A New Approach to Picture Quality-independent Robust Text Binarization using Differential Binning

The Overview of the steps involved

1. **Foreground – Background Pixels split**: The first step involves identifying and splitting the pixels of the image into sets of candidate foreground and candidate background pixels. We used a differential binning approach which gives different sets of candidate foreground and background splits. This is the critical step as the next set of steps depends on the fact that there is at least one split where the text pixels fall into one set of pixels and with very little background pixels in the same set.
2. **Homogeneous region detection:** Step 1 gives several sets of candidate foreground and background pixels. This step involves identifying clusters of connected pixels which represent a homogenous region in the image. A homogeneous region is defined as a set of connected pixels in an image whose deviation in RGB values is less than the deviation from the surrounding pixels of that cluster.
3. **Text, non-Text Separation:** The Homogeneous regions extracted in the form of connected set of pixels from Step 2 are now classified into text and non-text using a classifier model. Regions classified as Non-Text are removed and candidate Text regions are retained for combination
4. **Probabilistic Combination:** Step 3 gives us multiple sets of candidate Text pixels across multiple bin sizes used in the differential binning step. We identify the probability of a pixel being a text pixel based on its occurrence across the different bin sizes. We label Pixels in the final binarized image based on a probability outcome
5. **Foreground – Background Pixels split: Differential Binning**

The first target is to identify sets of foreground and background pixels and split them. We use two-level overlapped binning as described below.

1. **Two-Level Overlapped Binning**

For a given Bin Size s,we divide the entire range of rgb Euclidian distances into two levels of Bins

*For a given Bin size s:*

*Level 1={ 0 to (s-1),*

*(s) to (2\*s – 1),*

*(2\*s) to (3\*s – 1) ,*

*……*

*(k-1)\*s to (k\*s - 1) }*

*Total : k elements*

*Level 2={ k to (k+s-1),*

*(k+s) to (k +2\*s – 1),*

*(k + 2\*s) to (k + 3\*s – 1) ,*

*……*

*(k + (k-2)\*s) to (k + (k-1)\*s - 1) }*

*Total : (k-1) elements*

*Where k = upper(442/s)*

Example :

suppose Bin size = 56

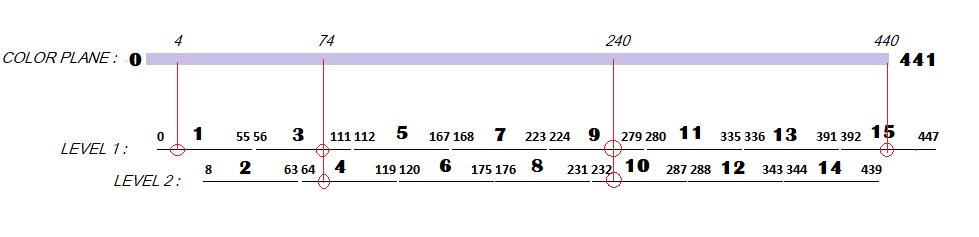
Then

k = upper(441/56) = 8

Level 1 = { 0 to 55,56 to 111,112 to 167,168 to 223,224 to 279,280 to 335,336 to 391,392 to 447 }

Level 2 = { 8 to 63,64 to 119,120 to 175,176 to 231,232 to 287,288 to 343,344 to 439 }

The following figure denotes the above example visually



Elements of Level 1 set are labelled with alternate odd numbers and Elements of Level 2 set are labelled with alternate even numbers

Each element *(x to y)* belonging to Level 1 set or level 2 Set is represented by a binary image.

For a Binary Image representing *(x to y)*

1 if x <= image(a,b) <= y

Pixel-Value(a,b) =

0 otherwise

*Pixel-Value(a,b) denotes the value of the binary image at row number a and column number b*

*image(a,b) denotes the Euclidean RGB values at row number a and column number b of the original image*

A single point on the original image can map to two Bins(example a point with Euclidean RGB value 74 maps to both Bin no. 3 and Bin no. 4 in the example above) or to just one bin ( a point with Euclidean RGB value 4 maps to just bin no. 1 in the example above )

1. **Multiple Bin Sizes**

In order to achieve robustness and quality independence, we use multiple bins sizes, separated at small intervals. Each Bin size has a significance – The aim of each bin size is to have pixels of all homogenous regions whose difference between the maximum RGB Euclidean distance and minimum RGB Euclidean distance is less than that bin size and the difference between each pixel’s RGB Euclidean value inside the homogenous region and immediate surrounding background pixels is greater than bin size

Bin Sizes are generated using the following recursive technique

, if n = 0

Xn = Xn-1+ - Xn-1 ) ,if Xn-1  < (MAX\_PIXEL\_VALUE/2)

∞ , otherwise

MAX\_PIXEL\_VALUE is the maximum value a pixel can have. For gray scale it is 255 while for RGB Euclidean distance we used,it will be 441

We do not use any Bin Size which are ∞ thus the calculation can end once a Bin size exceeds Half the MAX\_PIXEL\_VALUE, thus calculations can stop.

The above formula is used for a particular desirable property – The interval Xn – Xn-1 decreases with increasing values of n. This is needed because in low quality images, the homogeneous regions have a much larger range of values and much smaller difference to its background. Best possible results can be obtained if we have an arithmetic series a + (n-1)\*d with a small a and d values( a around 21 and d around 11 if using RGB Euclidean distances) to capture as many bin sizes as possible however that would lead to a higher computation cost and running time to process so many binary images. The above formula gave us an optimal balance between running time and accuracy.

1. **Solving the boundary problem: Delta Bins**

