Universitatea „Dunărea de Jos” din Galaţi

Școala doctorală de inginerie

REZUMAT TEZĂ DE DOCTORAT

Contribuţii privind clasificarea imaginilor în funcție de conţinut (content based image retrieval)

Doctorand
Mihai-Bogdan Ilie

Conducător științific, Prof. univ. dr. ing. Luminita Dumitriu

Seria I Nr. 2

GALAŢI

2014

Universitatea „Dunărea de Jos” din Galaţi

Școala doctorală de inginerie

REZUMAT TEZĂ DE DOCTORAT

Contribuţii privind clasificarea imaginilor în funcție de conţinut (content based image retrieval)

Doctorand
Mihai-Bogdan Ilie

Conducător științific, Prof. univ. dr. ing. Luminita Dumitriu

Seria I Nr. 2

GALAŢI

2014

Elaborarea şi finalizarea unei teze de doctorat nu poate avea loc decât în condiţiile unei îndrumări ştiinţifice profesioniste, acordată cu tact şi meticulozitate. Datorez mulţumiri şi port o deosebită recunoştinţă doamnei Prof. Univ. Dr. Luminita Dumitriu, atât în calitate de îndrumător ştiinţific, cât şi pentru încrederea acordată.

Doresc să mulţumesc cu această ocazie domnului Prof. Dr. Bart Lamiroy din cadrul Institutului LORIA, Universitatea Nancy, pentru ospitalitatea manifestată şi pentru sprijinul de care am beneficiat în înţelegerea problemelor întâlnite în domeniul procesării de documente.

Mulţumesc de asemenea pentru sprijinul material acordat şi pentru activităţile întreprinse, proiectului SOP HRD /107/1.5/S/76822- TOP ACADEMIC, din cadrul Universităţii Dunărea de Jos.

Nu în ultimul rând, mulţumesc şi sunt adânc recunoscător familiei mele pentru încurajări, pentru sprijinul moral şi pentru înțelegerea de care au dat dovadă, ajutându-mă astfel să duc la bun sfârşit această teză.

Cuprins

Introducere

Domeniul de cercetare

Definiţie: În domeniile ingineriei electrice şi al ştiinţei calculatoarelor, **procesarea de imagini** reprezintă orice formă de procesare a semnalului, unde intrarea este reprezentată de o imagine (cum ar fi o poză, sau un instantaneu din cadrul unei secvenţe video), iar ieşirea e reprezentată de o altă imagine, sau de un set de caracteristici, sau parametri ai acesteia. Majoritatea tehnicilor de procesare de imagini tratează imaginea ca un semnal bidimensional, pe care aplică tehnici standard de procesare a semnalelor.

Definiţie: **Clasificarea imaginilor în funcție de conţinut (Content-Based Image Retrieval – CBIR)** este cunoscută şi sub numele de Query By Image Content (QBIC), sau Content-Based Visual Information Retrieval (CBVIR) şi reprezintă o soluţie la problema căutării unor imagini digitale într-o bază de date de dimensiuni ridicate. Termenul de "Content-Based" se referă la faptul că tehnicile CBIR folosesc conţinutul real al imaginii, spre deosebire de datele extrase din metadata, cum ar fi cuvinte cheie, etichete, sau descrieri asociate imaginii.

Procesarea imaginilor este folosită preponderent în două zone distincte:

- Îmbunătăţirea calităţii unor imagini pentru a fi observate/analizate de un utilizator uman. Din această categorie fac parte inclusiv imprimarea și transmisia imaginilor.
- Analiza imaginilor în vederea obţinerii unor caracteristici şi structuri. Descompunerea imaginilor este bazată în special pe tehnici de identificare a marginilor, intensitate luminoasă unică, textură, culoare, sau o combinaţie între acestea. Forma e un concept mai dificil, dar există modalități de a o descrie matematic. Toate aceste tehnici sunt de obicei folosite pentru a clasifica, sau recunoaște obiecte. Lucrarea de față tratează această zonă, în contextul general al CBIR.

Necesitatea apariției fenomenului de CBIR a fost dictată de mai multe zone de interes. În faza incipientă, clasificarea imaginilor se făcea pe bază de etichete text. Acest lucru s-a dovedit a fi foarte anevoios, sau chiar imposibil, în unele cazuri.

Din această cauză, tehnicile de procesare de imagini au fost combinate şi extinse, pentru a acoperi o gamă largă de aplicaţii, pornind din zona utilizatorului comun până la zone foarte complexe, după cum urmează:

- detectarea de duplicate, în colecțiile de poze de acasă, dar şi pentru copyright;
- organizarea imaginilor;
- căutarea de imagini care conțin subimagini asemănătoare cu una dată;
- căutarea de imagini care se potrivesc cromatic într-un ambient;
- aplicaţii medicale - zonă separată de CBIR, în care de obicei are loc un proces multiplu de clasificare, dublat la un moment dat de expertul uman;
- procesarea documentelor - o altă zonă separată a procesării imaginilor, atât în ceea ce privește tipul și structura copiilor, cât şi zona țintă căreia i se adresează;
- securitate, supraveghere video;
- face recognition, amprente;
- industrie - pentru detectarea de produse greșite
    - lipsește un controller
    - cantitatea de lichide din recipiente
    - numărul de pastile din folii
    - cantitatea (bulele) de aer din plastic
    - aproximarea numărului de fulgi de porumb din pungă
- militare
    - tracking
    - țintire
    - reconstrucție digitală a unei zone, pornind de la o poză

Definiţie: **Procesul de analiză şi recunoaștere a documentelor (eng. *Document analysis and Recognition/Retrieval - DAR*)** reprezintă totalitatea etapelor implicate în extragerea de informații logice şi semnificative din copii ale documentelor imprimate pe hârtie. Până în prezent au fost propuse multiple abordări care încearcă recunoașterea diferitor categorii de documente, prin extinderea problemei OCR, sau prin analiza aspectului documentelor, pornind de la un set variat de cerințe. Scopul final al acestor implementări este de a stabili o structură fizică, sau logică, şi de asemenea de a identifica elemente distincte, de sine stătătoare, sau interconectate într-un anumit context.

În general, aplicațiile DAR sunt asociate unor cerințe specifice, dictate de necesităţile apărute în diferite zone de interes.

De cele mai multe ori cercetătorii se concentrează pe *procesarea documentelor vechi*, care au fost digitalizate, respectiv pe extragerea informațiilor relevante din aceste documente. În acest context se intenționează rularea unor algoritmi de OCR, care încearcă să extragă în mod automat textul cuprins în aceste copii. Sub-problemele ridicate de acest domeniu sunt variate; în cele ce urmează menționăm câteva din categoriile ţintă:

- preprocesarea documentelor, în vederea binarizării lor – ca o categorie aparte menționam documentele vechi, deteriorate, în care este necesara introducerea de noțiuni logice pentru a putea distinge elementele de zgomot de cele grafice. În absenta unui algoritm de binarizare eficient, etapele ulterioare (inclusiv cea de OCR) eșuează de cele mai multe ori;
- implementarea de algoritmi OCR pentru diferite limbi/dialecte – menționăm aici problemele ridicate de variația direcției şi a sensului de scriere;
- segmentarea de caractere – o problemă extrem de dificilă în contextul documentelor scrise manual. Menționăm în acest context şi problema de detectare şi înțelegere a adnotărilor.

O altă zonă de interes este dictată de *analiza automata a formularelor oficiale*, în care se încearcă stabilirea structurii fizice a unor document cu format bine definit, în vederea extragerii informației introduse de factorul uman. O zonă similara este cea a *clasificării unor obiecte pe baza unor etichete* care conțin informații textuale. Cel mai întâlnit scenariu este redirecționarea coletelor poștale pe baza informațiilor introduse de expeditor. Desigur, în ambele situații algoritmii de segmentare textuală (coloane, paragrafe, linii, caractere) şi OCR joacă un rol foarte important.

În cazul companiilor mari, sau al băncilor, se încearcă *procesarea şi clasificarea automată a documentelor de același tip* (cecuri, cărti de vizită, formulare, embleme etc.). În aceasta situație este introdusă o altă constrângere – cea de execuție cât mai rapidă, pentru a reduce cât mai mult latentele procesului per ansamblu.

O categorie aparte este cea a *literaturii tehnice*, unde se dorește segmentarea elementelor grafice şi extragerea unor informații logice. Cel mai des întalnite scenarii sunt procesările de diagrame, hărţi, grafice etc. şi asocierea lor cu descrierile oferite de autor.

Procesarea documentelor artistice se concentrează pe segmentarea imaginilor în vederea formării de *colecții de documente asemănătoare*. O situație similară, dar de o complexitate mai scăzută, se întâlneşte şi în cazul clasificării documentelor conform emblemelor din antet.

Obiectivele cercetării

Abordările tradiționale de CBIR încearcă rezolvarea acestei probleme prin extragerea unor caracteristici reprezentative din diferite spații (culoare, textură, formă etc.), urmată de compararea acestora cu un set obținut anterior. Rezultatele obținute până acum sunt promițătoare, dar aceste implementări sunt departe de a acoperi toate necesitățile ridicate de un scenariu real. Din această cauză, comunitatea cercetătorilor în acest domeniu converge spre ideea de a cumula caracteristicile obţinute în spații diferite pentru a rafina rezultatele obţinute şi a îmbunătăți performanţele. Cu toate acestea, nu există implementări care să realizeze acest lucru.

De asemenea, abordările curente se adresează unor probleme specifice, aparte în cadrul general al CBIR. Din acest motiv, motoarele de acest gen funcționează relativ asemănător, în sensul în care se folosesc de un set de date de antrenare pentru a clasifica ulterior imaginile primite ca input de la un utilizator oarecare. Menționăm aici că atât imaginile folosite în cadrul etapei de antrenare, cât şi cele folosite pentru a testa/valida corectitudinea clasificării provin din aceeași zonă. Din acest motiv, organizațiile care preiau aceste implementări pentru a le testa pe imagini în contextul general al CV (eng. *computer vision*) obțin performanțe mult mai scăzute.

Prezenta lucrare își propune implementarea unui sistem CBIR antrenat pe un set de imagini "tradiționale", care să poată recunoaște şi clasifica atât imagini de același gen, cât și imagini extrase din documente. Subproblemele ridicate variază, după cum urmează:

- preprocesarea imaginilor;
- extragerea de caracteristici din diferite spații;
- implementarea unui modul de învățare supervizată;
- segmentarea imaginilor din documente;
- încadrarea rezultatelor obţinute.

Concluzia la care vrem să ajungem este că un motor de CBIR obține rezultate superioare în condițiile prezenței mai multor descriptori (chiar de aceeași proveniență), ajutându-l astfel să clasifice corect imaginile extrase din documente, deci dintr-o altă zonă decât cea din care provin imaginile folosite ca date de antrenare.

Introduction

Research field

Definition: In the fields of electrical engineering and computer science, the **image processing** concept is composed of any signal processing technique, where the input is represented by an image (a standalone picture or a snapshot taken from a video sequence) and the output is represented by a different image or by a set of characteristics or parameters which describe it. Most of the image processing algorithms treat the image as a bi-dimensional signal on which they apply standard signal processing techniques.

Definition: **The Content-Based Image Retrieval – CBIR** concept is also known as Query by Image Content (QBIC) or Content-Based Visual Information Retrieval (CBVIR) and represents a solution to the problem of searching digital images in a large database. The "Content-Based" term implies that the CBIR techniques use the real image content, as opposed to the information extracted from the metadata fields, like key words, labels, tags or any other form of description associated to the image.

The image processing techniques are mainly used in two distinct areas:

- Improving the image quality in order for it to be observed and analyzed by a human person. We are mentioning in this category the image printing and image transmission fields;
- Image analysis – in order to obtain a set of characteristics and structures. The image decomposition algorithms are usually based on techniques which are targeting edge detection, unique light intensity, texture or colour recognition, or a combination of all. Shape is a more difficult concept but there are some methods to describe it mathematically. All these techniques are used to classify or recognize objects; this paper is located in this area, in the general concept of CBIR.

The necessity of CBIR was imposed by the needs in multiple fields. Primarily the image classification was conducted based on text labels. This was proven to be error prone, very hard to achieve or even impossible in some cases.

Given the above, the image processing techniques have been combined and extended in order to cover a large application area, targeting the common end user needs or very complex scenarios, as it follows:

- Duplicate detection, useful for home picture collections and for copyright detection;
- Image classification;
- Searching for images that match certain sub-areas from one provided as a query;
- Searching for images which match chromatically;
- Medical applications – a separate CBIR area, where the classification process is usually doubled by human expertise;
- Document processing – another separate area in the image processing field, in what regards the image type and structure and the targeted field;
- Security and video surveillance;
- Face and print recognition;
- Industry – for detecting erroneous products:
    - Missing controllers;
    - Measuring the liquid levels in containers;
    - Counting the air bubbles in plastic products;
    - Approximating the corn flakes in a bag;
- Military
    - Tracking;
    - Aiming;
    - Digital reconstruction of an area, starting from a picture.

Definition: **The process of Document Analysis and Recognition/Retrieval – DAR** is represented by all the stages involved in extracting logical and meaningful information from paper (or other similar materials) print scans. Until today, many methods/approaches have been developed targeting the recognition of different document categories, by extending the OCR problem or by analyzing the documents' aspect, starting from a complex requirements set. The final purpose of these implementations is to establish a physical or logical structure and to identify distinct elements, standalone or interconnected in a certain context.

Generally, the DAR applications are associated with specific needs, dictated by the necessities emerged in different fields.

Most frequently, the researchers focus on *processing old documents* which have been digitalized or by extracting the relevant information from these documents. Given this context, the purpose is to run specific OCR algorithms which are trying to extract the text included in these scans automatically. This field raises a numerous set of problems; we will mention below some of the targeted categories:

- Document preprocessing, targeting the document binarization – as a specific category we are mentioning the old, deteriorated documents, where the algorithms need to make use of logical elements in order to distinguish the graphical details from the noise. In the absence of an efficient binarization algorithm all the subsequent stages (OCR included) will usually fail;
- Implementing OCR algorithms for different languages – we are mentioning here the problems raised by the text direction variation;
- Character segmentation – a task which is very difficult to achieve in the context of hand written documents. We are also mentioning the problem of detecting and understanding annotations.

Another DAR field is represented by *the automated official document analysis*, where the target is to establish the physical structure of a set of documents with a well defined format; the final purpose is to extract the information entered previously by a person. A similar area is the one of *object classification based on label processing*, where the labels contain textual information. The most common scenario is the automated package shipping in the post offices, based on the information entered by the expeditor. Of course, in both cases the OCR and textual segmentation (columns, paragraphs, lines, characters) algorithms play a very important role.

In the case of large companies or banks the researchers are trying to *automatically process and classify the documents with the same characteristics* (checks, business cards, official forms, logos etc.). This area enters a new constraint, of minimizing the execution time, in order to reduce the overall process latencies.

In the *technical literature* category the target is to segment the graphical elements and extract logical information out of them. The most common scenarios involve processing diagrams, charts, maps etc. and associating them with the captions provided by the authors.

Processing art related documents involves image segmentation techniques with the main purpose of *creating collections of similar documents*. A similar but less complex situation can be encountered in the field of document classification according to the heading logos.

Research objectives

The traditional CBIR approaches are trying to solve this problem by extracting a set of representative characteristics from various search spaces (colour, texture, shape etc.) followed by comparing those with a set computed previously. The results are promising so far but these implementations are far from satisfying all the needs raised by a real world scenario. Given this situation the research community converges to the idea of combining the characteristics obtained in multiple spaces in order to improve the results and the overall performance. Even so, there are no current implementations which can achieve this goal.

Also, the current approaches are addressing specific problems in the general CBIR field. This context causes all the CBIR engines to work in a similar manner, by using the data in a particular training set in order to later classify the images received as input from the users. Usually, both the images in the training set and the ones used to test and validate the classifier are originated in the same area. Because of this approach, the organizations which test these implementations on different types of images in the general context of computer vision obtain much lower performances.

This paper targets to implement a CBIR system trained on a set of "traditional" images, which is able to recognize and classify both similar images and document scans. This purpose imposes a complex set of problems as described below:

- Image preprocessing;
- Extracting characteristics originated in different spaces;
- Implementing a supervised machine learning module;
- Document image segmentation;
- Result comparison.

We are aiming to reach the conclusion that a CBIR engine can obtain superior results when the classifier is trained with multiple descriptors (even from the same search space) leading to a correct classification process when exposed to images extracted from documents, thus originating in a different area than the one used for training.

Notații şi abrevieri

BOW - Bag Of Words
CBIR - Content Based Image Retrieval
CIE - International Commission of Illumination
CMYK - Cyan Magenta Yellow Key (colour space)
CV - Computer Vision
CVSEG - Cluster Variance Segmentation
DAR - Document Analysis and Retrieval
FD - Fourier Descriptor
GCH - Global Colour Histogram
HOG - Histogram of Oriented Gradients
HSL - Hue Saturation Lightness
HSV - Hue Saturation Value
ICA - Independent Component Analysis
LBP - Local Binary Pattern
LCH - Local Colour Histogram
LESH - Local Energy based Shape Histogram
LoG - Laplacean of Gaussian
LSH - Local Sensitivity Hashing
MI - Moment Invariant
MRF - Markov Random Fields
NLBIN - Non-Linear Binarization
NN - Neural Network
PCA - Principal Component Analysis
RGB - Red Green Blue (colour space)
RGVF - Radiating Gradient Vector Flow Snake
SIFT - Scale Invariant Feature Transform
SOM - Self Organising Map
SURF - Speeded Up Robust Feature
SVM - Support Vector Machine
GLOH - Gradient Location and Orientation Histogram
VP - Vantage Point

Lista figuri

Lista tabele

1 Abordări curente în CBIR

Fenomenul CBIR încearcă să imite modul de funcționare al clasificării umane, prin urmare vom descrie tipurile de interogări, domenii de interes, caracteristicile extrase și performanța per ansamblu a sistemelor.

Interogările specifice CBIR pot avea loc la mai multe niveluri (1) nivelul caracteristicilor, nivelul semantic și nivelul afectiv.

Pe măsură ce importanța semanticii şi a nivelului afectiv crește, problema CBIR devine din ce în ce mai dificilă, spre imposibilă. Încă nu există o soluție completă de interpretare a unei interogări la nivelul afectiv. Lucrarea de față se va concentra pe analiza posibilelor abordări pentru celelalte două niveluri.

Pentru analiza unei imagini, un algoritm de procesare a imaginilor trebuie să extragă un anumit număr de caracteristici (eng. pl. *features*). Acestea se împart în două mari categorii - **textuale** și **vizuale**.

Cele **textuale** se regăsesc de obicei în adnotări şi metadata asociată imaginii - etichete, cuvinte cheie, data preluării, etichete geografice (eng. pl. *geo-tags*), numele fișierului, condiții de preluare (expunere, apertură, bliț etc.).

Cele **vizuale** sunt extrase prin analiza pixelilor constituenți – descriptori ai culorii, texturii, formei şi ai așezării în spațiu. La rândul lor, aceste caracteristici se împart în două mari categorii – globale şi locale. Cele globale se referă la întreaga imagine şi conțin informații mediate în urma analizei tuturor pixelilor (intensitate luminoasă medie, cantitatea medie de roșu etc.). Caracteristicile locale descriu zone anume ale unei imagini, obţinute în urma unui proces de segmentare.

Într-o fază premergătoare determinării caracteristicilor se stabilesc punctele de interes (eng. pl. *key points*) din imagine, care vor fi folosite ca și centri pentru zonele de căutare ale algoritmilor de extragere (*feature extraction*).

Toate aplicațiile care folosesc tehnici de procesare a imaginilor sunt clasificate în funcție de două criterii - viteza de calcul, respectiv precizia.

Unul din factorii determinanți ai apariției domeniului de procesare a imaginilor a fost cel uman. În fază incipientă s-a încercat clasificarea imaginilor pe bază de etichete. Pe lângă dezavantajele evidente ale acestei abordări, una din problemele majore a fost subiectivismul celor care atribuiau aceste etichete. Oamenii încearcă întotdeauna să găsească un sens pentru imaginea pe care o vad. Biologia încă studiază fenomenele optice din cadrul aparatului vizual uman, precum și modul de interpretare și asociere a datelor culese cu informațiile anterioare.

De asemenea, creierul poate cataloga cu ușurință obiectele prezente în imagini, chiar dacă nu sunt neapărat evidente, mai ales în cazurile în care categoriile respective sunt foarte familiare. Acest lucru poate fi realizat în absența unui fundal uniform, sau în absența contururilor; ba mai mult, creierul uman poate intui starea în care se aflau obiectele, atunci când au fost imortalizate. În imaginile de mai jos putem stabili cu ușurință poziția dinamică, chiar dacă în realitate obiectele sunt statice:

Figura 1. Dinamica în imagini

Din păcate, aceleași mecanisme care ne ajută să catalogăm corect anumite imagini ne induc în eroare în alte situații. Cel mai des întâlnite cazuri sunt legate de modul în care

asociem imaginile vizionate experienței noastre anterioare, memoria funcționând pe bază de input de mediu și cultural. Pentru a exemplifica, menționăm cateva exemple:

- perspectiva şi lipsa celei de-a treia dimensiuni influențează modul în care apreciem distanța dintre obiecte situate în planuri diferite;
- europenii nu disting bine fețele asiaticilor - asta nu înseamnă că nu sunt diferite;
- scările urcă pentru europeni şi coboară pentru arabi (care citesc invers);

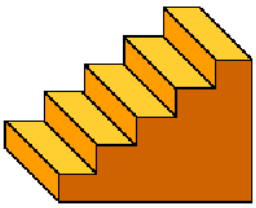


Figura 2. Scări

- creierul are tendința de a completa anumite detalii obturate în momentul capturării imaginii, nu întotdeauna în mod corect;

În acest context, motoarele de CBIR sunt expuse publicului, pentru a culege informații legate de relevanța rezultatelor evaluată uman, nu prin analiză de valori calculate de mașină.

1.1 Spațiul culorilor

Cele mai primare caracteristici ale imaginilor sunt extrase din spațiul culorilor.

Motoarele de CBIR trebuie să țină cont de anatomia ochiului uman în cadrul clasificării. Biologia încă studiază elementele care afectează interpretarea unei imagini de către om, respectiv intensitatea luminoasă și cantitatea de culoare din fiecare imagine. Arhitectura tuturor instrumentelor de înregistrare și stocare a informației vizuale pornește de la capacitatea ochiului uman de a percepe caracteristici precum cantitatea de culoare sau intensitatea luminoasă, chiar dacă poate fi ușor păcălit.

Analiza caracteristicilor unei imagini la nivelul culorii pare o sarcina simplă. Cu toate acestea, problemele ridicate de această analiză sunt foarte complexe; ba mai mult, toate nivelurile superioare de analiză a caracteristicilor se bazează pe informațiile obținute aici.

Există multe abordări în ceea ce privește reprezentarea informației dintr-o imagine din punct de vedere al culorii, dar toți autorii pornesc de la același triplet:

- lumină (unghi, culoare etc.);
- obiect (formă, culoare, suprafață etc.);
- senzor (expunere, rezoluție etc.).

Scopul unui spațiu (sau model, sau sistem) al culorilor este de a facilita determinarea unei culori într-un mod standard. Informația aflată în culoarea unui obiect este determinată de trei factori - sursa luminii, caracteristicile obiectului, respectiv caracteristicile senzorului.

Un astfel de spațiu conferă un sistem de coordonate și un subspațiu, în care fiecare culoare poate fi descrisă printr-un punct. Cele mai des întâlnite spații ale culorilor sunt:

- RGB (pentru monitoare și camere de luat vederi);
- CMY/CMYL (pentru imprimante);
- HSI/HSV/HSL/HSB (procesare de imagini);
- CIE Lab (procesare de imagini).

