



Facultatea de Automatică și Calculatoare,
Universitatea Politehnica, București

Documentul de Proiectare a Soluției Aplicației Software
(Software Design Document)

Indrumator:

Octavian Saca

Profesor:

Andrei Boianiu

Componenta echipei:

Andreea-Alisa Dragomir

Mihai-Bogdan Trifu

Dragos Petre Godeanu

Cuprins

1. Introducere	3
1.1. Scopul documentului	3
1.2. Definirea termenilor tehnici	3
1.3. Conținutul documentului	3
2. Obiective	4
2.1. Situația actuală	4
2.2. Scopul aplicației	4
2.3. Contextul aplicației	4
3. Modelul datelor	5
3.1. Structuri de date globale	5
3.2. Structuri de date temporare	5
3.3. Formatul fișierelor	5
4. Modelul arhitectural și modelul componentelor	5
4.1. Șabloane arhitecturale folosite	5
4.2. Diagrama de arhitectură	6
4.3. Descrierea componentelor	6
4.4. Restricții de implementare	6
4.5. Interacțiunea dintre componente	7
5. Elemente de testare	7
5.1. Componente critice	7
5.2. Tipuri de testare	7
5.3. Alternative	7

1.Introducere

1.1.Scopul documentului

Acest document are rolul de a descrie acurat și complet soluția proiectată pentru aplicația de binarizare a unei imagini, care se presupune a fi o pagină (veche) de ziar.

Documentul servește drept ghid unic de construire a soluției pentru echipa de dezvoltare a proiectului.

Acest document va fi înțeles și utilizat de către echipa de dezvoltatori ai aplicației, precum și de către utilizatori.

1.2.Definirea termenilor tehnici

Pixel = Cel mai mic element (punctual) în care se poate descompune o imagine obținută prin fotografiere (inclusiv prin teledetecție), tipărire sau creată pe un ecran.

Digitalizare = trecerea unui document dintr-un format fizic în unul digital.

BAM = Binarization Algorithm Module

VBAM = Voting Binarization Algorithm Module

Binarizare = Această operație are ca obiectiv obținerea unei imagini alb- negru dintr-o imagine care conține și alte nuanțe nedorite provenite din diverse motive tehnice (de exemplu copiere).

Testare white-box = strategie de generare a testelor pe baza structurii interne a codului.

Testare black-box(testare functionala) = strategie de testare care necesită cunoașterea comportamentului extern al programului pe baza specificațiilor.

IBS = Image Binarization System

1.3.Conținutul documentului

- În prima parte a documentului se realizează descrierea într-o manieră sumară a proiectului, conținutul și scopul acestuia.
- Cea de-a doua parte cuprinde modelul datelor folosit în aplicație. Acest model prezintă principalele structuri de date și formatul fișierelor folosite pentru schițarea soluției.
- Cea de-a treia parte a documentului cuprinde modelul arhitectural și cel al componentelor. Aceste modele prezintă șabloanele folosite, arhitectura sistemului și descrie componentele acestei arhitecturi.
- Următoarea parte va descrie modul de interacțiune al utilizatorului cu aplicația, iar partea a patra cuprinde elemente de testare care prezintă componentele critice și alternative de proiectare ale acestora.

2.Obiective

2.1. Situația actuală

În decursul ultimelor decenii, crearea și depozitarea documentelor a trecut de la formatul fizic la cel electronic. Acest fapt a schimbat radical modul în care oamenii interacționează cu datele.

Deși documentele create recent beneficiază atât de suport fizic cât și electronic, cele din trecut se regăsesc doar în forma lor fizică. Acest lucru poate fi un dezavantaj întrucât exemple de astfel de documente pot fi manuscrise sau tiparituri ce conțin informații importante ce s-ar dori să fie procesate în mod automat.

2.2. Scopul aplicației

Aplicația are drept scop digitalizarea unor documente prin supunerea lor unor algoritmi complexi de binarizare, având drept rezultat un output cât mai lizibil, indiferent de gradul de iluminare sau degradare al documentului original.

