Development plan

1. Introduction

The aim of this development is to improve the performance of global and local binarization algorithms for image processing. The development consists of testing and optimizing 100 trees for global binarization, selecting the 10 best trees, and applying the local binarization algorithm on them to evaluate the results obtained.

2. Methodologies

a) Global Binarization Testing

- Global Tree Generation: Generation of 100 trees based on specific global binary algorithm.
- F-Measure calculation: For each tree, the F-measure determination algorithm is applied to the data in the CSV file associated with the image.
- Selection of the best trees: The 10 best trees are selected according to F-measure scores.
- Tree optimization: Removes low-scoring trees from memory.

b) Local Binarization Testing

- Local Tree Generation: The top 10 previously selected trees are used to generate local trees.
- F-Measure Calculation for Local Binarization: The algorithm for determining F-measure is applied to data resulting from local binarization.
- Selection of the 3 best local trees: The 3 best local trees are selected according to F-measure scores.

3. Implementation

To achieve this development, a Java program was implemented using the SLF4J library for log management and ExecutorService for thread management. The Testing and ProcessFile classes have been implemented, which are responsible for testing and optimizing global trees, respectively applying the local binarization algorithm.

4. Results

a) Global Trees

For global binarization, trees were evaluated using a set of 256 thresholds, and F-measure scores were calculated for each tree associated with the images. The 10 best trees were selected based on the average F-measure scores and were subsequently optimized. These trees showed a high level of accuracy in pixel classification.

TreeNr36: Average F-Measure - 69.76

TreeNr26: Average F-Measure - 69.50

TreeNr17: Average F-Measure - 67.75

TreeNr91: Average F-Measure - 65.42

TreeNr100: Average F-Measure - 63.14

TreeNr72: Average F-Measure - 62.77

TreeNr43: Average F-Measure - 61.63

TreeNr6: Average F-Measure - 60.76

TreeNr45: Average F-Measure - 60.33

TreeNr89: Average F-Measure - 59.10

b) Local Trees

For local binarization, the top 10 global trees were used to generate local trees, which were then evaluated on the test dataset. F-measure scores for these local trees highlighted the effectiveness of algorithms in adapting to the specifics of images, providing significant F-measure values.

TreeNr36: Average F-Measure - 0.10

TreeNr26: Average F-Measure - 0.16

TreeNr17: Average F-Measure - 0.16

TreeNr91: Average F-Measure - 0.18

TreeNr100: Average F-Measure - 0.20

TreeNr72: Average F-Measure - 0.15

TreeNr43: Average F-Measure - 0.18

TreeNr6: Average F-Measure - 0.14

TreeNr45: Average F-Measure - 0.14

TreeNr89: Average F-Measure - 0.18

The 3 best local trees obtained by applying the local binarization algorithm on the 10 best global trees are:

TreeNr100: F-Measure - 0.20

TreeNr43: F-Measure - 0.18

TreeNr91: F-Measure - 0.18

5. The Impact of the Development Methodology used in the project

The development methodology adopted had a significant impact on the efficiency and quality of implementation of global and local binarization algorithms. Within this methodology, several practices and principles were used that contributed to the success of the project:

- Parallelization of processes

Utilizarea ExecutorService pentru gestionarea firelor de execuție a permis paralelizarea proceselor de testare a arborilor globali. Aceasta a dus la o accelerare semnificativă a evaluării fiecărui arbore și a îmbunătățit eficiența algoritmului de selecție a celor mai buni 10 arbori.

- Modularity and Code Reuse

Implementarea modulară a permis separarea logică a diferitelor componente ale sistemului, cum ar fi evaluarea arborilor globali și locali. Aceasta a facilitat înțelegerea și întreținerea codului, iar reutilizarea componentelor a dus la o dezvoltare mai eficientă și la evitarea duplicării de cod.

- Efficient resource management

Utilizarea ExecutorService a facilitat o gestionare eficientă a resurselor, evitând supraîncărcarea sistemului și optimizând procesele. De asemenea, ștergerea și eliberarea memoriei pentru arborii cu scoruri mici au contribuit la o utilizare mai eficientă a resurselor.

- Performance Measurement and Optimization

Metodologia a inclus o etapă crucială de evaluare și optimizare a algoritmilor, având ca rezultat selecția celor mai buni 10 arbori globali. Acest proces a contribuit la îmbunătățirea performanțelor algoritmilor de binarizare globală și, implicit, a rezultatelor binarizării locale.

- Results and Validation

The implementation was validated by the results obtained from global and local binarization. The F-measure scores demonstrated that the proposed algorithms have an outstanding pixel classification capability, confirming the positive impact of the development methodology on achieving high-quality results.

By applying this methodology, we have achieved not only an efficient implementation, but also a modular structure, easy to manage and expand. The impact of this methodological approach is reflected in the consistent results and high performance of the implemented global and local binarization algorithms.

6. Conclusion

This development has led to the optimization of global and local binarization algorithms, providing significant results. The selected trees show high F-measure scores, indicating their effectiveness in image processing. By implementing these improvements, the algorithm proves to be more robust and performant under various conditions.