Spațiul RGB poate fi descris printr-un cub, care are pe fiecare latură coordonate ale celor trei culori primare. Dacă aceste culori primare sunt reprezentate pe 8 biti (imagine pe 24 de biti RGB), numărul total de culori e $(2^8)^3$ = 16,777,216. Informațiile de pe canalele RGB sunt oferite direct de aparatele care înregistrează imaginile (aparate foto, camere de luat vederi etc.). Pe de altă parte, spațiul RGB nu e foarte precis şi de obicei se normalizează.

În funcție de necesitățile zonei de interes, se pot folosi diverse spații ale culorilor. Două din cele mai interesante sunt c1c2c3, respectiv l1l2l3.

c1c2c3 are proprietatea de a elimina atât umbrele, cât şi zonele de *highlight* din imagine. Acesta este descris prin formulele de mai jos:

$$c_1 = arctg\frac{R}{\max(G,B)}; c_2 = arctg\frac{G}{\max(R,B)}; c_3 = arctg\frac{B}{\max(G,R)}$$

Formula 1. Spațiul c1c2c3

l1l2l3 este folosit în special pentru scene în care zonele de *highlight* sunt importante, cu alte cuvinte pentru obiecte lucioase, deoarece elimină umbrele, dar păstrează zonele strălucitoare. Acesta este descris prin formulele de mai jos:

$$l1(R,G,B) = \frac{(R-G)^2}{(R-G)^2 + (R-B)^2 + (B-G)^2}$$

$$l2(R,G,B) = \frac{(R-B)^2}{(R-G)^2 + (R-B)^2 + (B-G)^2}$$

$$l3(R,G,B) = \frac{(B-G)^2}{(R-G)^2 + (R-B)^2 + (B-G)^2}$$

Formula 2. Spațiul l1l2l3

Histogramele reprezintă o colecție de statistici legate de culoare şi intensitate, prin procesarea fiecărui pixel. Modalitățile de abordare (tehnicile de calcul) pot varia, dar în marea lor parte folosesc matrice de asemănare între culori şi nu țin cont de poziția culorii în imagine. În prezent se încearcă abordări noi, respectiv histograme fuzzy și folosirea centrilor de culoare. Din păcate nu funcționează pentru mai multe zone cu aceeași culoare.

Tehnicile de *colour moments* țin cont de contextul pixelului analizat în cadrul întregii imagini. În principal, se calculează media și deviația standard a pixelului, pe baza de discretizare.

Se pot aplica tehnici fuzzy (2) pe niște zone prestabilite (de exemplu centru plus încă 4 pe margini), dar nu întotdeauna dau rezultate bune (de exemplu în cazurile în care imaginea nu este centrată, sau atunci când imaginea descrie mai multe subiecte/stări).

Momentele de culoare nu variază în cazul scalării, sau al rotirii. De obicei numai primele 3 momente de culoare sunt calculate pentru a determina caracteristicile unei imagini, deoarece în momentele de ordin jos se află cea mai multă informație (3). Acestea se calculează per canal, în funcție de spațiul de culoare ales (de exemplu, în cazul RGB vom folosi 9 momente, iar în cazul CMYK, 12). Momentele de culoare au dat rezultate bune în contextul schimbării intensității luminoase, dar nu funcționează foarte bine la intensități foarte reduse.

Conform unui studiu realizat asupra posibilelor implementări ale motoarelor de CBIR (4), cuantizarea avantajează tehnicile bazate pe momente de culoare. Cu toate acestea, recomandarea este să se folosească o metodă **mixtă** de extragere de caracteristici.

1.2 Spațiul texturilor

Textura este proprietatea intrinseca a unei suprafețe, de a descrie structuri vizuale repetitive (eng. pl. *patterns*), fiecare având aceleași proprietăți de omogenitate. Textura poate conține informații importante despre suprafețe şi poate descrie relația dintre suprafața analizată şi mediul înconjurător (5). Proprietățile texturale includ granulația, contrastul, direcția, liniaritatea, regularitatea și relieful (asperitate).

Exista patru categorii majore de algoritmi pentru determinarea texturilor:

- **tehnici statistice** – caracterizează texturile în funcție de diferitele niveluri de gri ale pixelilor care compun o suprafață.

- **tehnici geometrice** – caracterizează texturile ca fiind compuse din unități structurale primitive simple, numite texeli (eng. *texels*), așezate pe o suprafață în mod regulat.
- **tehnici spectrale** – bazate pe proprietăți ale spectrului Fourier pentru a descrie periodicitatea globală a nivelurilor de gri de pe o suprafață, prin identificarea punctelor de energie înaltă.
- **tehnici bazate pe model** – folosesc abordări statistice de distribuție, pornind de la parametri aleatori atribuiți unui pixel (Markov random fields), sau fractali.

In cadrul tehnicilor statistice, vom mentiona abordarea LBP (eng. *local binary patterns*):

- se împarte fereastra în celule (de exemplu 16x16 pixeli);
- în fiecare celulă se compară fiecare pixel cu cei 8 vecini ai lui;
- oriunde valoarea pixelului este mai mare decât a vecinului se marchează un 1, altfel - 0. Numărul binar obținut este de obicei transformat în zecimal.
- se calculează histograma numerelor astfel obţinute într-o fereastră;
- opțional se normalizează (în special pentru ferestre mari);
- se concatenează histogramele tuturor celulelor, de unde rezultă un vector de caracteristici.

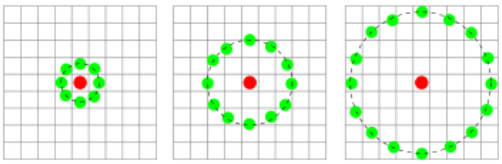


Figura 3. LBP pe celule de 9x9 pixeli

Pentru interogare se folosesc de obicei SVM (eng. pl. *support vector machines*).

A se observa că în cadrul algoritmului există tendințe de a discretiza imaginea (în ferestre mai mari) și de a aplica forme de *hashing*, ceea ce duce la scăderea timpilor de execuție.

1.3 Spațiul formelor

Caracteristicile formelor dintr-o anumită imagine sunt supuse unor serii constrângeri/cerințe, cum ar fi recunoașterea după translatare, scalare, rotație, stabilitatea după modificări mici ale formei, respectiv o serie de restricţii asupra complexităţii de calcul.

Abordările se împart în două mari categorii:

- bazate pe ***margini***;
- bazate pe ***regiuni***.

În ciuda restricţiilor impuse, descriptorii extrași din spaţiul formelor sunt sensibili la obturari și la prezenţa elementelor de zgomot, prin urmare nu vor fi folosiţi in implementarea proprie.

1.4 Segmentarea

Principalul scop al segmentării este obținerea regiunilor de interes în contextul unei anume aplicaţii și anume de a detecta regiuni omogene, sau de a detecta margini sau contururi. A fost aplicată cu succes pentru a detecta obiecte atât din scene statice, cât și dinamice; o altă zonă de interes este calcularea așezării obiectelor într-o scenă, pentru a calcula traiectoria unui robot.

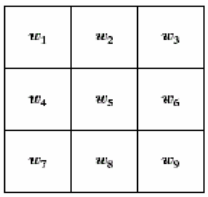
Abordările principale sunt:

- cele bazate pe detectarea marginii;
- cele bazate pe analiza unei regiuni.

Soluțiile găsite pot fi bazate pe intensitatea culorilor, textură, culoare, mișcare etc.

Subproblemele ridicate de acest domeniu sunt legate de **detectarea discontinuităților** – puncte, linii, margini.

Detectarea lor se face în mod obișnuit prin aplicarea unei măști pe imagine, ca în figura de mai jos:



$$R = w_1 z_1 + w_2 z_2 + \cdots + w_9 z_9 = \sum_{i=1}^{9} w_i z_i$$

Figura 4. Exemplu de mască aplicată pentru detectarea discontinuităților

1.5 Descriptori locali

În urma aplicării procesului de segmentare se obțin regiuni (zone) de imagine, care determină caracteristici. Aceste caracteristici pot fi margini, colțuri, regiuni de interes (eng. Regions of Interest - RoI), sau creste (eng. ridge).

Punctele de interes sunt acele puncte care nu sunt afectate de translatare, scalare sau rotire şi sunt minim afectate de zgomot, sau de modificări minore.

Principalele abordări în această zonă sunt:

- SIFT (eng. Scale Invariant Feature Transform);
- SURF (eng. Speeded Up Robust Feature);
- GLOH (eng. Gradient Location and Orientation Histogram);
- HOG (eng. Histogram of Oriented Gradients);
- LESH (eng. Local Energy based Shape Histogram).

1.6 Dimensionalitatea datelor

Toate abordările descrise mai sus funcționează pe baza aceluiași principiu - descompunerea unei imagini în seturi de caracteristici (eng. pl. feature vectors), care să fie ulterior comparate cu cele ale imaginilor din baza de date. Acești vectori au dimensionalități variate. De exemplu, dimensionalitatea spațiului RGB este de 3, pe când cea a histogramelor e de 256, iar a texturilor obţinute în urma aplicării unui algoritm ICA depășește 600.

Prin urmare, una din subproblemele majore ale CBIR este găsirea unui mod eficient și rapid de a stoca şi parcurge datele obţinute în urma aplicării diferiților algoritmi. Din fericire, aceste date nu sunt foarte dense și li se pot aplica algoritmi de indexare multidimensionala, care folosesc măsuri de similitudine sau metode de genul cel mai apropiat vecin (eng. nearest neighbour).

Exista doua abordări majore în ceea ce privește rezolvarea acestei probleme:

- structuri de date arborescente (eng. *tree data*) – R-tree, Quad-tree, k-D tree, VP tree;
- algoritmi de hashing - LSH.

2 Tehnici de prelucrare a documentelor

În contextul CBIR pentru copii (eng. pl. *scans*) de documente, procesul de Analiză şi Recunoaștere a Documentelor (eng. *Document Analysis and Recognition – DAR*) este crucial pentru a clasifica în mod corect elementele primite ca input.

Scopul principal al analizei de documente este de a recunoaște textul și elementele grafice din imaginile incluse într-un *scan*, așa cum ar face o persoană. DAR intenționează să reconstruiască în mod automat un set de informații logice şi reprezentative, inițial stocate pe documente imprimate pe hârtie, sau materiale asemănătoare.

Domeniul de cercetare DAR include mai multe direcții, cum ar fi binarizarea, reducerea zgomotului, segmentarea (pentru imagini, text, linii de text, cuvinte și caractere), determinarea factorului de distorsionare (eng. *skew estimation*), OCR (eng. *optical character recognition*) și multe altele.

Obiectivul **binarizării** este de a alege automat un prag care separă informația de pe fundal de cea din prim plan. Alegerea acestui prag este de obicei un proces empiric, de genul încercare/eroare (eng. *trial and error*). Deși cei mai recenți algoritmi folosesc praguri adaptive, aceștia încă depind de o largă varietate de imagini incluse în setul de date de antrenare. De multe ori autorii încearcă abordări de redimensionare a documentelor, pentru a filtra elementele de zgomot.

Algoritmii de segmentare a documentelor țintesc zone diferite de interes, conform domeniilor în care urmează să fie folosiți, după cum urmează:

- segmentare de **imagini**;
- segmentare de **paragrafe şi coloane** de text;
- segmentare de **rânduri** de text;
- segmentare de **cuvinte și caractere**.

Lucrarea de față își propune să analizeze în mod special **segmentarea imaginilor**. În această zonă există mai multe implementări, dar majoritatea lor folosesc aceleași abordări – binarizare, urmată de analiza texturilor, proiectarea pe axe, sau analiza redimensionării documentului. În unele cazuri sunt aplicați şi anumiți operatori pe grila de pixeli, cum ar fi dilatarea.

Exista mai multe moduri de a clasifica abordările DAR curente, dar unul din cele mai complete studii (6) stabileşte următoarea taxonomie de algoritmi:

- bazați pe **caracteristici ale imaginii** – descriptori locali şi globali pentru culoare, formă, textură, gradient s.a.m.d.;
- bazați pe **structura fizică** – care stabilesc ierarhia geometrică a documentului;
- bazați pe **caracteristici logice** – care stabilesc ierarhia logică a documentului;
- bazați pe **caracteristici textuale** – cuvinte cheie, determinate în urma unui algoritm OCR.

În funcție de strategia de clasificare, algoritmii se împart în două clase principale:

- **bottom-up** – care încep prin a analiza pixeli, zone, sau regiuni adiacente; obiectele obţinute astfel sunt apoi reunite şi clasificate ca zone ale documentului;
- **top-down** – care încep analiza de la nivelul întregii imagini şi apoi încearcă împărțirea ei în regiuni unitare.

Eforturile depuse în ultima perioadă în acest domeniu se adresează unor probleme anume, concrete, iar soluțiile finale sunt date de intersecția diferiților algoritmi implicați în diferite faze de procesare. În acest context, rezultatele algoritmilor de ultimă generație variază mult, în funcție de zona de interes. Mulți autori au folosit copii de ziare și reviste, fotografii ale unor documente istorice, documente tehnice, sau formulare oficiale. Scopul algoritmilor variază de asemenea, dar majoritatea tind spre tehnici OCR, sau de analiză a ierarhiei. Rezultatele procesării de documente strict în zona segmentării de imagini gravitează undeva în jurul plajei de 80%-90%.

3 Contribuții privind tehnicile de procesare a documentelor

După cum am menționat anterior, majoritatea implementărilor din acest domeniu vizează o zonă aparte, cu un context încadrat precis OCR pentru caractere latine, OCR pentru caractere arabe, sau asiatice, clasificarea automată a scrisorilor, revistelor, sau a documentelor oficiale, segmentarea imaginilor, procesarea documentelor vechi etc.). Unul din obiectivele acestei lucrări este clasificarea imaginilor prezente în documente; având în vedere acest lucru, a trebuit sa implementam un submodul responsabil de procesarea documentelor, în privința segmentării imaginilor din documente.

Prin urmare, am implementat o serie de algoritmi de binarizare, respectiv un algoritm de segmentare, care au fost testați pe documente provenite din zone diferite:

- copii de calitate scăzută ale unor cărți vechi;
- copii de calitate, provenite din convertirea unor documente PDF.

În cele ce urmează, vom oferi detalii legate de rezultatele obţinute, respectiv problemele întâlnite de-a lungul etapei de dezvoltare a acestui submodul.

3.1 Segmentarea imaginilor prezente în documente

3.1.1 Algoritmul de binarizare

Etapa de binarizare este un pas premergător al etapei de segmentare, pentru a facilita complexitatea calculului. Acest stadiu de preprocesare este necesar, deoarece, în cele mai multe cazuri, documentele sunt *scan*-ate în mod color, sau *grayscale*.

Obiectivul final al unui algoritm de binarizare este de a converti o astfel de imagine în spațiul *black and white* (BW), în așa fel încât să fie păstrate doar elementele importante, eliminând totodată diferitele tipuri de zgomot:

- caractere sau imagini de pe cealaltă pagină, vizibile datorită transparenței paginii;
- curburi ale paginii;
- iluminare incorectă;
- defecțiuni ale aparatului cu care a fost realizată înregistrarea;
- documentele prezintă urme ale întrebuințării de-a lungul timpului etc.

În faza incipientă, imaginea oferită ca *input* este analizată din punct de vedere al numărului de culori prezente. În cazul în care imaginea nu a fost înregistrată în spațiul *grayscale* (deci avem de-a face cu o imagine color), aceasta este convertită. Apoi, această imagine este supusă mai multor tipuri de binarizare, în vederea facilitării etapei ulterioare de segmentare.

În cele ce urmează, vom prezenta variantele de algoritmi de binarizare folosiți in cadrul testelor preliminare:

- **simplă**, prin aplicarea unui prag global de 127 pixelilor componenți – algoritmul este foarte rapid și, în aceeași măsură, foarte sensibil;
- **normalizată**, prin determinarea unui prag global calculat în funcție de intensitățile mediate ale pixelilor constituenți – de asemenea un algoritm rapid, dar mai robust în ceea ce privește documentele cu probleme;
- **mediată** (eng. *average*) - algoritmul este de o complexitate mai ridicată, dar oferă rezultate mult mai bune pe copiile documentelor degradate. Produce imagini mai clare, iar caracterele din text sunt mult mai lizibile;

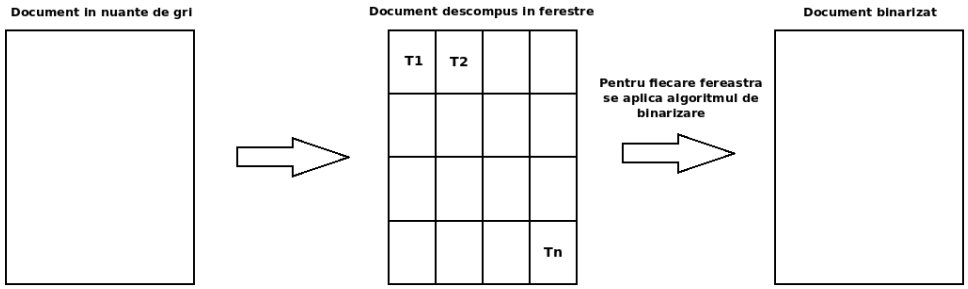


Figura 5. Binarizare mediată

- **interpolată** – se procedează ca în cazul anterior, cu diferența că pragul nu se aplică pe *tile*, ci se reține pentru a calcula valorile interpolate de prag şi pentru a se aplica ulterior. Am renunțat la această abordare, deoarece prezenta o complexitate de calcul mult mai ridicată, iar performanțele obţinute erau foarte asemănătoare cu cele obţinute în urma aplicării binarizării mediate;
- algoritmul de binarizare al lui **Sauvola** - complexitate ridicată, rezultate foarte bune. Pentru a obține rezultate cât mai relevante, a fost folosită o implementare îmbrățișată de comunitate; prin urmare, acest algoritm a fost preluat din biblioteca *ocropus*;
- algoritmul **NLBIN** (eng. *non-linear binarization*) - complexitate ridicată, rezultate excelente. În urma experimentelor efectuate atât pe documente degradate, cât și pe documente de calitate ridicată, acest algoritm a produs rezultate foarte bune, având tendința de a elimina majoritatea formelor de zgomot prezente în imagine şi de a reconstrui detaliile incomplete. Din păcate, datorită caracterului neliniar, complexitatea de calcul este cea mai ridicată, în comparație cu restul algoritmilor prezentați.

3.1.2 Algoritmul de segmentare - Clustering Variance Segmentation (CVSEG)

Acest algoritm preia grila de pixeli binarizati anterior şi încearcă descoperirea imaginilor din interiorul documentului. În cele ce urmează vom descrie și exemplifica etapele implementării algoritmului.

Pornim de la presupunerea că textul are o structură mult mai variată decât o imagine, sau decât fundalul. Din acest motiv, ferestrele care prezintă o varianță ridicată sunt filtrate din imaginea inițială.

Modalitatatea de stabilire a variației se bazează pe calculul a doi parametri:

- abaterea fiecărui pixel de la media nivelului de gri dintr-o fereastra. În faza incipientă s-a încercat ca şi criteriu de varianță folosirea numărului de pixeli activi (negri), dar în acest caz erau filtrate prea multe din ferestrele închise;

$$G_L = \frac{1}{w * h}\sum_{i,j} p_{i,j}\,, nivelul\ mediu\ de\ gri$$

$$\Delta_G = \frac{1}{w * h}\sum_{i,j} |p_{i,j} - G_L|, unde\ w\ reprezinta\ latimea, iar\ h\ reprezinta\ inaltimea\ ferestrei$$

Formula 3. Stabilirea abaterii de la medie într-o fereastră

- diferența mediată a nivelurilor de culoare (0, sau 255) dintre pixelii învecinați:

$$C_L(p_1, p_2) = \begin{cases} 0, daca\ p1 = p2 \\ 1, daca\ p1 != p2 \end{cases}$$

$$\Delta_C = \frac{1}{w * h}\sum_{i,j}\frac{1}{N_{i,j}}\sum_{m,n} C_L\left(p_{i,j}, p_{m,n}\right), m != i, n != j$$

$$N_{i,j} - \ numarul\ de\ vecini\ ai\ pixelului\ p_{i,j}$$

Formula 4. Stabilirea variației contrastului

În cele din urmă, fereastra este invalidată atunci când $\Delta_G > T_G$, respectiv $\Delta_C > T_C$. Determinarea pragurilor a fost efectuată experimental, obținându-se valorile de 0.1, respectiv 0.37. Cu toate acestea, în unele cazuri această etapă nu a fost suficientă pentru a filtra toate ferestrele invalide, așa cum se poate observa şi în imaginea de mai jos:

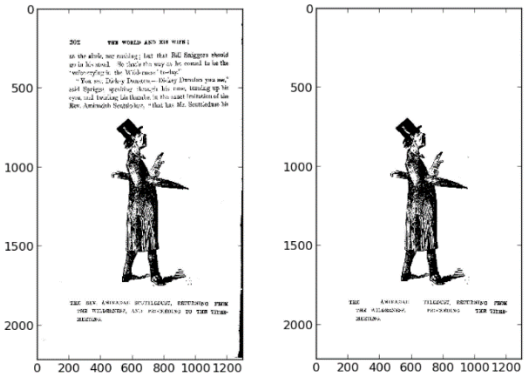


Figura 6. Ferestre cu text

Prin urmare, s-a impus adăugarea unui pas premergător, de eliminare a textului. Această etapă este de fapt o variantă simplificată a algoritmului RXYC. Alegerea acestui tip de algoritm a fost determinată de complexitatea de calcul scăzută, respectiv de viteza ridicată de procesare a documentului. Menționăm aici şi faptul că algoritmul RXYC nu este foarte eficient atunci când avem de-a face cu documente în care textul este înscris intr-un chenar. Imaginea de mai jos arată rezultatele aplicării acestei etape intermediare:

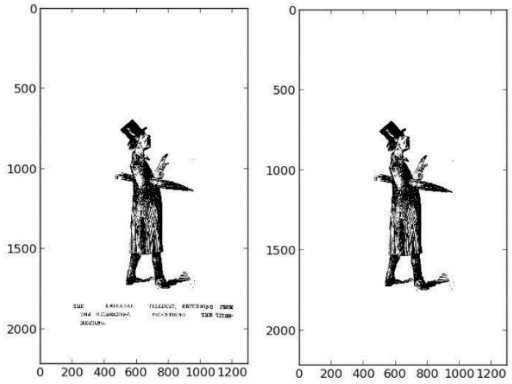


Figura 7. Aplicarea algoritmului de filtrare a textului

După parcurgerea etapelor de filtrare a textului şi de filtrare conform variantei, am observat că, în unele cazuri, imaginea rezultat poate conține ferestre singulare (sau cu foarte puțini vecini), de obicei cauzate de pete, sau alte imperfecțiuni prezente în document. Din acest motiv, a fost necesară introducerea unei alte etape intermediare, care să filtreze aceste elemente de zgomot. Acest filtru calculează un scor pentru fiecare fereastră, ca funcție de ferestre vecine active (validate anterior). Ca și în cazul anterior, pragul a fost stabilit experimental la valoarea de 3 (orice fereastră cu 2, sau mai puțini vecini este eliminată). Rezultatele pot fi observate în imaginile de mai jos:

- prima imagine e rezultatul aplicării algoritmului de binarizare (în cazul de față, NLBIN);
- a doua imagine reprezintă ferestrele validate de primele două etape ale algoritmului de segmentare;
- a treia imagine conține rezultatul filtrării - a se observa prezenta elementelor de zgomot, atât în colțurile din stanga jos şi dreapta sus, cât și reminiscentele filtrării anterioare, împrăștiate pe toată pagina;

- ultima imagine conține rezultatul aplicării filtrului menționat anterior. A se observa lipsa zgomotului.

