those pixels are mapped into two clusters considering their own and neighbouring pixel intensities. Finally some pixels from each cluster are chosen as Known patterns for training the MLP. After that the trained MLP is applied to the remaining pixels for mapping their intensities to either white or black. Finally an Output showing the binarized form of Input image is produced. Semi-Supervised [3]-[7] approach is used here for Image Binarization.

The Section II describes the strategies and concepts been used for the proposed work. The Section III describes the problem statement on which we worked. The Section IV describes the procedure of work. Section V shows the measures of experimental verification and experimental results obtained. Finally Section VI describes the conclusion drawn from work.

II. STRATEGY AND CONCEPTS

Here the work progressed in a strategic manner by the principle used in [8]. The steps are discussed as —

- Extraction of pixel intensities from an Input RAW image and creation of Input feature for K-means clustering.
- Through K-means clustering algorithm all the input patterns are partitioned into two classes (Black and White class).
- Based on some criterion (discussed in Section IV), some patterns are taken (known as known patterns) from both the classes for training in MLP.
- Training of MLP through known patterns and applying it to remaining patterns.
- Conversion of MLP output into output image which is the desired Binarized image.

Here Learning [2] should be concerned. In Supervised Learning a trainer supplies input-output instances to the system and through this the system adapts its parameters so as to produce desired output for a given input pattern. In Unsupervised Learning no trainer is there. Also only input patterns are given. No output patterns are known. Learner adapts its parameters autonomously.

For the completion of the work the strategy includes the choice of different concepts which will be tied together to fulfill the objectives. The base of this strategy is the semi-supervised learning. In case of supervised learning data sets are provided and known to the learner. But for unsupervised learning no initial data/pattern are provided. Here in this work, semi-supervised approach is employed, where a few data/pattern will be provided to the learner’s knowledge base, but not all data. The learner will use that data to update its parameters and will learn progressively. It means the learner will use that knowledge to discover the unknown patterns and recognizing those patterns will add those to its knowledge base. The knowledge base will expand as enrichment continues and will continue until all the data/patterns are recognized. Here in case of Image Binarization, until all the pixels are classified into black or white. So learner will iteratively and simultaneously learn and apply its knowledge to unknowns and will learn more.

Here the strategy will be to create a data set where some of training examples will be known and the others will be unknown. The MLP will be trained with the known samples through Backpropagation algorithm to a certain
extent and will be applied to the remaining unknown samples to classify them. Definitely some correct and incorrect results will be discovered. Now these correct results will be added to the knowledge base of the MLP. It means the knowledge base will contain the Provided examples and the Correct Examples discovered. Through this the knowledge base of the MLP will be enriched gradually and the MLP will start producing more accurate results. This process continues cumulatively until some terminating condition arrives.

Another important thing to mention here that the data set (initially mentioned) is created through unsupervised learning. It means though K-means we shall perform the two-cluster clustering in an unsupervised way and will take the data sample (which has the highest possibility of being a member of the cluster to which it belongs) as known patterns and remaining data samples will be treated as unknown patterns to be classified. The process is clarified in the next sections.

III. PROBLEM STATEMENT

From the histogram of Lena image which is shown in Fig. 1(b), it is found that the pixel counts at various intensities but cannot propose the required threshold value for which the Binarized image will be most appealing and will convey sufficient visual information captured from original image. That’s why it is need to do clustering of pixels according to intensity values and train the MLP with obtained information so that through learning MLP can determine the suitable threshold value and produce the Binarized Image.

IV. WORK DONE

A. 3 x 3 Matrix formation

For the application of K-means algorithm the input features have been created for this. An image has been taken as source image. It is a 256 x 256 gray-scale image where there are 65536 pixels each having the intensity within [0, 255] as integers. Each pixel has one of the 256 intensity levels.