2.3. Contextul aplicației

Acest proiect are drept scop dezvoltarea unui sistem IBS (de binarizare a imaginilor) care include două module:

- i) Modulul de binarizare care transformă imaginea în una binară.
- ii) Modulul de votare care compune o singură imagine binară folosind mai multe rezultate ale mai multor module de binarizare.

Binarizarea unui document este un proces complex ce presupune mai mulți pași de procesare, dar care nu se mai aplică datorită posibilității mari de apariție a erorilor. De aceea, se pleacă de la o variantă alb-negru a imaginii pentru a diferenția cât mai bine fundalul de conținut prin binarizare.

3. Modelul datelor

3.1. Structuri de date globale

În program vom introduce și folosi o structură de date globală în care vom reține fișierul de intrare asupra căruia se vor aplica algoritmul de binarizare.

3.2. Structuri de date temporare

- Pentru stocarea în memorie a imaginilor se vor folosi matrici alocate dinamic.
- Pentru reținerea rezultatului obținut în urma citirii și prelucrării datelor de intrare se va folosi o clasă sau o structură.
- Pixelii vor lua valori în intervalul 0-255 și vor avea valoarea 0 la inițializare.
- Acești pixeli vor fi salvați într-o matrice.
- Inițial avem salvate valorile pixelilor din imaginea originală.
- Asupra acestei matrici vom aplica mai întâi o conversie la gray scale și vom salva noile valori ale pixelilor.
- Apoi, se aplică algoritmul de binarizare Bernsen asupra matricii convertite la gray scale.
- Când se termină prelucrarea structurii ce presupune imaginea originală, celei convertite la gray scale sau a imaginii obținute în final, se șterge, întrucât nu mai este necesară.

3.3 Formatul fișierelor

- În realizarea proiectului am pornit de la frameworkul oferit și accepta orice imagine ca output care respecta regulile impuse.
- Ca și date de intrare, am testat pornind de la imagini de tip jpg sau png.