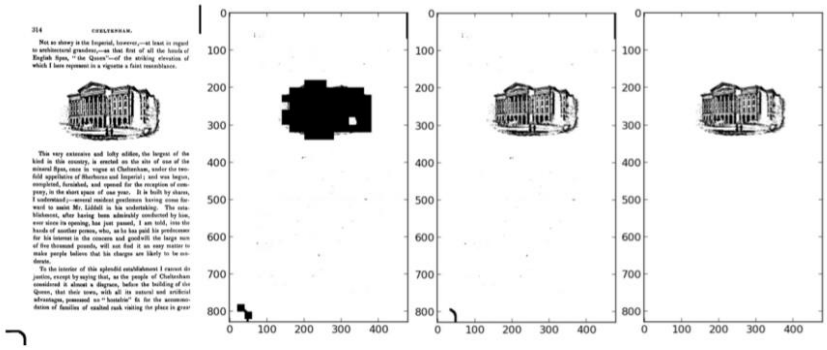


Figura 8. Filtrarea ferestrelor care conțin zgomot

Următoarea etapă este cea de aplicare a algoritmului de clustering. Acest algoritm oferă mai multe avantaje:

- posibilitatea de a elimina *cluster*-e invalide, în cazul în care etapele anterioare au eșuat în procesul de filtrare a tuturor ferestrelor care nu fac parte dintr-o imagine;
- determinarea imaginilor distincte în cadrul unui document care conține mai multe obiecte. Acest lucru este foarte folositor în cadrul clasificării ulterioare, din cadrul modulului de CBIR.

În unele cazuri, datorită transparenței paginii, sau diferențelor de imprimare şi/sau scanare dintre unele zone ale paginii, există ferestre care nu sunt invalidate de pașii anteriori. Desigur, după aplicarea algoritmului de clustering, aceste ferestre sunt grupate și oferite ca rezultat. Pentru a evita acest inconvenient, înainte de a oferi rezultatele utilizatorului, am introdus o etapa de analiză a fiecărui cluster, în care calculăm doi indici - de rarefiere, respectiv de conectivitate.

Indicele de rarefiere se obține astfel:

- se calculează aria suprafeței dreptunghiulare minime, care include toate ferestrele valide dintr-un anumit *cluster* (A);
- se determină indicele de rarefiere, ca fiind raportul dintre suprafața tuturor elementelor *cluster*-ului şi suprafața A.

$$sparsity_{\mathrm{k}} = \frac{\mathrm{w} * \mathrm{h} * \mathrm{S_k}}{\mathrm{A_k}}$$

w și h determină dimensiunea ferestrei,

S_k reprezintă numărul de ferestre din cluster-ul k,

A_k reprezintă cea mai mică fereastră dreptunghiulară, care poate include toate sub-ferestrele

Formula 5. Coeficientul de rarefiere

Indicele de conectivitate se obține astfel:

- pentru fiecare fereastră care are mai mult de un vecin validat, se incrementează un contor K_c;
- se determină indicele de conectivitate ca fiind raportul dintre contorul definit anterior şi numărul de elemente din *cluster* S_k.

$$conn_k = \frac{K_c}{S_k}$$

K_c reprezintă numărul de ferestre care au cel puțin 2 vecini valizi,

S_k reprezintă numărul de ferestre din cluster-ul k

Formula 6. Coeficientul de conectivitate

Pragurile corespunzătoare celor doi indici au fost determinate experimental, după cum urmează: T_{sparse}=0.33, respectiv T_{con}=0.5.

Efectele aplicării acestui mecanism de analiză a cluster-elor poate fi observat în imaginile de mai jos. După cum se poate observa, în cea de-a treia imagine exista 3

cluster-e invalide, care ulterior sunt eliminate, păstrându-se doar imaginea corectă. Prezența celor 3 cluster-e a fost determinată de:

- imperfecțiuni ale algoritmului de binarizare, care a introdus două benzi negre pe marginile documentului;
- imprimarea neuniformă a textului (în partea dreaptă);
- transparența paginii în partea stângă.

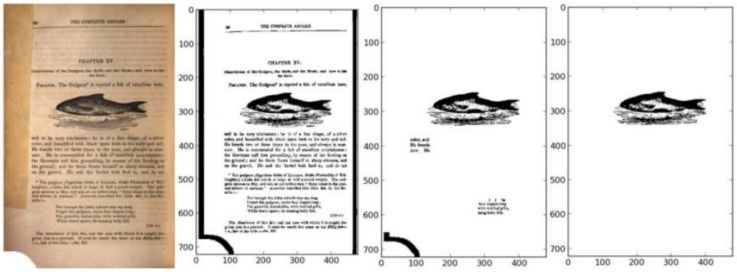


Figura 9. Analiza *cluster*-elor

După adăugarea tuturor acestor etape, am observat că în unele cazuri, anumite ferestre lipseau din obiectul țintă. Acest lucru avea loc în zonele limitrofe imaginii şi era datorat modalității de împărțire a documentului în ferestre. Prin urmare, am introdus etapa de reconstrucție, care are rolul de a extinde imaginea obținută anterior până la o distanță maxim egală cu diagonala unei ferestre. Rezultatele pot fi observate în imaginile de mai jos; se poate observa că:

- în cea de-a doua imagine, datorită modului în care sunt definite ferestrele, unele elemente sunt filtrate din imaginea finală (a se observa crengile bradului din stânga, respectiv detaliile drumului, în partea de jos a imaginii);
- în cea de-a treia imagine este evidentă lipsa crengilor din vârful bradului din partea stângă;
- în cea de-a patra imagine, elementele colorate cu roșu sunt introduse în răspunsul final.

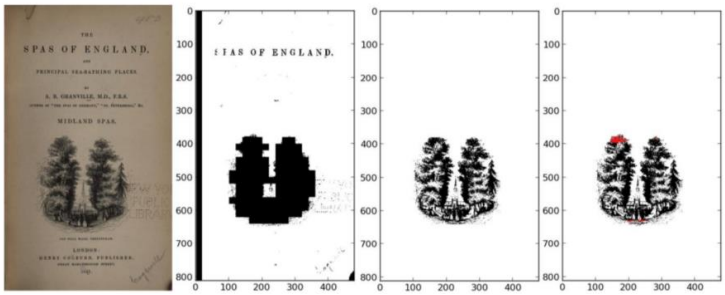


Figura 10. Etapa de reconstrucție

Având în vedere cele de mai sus, varianta finală a algoritmului include următoarele etape:

- se definește o anumită dimensiune de eșantionare a documentului (dimensiunea ferestrelor constituente - w*h);
- se elimină o parte din textul din imagine, pentru a facilita procesarea zonelor rămase;
- momentan se folosește un algoritm bazat pe proiecție pe axele XY;
- se împarte imaginea în ferestre de dimensiuni w*h;
- pentru fiecare fereastră din document:
  - se calculează parametrul Δ_G şi se verifică inegalitatea $\Delta_G > T_G$
  - se calculează parametrul Δ_C şi se verifică inegalitatea $\Delta_C > T_C$
- se elimină ferestrele singulare, care nu sunt conectate la alte ferestre validate anterior, obținându-se astfel mulțimea M a tuturor ferestrelor care sunt părți constituente ale unei imagini;

- pe mulțimea M se aplică un algoritm de K-Means clustering, care are ca metrică de decizie distanța dintre ferestre;
- fiecare din cluster-ele obţinute anterior este analizat, după cum urmează:
  - se calculează scorul de conectivitate al ferestrelor constituente şi se verifică inegalitatea $conn_k > T_{con}$;
  - se calculează scorul de rarefiere și se verifică inegalitatea $sparsity_k > T_{sparse}$;
- rezultatul obținut este expus unui algoritm de reconstrucție, care adaugă iterativ pixelii conecși până la o distanță maxim egală cu diagonala unei ferestre;
- în cele din urmă, cluster-ele sunt grupate pentru a forma imaginile complete, care sunt afișate utilizatorului, salvate, sau trimise mai departe către modulul de extragere de descriptori.

Schema logică a algoritmului este descrisă în figura de mai jos:

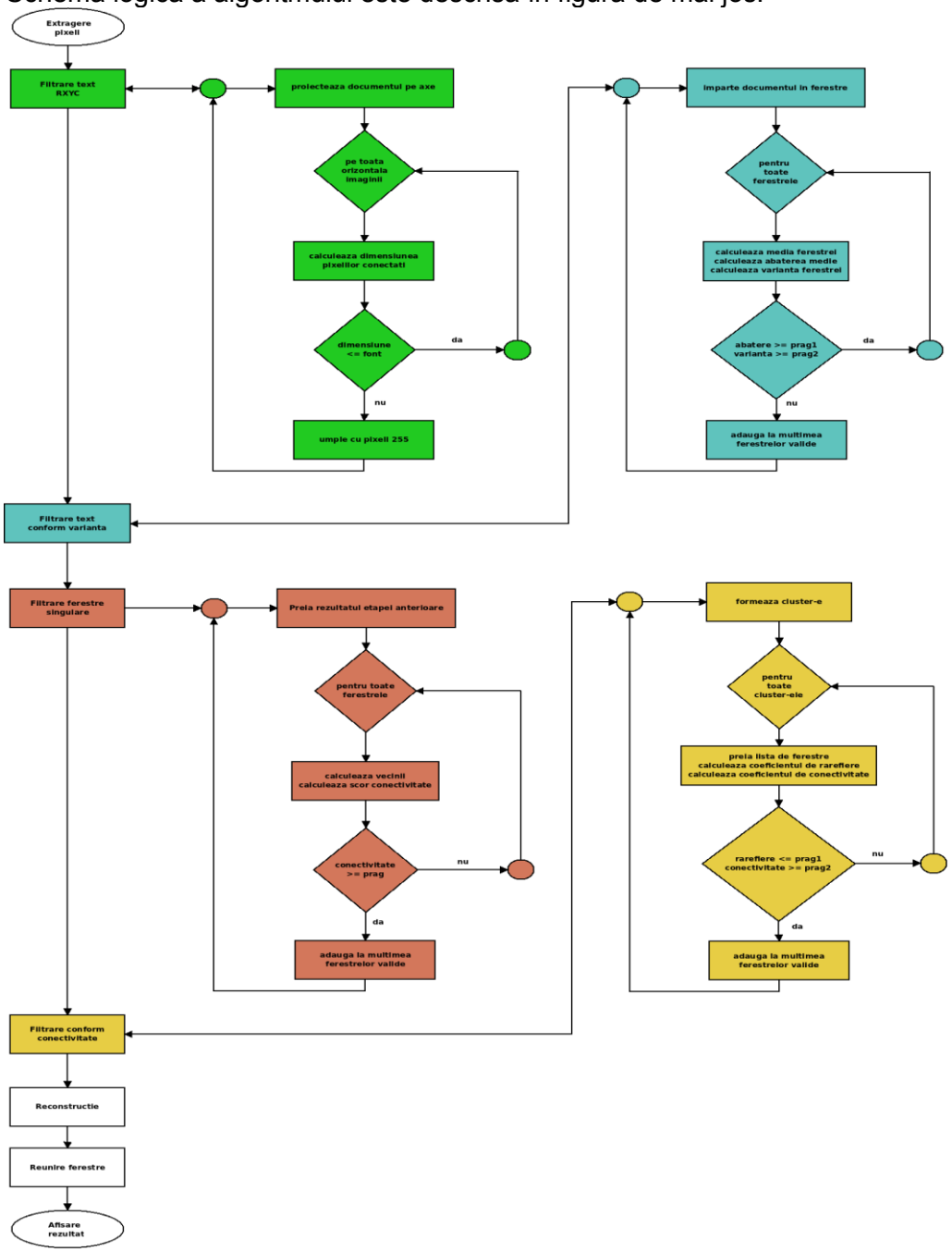


Figura 11. Schemă logică algoritm segmentare CVSEG

Algoritmul a dat rezultate foarte bune pe ambele tipuri de imagini provenite din sursele menționate anterior, așa cum se poate observa în imaginile de mai jos. A se nota eliminarea elementelor de zgomot din documentul vechi şi segmentarea corectă, chiar în prezența chenarului şi a caracterelor cu fundal inversat (alb/negru).

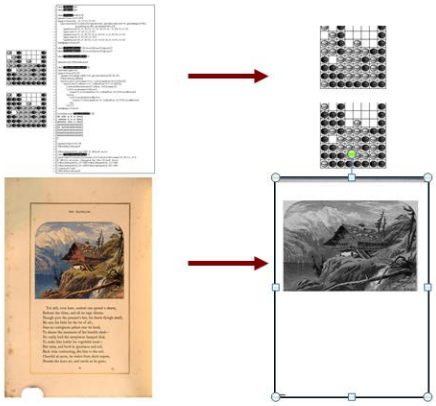


Figura 12. Rezultate segmentare pe două tipuri de documente

3.2 Analiza comparativă a performanței algoritmului de segmentare

Lucrarea de față își propune să analizeze în mod special **segmentarea imaginilor**.

În zona DAR algoritmii de segmentare vizează preponderent **segmentarea de text** şi nu de imagine. Singurul algoritm de segmentare de imagini este cel al lui Bloomberg, reluat ulterior de Syed Saqib Bukharia, care a adăugat unele etape de rafinare a celor două faze de redimensionare. Din acest motiv este necesară o modalitate de a compara rezultatele celor două tipuri de algoritm.

Pornim de la presupunerea că un document este compus din 2 mari zone: prim plan şi fundal.

$$D = B \cup F$$

unde B reprezintă fundalul (eng. *background*), iar
F reprezintă prim planul (eng. *foreground*)
Formula 7. Zone document

La rândul său, prim planul este descris de 2 proprietăți, respectiv compus din 2 zone: imagine şi text:

$$F = T \cup I$$

unde T reprezintă zonele de text, iar
I reprezintă zonele de imagine
Formula 8. Zone prim plan

Prin urmare, ajungem la formula:

$$D = B \cup T \cup I$$

Formula 9. Zone document detaliat

În continuare, pentru a avea un criteriu de comparație, trebuie sa definim o funcție, care să descrie documentul în cele 3 zone componente. Fie:

- f_{Dx} funcţia prin care descriem documentul printr-un algoritm de segmentare x;
- n numărul total de proprietăți prin care vrem să descriem documentul;
- P_i proprietățile propriu zise. În cazul de mai sus, n=3, iar P_i vor fi desigur zonele de text, imagine, sau fundal.

Pentru a descrie fiecare proprietate în parte, vom folosi eroarea de determinare a acestei proprietăți în urma folosirii unui anumit algoritm. Deoarece proprietățile sunt similare în ceea ce privește zona lor de apartenență, nu este necesar să folosim funcții de eroare pentru fiecare. Prin urmare, vom defini funcţia ε_{Pi} prin care vom descrie eroarea de determinare a proprietății P_i, reprezentată prin diferența dintre suprafața reală şi suprafața descrisă de proprietatea rezultată în urma aplicării algoritmului. Deoarece toate proprietățile sunt conținute în documentul D şi nu vrem să influențăm calculul celorlalte erori, eroarea

rezultată va fi împărțită la 2, conform stărilor de apartenență la mulțimea de pixeli determinată de suprafața reală. În concluzie,

$$\varepsilon_{Pi} = \frac{|P_i^r - P_i|}{2D}$$

unde P_i^r reprezintă proprietatea reală,
P_i reprezintă proprietatea calculată, iar
D suprafața totală a documentului

Formula 10. Eroarea de determinare a unei proprietăți

Așadar, funcţia prin care descriem documentul cu ajutorul algoritmului de segmentare x va deveni:

$$f_{Dx} = 1 - \frac{\sum_{i=0}^{n} \varepsilon_{Pi}}{2D}$$

respectiv,

$$f_{Dx} = 1 - \frac{\sum_{i=0}^{n} |P_i^r - P_i|}{2D}$$

unde $\sum_{i=0}^{n} P_i^r = \sum_{i=0}^{n} P_i = D$
iar $P_i \in \{B, T, I\}$

Formula 11. Funcția de descriere (scor) a unui algoritm de segmentare

3.2.1 Determinarea proprietăților lipsă

Majoritatea algoritmilor de segmentare existenți oferă ca rezultat documentul inițial, din care au fost eliminate zonele considerate ca neaparținând zonei țintă. De exemplu, în cazul unul algoritm de segmentare de imagini aplicat pe un document mixt, care conține atât imagini cât şi text, rezultatul va fi documentul inițial, din care a fost eliminat textul.

În principiu, am putea analiza documentul inițial, rezultatul algoritmului şi decide pe baza pixelilor activi cât din rezultat este zona țintă, respectiv zona filtrată şi zona de fundal. Cu toate acestea, o imagine nu este compusă numai din pixeli activi, ci şi din elemente de fundal. Prin urmare, vom aplica următorul algoritm:

1. pe documentul inițial se aplică modificatorul de expandare (dilatare) *dil*. Scopul acestui operator este de a accentua pixelii activi şi este definit pe o fereastră de document astfel:

Figura 13. Documentul inițial

$$dil(D[i:i+s, j:j+s]) = \begin{cases} D[i:i+s, j:j+s] = 0, \frac{1}{s^2}\sum D[i:i+s, j:j+s] * \alpha < 255 \\ D[i:i+s, j:j+s] = 255, otherwise \end{cases}$$

unde D reprezintă documentul inițial
i, j reprezintă indicii ferestrei curente,
s reprezintă dimensiunea ferestrei, iar
α reprezintă pragul de la care considerăm întreaga fereastră "activă"

Formula 12. Operatorul de dilatare

Astfel se obține imaginea expandată:

$$dil(D) = \sum_{i=0}^{w/s} \sum_{j=0}^{h/s} dil(D[i:i+s, j:j+s])$$

unde w şi h reprezintă lățimea, respectiv înălțimea documentului D

Formula 13. Dilatarea întregului document

2. se aplică același operator *dil* pe imaginea obținută ca rezultat al segmentării, obținându-se *dil*(S);

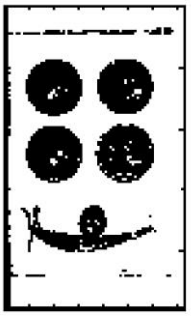


Figura 14. Imagine expandată

3. se determină proprietatea de prim plan lipsă, ca fiind diferența celor două documente. Pentru aceasta definim operatorul delta, astfel:

$$delta\big(dil(D), dil(S)\big) = \begin{cases} 0, dil(D)[i,j] == 0, dil(S)[i,j] == 255 \\ 255, \quad altfel \end{cases}, pentru\ \forall i \in [0, w/s],$$
$$\forall j \in [0, h/s]$$

Formula 14. Operatorul delta

Prin urmare, proprietatea lipsă *dil*(P) = *delta*(*dil*(D), *dil*(S)).

Figura 15. Text expandat obținut prin aplicarea operatorului *delta*

4. se determină fundalul, ca fiind diferența B = *dil*(D) - *dil*(S) - *dil*(P). Ulterior, acest fundal va fi marcat cu pixeli activi pentru ușurarea calculelor de determinare a erorii.

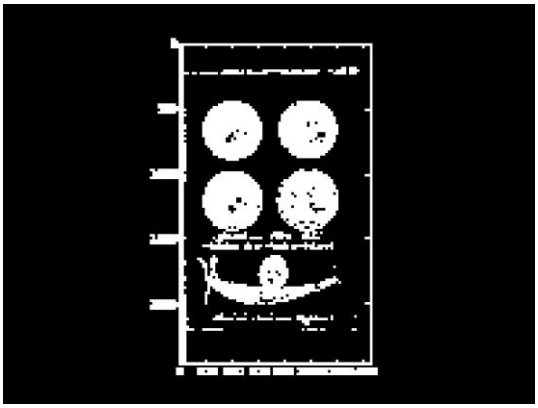


Figura 16. Determinarea fundalului

Pentru a exemplifica determinarea proprietăților lipsă în cazul unui algoritm de segmentare a textului:

a. în urma pasului 1 se va obține documentul expandat;
b. în urma pasului 2 se va expanda rezultatul algoritmului (zona de text);
c. în urma pasului 3 se va determina imaginea conținută în document;
d. în urma pasului 4 se va determina fundalul, marcat cu pixeli activi;
e. la final vom avea 4 imagini, câte una pentru fiecare din pașii de mai sus.

Desigur, în cazul unui algoritm de segmentare a imaginilor, pașii b şi c se vor inversa.

3.2.2 Calculul efectiv al funcţiei scor

Implementarea algoritmului oferă o interfață care necesită introducerea a cinci parametri:

- fișierul care descrie valorile reale ale proprietăților – necesar pentru stabilirea elementelor de *ground truth*;
- fișierul care conține documentul care trebuie analizat (documentul preprocesat cu același tip de algoritm de binarizare folosit în cadrul execuției celor doi algoritmi de segmentare);
- fișierele rezultat ale aplicării algoritmilor de segmentare (text și imagine) pe documentul de mai sus;
- calea în care stocăm rezultatele.

În cele ce urmează vom descrie și exemplifica sub-etapele constituente ale algoritmului, precum şi operatorii necesari pentru a determina precizia (valoarea funcţiei scor) fiecărui algoritm de segmentare în parte.

3.2.3 Valorile reale

În primul rând, pentru a putea determina coeficientul de eroare al unui algoritm, trebuie să știm care sunt **valorile corecte/reale**. Mai exact, avem nevoie de un set de date marcate, pe care vom rula ulterior algoritmii cărora dorim să le măsurăm performanța. În momentul de față, implementarea algoritmului descris mai sus permite introducerea măsurătorilor reale doar sub forma unui fișier text, care poartă același nume cu imaginea corespunzătoare, împărțit în două zone - imagine şi text. Fiecare suprafață este descrisă de un tuplu care descrie o zonă dreptunghiulară de forma:

(Xul, Yul, Xdr, Ydr)

unde (X, Y)ul reprezintă coordonatele colțului din stânga sus, iar
(X, Y)dr reprezintă coordonatele colțului din dreapta jos

Formula 15. Determinarea valorilor unei proprietăți reale

În urma parcurgerii acestui fișier, algoritmul va putea extrage toate informațiile necesare pentru a determina toate proprietățile reale ale documentului de analizat - P_i^r.

3.2.4 Analiza imaginilor

Toate cele 3 documente implicate în calculul erorii vor fi expuse algoritmului descris anterior, respectiv pașilor de dilatare, calcul delta, calcul fundal, calcul eroare. Implementarea algoritmului a fost realizată în python, pentru că oferă o interfață asemănătoare cu cea Matlab şi pentru facilităţilor oferite în ceea ce privește calculul cu matrice. Nu există restricții în ceea ce privește formatul imaginilor.

Un scenariu valid de folosire a acestui algoritm este:

- utilizatorul încearcă să determine precizia unui anume algoritm de segmentare x, comparând rezultatele acestuia cu cele ale unui alt algoritm de segmentare y (menționăm aici că algoritmii x şi y segmentează zone diferite ale documentului);
- utilizatorul oferă o serie de date necesare rulării algoritmului de comparare:
  - imaginea originală (binarizată);
  - rezultatul segmentării unei proprietăți cu ajutorul algoritmului x (de exemplu text);
  - rezultatul segmentării cu ajutorul algoritmului y (de exemplu imagine);
  - o modalitate de a stabili zona de *ground truth*, respectiv asociază imaginii originale o serie de etichete prin care marchează zonele de text și de imagine;
- algoritmul de comparare preia toate aceste date, calculează precizia celor doi algoritmi şi oferă utilizatorului rezultatele obţinute.

Așa cum am explicat anterior, rolul **operatorului de dilatare (dil)** este de a include în analiza proprietăților constituente atât pixelii activi, cât și cei de context, pentru a minimiza eroarea scorului final. Operațiile cele mai sensibile în această zonă sunt:

- determinarea dimensiunii ferestrei de căutare. Alegerea unei ferestre mai mari determină o îmbunătățire a timpilor de calcul, dar şi o creștere a coeficientului de eroare, prin includerea unei zone foarte mari de fundal. În aceeași măsură, alegerea unei ferestre foarte mici va crește eroarea prin eliminarea zonelor din interiorul imaginii şi adăugarea lor la fundal;
- determinarea pragului α de la care considerăm întreaga fereastră ca parte constituentă a zonei țintă. Desigur, pragul trebuie sa fie reprezentat procentual față de dimensiunile ferestrei curente.