For the extraction of intensity of each pixel a nine-dimensional feature space has been considered. The aim of such imagination is to consider a pixel’s intensity as well as that of its neighbouring pixels because we know that image is the visual representation of data and that’s why in that representation the intensities of neighbouring pixels has a great impact on the actual pixel’s intensity. That’s why during image binarization impact of neighbouring pixels’ intensities should be taken into account. For this a 3 x 3 matrix has been imagined that will capture the intensity of a target pixel and that of all its neighbours.

In Fig. 2, I_ij is the intensity of the target pixel (i, j). The remaining intensities are of all the neighbouring pixels of the target pixel. A data set file containing 65536 records (or patterns) has been prepared where the intensity of each pixel and that of its neighbouring pixels are recorded in the following format:

\[
\begin{bmatrix}
I_{i,j}, (I_{i+1,j}), (I_{i,j+1}), (I_{i-1,j}), (I_{i,j-1}), (I_{i+1,j+1}), (I_{i+j}, j), (I_{i+1,j+1})
\end{bmatrix}
\]

B. K-means application

It is known that K-means algorithm is a clustering method. So K-means clustering (K = 2) [9] is used here for mapping the intensities into two clusters known as black cluster and white cluster without knowing which one is black or white. Each cluster has its own centroid which is a 3 x 3 matrix (conveying nine-dimensional input features). Initially the centroid of each cluster has been chosen randomly from the available nine-dimensional records/patterns such that centroids of all the clusters are distinct. Now the clusters are populated with records which are closest to it. The closeness or vicinity is measured between centroid and chosen record through Euclidian distance formula.

\[
\text{Distance}_d = \sqrt{\sum_{i=1}^{n} (X_{ik} - C_{ij})^2}
\]

Where X is i-th intensity of k-th pattern/record and C is the i-th intensity of l-th (actually 1 = 1, 2 here) centroids where n = 9 (since nine-dimensional input pattern). Here i-th intensity of a pattern/centroid means it is the intensity of i-th cell in the nine-dimensional matrix representation of the pattern/centroid. In the 3 x 3 matrix the cells are numbered as Fig. 3.

\[
\begin{align*}
I_{i,j}^{(1)} & \quad I_{i,j+1}^{(2)} & \quad I_{i,j+2}^{(3)} \\
I_{i+1,j}^{(4)} & \quad I_{i+1,j+1}^{(5)} & \quad I_{i+1,j+2}^{(6)} \\
I_{i+2,j}^{(7)} & \quad I_{i+2,j+1}^{(8)} & \quad I_{i+2,j+2}^{(9)}
\end{align*}
\]

Fig. 4. Reading pattern of Cells
And they are read sequentially in this format cell by cell. Due to such sequential reading the patterns will look like Fig. 4. It means the original pixel’s intensity will be at cell – 1.

After K-means algorithm is over we get two different clusters populated with input patterns. Now by some certain analysis of the intensity values in the clusters we can determine the black and white clusters.

C. Concept of Hypersphere

Let lc and uc be the two centroids obtained by K-means algorithm in nine-dimensional feature space. Two other points lb (0, ..., 0), the possible minimum component values of the patterns which is near to lc and ub (255, ..., 255), the possible maximum component values of the patterns which is near to uc in the same feature space have been considered here.

A pattern can be assigned to the black class if it is inside the Hypersphere [1] whose center is at lb and radius is the Euclidian distance between lb and lc or it can be assigned to white class if it is inside the Hypersphere whose center is at ub and radius is the distance between ub and uc or else it is considered as unlabeled patterns. These unlabeled patterns are the patterns which will be considered as unknowns in the semi-supervised learning.

In Fig. 5 the Hypersphere has been shown. The patterns within the Hypersphere have the strong possibility of being mapped into black or white pixels accordingly. Next the $x\%$ ($0 \leq x \leq 25$) of the patterns which resides within the Hypersphere are selected. Those selected patterns will be considered as the Known patterns for the semi-supervised learning of MLP.

