Generating functions/trees for evaluating optimal binarizations

MPS Team Project

1st Asmaa Alaghbari	2nd Mihai Ilinca	3rd Ioana Rusu	4th Andreea-Maria Piciu	5th Cerasela Enus
Team Leader	*Developer*	*Developer*	*Project Manager*	*Tester*
343C5	*343C5*	*342C3*	*341C5*	*343C5*

***Abstract*—This project proposes a new image binarization method that combines global and local thresholding. The proposed method was evaluated on a variety of benchmark datasets and outperformed existing methods in terms of accuracy, recall, and F-measure.**

***Index Terms*—Image binarization, Global thresholding, Local thresholding.**

I. Introduction

Image binarization is the process of converting a grayscale image to a binary image, where each pixel is assigned a value of either black or white. Binarization is a fundamental image processing task that is used in a variety of applications, such as text recognition, document analysis, and medical imaging. There are a variety of different binarization methods available, each with its own advantages and disadvantages. Global thresholding methods use a single threshold value to binarize the entire image. This approach is simple and efficient, but it can be sensitive to noise in the image.
Local thresholding methods use different threshold values for different parts of the image. This approach is more robust to noise than global thresholding, but it is more complex and computationally expensive.

II. Solution

The solution implemented for the image binarization challenge encompasses both global and local thresholding techniques. The main.py script serves as the entry point, orchestrating the overall process and calling the appropriate functions. Our method iteratively computes the optimal thresholding parameters that best segment the foreground text from the background, balancing the contrast between the two.
The goal is to use a set of different algorithms to determine thresholds for binarization at both the global and local levels, and then to intelligently combine these thresholds to improve the quality of the binarized digital documents.

III. Architecture

The architecture of our solution is modular, with each component responsible for a particular aspect of the binarization process. The ‘tree.py‘ script embodies the core algorithm, which constructs a decision tree to evaluate different binarization thresholds. Each node in the tree represents a computation step, employing either a global or a local thresholding operation, derived from the project’s extensive research and experimentation phase.

IV. Intermediate Results

The main.py program uses the generated trees to evaluate the performance of binarizations on input data. The F-measure is used to quantify how well the optimal binarizations generated by the trees match the provided references. The average F-measure for all lines provides an overall estimate of the performance of the binarizations.

- Variability of Results: Because the trees are randomly generated, results may vary between different runs of the program. Variability may reflect the fact that, under certain circumstances, certain functions or values may lead to better binarizations than others.
- Possible Problems: If input data and references are not available, it is difficult to assess whether the binarizations are truly optimal or not. The use of representative datasets and appropriate references is crucial to ensure that the assessment is relevant and understandable.
- Parameter Optimization: If the performance of the binarizations is not satisfactory, one possibility would be to adjust the tree generation parameters or change the way functions and values are randomly chosen. Deeper exploration of function types and tree structure can provide insights into more effective binarization strategies.

V. Intermediate Conclusions

Our intermediate conclusions suggest that the integration of global and local thresholding, guided by a decision tree mechanism, is superior to traditional methods. The adaptability of local thresholding, combined with the simplicity of global thresholding, yields a robust binarization process. The next steps involve refining our algorithms and conducting further tests to validate these findings.

REFERENCES

[1] CS Open CourseWare. (2023). Proiect [CS Open CourseWare]. Retrieved from https://ocw.cs.pub.ro/courses/mps/proiect

[2] Gonzalez, R.C., & Woods, R.E. (2008). Digital Image Processing (3rd Edition).

[3] Otsu, N. (1979). A Threshold Selection Method from Gray-Level Histograms. IEEE Transactions on Systems, Man, and Cybernetics, 9(1), 62-66.

[4] Niblack, W. (1985). An introduction to Digital Image Processing.