Prin urmare, atât dimensiunea cât şi pragul ferestrei, trebuie alese cu grijă, ținându-se cont de dimensiunile documentului global în așa fel încât rezultatele să fie optime. În urma testelor efectuate pe copii digitale ale unor documente vechi, am observat ca:

- dimensiunea optimă a ferestrei este de 5% din dimensiunea cea mai mare a documentului;

$$s = 0.05*MAX(w, l)$$

Formula 16. Determinarea dimensiunii ferestrei

- dimensiunea optimă a pragului α este de 25% din suma pixelilor activi din fereastră. Prin urmare, o fereastră va fi considerată activă dacă:

$$\frac{1}{4s^2}\sum D[i{:}\, i+s, j{:}\, j+s] < 255,\ \ \forall i \in [0, w/s], \forall j \in [0, h/s]$$

Formula 17. Determinarea pragului de activare a ferestrei

În concluzie, pentru a determina valoarea operatorului *dil*, vom inițializa o matrice de pixeli cu dimensiunile documentului inițial, în care toate valorile vor fi 255 (deci cu toți pixelii inactivi). Ulterior, vom parcurge documentul de analizat, vom aplica formula de mai sus şi îi vom atribui valori de 0, în funcție de rezultatul obținut.

Operatorul de diferență (delta) este definit pe două matrice cu dimensiuni egale cu cele ale documentului inițial şi este necesar pentru a determina valorile proprietăților lipsă, respectiv a fundalului. În faza incipientă vom iniţializa o matrice cu valoarea diferenței dintre cele 2 matrice. Ulterior, vom inițializa rezultatul cu o matrice în care toți pixelii sunt albi, ca în cazul operatorului *dil*. Pe acest rezultat se va aplica o mască, în așa fel încât fiecărui pixel

din rezultat îi vom atribui valoarea 0 (îl vom activa), atunci când pixelul corespunzător din matricea diferență va avea valoarea -255 (ceea ce înseamnă că în prima matrice pixelul e activ, iar în cea de-a doua inactiv).

Operatorul de eroare ε este definit, ca și *delta*, pe două matrice de dimensiuni egale cu cele ale documentului inițial. Pentru calculul acestuia vom folosi o matrice sumă a proprietății în cauză și a proprietății reale, ambele expuse operatorului *dil*. Rezultatul final va fi inițializat cu matricea care reprezintă documentul alb, pe care vom aplica o mască, în așa fel încât fiecare pixel din matricea rezultat va fi activat (=0) în cazul în care pixelul corespunzător din matricea sumă va fi 255. Mai concret, vom activa în matricea rezultat toți pixelii pentru care în matricea proprietate sunt diferiți de matricea care descrie proprietatea reală.

În final, eroarea per proprietate va fi calculată ca numărul pixelilor activi din matricea rezultat, iar eroarea totală va fi suma erorilor împărțită la dublul suprafeței documentului, așa cum arata formula de mai sus (Formula 10. Eroarea de determinare a unei proprietăți).

3.2.5 Arhitectura algoritmului

Pentru o mai bună înțelegere a algoritmului vom exemplifica fiecare etapă printr-o secvență din schema logică:

- în faza incipientă utilizatorul introduce datele necesare algoritmului (fișierul *ground truth*, respectiv cele 3 imagini). Ulterior, imaginile vor fi expuse operatorului de dilatare;

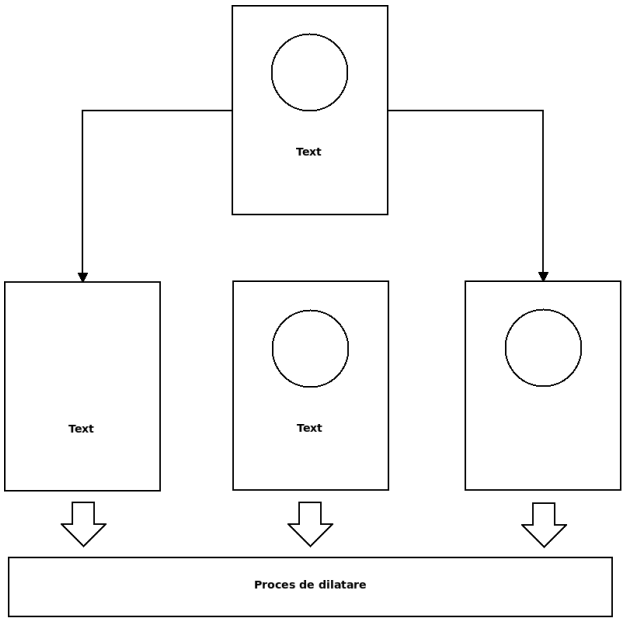


Figura 17. Algoritmul de comparație - introducerea datelor și etapa de dilatare

- după ce caracteristicile imaginilor au fost dilatate, se determină caracteristicile lipsă prin aplicarea operatorului delta (fundalul, imaginile în cazul segmentării de text, respectiv blocurile de text viceversa);

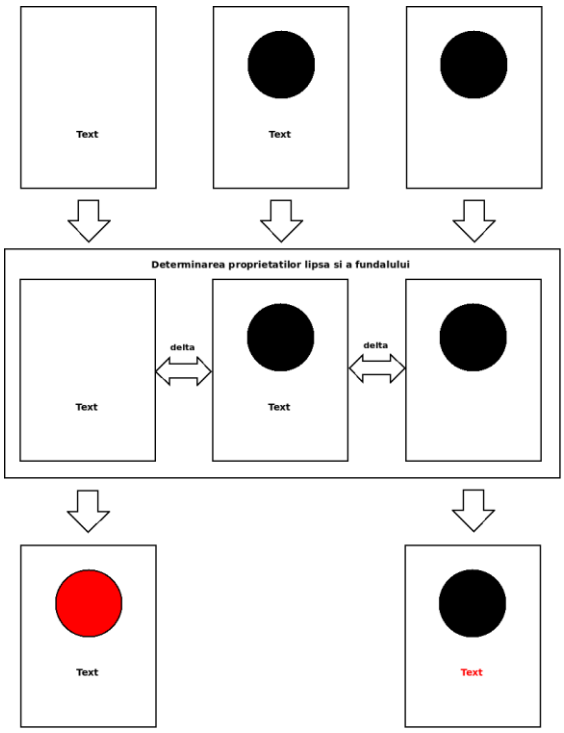


Figura 18. Algoritmul de comparație - determinarea proprietăților lipsă

- în cele din urmă se determină performanta fiecărui algoritm, prin aplicarea operatorului de eroare pe fiecare din proprietăți (inclusiv fundal).

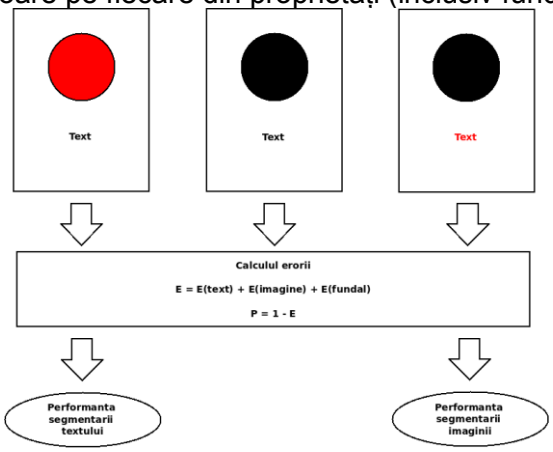


Figura 19. Algoritmul de comparație - determinarea performanței

3.3 Rezultate obţinute

Experimentele au fost realizate pe un set de 1380 de imagini, obţinute din două surse:

- copii ale unor documente vechi, aflate într-o condiție proastă, folosite ca *benchmark* în cadrul conferinței ICDAR 2007 (7). Menționăm că documentele au fost păstrate în format inițial, pentru a reține cât mai mult din informația inițială:
- copii de foarte bună calitate, conținând în mare parte manuale şi documentație pentru sistemul de operare Ubuntu 10.04. Acestea au fost obţinute prin conversia unor fișiere în format PDF. Pentru convertirea documentelor pdf într-un format tradițional, s-a folosit utilitarul *convert* din biblioteca ImageMagick.

Datorită structurii modulare am putut schimba cu ușurință dimensiunile ferestrelor de parcurgere a documentelor, respectiv algoritmii de binarizare.

În cadrul domeniului *pattern regonition*, problema de clasificare trebuie analizată determinand valorile *true positives*, *false negatives*, respectiv *false positives*.

Prin urmare, pentru fiecare din combinațiile menționate anterior s-a calculat scorul F1 (8), definit conform formulei de mai jos:

$$F_\beta = \frac{(1+\beta^2) * true\ positives}{((1+\beta^2) * true\ positives + \beta^2 * false\ negatives + false\ positives)}$$

Formula 18. Scorul F1

3.3.1 Binarizarea documentelor

În cadrul etapei de binarizare a fost folosită o variație a scorului F1, definită ca fiind o medie armonică a celor două caracteristici descrise mai sus. Măsura F (eng. *F measure*) este descrisă de formula de mai jos:

$$F\ measure = \frac{2 * Recall * Precision}{Recall + Precision}$$

Formula 19. F measure

Setul de date de referință (eng. *ground truth data*) a fost determinat după cum urmează:

- Cei 6 algoritmi de binarizare au fost rulați pe un set restrâns de imagini reprezentative pentru toate problemele specific zonei DAR;
- S-a observat că algoritmul NLBIN a avut cele mai bune rezultate;
- Un eșantion de 1000 de imagini mixte din setul ICDAR, respectiv din documentația Ubuntu au fost binarizate cu ajutorul algoritmului NLBIN;
- În cazurile în care rezultatele nu erau 100% corecte, acestea au fost modificate manual.

Considerând imaginile în forma de matrice bidimensională, cu elemente care pot avea valori în mulțimea discretă {0, 255}, corespunzând culorilor negru, respectiv alb, caracteristicile rezultatelor au fost determinate după cum urmează:

- Valoarea *true positive* a fost determinată de pixelii care și-au păstrat valorile din imaginea de referință după binarizare, în imaginea rezultat;
- Valoarea *false positive* a fost determinată de pixelii care erau albi în imaginea de referință și care au devenit activi în imaginea rezultat;
- Valoarea *false negative* a fost determinata de pixelii care erau activi în imaginea de referință și care au fost dezactivați în imaginea rezultat.

Rezultatele experimentelor sunt prezentate în graficul de mai jos:

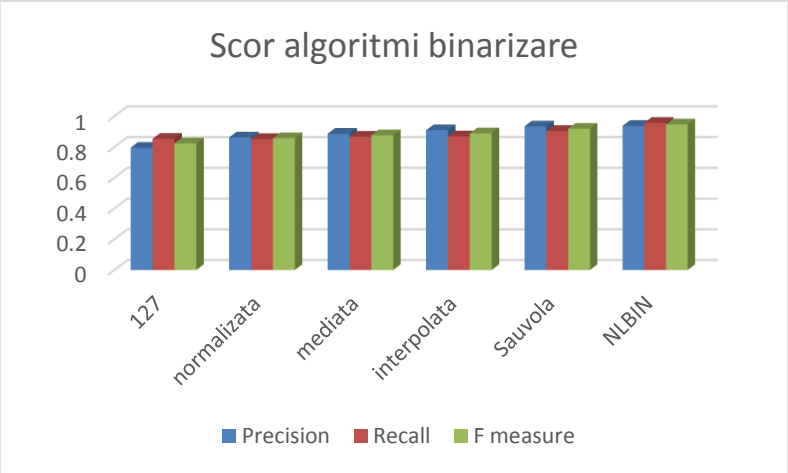


Figura 20. Scor algoritmi binarizare

Așa cum am arătat anterior, experimentele din această zonă au folosit 6 implementări diferite de algoritmi de binarizare:

- binarizarea **simplă**, cu prag fix de 127 este utila doar în cazul în care avem de-a face cu documente de calitate ridicată. În cazul documentelor vechi este foarte ineficientă;

prin urmare acest tip de binarizare a fost folosit doar în testele desfășurate în cursul dezvoltării modulului de procesare de documente, după care a fost abandonat;

- binarizarea **normalizată** a fost folosită pentru o scurtă perioadă de timp. Aceasta a produs îmbunătățiri în cazul documentelor cu contrast scăzut, dar a fost abandonată rapid, datorită tendinței de a subția prea mult liniile cu care sunt desenate caracterele și imaginile; din această cauză algoritmul de segmentare filtra părți mari din documentul inițial, implicit din imaginile incluse;
- binarizarea **mediată** a fost folosită pentru o perioadă mare de timp, datorită echilibrului intre timpii de calcul şi performanțele obţinute. Această abordare este foarte eficientă mai ales în cazul nostru, deoarece nu ne interesează în mod special calitatea ridicată a imaginilor rezultat; de exemplu, prezența unor elemente de zgomot în imaginile binarizate este un obstacol care poate fi eliminat atât de algoritmul de segmentare, cât şi de clasificatorul final, care poate ignora aceste detalii
- binarizarea **interpolată** a produs rezultate foarte asemănătoare cu cele ale binarizării mediate, dar la un cost computațional mai ridicat. Din acest motiv, această abordare a fost abandonată rapid. Menționăm că în cazul algoritmilor care împart documentul inițial în mai multe eșantioane (binarizare mediată și interpolată), alegerea unor dimensiuni mai mici ale ferestrelor de parcurgere a documentelor crește atât timpul de procesare, cât şi precizia rezultatelor obţinute. Experimentele descrise mai sus au folosit ferestre de 20x20 pixeli;
- binarizarea **Sauvola** a produs rezultate mai bune decât cea mediată, pe documente mai variate și mai expuse la zgomot. Din păcate, această abordare prezintă o serie de dezavantaje, legate de timpii de calcul mai ridicați, respectiv de modalitatea de stabilire a unghiului de rotație;
- binarizarea **NLBIN** a produs cele mai bune rezultate pe toate tipurile de documente, afectate de toate variantele de zgomot. Documentele preprocesate cu ajutorul acestui tip de binarizare au produs cele mai bune rezultate și în cadrul algoritmului de segmentare aplicat ulterior. Prin urmare, această abordare este opțiunea noastră finală ca etapă intermediară în cadrul procesului complet de procesare de documente, în ciuda timpilor de calcul ridicați. Menționăm aici că implementarea preluată din modulul ocropy oferă posibilitatea de a împărți sarcinile de binarizare pe mai multe fire de execuție; din păcate acest lucru se poate realiza numai atunci când se încearcă procesarea mai multor documente (binarizarea unui singur document are loc intr-un singur fir de execuție).

Principalele probleme întâlnite au fost cauzate în general de copiile de calitate scăzută ale documentelor vechi. Chiar şi folosirea unor algoritmi de binarizare complecși s-a dovedit a fi ineficientă, deoarece aceasta a condus la obținerea unor zone mari de culoare neagră în imaginea rezultat, ceea ce provoca erori la nivelul algoritmului de segmentare, care clasifică aceste zone ca şi imagini.

3.3.2 Segmentarea documentelor cu ajutorul algoritmului CVSEG

Algoritmul CVSEG a fost testat preponderent pe documente de calitate scăzută pentru a ne asigura că se comportă corect atunci când trebuie să proceseze imagini afectate de zgomot. Categoriile de documente au fost variate, asa cum vom arăta in cele ce urmează.

În cazul documentelor **închise la culoare cu contrast scăzut**, algoritmul de binarizare joacă un rol foarte important, deoarece este responsabil de stabilirea unui prag corect şi de a "clarifica" în acest fel imaginea. Așa cum am arătat mai sus, în unele cazuri algoritmii de binarizare eșuează în etapa de preprocesare a documentului, ceea ce duce la rezultate foarte proaste în cadrul etapei de segmentare.

Algoritmul de binarizare joacă un rol foarte important şi în cazul documentelor cu **contrast scăzut datorat estompării cernelii**, sau imprimării defectuoase. Atunci când pragul de binarizare este determinat incorect, există pericolul de a subția liniile prin care a fost desenată imaginea, sau de chiar de a eluda porțiuni din obiect.

În ceea ce privește numărul de imagini dintr-un document, acesta nu pare să aibă o influență foarte mare asupra algoritmului de segmentare. Documentele cu o singură imagine sunt segmentate în general corect, atât timp cât nivelul de zgomot din cadrul imaginii se află sub un prag acceptabil. În ceea ce privește documentele cu mai multe imagini, există posibilitatea ca algoritmul CVSEG să includă şi zone adiacente în imaginea rezultat, datorită tehnicii de *cluster*-izare

Documentele care conțin **imagini strâns încastrate în blocul de text** ridică o altă serie de probleme:

- etapa de proiecție pe axele XY va fi afectată datorită prezentei imaginii. În general, atât timp cât imaginea are o formă apropiată de un patrulater dreptunghic, textul va fi filtrat în continuare;
- datorită distanței mici intre zonele de text și imagine, algoritmul de clustering poate asocia imaginilor elemente invalide, în special atunci când este afectat şi algoritmul de proiecție.

În ceea ce privește **documentele cu chenar**, acestea sunt segmentate corect, în proporție majoritară. În faza incipientă, algoritmul CVSEG avea tendința de a adăuga tot conținutul chenarului în imaginea rezultat, dar odată cu adăugarea verificărilor de conectivitate şi rarefiere, atât la nivelul ferestrelor, cât și la nivelul cluster-elor, performanțele au crescut simțitor. În momentul de față, chenarele sunt eliminate odată cu elementele de zgomot, sau cu zonele de text, iar imaginea rezultat este corectă.

Unghiul de rotație nu provoacă mari probleme în cadrul etapei de segmentare. Exista totuși situații în care acest fenomen este asociat şi altor impedimente, caz în care algoritmul CVSEG poate fi afectat. De exemplu, atunci când imaginile incluse în document sunt incastrate în text, imaginea finală va include ferestre invalide. Mai menționam aici că setul de date de antrenare nu includea imagini afectate de acest tip de zgomot; prin urmare, am fost nevoiți să testăm algoritmul pe date generate artificial, prin folosirea unui editor de imagini (GIMP). Unele imagini au putut fi preluate din rezultatele aplicării algoritmului lui Sauvola, care, așa cum am arătat înainte, are tendința de a introduce un unghi de rotație în anumite situații.

În unele cazuri, folosirea **de caractere de dimensiuni şi stiluri diferite** a condus la erori la nivelul algoritmului de segmentare, care includea aceste zone în rezultatul final. Comportamentul este foarte asemănător cu cel întâlnit în cazul documentelor cu imagini strâns încadrate de blocuri de text.

Problemele întâlnite în cazul documentelor cu un **grad ridicat de transparență** nu pot fi rezolvate la nivelul algoritmului de segmentare. Din păcate, în această situație, algoritmii de binarizare nu reușesc întotdeauna să diferențieze prim planul (elementele active ale documentului) de fundal. Prin urmare, atunci când binarizarea eșuează pe documentele afectate de transparență, rezultatele segmentării sunt de obicei eronate, deoarece imaginea rezultat va include elemente de pe ambele pagini.

O categorie aparte în cadrul documentelor cu probleme este reprezentată de prezența mai multor pagini în scan. De obicei, această situație este întâlnită în cazul copierii cărților, în special cele cu un număr mare de pagini. În acest caz, ne lovim de o serie de probleme, care au loc simultan:

- datorită numărului mare de pagini, cartea nu poate fi fixată corespunzător în aparatul cu care se realizează copierea, ceea ce duce de obicei la apariția problemelor de rotație și translatare; implicit, pagina copiată nu va conține întotdeauna toate elementele reprezentative;
- apariția altor elemente în copia documentului duc la probleme de iluminare diferită, deoarece aceste elemente au alte culori, sau sunt poziționate la distanțe diferite (marginile celorlalte pagini, suprafața pe care este așezată cartea, cotorul cărții, elemente de fixare a cărții – inclusiv degete etc.);
- curbura paginii - această problemă este mai des întâlnită în apropierea cotorului cărții şi este cu atât mai pregnantă cu cât cartea este mai groasă;

- zgomot cauzat datorită fixării imprecise a cărții. Acesta se manifestă sub forma estompării elementelor prezente pe pagină, datorită mișcării în timpul copierii;
- prezența oricărui alt element de zgomot, care este mult mai nocivă, în condițiile descrise mai sus.

În afara elementelor perturbatoare descrise mai sus, mai există şi alte **variații de zgomot** în imaginile care reprezintă copii de documente. Trei dintre cele mai intalnite sunt:

- pigmentări, sau depigmentări ale suportului pe care a fost imprimat documentul, în mare parte datorate efectelor trecerii timpului, sau expunerii la diferiți factori (căldură, umezeală etc.). În cadrul acestei categorii, rezultatele algoritmului de segmentare depind foarte mult de performanța algoritmului de binarizare, care este responsabil de filtrarea acestor elemente de zgomot. Din păcate, atunci când documentele sunt foarte deteriorate, algoritmii de binarizare au tendința de a elimina complet detaliile care au fost supuse depigmentării, sau de a valida zonele care au fost pigmentate în exces;
- urme, sau adnotări efectuate de foști cititori. Acestea pot fi de diferite forme:
  - pete cauzate de vărsarea diferitor lichide pe document - această sub-categorie prezintă aceleași caracteristici ca și categoria precedentă;
  - pete de cerneală apărute în momentul imprimării documentului. În aceasta situație, în funcție de dimensiunea şi nivelul de transparența al petelor, algoritmul de binarizare poate elimina parte din zgomot; cu toate acestea, de obicei petele sunt incluse în imaginea binarizată. În cazul în care petele sunt de dimensiuni reduse, algoritmul de segmentare le poate filtra şi produce un rezultat corect;
  - note ale cititorilor anteriori efectuate în spațiile libere ale paginii, inclusiv semnături, sau mâzgălituri. Algoritmul de segmentare este exclusiv responsabil de eliminarea acestor elemente din imaginea rezultat;
- mizerie prezentă pe document, sau pe obiectivul aparatului cu care se efectuează copierea. Ca şi în cazul petelor de cerneală, dimensiunea elementelor perturbatoare este foarte importantă în determinarea unui rezultat corect, atât în etapa de binarizare, cât și în cea de segmentare.

Documentele de calitate ridicată nu au ridicat probleme în cadrul etapei de segmentare, în principal datorită absenței elementelor de zgomot prezente în imaginile din primul set de documente. Importanța algoritmului de binarizare ales a scăzut, rezultatele fiind în mare parte corecte. Singurele cazuri în care am întâmpinat dificultăți au fost cele în care documentele includeau grafice şi diagrame desenate cu linii subțiri; în aceste situații CVSEG are tendința de a filtra unele ferestre valide din imaginea rezultat.

3.3.3 Rezultate comparative

În zona procesării de documente, segmentarea imaginilor este ignorată de cea mai mare parte a cercetătorilor. Singurul algoritm relevant în acest domeniu îi aparține lui Bloomberg (9). Prin urmare, pentru a încadra cât mai corect rezultatele CVSEG, au fost efectuate doua serii de teste comparative:

- față de rezultatele algoritmului lui Bloomberg;
- față de rezultatele unor algoritmi de segmentare de text, folosindu-ne de algoritmul descris în capitolul Analiza comparativă a performanței algoritmului de segmentare.

Pentru a obține rezultate cât mai concludente, în desfăşurarea **testelor comparative faţă de algoritmul lui Bloomberg** a fost folosită implementarea autorului din biblioteca *libleptonica* (10), disponibilă în sistemul de operare Ubuntu 12.04. Deoarece algoritmul nu funcţionează pe orice set de imagini, acestea au fost supuse unei convertiri; imaginile au fost binarizate şi convertite în modul indexat - pentru a realiza acest lucru, s-a folosit un script GIMP CLI.