4. Modelul arhitectural și modelul componentelor

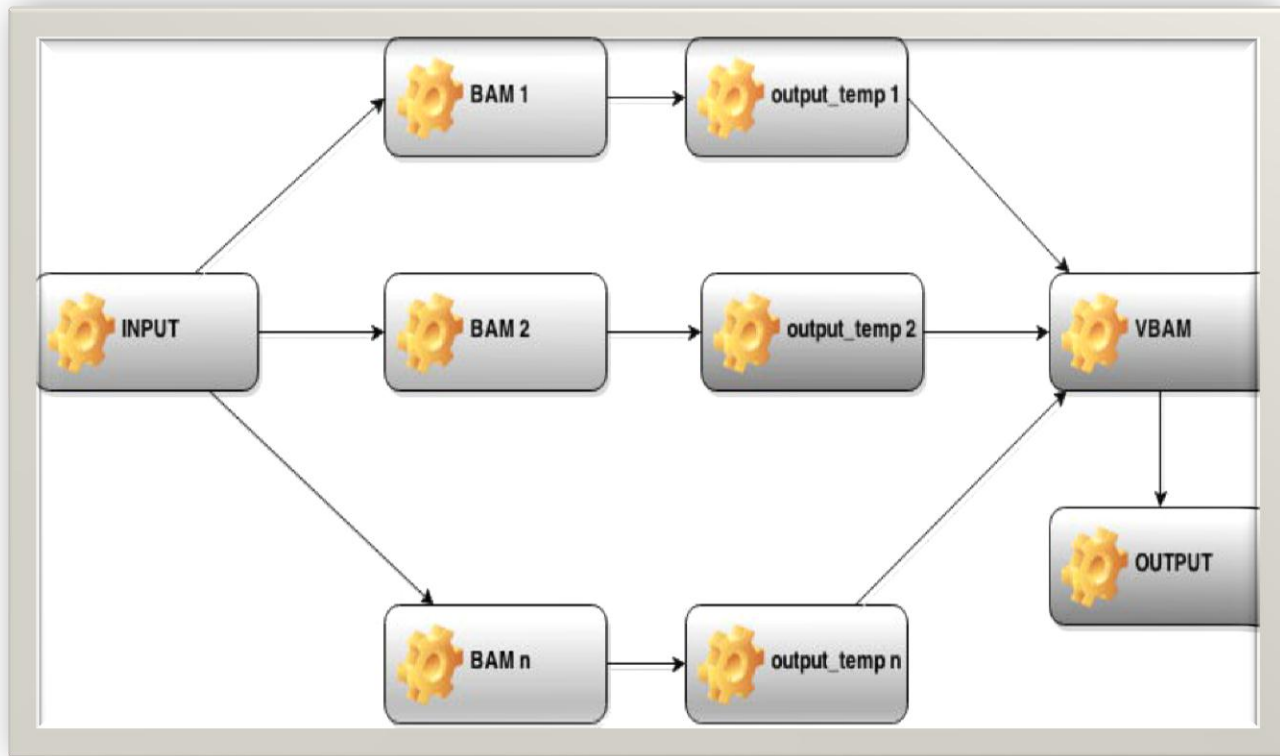
4.1 Șabloane arhitecturale folosite

Aplicația conține două tipuri de module: VBAM și BAM.

BAM-urile preiau ca input o imagine, aplică un algoritm de binarizare și întorc drept output două imagini.

Prima imagine reprezintă soluția după aplicarea algoritmului, iar cea de-a doua reprezintă factorul de încredere pentru corectitudinea imaginii output.

4.2 Diagrama de arhitectură



4.3 Descrierea componentelor

- **BAM:** aplică un algoritm de binarizare asupra imaginii primite ca input și realizează imaginea binarizată și imaginea conținând factorul de corectitudine pentru fiecare pixel.

- **VBAM:** preia output-ul modulelor BAM (imaginile binarizate și imaginile factor de corectitudine) și le supune unui algoritm de votare în urma căruia va desemna o singură imagine, presupusă corectă (sau cea mai apropiată de corectitudine).

4.4 Restrictii de implementare

- Modulele BAM și VBAM vor fi dezvoltate utilizând limbajul de programare C/C++.
- Formatul imaginilor de intrare va fi orice tip accepta frameworkul și respecta condițiile impuse

4.5 Interactiunea dintre componente

Modulul VBAM va apela modulele BAM si va genera imaginea binarizată pe baza algoritmului de votare.

5. Elemente de testare

5.1 Componente critice

Pentru a se verifica corectitudinea aplicatiei se vor introduce parametrii de intrare gresit, se vor introduce fisiere de un alt format, altul decat cel suportat de aplicatie, imagini corupte, numar gresit de BAM-uri.

Fiecare componentă va fi testată individual.

5.2 Tipuri de testare

Testarea se poate realiza ca:

1. **Testare White-Box** ce presupune ca testerul care acces in sursele programului. De multe ori, testarea prin aceasta metodă implică scrierea de cod sau cel puțin, urmărirea celui existent. Se testează fiecare metodă in parte inițial, cât si interoperarea metodelor.

2. **Testare Black-Box** ce presupune testarea aplicatiei la nivelul user-ului, plecand de la premisa ca nu se cunoaste modul in care functioneaza. Pentru acest tip de testare, testerul trebuie sa cunoască rezultatele ce se așteaptă din partea aplicației.

Pentru testarea am folosit imagini de tip jpg sau png. Pentru fiecare imagine data la intrare, in urma rularii programului vom avea 2 imagini:

- ➔ Imaginea binarizata -> imaginea obtinuta in urma rularii algoritmului
- ➔ Imaginea de confidenta -> imaginea obtinuta astfel:
 - Pixeli intre 255 si 0 astfel incat 255 reprezinta ca sigur a fost binarizat corect si 0 cel mai putin sigur.

5.3 Alternative

Modulul VBAM poate executa in paralel evaluarea fiecarui modul BAM prin pornirea mai multor procese.