That’s why the selected patterns from black class’s Hypersphere are mapped to black pixels. Similarly for selected patterns in white class’s Hypersphere are mapped to white pixels.

Finally an output is generated containing information of all the 65536 pixels where some of them have been mapped to black or white intensity and the other pixels’ intensities remain unchanged.

Ultimately after mapping of selected pixels an output of the following format is created:

- [RowIndex, ColumnIndex, PixelIntensity]
- $0 \leq \text{RowIndex} \leq 255, 0 \leq \text{ColumnIndex} \leq 255, 0 \leq \text{PixelIntensity} \leq 255.$

D. Data Set Creation

For utilizing this obtained information it is needed to convert it into a data set so that it can be fed to MLP. For converting this into data set the following format has been used:

- [Pixel ID, Intensity Vector, Pixel mapping information]
- where $1 \leq \text{Pixel ID} \leq 65536$
- 
- $\text{Intensity Vector} = [(l_{ij}), (l_{i,j+1}), (l_{i, j}), (l_{i-1,j+1}), (l_{i,j-1}), (l_{i-1, j-1}), (l_{i+1,j}), (l_{i,j+1})]$

It is actually the nine-dimensional representations of the original pixel intensities fetched directly from the image where $l_{ij}$ denotes the original pixel intensity and the remaining depicts the neighbouring pixel intensities (same as discussed before).

-Pixel mapping information denotes whether the corresponding pixel has been already mapped as black or white pixel or not. It includes the following values:
- “?” denotes that it is an unmapped pixel.
- “0” denotes that it has been mapped as a black pixel with intensity value 0.
- “1” denotes that it has been mapped as a white pixel with intensity value 255.

E. Concept of Soft Class

At first, training is done in the network using $x\%$ ($0 < x \leq 25$) already classified inputs (known patterns) classified by K-Means algorithm. After training we fed the remaining unclassified patterns to the network and check if it satisfies any of its outputs or can it be classified to a particular class? If it satisfies a particular class then we classified that pattern and called it a Soft Class [8]. Then we again train the network using the all the classified patterns and the soft classes and repeat the processes until all the patterns get classified.

F. Backpropagation Algorithm, MLP and Associated Implementation

Before applying the Backpropagation algorithm it is need to frame the Artificial Multilayer Neural Network. The design of the ANN [10] is as follows:

- Number of Input Layer Units: 9
- Number of Hidden Layer Units: 15
- Number of Output Layer Units: 2
- Learning Rate used: 0.1

The algorithm used as follows:

BEGIN:
- 
- $Step 1$: Generate the input patterns from the gray-scale intensity values of a pixel and its corresponding eight neighbouring pixels.

- $Step 2$: Since here the learning algorithm that means the Backpropagation algorithm works best while the inputs of the Neural Network are real values between 0 and 1 so each input is divided by 255 so that the maximum input value become 1 (for intensity 255) and minimum input value become 0 (for intensity 0).

- $Step 3$: Learn the network by $x\%$ ($0 < x \leq 25$) already classified inputs (by K-Means) as following:

  - $Step 4$: For all classified inputs do

Fig. 5. Hypersphere
i> First initiate the weights of the network using random values.

ii> Fed the inputs through the network and calculate the linear combination of the weights and the inputs for each of the hidden units as

$$O_h = \sum_{i=0}^{m} W_i X_i$$

(where $W_i$ is the edge weight from input to hidden layers and $X_i$ is the inputs.)

iii> Then calculate the Sigmoidal Output for each of the hidden unit as

$$Y_h = \frac{1}{1 + e^{-(\lambda O_h)}}$$

(where $\lambda$ is the Sigmoidal Gain and here $\lambda = 0.5$ has been considered)

iv> Fed the hidden units values to the Output Layer and calculate the linear combination of the hidden-output weights and the inputs to the output layers as

$$O_o = \sum_{i=0}^{m} W_i X_i$$

(where $W_i$ is the edge weight from hidden to output layers and $X_i$ is the inputs to output layer)

v> Then calculate the Sigmoidal Output for each of the Output unit as

$$Y_o = \frac{1}{1 + e^{-(\lambda O_o)}}$$

(where $\lambda$ is the Sigmoidal Gain and here $\lambda = 0.5$ has been considered)

vi> Now to increase the quality of the outputs at each of the output units perform the following operation:

If $Y_o \leq 0.5$ then

$$Y_o = Y_o^2$$

Else $Y_o = [1 - 2*(1 - Y_o)^2]$  

vii> Then Calculate the error at each of the output units as

$$Error_o = Y_o * (1 - Y_o) * (Target_o - Y_o)$$

viii> Then fed the error backwards and modify the edge weights between the hidden layer and output layer as

$$W_{ji} = W_{ji} + \eta * Error_o * X_{hi}$$

(where $\eta$ is the learning rate and $X_{hi}$ is the inputs to the output layer)

ix> Then calculate the error at each hidden layer unit as

$$Error_h = Y_h * (1 - Y_h) * \sum_{k=0}^{kmax} (W_{khi} * Error_k)$$

x> Update the edge weights between input to hidden layer as

$$W_{ij} = W_{ij} + \eta * Error_h * X_{ij}$$

(where $\eta$ is the learning rate and $X_{ij}$ is the inputs to the hidden layer)

xi> Continue the process from step <ii> for next set of inputs.

Until total error for all input patterns in (previous iteration - current iteration) become < 0.01

Step 5: Now for all unclassified inputs do

i> for each input patterns

a) calculate its’ outputs at output units.

b) if $Y_o > 0.9$ then the input pattern is classified as target = 0

(where 0 ≤ o < 2 and this patterns are termed as Soft Class)

else the pattern remains unclassified.

Step 6: Go to Step 4 until all the input patterns are classified or no more unclassified input patterns are being classified.

Step 7: For remaining unclassified input patterns Do

i> for each input patterns

a) calculate its’ outputs at output units.

b) For each output unit find out the unit having maximum value, i.e. $max_o$ .

c) Target = $max_o$

END.

How the training is done in MLP for the dataset that is shown in Fig. 6.

G. Binarized Output creation:

Finally the output of the MLP is to be converted into the image back. So MLP produces an output of the following format:

[Pixel ID, Intensity Vector, Mapped Class]

Where 1 ≤Pixel ID ≤65536

Intensity Vector = [(I_{h1}, I_{h1,j1}), (I_{h1}, I_{h1,j1}), (I_{h1,j1}), (I_{h1}, I_{h1,j1}), (I_{h1,j1}), (I_{h1,j1})]

It is actually the nine-dimensional representations of the original pixel intensities fetched directly from the image where $I_{hj}$ denotes the original pixel intensity and the remaining depicts the neighbouring pixel intensities (same as discussed before).

Mapped Class denotes whether the corresponding pixel has been mapped as black or white pixel class. It includes the following values:

“0” denotes that it has been mapped as a Black Pixel with intensity value 0.

“1” denotes that it has been mapped as a White Pixel with intensity value 255.

Training of MLP

Fig. 6. Training of MLP
B. Reference Image

A reference image has been created to evaluate the performance of different binarization methods. But for evaluate graphics image there is no standard method for creating a proper reference image. In this paper, the reference image is created by following a major voting scheme [12], as follows: five binarized images have been created by using K-Means algorithm, using proposed MLP method with 10% training data, using proposed MLP method with 15% training data, using proposed MLP method with 20% training data, and using proposed MLP method with 25% training data. For generating the reference image, all the five binarized images are consulted pixel-by-pixel. Each pixel in the reference image is set to black if the majority of the five methods agree that the corresponding pixel is black. Otherwise the pixel of the reference image is set to white.