Scenariul de testare s-a desfăşurat pe două planuri paralele:

- imaginile din setul inițial au fost etichetate pentru a stabili elementul de *ground truth*;
- ulterior, imaginile au fost expuse în faza incipientă aceloraşi algoritmi de binarizare, ca etapă premergătoare a algoritmului de segmentare (algoritmii NLBIN și Sauvola au fost preluați din biblioteca *ocropus* (11); celelalte două forme de binarizare – mediată şi cu prag fix – sunt oferite de implementări proprii; imaginile au fost păstrate la dimensiunile originale);
- rezultatele obţinute au fost transformate în modul indexat pentru a putea fi folosite de algoritmul lui Bloomberg;
- cele două seturi de imagini astfel obţinute au fost procesate cu ajutorul celui doi algoritmi de segmentare (menţionăm aici că algoritmul CVSEG a folosit o dimensiune fixă a ferestrei, de 20 de pixeli);
- procesarea documentelor a fost realizată automat, cu ajutorul unor scripturi care au înregistrat timpii de execuție;
- în final au fost colectate și centralizate rezultatele.

Timpii de execuție sunt descriși în tabelul de mai jos:

Tabel 1. Timpi de execuție pentru etapele premergătoare segmentării

Tip algoritm	Timp mediu de execuție
Binarizare simplă (127)	0.23s
Binarizare mediată (average)	3.14s
Binarizare Sauvola	**8.86s**
Binarizare NLBIN	**10.74s**
Conversie GIMP	3.95s

După cum se poate observa, în cadrul procesului de preprocesare, cele mai scumpe activități din punct de vedere computaţional sunt algoritmii de binarizare NLBIN și al lui Sauvola. A se nota de asemenea durata conversiei la modul indexat, cu ajutorul GIMP.

Rezultatele algoritmului Bloomberg sunt descrise în tabelul de mai jos. Timpul mediu de execuție a algoritmului, excluzând timpii de conversie, respectiv de binarizare, pe toate eşantioanele a fost de 0.944 secunde.

Tabel 2. Rezultate segmentare Bloomberg

Tip algoritm	Rezultate obţinute	Timp de execuție
Bloomberg + GIMP + 127	72.2%	5.205s
Bloomberg + GIMP + average bin	73.2%	8.012s
Bloomberg + GIMP + Sauvola	74.1%	13.205s
Bloomberg + GIMP + NLBIN	**74.5%**	15.429s

Algoritmul CVSEG a fost aplicat pe aceleaşi imagini, în aceleaşi condiţii; dimensiunea ferestrei a fost păstrată fixă, de 20x20 de pixeli. Rezultatele obţinute sunt descrise în tabelul de mai jos (timpul mediu de execuție a algoritmului a fost de 4.431 secunde):

Tabel 3. Rezultate comparative

Tip algoritm	Rezultate obţinute	Timp de execuție
CVSEG + 127 bin	80.2%	0.420s
CVSEG + average bin	81.2%	0.614s
CVSEG + Sauvola	84%	14.500s
CVSEG + NLBIN	**84.1%**	28.762s

Rezultatele obţinute în urma aplicării celor doua variante de algoritmi au demonstrat următoarele:

- performanţele cresc odată cu calitatea algoritmului de binarizare folosit;
- timpii de execuție cresc odată cu calitatea algoritmului de binarizare folosit;
- există un salt foarte mare în ceea ce privește timpii de execuție, atunci când se trece la algoritmii de binarizare superiori (Sauvola şi NLBIN);

- algoritmul CVSEG a obținut performanţe superioare algoritmului lui Bloomberg, cu până la 9.6%;
- timpii de execuție sunt aproximativ aceiaşi, cu menţiunea că în cazul algoritmului lui Bloomberg a fost necesară conversia la modul indexat, care adaugă încă 4 secunde la timpul final de execuție;
- în ceea ce privește execuţia strictă a algoritmilor pe o anumită imagine, CVSEG este mai încet decât implementarea algoritmului lui Bloomberg.

În ceea ce privește timpii de calcul raportaţi la performanţe, evoluţia acestora este oarecum firească - un algoritm de binarizare mai bun produce rezultate mai bune, dar intr-o perioadă mai mare de timp. Cu toate acestea, dacă privim diferenţele dintre timpii de execuție și performanţele obţinute atunci când au fost folosiţi algoritmul de binarizare simplă, respectiv NLBIN, observăm că unei creşteri de peste 10 secunde îi corespund îmbunătăţiri ale performanţei de 2.3% în cazul algoritmului lui Bloomberg, respectiv 3.9% în cazul CVSEG. Deoarece scopul acestui modul e de a produce imagini care urmează să fie clasificate ulterior de un motor CBIR, există posibilitatea să renunţăm la algoritmii de binarizare superiori, în cazul în care rezultatele acestora nu produc îmbunătăţiri simţitoare de precizie a clasificării.

În ceea ce privește timpii de calcul ai algoritmului CVSEG, se poate observa că algoritmul lui Bloomberg segmentează imaginile mult mai repede. Acest lucru este datorat faptului că biblioteca leptonica foloseşte structuri optimizate de calcul, spre deosebire de implementarea proprie, bazata pe python. Chiar dacă modulele scipy şi numpy oferă facilităţi asemănătoare, *overhead*-ul introdus este mult mai mare.

Algoritmul CVSEG a obținut rezultate cu până la 9.6% mai bune decât algoritmul lui Bloomberg. Erorile întâlnite în analiza acestuia gravitează în jurul celor de mai jos:

- nicio imagine în rezultatul segmentării. Acest lucru este cauzat în special de grosimea liniilor folosite în cadrul desenului/imaginii. Din această cauză, în cadrul procesului de eliminare a textului sunt eliminate şi părţi din imagine;
- documentul complet este inclus în rezultat, sub forma negativă. Acest lucru este cauzat de zgomotul prezent în document, sub formă de transparenţă a paginii, sub formă de imperfecţiuni din cadrul procesului de scanare sau pur şi simplu calitate scăzută a documentului în momentul înregistrării imaginii.
- rezultatul segmentării este incomplet. Acest lucru este cauzat de contrastul scăzut al imaginii în cadrul documentului, sau de încastrarea strânsă a acesteia într-un bloc de text.

Având în vedere opţiunile limitate din zona segmentării de imagini, pentru a încadra mai precis rezultatele algoritmului CVSEG, s-a impus efectuarea unor **teste comparative faţă de alţi algoritmi de segmentare**, care vizează extragerea zonelor de text din documentul inițial.

Prin urmare, au fost necesare două variante de teste:

- validarea algoritmului de calcul al performanţei al unui anumit algoritm de segmentare;
- teste comparative, pentru a stabili diferența de performanţă între CVSEG şi un alt algoritm de segmentare.

Testele au fost efectuate pe o serie de documente mixte (degradate şi de bună calitate), care au fost în prealabil analizate pentru a putea specifica valorile reale ale proprietăților în fișierele text corespunzătoare, așa cum am menţionat anterior, în capitolul Valorile reale.

Pentru **validarea algoritmului comparativ** au fost parcurse următoarele etape:

- a fost stabilită o dimensiune fixă a pasului de dilatare (5 pixeli de context);
- au fost stabilite valorile de ground truth pentru fiecare din documentele iniţiale, pe zonele de imagine, respectiv text;

- au fost extrase manual imagini reprezentative ale zonelor țintă din cele două categorii;
- a fost executat algoritmul de calcul al performanţei.

Rezultatele obţinute sunt descrise în tabelul de mai jos:

Tabel 4. Rezultate algoritm comparaţie

Caracteristică	Rezultate obţinute
Imagine	95.6%
Text	97.7%

O alta serie de teste a fost efectuată pe documente marcate eronat pentru a verifica scăderea de performanţă. Pentru acestea, fișierele care descriu marcajele au fost modificate după cum urmează:

- Fie o caracteristică C, determinată de coordonatele (x_1y_1), (x_2y_2), (x_3y_3), (x_4y_4)
- Coordonatele caracteristicii au fost translatate într-o direcţie aleatoare, obținându-se setul $(x'_1y'_1)$, $(x'_2y'_2)$, $(x'_3y'_3)$, $(x'_4y'_4)$, ca în figura de mai jos:

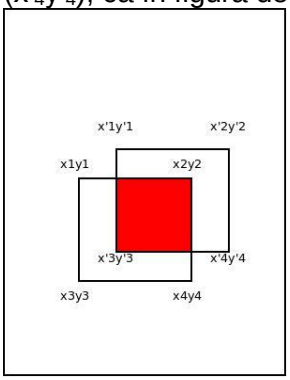


Figura 21. Generarea de erori pentru validarea algoritmului de comparaţie

- Suprafaţa caracteristicii iniţiale este $Si = (x_4-x_3)*(y_2-y_4)$
- Suprafaţa caracteristicii modificate este $Sm = Si = (x'_4-x'_3)*(y'_2-y'_4)$;
- Suprafaţa comună celor 2 zone (intersecţia) este $Sc = |x_4-x'_3|*|y_2-y'_4|$
- În cadrul procesului de generare de marcaje eronate s-a urmărit raportul err, pentru a putea determina eroarea introdusă.

$$err=(Si-Sc)/Si$$

Formula 20. Raport eroare algoritm comparaţie

Experimentele au fost realizate în două situaţii, cu eroare de 10%, respectiv de 20%. Nu au fost realizate experimente pentru err>20%, deoarece translatarea unei caracteristici pe distanţe prea mari duce la intersectarea tuturor caracteristicilor (imagine, text, respectiv fundal); în acest caz rezultatele nu sunt la fel de relevante. Rezultatele obținute sunt descrise în tabelul de mai jos:

Tabel 5. Erori algoritm comparaţie

Caracteristică	err =10%		err = 20%	
	Rezultat	Eroare algoritm	Rezultat	Eroare algoritm
Imagine	86%	**11%**	76%	**11%**
Text	88%	**10%**	78%	**21%**

După cum se poate observa, erorile algoritmului de comparaţie sunt egale sau foarte apropiate de valorile erorilor introduse, ceea ce validează implementarea şi în cazul testelor negative.

Următorul test efectuat a ţintit stabilirea diferenţei de performanţă dintre CVSEG şi un algoritm de segmentare de text. Prin urmare, am folosit opţiunea recomandată de autorii bibliotecii ocropy, respectiv un algoritm de segmentare de text bazat pe calcul de gradienţi (ocropus-gpageseg). Scenariul experimental a inclus următoarele etape:

- a fost selectat un eşantion de 500 de imagini din setul inițial de date;
- aceste imagini au fost binarizate cu ajutorul algoritmilor descriși anterior;
- pe imaginile binarizate au fost aplicaţi cei doi algoritmi de segmentare;
- rezultatele obţinute au fost folosite ca input pentru algoritmul comparativ.

În urma execuţiei algoritmului de segmentare de text au fost observate următoarele manifestări:

- algoritmul nu reuşeşte întotdeauna să elimine toate elementele de zgomot;
- imaginile care nu sunt compacte sunt filtrate parţial;
- în unele situaţii, algoritmul filtrează zone de text valide.

Influenţa algoritmilor de binarizare este evidentă în ceea ce privește performanţa algoritmului de segmentare de text. Cu toate acestea, chiar şi în cazul folosirii unor documente binarizate corect, segmentarea zonelor de text prezintă probleme în cadrul etapei de filtrare a zgomotului. Ca element constant, algoritmul elimină doar segmente din elementele de chenar introdus în cursul etapei de binarizare.

O altă categorie de documente care ar putea ridica probleme este reprezentată de imaginile care prezintă un anumit unghi de rotaţie; în aceasta zonă, algoritmul de segmentare de text a avut rezultate bune.

În ceea ce privește documentele care includ imagini, acestea au fost segmentate preponderent incorect. Majoritatea imaginilor au fost filtrate într-o oarecare măsură, dar rezultatul a inclus, de cele mai multe ori, o serie de elemente invalide.

O altă problemă întâlnită frecvent în cadrul segmentării zonelor de text este reprezentată de eludarea unor anumite blocuri. Acest fenomen are loc în special în cazul documentelor cu mai multe tipuri de caractere, de dimensiuni diferite, îngroşate sau nu etc.

În cazul documentelor care prezintă adnotări, algoritmul de segmentare de text a avut rezultate bune, majoritatea acestora fiind prezente în imaginea rezultat, în condițiile în care algoritmul de binarizare nu le-a filtrat anterior. În unele cazuri însă, atunci când adnotările prezintă un grad de lizibilitate scăzut, algoritmul eșuează în a le identifica.

În cele ce urmează vom prezenta şi interpreta rezultatele experimentale finale, obţinute în urma aplicării celor doi algoritmi pe imagini binarizate:

Tabel 6. Rezultate comparative faţă de un algoritm de segmentare de text

Tip algoritm	Valoare minimă	Valoare maximă	Rezultate finale
Segmentare text – 127 bin	61%	75%	67%
Segmentare text – ave bin	65%	82%	73%
Segmentare text – Sauvola	69%	89%	80%
Segmentare text – NLBIN	74%	**94%**	**82.57%**
CVSEG – 127 bin	73%	79%	75%
CVSEG – ave bin	75%	83%	79%
CVSEG – Sauvola	78%	89%	83%
CVSEG – NLBIN	81%	**91%**	**85.14%**

Așa cum se poate observa, algoritmul CVSEG a obținut rezultate mai bune decât algoritmul de segmentare de text indiferent de modalitatea de binarizare folosită. Sunt interesante şi valorile de minim și de maxim ale celor doi algoritmi; în urma analizei rezultatelor obţinute, am observat că:

- rezultatele cele mai bune au fost obţinute atunci când a fost folosit algoritmul de binarizare NLBIN;
- atunci când au fost folosiţi algoritmi de binarizare superiori, segmentarea zonelor de text a avut valori maxime apropiate de cele ale algoritmului CVSEG. În aceeași măsură există şi multe valori de minim, ceea ce a condus la o medie scăzută;
- folosirea unor algoritmi de binarizare mai puțin calitativi a dus la scăderea drastică a performanţelor algoritmului de segmentare de text, inclusiv a valorilor de maxim. De asemenea, se poate observa că diferența de performanţă este maximă în cazul

binarizării simple (8%), după care scade la 6% în cazul binarizării mediate; ulterior aceasta se stabilizează la aproximativ 3% în cazul algoritmilor de binarizare cu rezultate superioare;

- diferenţele de performanţă în cadrul aceluiaşi algoritm sunt de 15% în cazul segmentării de text, respectiv de 10% în cazul segmentării de imagini. Corelând aceste valori cu afirmaţiile de la punctul anterior, putem observa că segmentarea de text este mult mai sensibilă la forme imprecise de binarizare;
- indiferent de tipul de binarizare folosit, algoritmul CVSEG a avut valori mai puțin oscilante decât cele ale algoritmului de segmentare de text, cu minime de peste 80% în cazul NLBIN, ceea ce denotă consistenţa abordării.

Graficul de mai jos descrie diferenţele de performanţă dintre cei doi algoritmi, relativ la cele 4 metode de binarizare folosite (a se observa valoarea de sub axa în cazul NLBIN, acolo unde maximele segmentării de text au depășit maximele segmentării CVSEG):

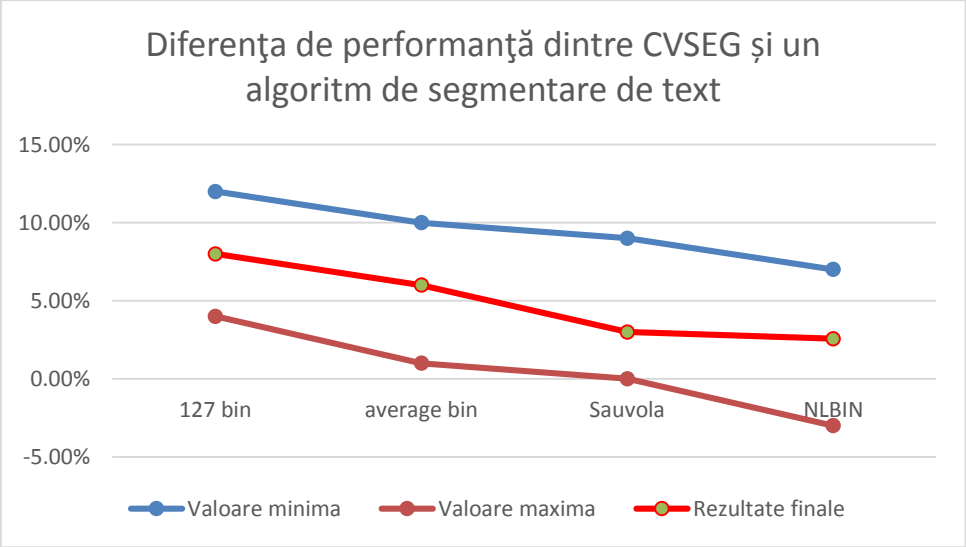


Figura 22. Diferența de performanţă dintre CVSEG şi un algoritm de segmentare de text

3.3.4 Concluzii

În ceea ce privește algoritmul CVSEG, problemele întâlnite au fost cauzate în mare parte de:

- etapa de eliminare a textului nu funcţionează în toate cazurile; cu toate acestea, etapele ulterioare de calcul de variaţie și conectivitate corectează majoritatea erorilor;
- algoritmii de binarizare afectează rezultatele segmentării, în special în cazurile în care sunt introduse zone mari cu zgomot, sau în cazul în care imaginea este rotită;
- algoritmul de cluster-izare poate asocia eronat zone invalide unui cluster determinat de zgomote (erori de înregistrare, iluminare incorectă etc.);
- algoritmii de binarizare joacă un rol foarte important în cazul documentelor cu contrast scăzut;
- numărul imaginilor prezente în document nu afectează foarte mult rezultatele segmentării;
- elementele de chenar sunt în cea mai mare parte eliminate. Singurele probleme întâlnite în această zonă au fost cauzate de chenarul introdus de etapa de binarizare (de obicei gros şi cu un caracter pregnant în imagine);
- modul în care imaginile sunt incluse în document afectează rezultatele algoritmului de segmentare, deoarece afectează atât etapa de filtrare prin proiecţie pe axe, cât și etapa ulterioara de *cluster*-izare;
- gradul de rotaţie e un alt factor care poate influenţa rezultatul segmentării. În acest caz, este foarte important unghiul la care este rotit documentul - unghiurile mici sunt

tratate corect de CVSEG. Un alt motiv de îngrijorare în această zonă este absenţa unui set de imagini de *benchmark*; testele au fost efectuate preponderent pe imagini binarizate eronat de algoritmul lui Sauvola sau pe imagini rotite artificial;

- variaţia caracterelor în cadrul documentului poate influenţa rezultatul segmentării, în sensul în care caracterele mai mari, sau desenate cu linii mai groase pot fi acceptate de CVSEG ca și valide;
- gradul de transparenţă al paginii este un alt tip de zgomot în care algoritmul de binarizare joacă un rol foarte important. În cadrul acestei categorii algoritmul NLBIN a avut rezultate foarte bune;
- documentele provenite din copierea cărţilor cu un număr mare de pagini prezintă un complex de probleme, cum ar fi curbura paginii, iluminare diferită, eventuale obiecte suplimentare menite fixării paginii. Chiar şi în aceste condiţii, algoritmul CVSEG a produs rezultate performante, așa cum am arătat mai sus;
- setul de documente folosit pentru validarea algoritmului CVSEG a inclus imagini afectate de toate tipurile de zgomot cunoscute în domeniul DAR – elemente perturbatoare, cum ar fi mizeria prezentă pe obiectivul aparatului de înregistrat sau pe pagină, adnotările, petele de diferite provenienţe (cerneală, întreţinere necorespunzătoare), pigmentări sau depigmentări ale hârtiei etc. Indiferent de condiţia în care se afla documentul, algoritmul a oferit rezultate bune, de obicei incluzând în totalitate imaginea cuprinsă de document. În unele cazuri, algoritmul include şi alte ferestre adiacente, dar, după cum am menţionat anterior, elementele de inteligenţă artificială și de vot majoritar ar trebui să poată filtra cu ușurință aceste inconveniente.

Algoritmul de stabilire a performanţei segmentării unui document reprezintă o variantă rapidă şi precisă de a determina calitatea unei abordări în această zonă. În urma experimentelor întreprinse am observat următoarele:

- rezultatele nu au atins niciodată valoarea maximă de 100%. Acest lucru se datorează modului imperfect de a marca zonele care fac parte din imagine, respectiv text;
- modalitatea de stabilire a elementelor de *ground truth* este limitată la forme geometrice simple, dreptunghiulare, ceea ce nu reflectă întotdeauna realitatea. De exemplu, în cazul formelor cu elemente curbilinii, utilizatorul poate doar să aproximeze suprafaţa acoperită de acestea prin introducerea succesivă de elemente dreptunghiulare de dimensiuni mici. Acest proces este dificil și supus erorii, chiar şi în cele mai bune condiţii;
- algoritmul ia în considerare zonele adiacente pixelilor activi, ceea ce poate atenua posibilele perturbaţii, dar eroarea introdusă de factorul uman nu poate fi niciodată nulă. Din acest motiv, având în vedere că documentele folosite în cadrul experimentelor provin din mediul real și că elementele incluse în acestea au forme variate, considerăm că rezultatele obţinute sunt o aproximare suficient de bună a performanţei algoritmilor;
- rezultatele obținute în cadrul experimentelor efectuate pe algoritmii de segmentare de imagini sunt inferioare segmentării de text. Interpretarea acestor rezultate este legată de forma blocurilor de text, care este mai apropiată de un patrulater dreptunghic, spre deosebire de imagini, care au forme mult mai variate;
- având în vedere faptul că modalitatea de determinare a stării de *ground truth* nu poate fi ameliorată, una din posibilele abordări prin care putem îmbunătăţi performanţa algoritmului e aplicarea unui algoritm de segmentare în zonele stabilite pentru a determina suprafaţa exactă pe care încearcă să o indice utilizatorul.

4 Contribuţii privind CBIR

4.1 Abordarea propusă a sistemului

Lucrarea de faţă încearcă implementarea unui sistem CBIR mixt, antrenat pe imagini din afara sferei procesării de documente, dar care primeşte ca *input* atât *scan*-uri, cât şi orice altfel de imagini. În cele ce urmează vom descrie arhitectura sistemului, rolul fiecărui modul în parte, cât şi rezultatele obţinute în urma experimentelor.

Scopul final a fost dezvoltarea unei aplicaţii scalabile, portabile între diferite tipuri de arhitecturi, respectiv sisteme de operare. Prin urmare, s-a evitat folosirea Matlab, sau Octave, din considerente practice.

O altă zonă de interes a fost expunerea funcţionalităţilor sistemului spre a fi accesate de mai mulți utilizatori simultan. Aşadar, am optat pentru o aplicaţie de tip server/client; deocamdată nu s-a acordat atenţie problemelor de management a utilizatorilor sau a altor probleme asemănătoare – acestea sunt considerate în afara scopului acestei lucrări, cel puţin pentru acum. Cu toate acestea, partea de server include servicii de genul *webservices*.

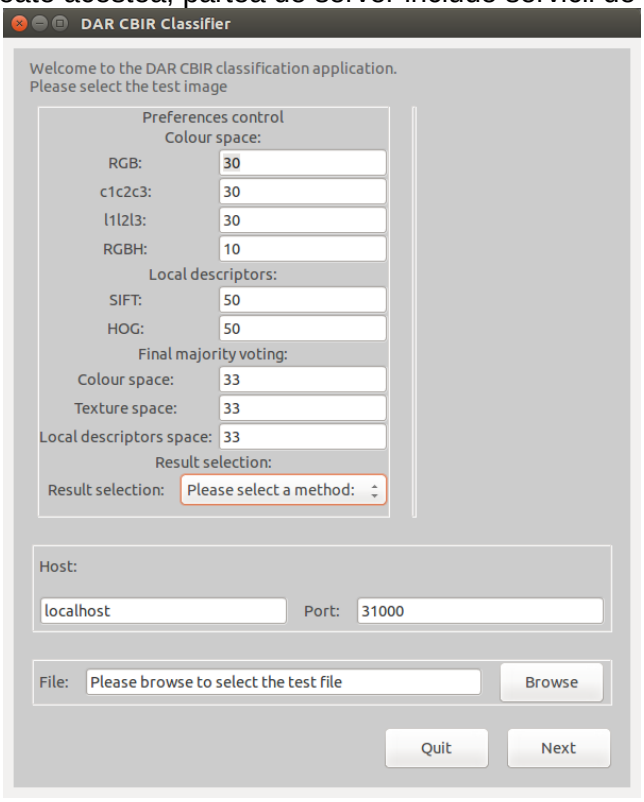


Figura 23. Interfaţa UI

Limbajul de programare folosit este python, în special din cauza facilităţilor oferite în zona interfeţei grafice, comunicării pe bază de *sockets* şi a procesării de date (bibliotecile de calcul cu matrice, de exemplu). Sistemul de operare este linux, iar arhitectura procesorului este de 32 de biți; cu toate acestea, nu există impedimente în a transforma aplicaţia într-una *multi-platform*. Stocarea datelor este realizată cu ajutorul unei baze de date MySQL, bazată pe o arhitectură MyISAM.

Arhitectura este modulară, pentru a facilita ulterioarele modificări aduse aplicației; fiecărui submodul îi corespunde o clasă. Pentru o mai buna gestionare *hardware*, atât implementarea *socket server*-ului, cât şi a sub-modulelor care necesită intensiv resurse *CPU*

şi memorie, folosesc tehnici *multi-threading*, respectiv *multi-processing*, în funcție de cantitatea de memorie şi de numărul de nuclee şi disponibile.

În forma ei finală, aplicația va oferi utilizatorului posibilitatea de a alege între mai multe tipuri de algoritmi pentru o configurare granulară şi cât mai potrivită zonei de căutare.

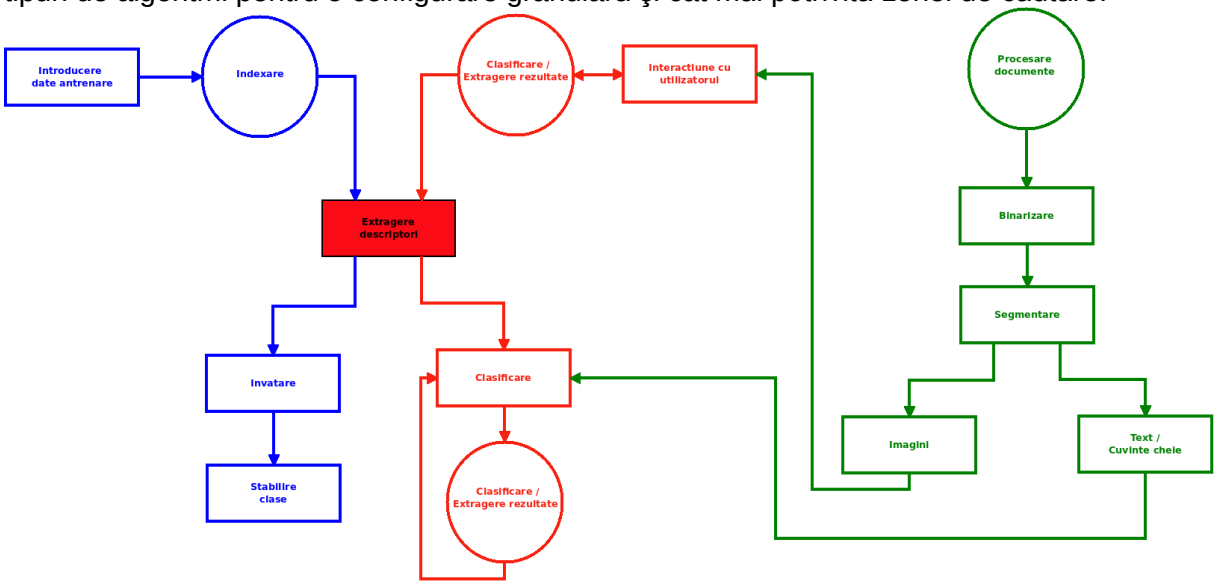


Figura 24. Arhitectura sistemului

Sistemul este compus din trei subansamble interconectate:

- partea de **antrenare şi învăţare automată** (de culoare albastră);
- partea de **clasificare şi de extragere a rezultatelor**, eventual asistată de utilizator (de culoare roșie);
- partea de **analiză şi clasificare a documentelor** (de culoare verde).

Un scenariu valid de operare pentru această aplicaţie trece prin următorii pași:

- sistemul este antrenat pe un set variat de imagini;
- fiecare imagine este analizată şi descompusă în descriptori relevanţi;
- descriptorii vor fi transmiși unui modul de învăţare automată, pentru a stabili clasele de bază;
- fiecare imagine nouă va fi descompusă la rândul ei şi clasificată conform categoriilor stabilite anterior;
- sistemul va extrage apoi cele mai relevante 10 rezultate şi i le va oferi utilizatorului ca răspuns la interogare;
- modulul de procesare a documentelor supune în prima fază imaginile unor etape de preprocesare premergătoare segmentării;
- imaginile astfel obţinute vor fi analizate şi clasificate conform algoritmului de mai sus.

Pentru a îmbunătăţi performantele şi robustețea sistemului, se dorește adăugarea următoarelor module:

- utilizatorului îi va fi oferită posibilitatea de a atribui un scor fiecărui rezultat; modulul de învăţare va pondera pe viitor rezultatele în funcție de scorul total obținut în urma interogărilor anterioare;
- în zona clasificării de documente se dorește extragerea a două seturi de informații – imagini şi text; aceste informații vor fi transmise modulului de clasificare, care va asocia rezultatele obţinute în urma procesării imaginilor cu informația textuală, pentru a oferi rezultate mai precise.

Ținta propusă este restricționată la zece clase de bază. Antrenarea se desfăşoară cu ajutorul a două tipuri de imagini:

- imagini folosite ca *benchmark* în competițiile internaționale pentru algoritmi de procesare a imaginilor (12). Deoarece imaginile au fost împachetate într-un format proprietar, a fost necesară convertirea lor într-un format comun, ușor de interpretat de către orice sistem de operare;
- imagini din setul ICDAR, folosit în cadrul implementării modulelor descrise în capitolul Contribuții privind tehnicile de procesare a documentelor.

Clasificatorul este compus dintr-o suită de rețele neuronale şi module de vot majoritar. În faza incipientă sistemul a fost antrenat numai cu primul tip de imagini, pentru ca apoi sa fie introduse şi copiile de documente, pentru a observa fluctuațiile de performanţă.

Caracteristicile de culoare sunt extrase folosind descriptori globali, bazați pe calculul histogramelor. Principalele probleme întâlnite în cadrul acestui spațiu de caracteristici au fost alegerea modelului de culoare, a numărului de zone de culoare (eng. *bins*), respectiv determinarea gradului de încredere asociat acestui tip de descriptori.

Ca urmare a studiului întreprins propunem aplicarea a trei tehnici de prelucrare a imaginilor (desigur, aceste abordări țin cont de proprietățile imaginii analizate - color, *grayscale*, alb-negru):

- În faza incipientă vom schimba spațiul RGB cu spatiile c1c2c3/l1l2l3 pentru a elimina zonele de umbra şi *highlight*, care ar putea vicia rezultatele clasificării, prin adăugarea de caracteristici irelevante şi prin mascarea celor eventual importante;
- Un alt set de caracteristici va fi oferit de întreaga imagine, în coordonate RGB, eventual scalată;
- De asemenea, în cadrul extragerii de descriptori globali din acest spațiu, vom folosi histograme tradiționale RGB. Menționăm că în etapa de clasificare acești descriptori vor avea o pondere redusă, datorită faptului că sistemul CBIR propus urmărește preponderent clasificarea pe bază de caracteristici locale, invariante la schimbările de culoare.

În **spațiul texturilor** am folosit LBP (eng. *local binary patterns*) pentru a extrage un alt set de descriptori globali. Folosim LBP în principal datorită complexității reduse de calcul şi invarianței la fenomene cum ar fi rotație, sau iluminare diferită. De asemenea, datele obţinute sunt unidimensionale, ceea ce facilitează etapa de antrenare în cadrul procesului de învăţare supervizată. Principalele probleme întâlnite în această zonă au fost legate de:

- alegerea tipului de celulă folosită în cadrul implementării algoritmului;
- alegerea dimensiunii celulei;
- alegerea numărului de puncte pe baza cărora vom stabili ulterior textura.

În condiții de laborator, unde intensitatea luminoasă, cantitatea de culoare, gradul de reflexie a suprafețelor şi transparenţa mediului sunt măsurate cu atenție, majoritatea metodelor de extragere a **caracteristicilor de formă** dau rezultate foarte bune. Din păcate, aceleași metode sunt extrem de sensibile în ceea ce privește problemele de obturare şi *cluttering*; cele mai performante sisteme de acest gen ating procente de reușită de 60% atunci când sunt aplicate pe imagini obţinute în afara laboratorului. Prin urmare, implementarea curentă nu se folosește de descriptori din acest spațiu. Pe viitor intenționăm să testăm fluctuațiile de performanţă ale sistemului prin introducerea unor tehnici de *moment invariants*, momente Zernike sau descriptori Fourier. Indiferent de alegerile făcute, acești descriptori vor fi sub-ponderați în cadrul modulului de clasificare.

Considerăm **descriptorii locali** ca fiind cei mai importanți în determinarea unui proces de clasificare corect. Sistemele CBIR au obținut rezultate foarte bune bazându-se numai pe descriptori locali; așteptările noastre sunt ca prin combinarea vectorilor de caracteristici din această zonă cu rezultate din celelalte spaţii implicate în procesarea de imagini să obținem rezultate superioare.

Descriptorii locali sunt calculați cu ajutorul algoritmilor de mai jos:

- HOG (eng. *Histogram of Oriented Gradients*) (13);
- SIFT (eng. *Scale Invariant Feature Transform*).

Învăţarea automată este realizată cu ajutorul unor rețele neuronale multi-strat de tip *feed-forward*, cu *back-propagation* (funcţie de transfer sigmoidă). Imaginile din setul de antrenare sunt împărţite în trei grupuri:

- 60% pentru antrenare;
- 20% pentru *cross-validation*;
- 20% pentru testare.

Etapa de antrenare este compusă din două faze, în funcție de datele de intrare – imagini obișnuite, respectiv copii de documente. Fiecărui descriptor i se asociază o rețea neuronală, pentru a determina procentul de apartenenţă la o anumită clasă după cum urmează:

- 4 rețele neuronale corespunzătoare spațiului culorilor;
- 2 rețele neuronale corespunzătoare descriptorilor locali;
- o rețea neuronală corespunzătoare descriptorului de textură.

Ulterior, rezultatele rețelelor neuronale sunt colectate de două module de vot majoritar ponderat, care au rolul de a stabili rezultatul final al clasificării, după cum urmează:

- rețelele neuronale din spațiul de culoare sunt colectate de primul modul de vot majoritar;
- cele două rețele neuronale corespunzătoare descriptorilor locali sunt ponderate cu 50%;
- rezultatele clasificării din spaţiile de culoare, al descriptorilor locali, respectiv din spațiul texturilor sunt apoi ponderate de cel de-al doilea modul de vot majoritar.

4.1.1.1 Etapa de analiză a documentelor

Documentele trec printr-o serie de etape de preprocesare înainte de a fi înaintate către modulul de clasificare, şi anume redimensionare, binarizare şi segmentare.

Redimensionarea nu este necesară în totalitatea cazurilor, dar unele *scan*-uri sunt realizate la rezoluții foarte mari, ceea ce creşte foarte mult timpul de procesare.

Binarizarea ajută ulterior, în procesul de segmentare. Așa cum am menționat în capitolul Contribuții privind tehnicile de procesare a documentelor, algoritmul de binarizare are o importanţă deosebită şi poate aduce scăderi majore de performanţă în cadrul procesului de clasificare. Prin urmare, algoritmul de binarizare a fost ales cu atenție, în urma testării a 6 implementări diferite; algoritmul care a avut rezultatele cele mai bune este NLBIN.

Segmentarea imaginilor din document este oferită de o implementare proprie – algoritmul CVSEG.

Nu sunt efectuate alte modificări asupra documentelor în cadrul procesului de clasificare, sau de extragere a descriptorilor. Cu toate acestea, există o serie de diferențe între o imagine standard şi o copie de document:

- Poate cea mai importantă caracteristică a copiilor de documente este lipsa culorii în cele mai multe cazuri. Prin urmare, relevanţa descriptorilor din spațiul culorilor scade foarte mult;
- Algoritmul de segmentare poate împărţi documentul în mai multe imagini, ceea ce înseamnă că la finalul procesului de clasificare vor fi oferite mai multe rezultate;
- Așa cum am arătat anterior, există posibilitatea prezenţei unor zone de zgomot în rezultatele segmentării, dar de obicei acesta nu afectează rezultatul clasificării.

4.2 Rezultate obţinute

În faza incipientă, în **spaţiul culorilor** s-a încercat folosirea de histograme RGB pentru determinarea distanţei dintre imaginea primită ca *input* şi imaginile stocate în baza de date. Rezultatele nu au fost concludente, din mai multe motive:

- doua imagini pot avea histograme foarte asemănătoare, dar conținut complet diferit;
- imaginile cu obiecte din aceeași clasă, dar fotografiate în condiții de iluminare diferite erau clasificate eronat;
- în cazul imaginilor alb-negru sau în nuanțe de gri, histogramele nu sunt relevante.

Ulterior s-a încercat determinarea numărului optim de zone de culoare (eng. *bins*), dar nici aceste teste nu au dat rezultate satisfăcătoare, din aceleași motive. Singura concluzie a fost că în cazul cuantizării culorilor dintr-o imagine, numărul de zone de culoare afectează precizia rezultatului în funcție de doi parametri - numărul real de culori din imagine, respectiv dimensiunea imaginii. Prin urmare, histogramele RGB au fost păstrate doar ca un criteriu suplimentar, oferit opțional utilizatorului, care poate fi interesat în special de aceste caracteristici. Utilizarea spatiilor c1c2c3/l1l2l3 a ameliorat rezultatele obţinute, prin eliminarea zonelor de umbra şi de *highlight*. Din păcate, şi în acest caz există probleme, introduse în special de tendința acestor spații de a normaliza suprafețe mari din cadrul imaginii şi chiar de a elimina unele obiecte.

În concluzie, caracteristicile obţinute în spațiul culorilor vor avea o pondere mică în cadrul procesului de clasificare.

O etapă aparte în cadrul obținerii acestor descriptori este reprezentată de procesarea imaginilor alb-negru, respectiv a copiilor de documente în care lipsește culoarea în cele mai multe cazuri. Prin urmare, având în vedere că toți cei 4 descriptori au fost concepuți să ruleze pe imagini RGB, se impune o serie de transformări:

- având în vedere că imaginile *grayscale* sunt bidimensionale, iar cele RGB sunt tridimensionale, caracteristicile 2D sunt triplate, pentru a simula coordonatele 3D;
- ținând cont de formula transformării din RGB în *grayscale* (GS=(R+G+B)/3), primul set de teste a fost efectuat pe imagini în care toate cele 3 canale aveau aceeași valoare (R=G=B=GS). Testele au dat rezultate foarte slabe, din motivul că această transformare nu este corectă, deoarece estompează foarte multe detalii;
- Lumina naturală, observată de ochiul uman, este filtrată în proporții diferite pentru a elimina excesul de culoare albastră, de exemplu. Prin urmare, fiecare canal este ponderat, pentru a obține o aproximare acceptabilă a unei imagini percepute de ochiul uman, conform următoarelor valori, stabilite empiric: GS = R*0.32+G*0.57+B*0.11

După cum am mai spus, în **spațiul texturilor** problemele ridicate de abordarea bazată pe LBP au fost în special legate de forma şi dimensiunea celulei, respectiv de numărul de pixeli pe baza cărora stabilim caracteristicile de textură.

Au fost executate mai multe serii de teste, în special pe imagini cu texturi uniforme, pentru a determina corectitudinea rezultatelor:

- în ceea ce privește celulele radiale, rezultatele obţinute au fost îmbucurătoare, urcând chiar până la 100%;
- celulele pătrate au dat rezultate ceva mai slabe, dar peste 90%;
- s-a observat că în cazul celulelor radiale timpii de execuție au fost mult mai mari, în special datorită calculului trigonometric.

În concluzie, s-a stabilit că diferența de precizie este acceptabilă, comparativ cu scăderea de performanţă cauzată de folosirea celulelor radiale.

În ceea ce privește dimensiunea celulei, testele au arătat că în general:

- creșterea dimensiunii celulei duce la scăderea preciziei, datorită ignorării texturilor mici. Pe de altă parte, procesul de extragere a caracteristicilor texturale este accelerat;
- scăderea dimensiunii celulei duce la creșterea complexității calculului (şi implicit a timpului de răspuns), respectiv la îmbunătăţirea performanţei. Singurele cazuri în care algoritmul nu funcționează corect sunt intalnite în cazul imaginilor cu texturi mari, afectate de zgomot.

În urma testelor efectuate în cadrul acestei zone, am observat că rezultatele cele mai bune din punct de vedere computațional şi al preciziei au fost obţinute atunci când:

- celula are formă rectangulară (pătrat);
- dimensiunea celulei este de 5 pixeli per latură;
- numărul pixelilor cu care determinăm textura este de 8.

După cum am menționat anterior, modulul de extragere a **descriptorilor locali** folosește caracteristici SIFT şi HOG. Pentru a obține aceste caracteristici, am folosit chiar implementările autorilor, cu următoarele mențiuni:

- tradițional, descriptorii HOG sunt folosiți în probleme de clasificare binară. Prin urmare acești descriptori sunt folosiți pentru a antrena un clasificator SVM. Deoarece în contextul CBIR avem de-a face cu o problemă de clasificare multiplă, am folosit o rețea neuronală;
- descriptorii SIFT sunt extrași cu ajutorul unui fișier binar executabil şi stocați într-un fișier. Din cauză că nu avem posibilitatea de a interveni în codul sursă, a fost necesară *parse*-area fișierului rezultat şi folosirea descriptorilor astfel rezultați pentru a antrena rețeaua neuronală corespunzătoare.

4.2.1 Procesul de clasificare

Întregul sistem a fost testat pe un set de date CIFAR, oferit de (14). Setul de date include 60000 de imagini color, de dimensiuni mici (32x32 pixeli), împărţit în:

- 50000 imagini - pentru antrenare
- 10000 imagini - pentru testare

Acest set de date a fost păstrat numai pentru validarea clasificatorului în cazul în care sunt folosite imagini tradiţionale. Ulterior, în cazul în care au fost folosite documente scanate la rezoluţii ridicate, a fost folosită o combinaţie de imagini obţinută din mai mute surse, oferite de MLCOMP (15) – 983 seturi, respectiv UCI (16) – 298 seturi. Această modificare a fost dictată şi de numarul redus de documente care conţineau obiecte de tipul celor din setul CIFAR (de exemplu, având in vedere că documentele ICDAR sunt documente vechi, alese în aşa fel încât să expună cât mai multe din problemele DAR, nu puteau conţine imagini cu avioane, sau camioane).

În urma testelor efectuate de autor (rezultatele sunt reproductibile), prin folosirea unei rețele neuronale de tip *feed forward* cu *back propagation*, rezultatele obţinute au fost de 87%. În faza incipientă, antrenarea sistemului nostru a fost efectuată pe imaginile brute, cu ajutorul unei rețele neuronale de același tip. Rezultatele obţinute au fost foarte asemănătoare cu cele ale autorului (85%). Cu toate acestea, atunci când sistemului i-au fost prezentate imagini obţinute din afara setului de imagini oferite de autor, performanţele au scăzut considerabil, ceea ce a condus la necesitatea de a schimba arhitectura clasificatorului.

Rezultatele experimentului menționat anterior, în care ne folosim exclusiv de descriptori proveniți din spațiul de culoare RGB, au pendulat în jurul pragului de 70%, în condițiile expunerii la imagini provenite din afara setului obținut de la autor. Prin urmare, în testele ulterioare au fost introduse secvențial alte seturi de descriptori în cadrul **clasificatorului de culoare**, după cum urmează:

- imagini convertite la spațiul I1I2I3;
- imagini convertite la spațiul c1c2c3;
- histograme RGB.

Rezultatele obţinute sunt descrise în tabelul de mai jos:

Tabel 7. Rezultate obţinute pe diferite tipuri de descriptori culoare

Combinaţie descriptori	**Rezultate obţinute**
RGB+I1I2I3	82%
RGB+c1c2c3	**84%**
RGB+histograme RGB	69%
RGB	71%

După cum se poate observa, prezenţa spaţiilor de culoare c1c2c3, sau I1I2I3 produce îmbunătăţiri substanțiale, de peste 10 procente. Pe de altă parte, introducerea histogramelor RGB a dus la scăderea performanţelor. Prin urmare au fost trase următoarele concluzii:

- experimentele efectuate pe imagini provenite din lumea reală (nu din laborator) confirmă necesitatea introducerii unor spaţii suplimentare de culoare;
- spațiul c1c2c3 a produs cele mai solide îmbunătăţiri. În urma analizei setului de imagini s-a observat prezenţa multor zone de umbră, ceea ce confirmă eficienţa descriptorilor proveniți din acest spațiu, în aceste condiții;
- spațiul I1I2I3 este clasat pe locul secund. Eficienţa sa a fost mai redusa pe acest set de imagini, deoarece zonele de *highlight* sunt (de obicei) mult mai puține decât cele de umbră;
- introducerea histogramelor RGB a scăzut precizia clasificatorului. Interpretarea acestui rezultat este legată de condițiile în care au fost fotografiate obiectele. Mediul înconjurător poate afecta în mare măsură histograma de culoare, ceea ce a dus la o scădere drastica a performanţei, mai ales atunci când ne raportăm la rezultatele anunțate în faza incipientă. Totodată, prezenţa zonelor de umbră menționate anterior afectează histogramele şi implicit rezultatul final al clasificării.

În aceste condiții, s-a impus introducerea unui modul de vot majoritar, cu rolul de a pondera importanţa rezultatelor obţinute din cele 4 zone.

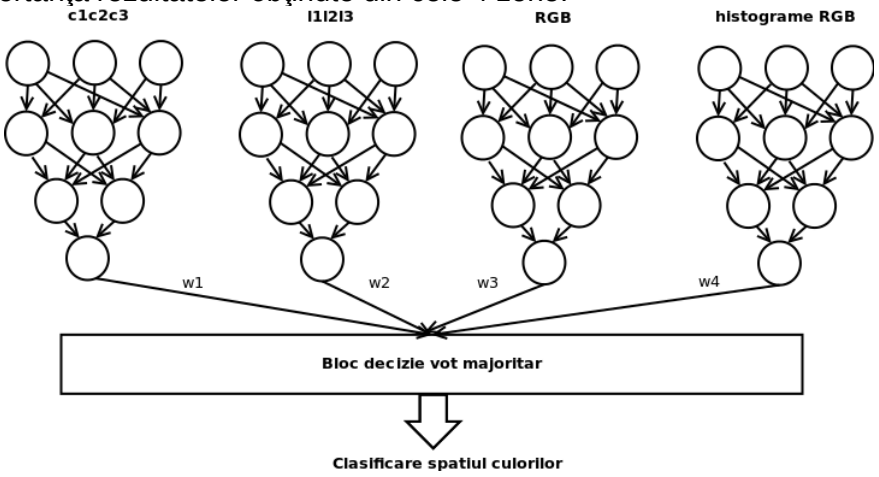


Figura 25. Modul vot majoritar în spațiul culorilor

Funcționarea acestui modul este descrisă mai jos:

- fie n numărul de clase acceptate şi k numărul de clasificatori, atunci
- fiecare rețea neuronală va avea pe nivelul de ieșire un vector de tipul Cx={c_1, c_2, ... c_n}, unde 1<=x<=k
- ponderea asociată vectorului de ieșiri va fi W={ w_1, w_1 ... w_k}
- rezultatul ponderat va fi dat de suma w_iC_i, conform formulei de mai jos:

$$R = idx\left(max\left(\sum_{i=1}^{k} w_i C_i\right)\right),$$

unde R ϵ [1, n],
max(C) reprezintă valoarea maximă obținută pentru o anumită clasă,
iar idx reprezintă poziția acelei clase în vectorul final
Formula 21. Determinarea rezultatului modulului ponderat

Experimentele ulterioare au folosit următoarele valori pentru ponderi:

- c1c2c3 - w1 = 30%;
- I1I2I3 - w2 = 30%;
- RGB - w3 = 30%;
- histograme RGB - w4 = 10%.