C. Experimental Results

The proposed method is tested for some of the input images. Here three input images and their output images are shown in Fig. 7 – Fig. 9. All the images are gray level images of size 256 × 256. The Lena image, the output binarized image by the proposed MLP method and the output binarized image using K-means method are shown in Fig. 7(a), 7(b) and 7(c) respectively. The Pepper image, the output binarized image by the proposed MLP method and the output binarized image using K-means method are shown in Fig. 8(a), 8(b) and 8(c) respectively. The Horse image, the output binarized image by the proposed MLP method and the output binarized image using K-means method are shown in Fig. 9(a), 9(b) and 9(c) respectively.

Table-1 shows the performance of the binarization techniques in terms of Misclassification Error (ME). From the table it is also shown that for all three images the proposed MLP method with 20% training data gives the best result.

### Table 1. Performance Evaluation of the Binarization Techniques in terms of ME

<table>
<thead>
<tr>
<th>Methods</th>
<th>Images Names</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Lena</td>
</tr>
<tr>
<td>K-Means</td>
<td>0.179</td>
</tr>
<tr>
<td>MLP (10% training data)</td>
<td>0.0151</td>
</tr>
<tr>
<td>MLP (15% training data)</td>
<td>0.0116</td>
</tr>
<tr>
<td>MLP (20% training data)</td>
<td>0.0009</td>
</tr>
<tr>
<td>MLP (25% training data)</td>
<td>0.0133</td>
</tr>
</tbody>
</table>

V. MEASURES OF EXPERIMENTAL VERIFICATION AND EXPERIMENTAL RESULTS

A. Misclassification Error (ME)

Here Misclassification error [11] is used to evaluate the result of the proposed work. It reflects the percentage of white pixels incorrectly assigned to black pixels, and conversely, black pixels incorrectly assigned to white pixels. For the two class segmentation problem, ME can be expressed as:

\[
ME = 1 - \frac{|B_O \cap B_I| + |W_O \cap W_I|}{|B_O| + |W_O|}
\]

where \(B_O\) and \(W_O\) denote the black pixel and white pixel of the original reference image, and \(B_I\) and \(W_I\) denote the black pixel and white pixel in the output binarized image. \(|\cdot|\) means the cardinality of the set. The ME varies from 0 for a perfectly classified image to 1 for a totally wrongly classified image.
VI. CONCLUSION

Conclusively we can say that in the field of artificial intelligence and machine learning semi-supervised approach has a great importance. Because real machines always have to deal with many complex data and also in reality a few amount of known data remains available. Through this approach we can tackle the real world problems where the efficiency of the machine increases progressively as it adapts to the environment. This paper presents an effective method for image binarization using semi-supervised approach. There are a lot of scopes for experimenting the proposed method with document images, analysis of medical images and change detection in remote-sensing images etc.

REFERENCES


AUTHOR’S PROFILE

Amlan Raychaudhuri received the B.Sc. degree (with honours) in Computer Science from the University of Calcutta, Kolkata, India, in 1999. He also received the M.Sc. degree in Computer and Information Science and the M.Tech. degree in Computer Science and Engineering from the University of Calcutta, Kolkata, India, in 2001 and 2003, respectively.

He was a Lecturer with the Department of Computer Science and Engineering, B. P. Poddar Institute of Management and Technology under West Bengal University of Technology, Kolkata, India, during 2004 – 2008. He is currently an Assistant Professor at the same department. His research interest include soft computing, image processing, pattern recognition and remote sensing image analysis. His email id is amlanrc@gmail.com.

Jayanta Dutta received the B.Tech. degree in Computer Science and Engineering from B. P. Poddar Institute of Management and Technology under West Bengal University of Technology, Kolkata, India, in 2011.

He is currently a Programmer Analyst Trainee (CSP-BPM) and Pega Certified PRPC System Architect in Cognizant Technology Solutions India Pvt. Ltd., Kolkata, India. His research interest include soft computing, image processing and cloud computing. His email id is jayanta1989@gmail.com.