În aceste condiții, rezultatele clasificării au urcat la 86%. Cu toate acestea, rezultatele sunt strict legate de identificarea obiectului ca fiind parte din imagine, ignorând în mare măsura noțiunea de culoare propriu-zisă. Din acest motiv, utilizatorului îi va fi pusă la dispoziție posibilitatea de a creşte relevanţa culorii în rezultatul final, prin alterarea ponderii asociate histogramelor RGB (de exemplu - w1=20%, w2=20%, w3=20%, w4=40%).

4.2.1.1 Arhitectura clasificatorului final

În cadrul procesului de clasificare final a fost introdus un alt modul de vot majoritar, care funcționează pe baza acelorași principii ca în cazul clasificatorului de culoare.

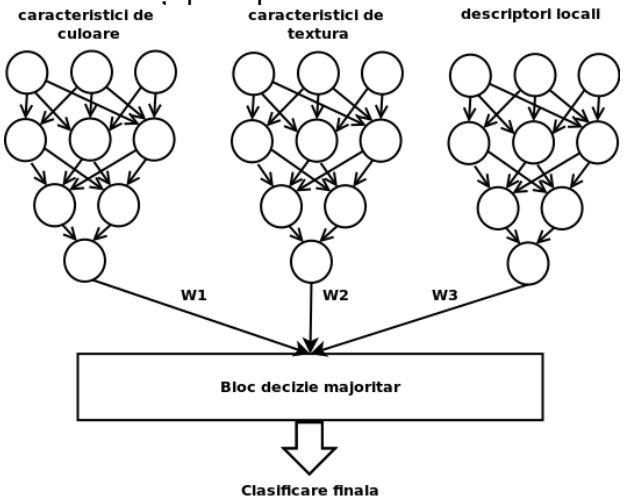


Figura 26. Modul vot majoritar clasificare finală

Ponderile folosite în cadrul experimentelor au fost:

- caracteristici de culoare: w_1=33%;
- caracteristici de textură: w_2=33%;
- descriptori locali: w_3=33%. Cele două seturi de descriptori locali au avut ponderi de 50% în cadrul determinării caracteristicilor locale per ansamblu.

Rezultatele agregate sunt descrise în tabelul de mai jos:

Tabel 8. Rezultate agregate în cadrul modulului de vot majoritar final

Tip clasificare	**Rezultat obținut**
Caracteristici culoare (RGB, histograme RGB, c1c2c3, l1l2l3)	86%
Caracteristici textură (LBP)	85%
SIFT	82%
HOG	85%
Toate cele de mai sus	**92%**

Rezultatele obţinute pe acest set de date sunt promițătoare şi arată o creștere de performanţă cu peste 5 procente. Cu toate acestea, există o serie de lucruri care trebuie menționate pentru a înțelege mai bine prelucrarea imaginilor de dimensiuni mari/rezonabile (640x480):

- prelucrarea unor imagini de dimensiuni standard (32x32 pixeli) cu ajutorul unei rețele neuronale tradiționale este relativ facilă, deoarece numărul de caracteristici este același (F=numărul de pixeli*numărul de culori). Așa se explică rezultatele similare obţinute în primele experimente, în care am folosit doar prelucrări în spațiul culorilor, respectiv caracteristici de textură;
- această soluție nu este aplicabilă în probleme reale, deoarece trecerea la dimensiuni ridicate ale imaginii presupune probleme majore de scalare arhitecturală. De exemplu, pentru o imagine de 640*480 pixeli, rețea neuronală ar trebui sa aibă 307200 neuroni în stratul de intrare. Totodată, având în vedere că o imagine color prezintă 3 canale de culoare, pentru fiecare imagine trebuie asociate 3 rețele neuronale, deci numărul total de neuroni ajunge la 10^6. Ca soluție alternativă, imaginea ar putea fi redusă la o scară mai mică, potrivită cu dimensiunile rețelei, dar în acest mod s-ar pierde date, iar imaginea ar putea fi alterată, afectând în acest fel rezultatul clasificării;

- în ceea ce privește folosirea algoritmului SIFT, nu se cunoaște dimensiunea caracteristicilor cu care trebuie antrenată rețeaua neuronală. Prin urmare, în momentul de faţă, numărul de *key points* este limitat la 1000;
- algoritmul HOG oferă posibilitatea de a căuta în imagini anumite obiecte pentru care a fost antrenat. Din păcate, descriptorii HOG sunt învăţaţi cu un SVM, care este un clasificator binar, ceea ce înseamnă că ar trebui adăugate module SVM pentru fiecare clasă. Chiar şi așa, tehnicile de *resampling* şi *sliding window* sunt foarte eficace.

4.2.1.2 Clasificarea mixtă

Toate experimentele descrise până acum în cadrul CBIR au folosit imagini tradiționale, sau documente procesate corect în proporție de 100%. În cele ce urmează vom prezenta rezultatele obţinute în cadrul procesului de clasificare, atunci când sunt analizate copii de documente afectate de diferite tipuri de probleme, specifice zonei DAR.

Prezenţa acestor factorilor de zgomot specifici DAR poate afecta rezultatul final al procesului de clasificare. Prin urmare, în cele ce urmează, vom descrie rezultatele unor teste efectuate defalcat pe fiecare etapă din cadrul procesului de procesare de documente.

Testele efectuate în cadrul implementării modulului de binarizare (pe 4 algoritmi diferiți) au arătat că rezultatele cele mai performante au fost oferite de NLBIN, respectiv algoritmul lui Sauvola. Ambii algoritmi au performanţe similare şi sunt afectați de același tip de probleme:

- Documentele vechi sunt binarizate incorect, ceea ce duce la includerea de zone negre mari în imaginea rezultat;
- Erori de estimare a unghiului de rotație;
- Erori de estimare a valorii de prag, cauzate de obicei de contrastul scăzut, sau de calitatea scăzuta a hârtiei. Din acest motiv, unele detalii sunt estompate, iar altele sunt augmentate.

Prin urmare, întregul sistem de clasificare a fost testat pe un set de imagini care produc problemele de mai sus, în contextul unei segmentări corecte în procent de 100%. Rezultatele sistemului sunt descrise în tabelul de mai jos:

Tabel 9. Rezultate clasificare pe imagini cu probleme în cadrul procesului de binarizare

Problema de binarizare	Rezultate obţinute
Zone negre în imaginea rezultat	75.50%
Erori estimare unghi rotație	90.80%
Erori estimare valoare de prag	89.10%

După cum se poate observa, cele mai mari probleme sunt cauzate de prezenţa zonelor negre în imaginea rezultat. Atunci când algoritmii de binarizare introduc asemenea erori, de obicei acoperă mare parte din document, mascând atât textul, cât şi imaginile, ceea ce afectează rezultatul segmentării şi al clasificării per ansamblu.

Următoarea cădere de performanţă a fost cauzată de erorile de estimare a valorii de prag. În unele cazuri detaliile sunt atenuate extrem de mult, sau chiar eliminate. Atunci când detaliile nu sunt eliminate complet, algoritmul de segmentare le poate considera ca fiind elemente de zgomot, ceea ce duce la filtrarea lor. În aceste circumstanțe, algoritmul de clasificare nu mai poate extrage suficienți descriptori, ceea ce a dus la o diferența de 3%.

Există de asemenea o scădere de performanţă în cazul imaginilor care nu sunt aliniate corect la axe, dar de obicei acestea sunt rezolvate de invarianţa la rotație a descriptorilor din modulul de clasificare.

În cadrul experimentelor legate de partea de segmentare, am folosit algoritmul CVSEG. Întregul sistem a fost testat cu imagini care introduc probleme, după cum urmează:

- Imagini care sunt binarizate greșit – în aceasta situație, rezultatul segmentării va include prea mult, sau prea puțin din imaginea originală;
- Folosirea de caractere cu dimensiuni variate – unele caractere pot fi incluse în imaginea rezultat;
- Probleme de structură a paginii, chenare, prezenţa unor obiecte care nu făceau parte din documentul inițial, documente nealiniate la axe.

Rezultatele experimentelor sunt descrise în tabelul de mai jos:

Tabel 10. Rezultate clasificare pe imagini cu probleme în cadrul procesului de segmentare

Problema de segmentare	Rezultate obţinute
Binarizare incorectă	74.20%
Caractere variate	91.00%
Structura pagină	91.10%

După cum se poate observa, şi în această situație, rezultatele binarizării au un impact puternic asupra întregului proces de clasificare. După cum am mai spus, algoritmul de segmentare nu poate extrage imaginea corectă dintr-un document binarizat incorect, deoarece în multe cazuri aceasta este mascata de zonele negre. Aceste zone pot fi cauzate în urma apariției problemelor de mai jos:

- Contrastul scăzut, fundalul cu nuanțe închise (în special în cazul documentelor vechi), sau transparenţa paginii determină algoritmul de binarizare să calculeze incorect pragul de activare, ceea ce duce la mascarea unor zone mari din imaginea rezultat (uneori chiar întregul document);
- În alte cazuri, datorită augmentării excesive a detaliilor, se ajunge la contopirea acestora. Această situație este întâlnită atunci când documentul conține zone mixte, cu detalii formate din linii subțiri şi linii groase; algoritmul de binarizare va încerca accentuarea detaliilor șterse, respectiv diminuarea celor pregnante, dar atunci când contrastul este scăzut, acest proces poate eșua.

În următoarele două situații, sistemul de clasificare prezintă scăderi de performanţă minime (sub 1%). În cazul caracterelor variate, prezenţa unor zone suplimentare în imaginea rezultat nu afectează procesul de clasificare. Elementele de chenar din structura paginii pot produce erori în cadrul algoritmului CVSEG, datorită filtrării zonelor de text prin proiecție pe axe, dar algoritmul de clasificare nu este afectat. Majoritatea problemelor din această zonă au fost cauzate de segmentarea incompletă a imaginilor din document.

Următoarele experimente au folosit un set mixt de imagini (toate provenind din arhivele ICDAR, sau din documentația sistemului de operare Ubuntu), după cum urmează:

- Imagini cu, sau fără probleme în zonele de binarizare (fără probleme de segmentare);
- Imagini cu, sau fără probleme în zonele de segmentare;
- Imagini mixte.

Rezultatele obţinute sunt prezentate în tabelul de mai jos:

Tabel 11. Performanţa generală a sistemului de clasificare

Tip imagini	Rezultate medii	Scădere de performanţă
Probleme binarizare	85.27%	6.73%
Probleme segmentare	85.52%	6.48%
Imagini mixte	89.58%	2.42%

După cum ne așteptam, cele mai mari scăderi de performanţă sunt cauzate de etapa de binarizare, care afectează totodată şi funcționarea modulelor ulterioare. Cu toate acestea, diferențele nu sunt foarte mari, în principal deoarece algoritmul de clasificare poate identifica imagini augmentate excesiv şi în unele cazuri, imagini incomplete.

Dacă analizăm rezultatele obţinute pe diferite seturi de date, obținem performanţe care variază între 85% şi 92%, după cum se poate observa în graficul de mai jos:

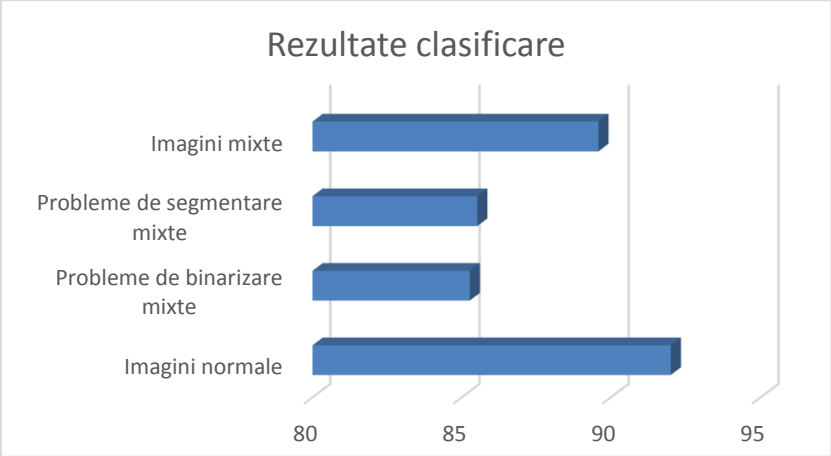


Figura 27. Fluctuații performanţă conform tip de date

Cu toate acestea, majoritatea scăderilor de performanţă au fost cauzate de prezenţa problemelor din cadrul etapei de binarizare; mai exact, chiar dacă rezultatele algoritmului de segmentare au introdus probleme în cadrul blocului de clasificare, acestea au fost cauzate preponderent de algoritmul de binarizare, în special de introducerea zonelor negre. Prin urmare, dacă analizăm rezultatele obţinute în cadrul etapei de clasificare, în absenţa erorilor de acest tip (de calculare a valorii de prag), obținem următorul grafic, cu valori între 89.5% şi 92%:

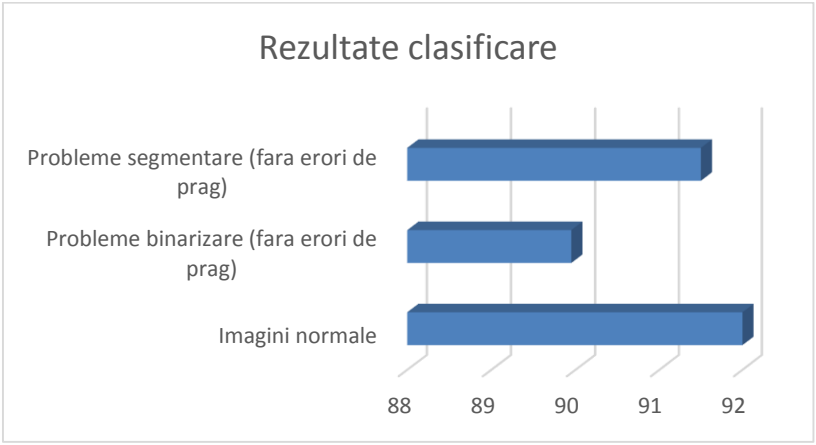


Figura 28. Rezultatele algoritmului de clasificare în absenţa erorilor de prag

Un alt aspect interesant al rezultatelor este determinat de calculul performanţelor medii ale algoritmilor de binarizare şi segmentare şi raportarea lor la performanţele maxime ale algoritmului de clasificare. Mai exact, dorim să analizăm rezultatele obţinute în contextul următor:

- algoritmul de binarizare are performanţa Pb;
- algoritmul de segmentare are performanţa Ps;
- performanţa maximă a algoritmului de clasificare este Pc;
- performanţa reală (obținută) a algoritmului de clasificare în zona DAR este Prc;
- ne interesează valorile Prc-Pb*Ps*Pc, respectiv Prc-Ps*Pc.

Tabelul de mai jos descrie rezultatele obţinute:

Tabel 12. Fluctuații performanţă

Etapa clasificare DAR	Performanţa
Binarizare - Pb	85.13
Segmentare - Ps	85.43

Clasificare reală - Prc	89.58
Clasificare optimă - Pc	92.00
Ps*Pc	78.60
Pb*Ps*Pc	66.91

După cum se poate observa, în condițiile în care algoritmii de segmentare şi binarizare ar produce erori de clasificare în situațiile (100-Pb), respectiv (100-Ps), iar algoritmul de clasificare ar eșua în toate situațiile (100-Pc), performanţa finală ar trebui să fie de 66.9%. Dacă dorim să excludem influenţa algoritmului de binarizare, deoarece algoritmul de clasificare procesează doar rezultatele algoritmului de segmentare, performanţa finală ar trebui să fie de 78.6%. Cu toate acestea, algoritmul a obținut o performanţă reală de 89.58%, ceea ce înseamnă că procesul de clasificare reușește să dea rezultate corecte, chiar în condițiile unor rezultate incorecte provenite din zona DAR.

4.2.2 Antrenarea clasificatorului cu imagini mixte

Testele descrise anterior au fost realizate după cum urmează:

- Clasificatorul este antrenat cu ajutorul imaginilor standard;
- Validarea sistemului a fost realizată cu imagini mixte.

Pentru a avea o imagine clară asupra performanţelor clasificatorului, trebuie să validăm şi scenariul în care antrenarea este realizată cu ajutorul imaginilor mixte. Prin urmare, setul de imagini ICDAR a fost împărţit la rândul lui în 3 eșantioane, ca în cazul imaginilor standard.

În faza incipientă a fost folosită o procedură de validare care implică folosirea de imagini mixte (imagini standard şi copii de documente). Rezultatele obţinute sunt prezentate în tabelul de mai jos:

Tabel 13. Validarea sistemului în condițiile antrenării mixte

Tip clasificare	Rezultat obținut
Caracteristici culoare (RGB, histograme RGB, c1c2c3, l1l2l3)	80%
Caracteristici textură (LBP)	85%
SIFT	82%
HOG	85%
Toate cele de mai sus	**90%**

Putem observa o scădere de 4 procente în cazul clasificării bazate exclusiv pe descriptori din spațiul culorilor. Acest rezultat vine ca o confirmare asupra faptului că relevanța acestor descriptori scade în cazul în care imaginile folosite provin din spațiul *grayscale*, din mai multe motive:

- Formula folosită la transformarea din *grayscale* în coordonate RGB este o aproximare;
- Relevanţa descriptorilor din spaţiile c1c2c3, respectiv l1l2l3 scade. Acești descriptori nu mai reușesc să elimine zonele de umbra şi *highlight*, estompează foarte multe detalii, respectiv introduc o zonă de zgomot la limita dintre doua zone disjuncte;
- Histogramele RGB îşi pierd din relevanţă, datorită cuantizării diferite a culorii prezente în fiecare canal, ceea ce duce la apariția unor diferențe chiar şi în aceasta etapă.

Cu toate acestea, rezultatele clasificării din acest spațiu sunt rezonabile, mai ales în contextul în care urmează să fie ponderată ulterior în cadrul modulelor de vot majoritar.

Atât rezultatele descriptorilor de textură, cât şi ale celor două seturi de descriptori locali nu au fost afectate de introducerea de documente în setul de date de antrenare. Acest lucru validează robustețea acestor algoritmi în condiții variate ale spațiilor de culoare.

Rezultatele finale ale procesului de clasificare prezintă o scădere de 2 procente, cauzată în principal de diferența de performanţă din zona descriptorilor de culoare. Considerăm acest rezultat îmbucurător, deoarece se situează în continuare în zona de precizie mai mare decât 90%.

Ulterior am continuat procesul de validare folosind imagini care provin numai din zona DAR. Ca şi în etapele anterioare, am împărţit setul de test în mai multe categorii, după cum urmează:

- Imagini cu probleme de binarizare;
- Imagini cu probleme de segmentare;
- Imagini mixte.

Ne interesează fluctuațiile de performanţă în cazul imaginilor mixte, respectiv impactul problemelor DAR în cadrul clasificării finale.

Rezultatele obţinute în cadrul imaginilor care prezintă probleme de binarizare sunt prezentate în tabelul de mai jos (ultima coloană prezintă acuratețea sistemului în cazul în care a fost antrenat numai cu imagini standard, iar etapa de segmentare a fost păstrata la o eficienta de 100%):

Tabel 14. Rezultate obţinute în urma validării sistemului cu imagini care prezintă probleme de binarizare

Problema de binarizare	Rezultate obţinute	Rezultate obţinute anterior
Zone negre în imaginea rezultat	76.28%	75.50%
Erori estimare unghi rotație	90.90%	90.80%
Erori estimare valoare de prag	89.10%	89.10%

În ceea ce privește prezenţa zonelor negre în imaginea rezultat, se poate observa o îmbunătăţire de 0.78%. Considerăm că această îmbunătăţire minoră este de fapt artificială şi este cauzată de prezenţa unor imagini similare în cadrul etapei de antrenare. În realitate, caracteristicile imaginii sunt foarte mult atenuate, atât în cadrul etapei de extragere a descriptorilor de culoare, cât şi a celor de textură şi locali.

Erorile de estimare a unghiului de rotație au în continuare un impact minim asupra rezultatului final al clasificării, datorită caracterului invariant al descriptorilor (în special în cazul celor de textură, respectiv locali). Se observă o creștere de 0.1% faţă de celălalt scenariu, care presupunea antrenarea exclusiv bazată pe imagini tradiționale. Cu toate că este un rezultat excelent, ne așteptăm la rezultate mai bune, cu minim 2 procente.

În cazul erorilor de estimare a valorii de prag, rezultatele sunt în continuare foarte bune. Chiar dacă unele detalii sunt atenuate, sau augmentate excesiv, clasificatorul poate extrage descriptori suficient de reprezentativi pentru a putea recunoaște obiectele din etapa de antrenare. Cu toate acestea, se poate observa că în acest caz antrenarea clasificatorului cu imagini mixte nu a produs îmbunătățiri, precizia stagnând.

Următorul scenariu a implicat validarea sistemului cu ajutorul imaginilor cu probleme de segmentare. Rezultatele obţinute sunt prezentate în tabelul de mai jos:

Tabel 15. Rezultate obţinute în urma validării sistemului cu imagini care prezintă probleme de segmentare

Problema de segmentare	Rezultate obţinute	Rezultate obţinute anterior
Binarizare incorectă	74.50%	74.20%
Caractere variate	91.20%	91.00%
Structură pagină	91.15%	91.10%

Ca şi în experimentul anterior, se poate observa o îmbunătăţire de 0.3% în cadrul problemelor de binarizare incorectă. Suntem în continuare de părere că această îmbunătăţire este artificială şi este cauzată de alterarea ponderilor rețelei neuronale în cadrul etapei de antrenare.

Prezenţa caracterelor variate pe pagină produce o creștere de performanţă de 0.2%, faţă de scenariul în care antrenarea s-a efectuat pe imagini mixte. Ca şi în cazul anterior, creșterea este minimă, ceea ce întăreşte suspiciunea că introducerea de copii de documente în setul de antrenare nu produce îmbunătăţiri substanțiale.

În cazul problemelor de structură de pagină creșterea este şi mai puțin semnificativă, de 0.05%.

Următorul set de teste din zona DAR a fost efectuat cu ajutorul unor imagini care provin din toate zonele de interes, după cum urmează:

- Imagini DAR cu probleme de binarizare;
- Imagini DAR cu probleme de segmentare;
- Imagini DAR fără probleme.

Scopul acestui test este de a determina performanţa clasificatorului în zona DAR, respectiv de a determina care din cele două etape influențează cel mai mult rezultatele finale. Rezultatele obţinute sunt prezentate în tabelul următor:

Tabel 16. Rezultate obţinute în urma validării sistemului cu imagini cu, sau fără probleme din toate zonele DAR

Tip date	Rezultate obţinute
Probleme binarizare	85.01%
Probleme segmentare	85.46%
Fără probleme	92.12%

După cum se poate observa, clasificatorul obține performanţe de peste 92% atunci când imaginile nu sunt afectate de zgomot, dar rezultatele scad cu peste 7% atunci când sunt introduse perturbațiile.

Așa cum am arătat anterior, algoritmul de binarizare este cuplat la cel de segmentare, care furnizează ulterior *input*-ul clasificatorului. Prin urmare, am vrut să verificăm, ca în cazul anterior, dacă problemele din cadrul binarizării sunt principalele cauze ale scăderii de performanţă. În mod special suntem interesați de influenţa determinării pragului de binarizare în etapa de clasificare finală.

Graficul de mai jos descrie rezultatele obţinute atunci când am testat sistemul numai prin imagini care nu sunt afectate de probleme de determinare de prag (în principiu, cu documente care prezintă un contrast bun).

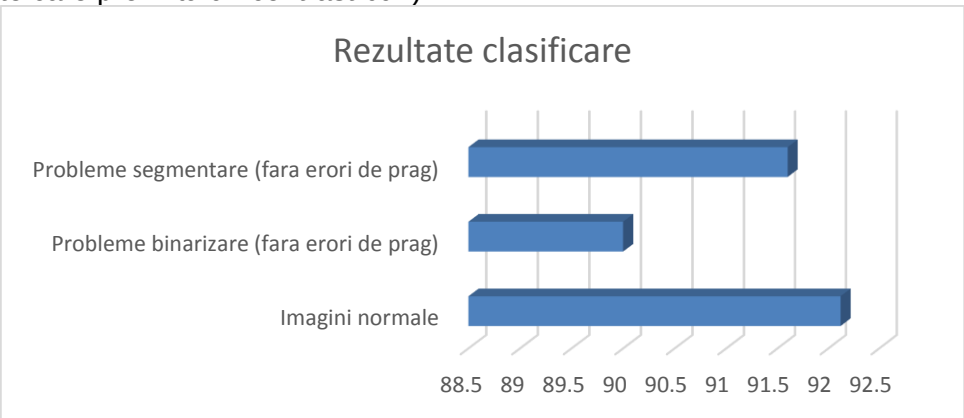


Figura 29. Performanţa sistemului în zona DAR, fără probleme de determinare a pragului

Așa cum se poate observa, cele mai slabe rezultate sunt obţinute în continuare atunci când imaginile sunt afectate de probleme de binarizare. Cu toate acestea, performanțele au

crescut cu minim 5% în cazul primelor două cazuri, aproape atingând pragul de 92% în cazul imaginilor cu probleme de segmentare.

Menționăm totuși că rezultatele ultimelor experimente nu sunt în totalitate corecte, deoarece s-a păstrat același clasificator ca în cazurile anterioare. Mai exact, nu s-a trecut printr-o nouă etapă de antrenare, în cadrul căreia să nu existe documente afectate de probleme de stabilire a pragului. Cu toate acestea, putem considera aceste rezultate ca fiind o bună aproximare a cazului real, în special datorită plasticității tipului de clasificator de bază folosit în arhitectura sistemului final – rețeaua neuronală.

Prin urmare, putem concluziona că problemele de stabilire corectă a pragului de binarizare determină cele mai mari scăderi de performanţă în cadrul procesului final de clasificare.

Ultimul set de experimente a fost efectuat numai cu ajutorul imaginilor tradiționale, pentru a putea determina variația performanţei sistemului între cele două scenarii de antrenare. Rezultatele sunt descrise în tabelul de mai jos:

Tabel 17. Rezultate comparative obţinute în urma validării sistemului cu imagini din toate zonele de interes

Tip date	Rezultate obţinute	Rezultate obţinute anterior
Imagini tradiționale	91.14%	92%
Documente	89.83%	89.58%
Imagini mixte	90.04%	90.25%

În cazul imaginilor tradiționale am obținut o scădere de acuratețe de 0.86%. Ne așteptam la acest rezultat după introducerea imaginilor din zona DAR în setul de date de antrenare. Cu toate acestea, rezultatul obținut este foarte bun, depășind pragul de 90%.

În cazul copiilor de documente se înregistrează o creștere 0.25% pe un set de test mixt, conținând atât imagini cu probleme, cât şi imagini obţinute din convertirea de manuale Ubuntu. Considerăm această creștere de performanţă nesatisfăcătoare, mai ales în contextul scăderii obţinute în cazul imaginilor tradiționale.

Per ansamblu, atunci când sistemul este validat cu imagini provenind din toate zonele de interes, înregistrăm iarăși o scădere de performanta de 0.21%, datorită scăderii din zona imaginilor tradiționale.

Având în vedere rezultatele experimentelor descrise în acest capitol, putem concluziona că introducerea de imagini din zona DAR în setul de antrenare nu produce îmbunătăţiri substanțiale:

- În cadrul problemelor de binarizare înregistrăm creșteri minime de performanţă, sau stagnări;
- În cadrul problemelor de segmentare nu există stagnări, dar creșterile au valori între 0.05%-0.3%;
- Documentele care nu prezintă probleme de niciun fel sunt clasificate corect, în proporție de peste 92%. Aceasta este cea mai importantă contribuție a setul de antrenare în cadrul procesului de clasificare. Menționăm că în scenariul anterior de antrenare am obținut performante de 89.58%;
- Imaginile tradiționale sunt clasificate corect, dar în acest caz înregistrăm o scădere de aproape 1%.

Prezentăm în graficul de mai jos rezultatele sistemului în diferite scenarii de testare, implicând imagini tradiționale, sau documente variate (cu, sau fara probleme în cadrul etapelor de binarizare şi segmentare), în ambele variante de antrenare:

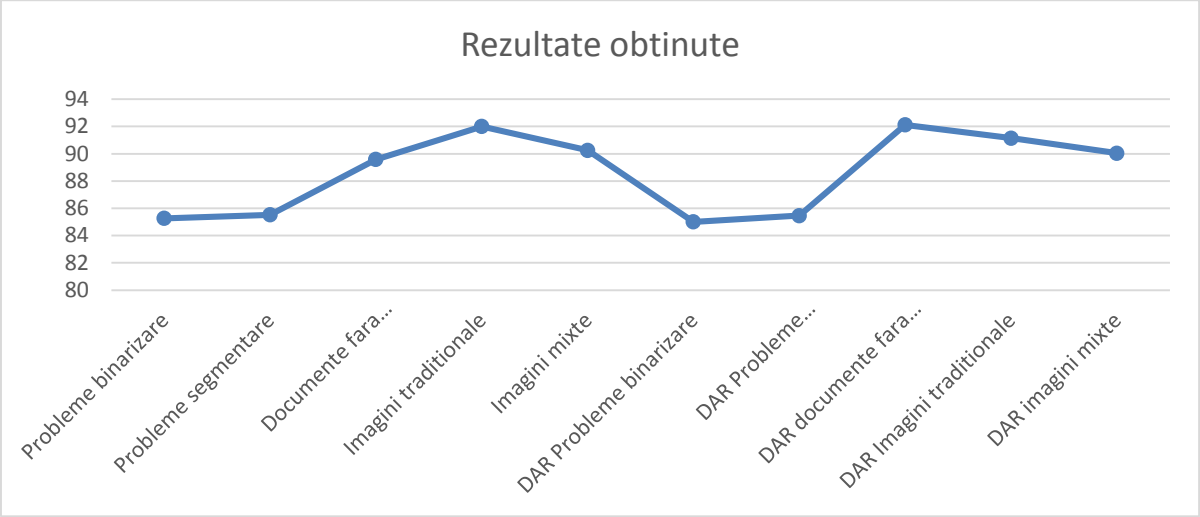


Figura 30. Fluctuații performanţă în diferite scenarii

În ceea ce privește procedeul de testare, respectiv performanţa sistemului, dorim să menționăm următoarele:

- Fiecare etapă de antrenare a unui clasificator bazat pe rețele neuronale produce de fapt un clasificator nou, datorită stării inițiale aleatoare a ponderilor rețelelor. Prin urmare, fluctuațiile mici de performanţă pot fi cauzate de aceste diferențe de stări interne. Pentru a exemplifica, chiar dacă sistemul ar fi fost antrenat de două, sau de mai multe ori doar cu imagini tradiționale este posibil să fi obținut creșteri de 0.05% în cadrul problemelor de segmentare;
- Performanţa maximă a sistemului pare că stagnează la un platou de 91%-92%, indiferent de tipul de date oferite. În cazul documentelor fără probleme de segmentare, sau de binarizare am obținut rezultate maxime de 92%, ca şi în cazul imaginilor tradiționale. Atunci când datele sunt mixte, performanţa scade undeva în jurul valorii de 90%;
- Există o mare problemă în ceea ce privește obținerea imaginilor de antrenare, respectiv test în zona DAR, în special în contextul segmentării de imagini. Din această cauză, nu am putut efectua teste în care să variem procentul de imagini DAR în cadrul procesului de antrenare.

4.2.3 Procesul de extragere a imaginilor asemănătoare

Problema CBIR presupune interacțiunea unui utilizator cu un modul *software*, în scopul obținerii unor imagini similare, sau conforme cu un criteriu de căutare. Acest proces implică mai multe etape, după cum urmează:

- Interogarea – în cadrul acestei etape utilizatorul trebuie să ofere sistemului CBIR un set de informații care să ajute la identificarea imaginilor ţintă;
- Determinarea noțiunii de *similar*;
- Oferirea rezultatului – problemele din cadrul acestei etape sunt de obicei legate de mașina de calcul şi interfața grafică.

Așa cum am menționat anterior, lucrarea de faţă îşi propune implementarea unui sistem în care interogarea se efectuează cu ajutorul unei imagini (tradiționale, sau DAR), iar procentul de similitudine este extras în urma analizei unor descriptori. Deoarece metrica de distanţă este de fapt stabilită de clasificator în sine (care determină procentul de apartenenţă la un set de clase), deocamdată nu s-a acordat o importanţă deosebită acestei etape.

Având în vedere caracteristicile tehnice ale sistemului, s-a impus folosirea unui motor de baze de date relaționale. Aplicația se dorește a fi ușor portabilă între diferite arhitecturi, aşadar a trebuit ca plaja de opțiuni să includă elemente care rulează atât pe Windows, cât şi

pe distribuții Linux/Unix. Comunitatea *open source* şi nu numai recomandă două mari variante – MySQL, respectiv PostgreSQL. În procesul de alegere între cele două opțiuni am studiat caracteristicile fiecărui motor, respectiv performanţele fiecăruia în funcție de necesităţile noastre şi ne-am hotarât asupra combinatiei MySQL/MyISAM.

Am considerat că nu este necesară inserarea imaginilor în baza de date, ci numai calea din sistemul de fișiere. Acest lucru se traduce într-o dimensiune fizică redusă a bazei de date. De asemenea, structura bazei de date nu este foarte complicată, fiind necesară o serie de 11 tabele, după cum urmează:

- IMAGE – pentru a centraliza toate informațiile legate de o imagine (inclusiv calea către fișier, tipul imaginii, valorile de aparetenenţă ale descriptorilor etc.);
- RGB, RGBH, c1c2c3, l1l2l3 – pentru spaţiile de culoare;
- LBP – pentru texturi;
- SIFT, HOG pentru descriptorii locali;
- CLASS – pentru a determina procentul de apartenenţă la diferite clase;
- MajorityVoting – pentru a stabili ponderile modulelor de vot majoritar;
- NN_LAYER – pentru a stoca straturile rețelei neuronale.

Într-o implementare ulterioară vom introduce încă o serie de tabele, printre care:

- TEXT_KEYWORD – în care sa păstrăm textul extras din documente, respectiv cuvintele cheie;
- USER – pentru a putea oferi posibilitatea de a administra accesul diferitor utilizatori la aplicaţie;
- USER_PREFERENCE – în care să salvăm preferințele legate de diferite aspecte ale sistemului (ponderile modulelor de vot majoritar, eliminarea unor caracteristici din procesul de clasificare etc.).

Atragem atenția asupra faptului că lipsa modulului de management al utilizatorilor provoacă o serie de probleme la nivelul tabelei *MajorityVoting* – de fiecare dată când un utilizator aduce modificări legate de ponderile clasificatorului din interfața grafică, acestea devin publice pentru restul utilizatorilor. După adăugarea tabelei de stocare a preferințelor, tabela *MajorityVoting* va fi înlocuită de combinația de mai jos:

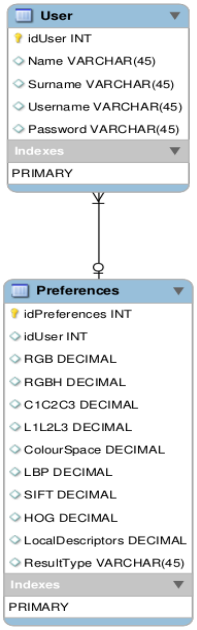


Figura 31. Tabele pentru stocarea preferințelor utilizatorului

Chiar şi după adăugarea acestor tabele, avem de-a face cu o arhitectură de dimensiuni reduse, care în corelație cu numărul relativ redus de intrări nu ar trebui să ridice probleme niciunui motor de stocare.

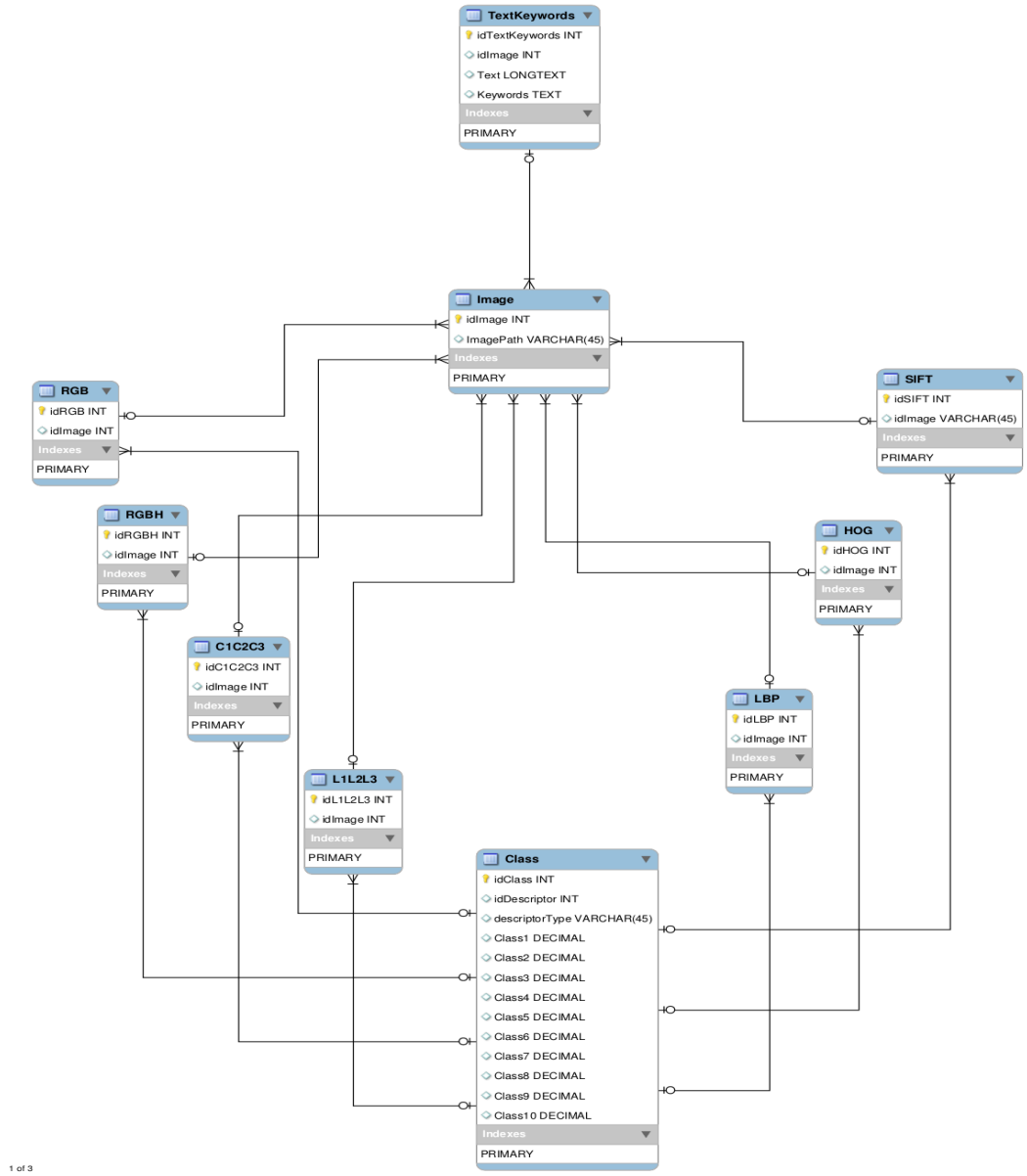


Figura 32. Structura bazei de date

Tabela *TextKeywords* nu este folosită în prezent; implementările ulterioare intenționează popularea ei cu informațiile extrase cu ajutorul modulului de OCR în cazul imaginilor de tip document.

4.2.3.1 Determinarea mulțimii rezultat

În urma procesului de clasificare sunt obţinute procentele de apartenenţă la diferitele clase determinate de-a lungul procesului de antrenare. Prin urmare, pornim de la următoarele notații:

$$C = \{c[1:N]\}, multimea\ claselor\ determinate\ in\ decursul\ procesului\ de\ antrenare$$
$$I = \{[p_a[1:N], path]\},$$
$$multimea\ tuturor\ imaginilor\ folosite\ in\ decursul\ procesului\ de\ antrenare$$
$$P = \{p_c[1:N]\}, multimea\ procentelor\ de\ apartenenta\ ale\ unei\ imagini\ de\ test$$

Formula 22: Notații folosite în cadrul procesului de determinare a mulțimii rezultat

Există două modalităţi principale de a determina mulțimea rezultat:

- Absolută - se consideră indexul P.idx(max(P)) şi se ignoră celelalte variante, ca fiind elemente de zgomot; mulțimea rezultat va conține primele k elemente ale clasei, după ce a fost în faza incipientă ordonată în funcție de scorul fiecărei imagini:

$$M_a = I.sort(p_a[\mathrm{P.idx(max(P))}])[0:k]$$

Formula 23: Varianta de rezultat absolut

- Relativă – în această situație se definește o funcție de tipul RMSE şi se urmărește minimizarea ei. Această abordare este utilă în cazul în care vectorul P conține elemente apropiate în special în partea superioară. De exemplu, pentru un vector P=[45, 44, 10, ...], clasificatorul absolut ar considera ca fiind valide numai elementele primei clase, ceea ce poate fi eronat în unele cazuri. Prin urmare, definim mulțimea rezultat astfel:

$$RMSE(\hat{\theta}) = \sqrt{(\theta - \hat{\theta})^2}, \text{ sau } RMSE(\hat{\theta}) = \frac{1}{N}\sqrt{\sum_{i=1}^{N}(y_i - \hat{y_i})^2}$$

Formula 24: Funcția RMSE

$$M_r = I.sort(RMSE(P), reverse)[0:k]$$

Formula 25: Varianta de rezultat relativ

Interfața grafică oferă utilizatorului posibilitatea de a alege între cele două modalităţi de a determina mulțimea rezultat. Prezentăm în imaginea de mai jos un exemplu de rezultat relativ în urma interogării clasificatorului cu o imagine care conţinea un cal:

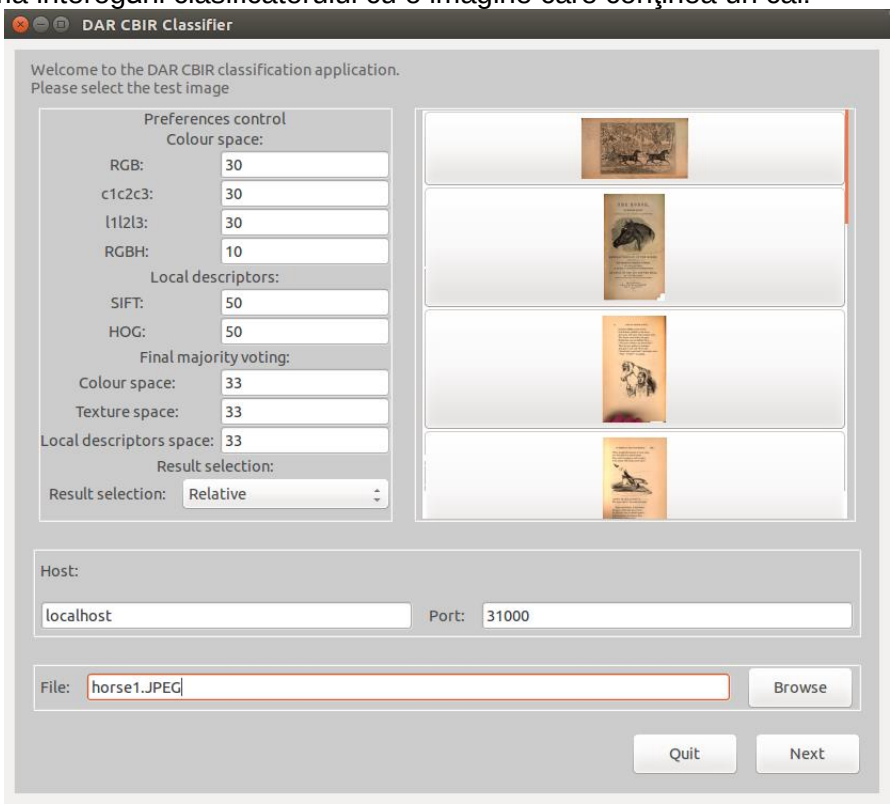


Figura 33. Rezultat interogare relativă

Nu putem fi siguri de corectitudinea rezultatelor afișate, deoarece interfața nu a fost expusă unei comunităţi de utilizatori, care să noteze relevanţa acestora. Reamintim aici importanţa factorului uman în procesul de clasificare, respectiv importanţa existentei unui modul care să fie responsabil de memorarea preferințelor fiecărui utilizator. Chiar dacă rezultatele obţinute depăşesc o acuratețe de 90% atunci când folosim acest clasificator, acestea nu se vor mula în același procent pe dorințele utilizatorului.

5 Concluzii, contribuții şi direcții viitoare de cercetare

În cadrul acestui capitol vom face o trecere în revistă a modului în care a fost structurat studiul abordat, concluziile, contribuțiile şi viitoarele direcții de cercetare.

5.1 Concluzii

Domeniul *computer vision* prezintă o complexitate deosebită şi implică folosirea unor metode de a achiziționa, procesa, analiza şi înțelege imaginile şi în general seturi de date în spații dimensionale ridicate, în vederea obținerii unui reprezentări numerice, sau simbolice. Subdomeniile cu care se află în tangenţă sunt la rândul lor foarte variate şi de o complexitate ridicată.

Capitolul 1 descrie tipurile de interogări, de domenii, dar mai ales importanţa factorului uman în determinarea unor rezultate care să fie acceptate ca fiind corecte de către un utilizator oarecare. Sub-capitolele 1.5 -1.10 descriu spaţiile din care se pot extrage descriptori globali şi locali în vederea descrierii unei anumite imagini. Fiecare sub-capitol punctează taxonomii, descrie abordări posibile şi implementări practice de actualitate. De asemenea, în final prezentăm un set de concluzii şi motivăm alegerile făcute în vederea finalizării sistemului CBIR propus.

Ne îndreptăm ulterior atenția către procesarea documentelor, în vederea segmentării/extragerii de imagini (capitolul 2). Sunt prezentate problemele generale din sfera DAR, abordările curente, respectiv etapele principale implicate în analiza imaginilor de tip document. Sunt prezentate atât tehnicile tradiționale de binarizare şi segmentare, cât şi cele recente, în cadrul cărora apar noțiuni de inteligenţă artificială.

În capitolul 3 prezentăm un algoritm nou de segmentare de imagini din documente, precum şi o modalitate nouă de a compara rezultatele unui algoritm de segmentare de imagini cu cele ale unuia care vizează segmentarea de text. Această etapă este necesară pentru încadrarea rezultatelor algoritmului în contextul DAR.

Capitolul 4 descrie structura şi implementarea sistemului CBIR rezultat, care ţine cont de mai multe seturi de vectori de caracteristici, din diferite spatii de interes.

În ceea ce privește contextul general al procesării imaginilor - există multe provocări, cele mai importante fiind:

- posibilitatea de a reprezenta caracteristicile unei imagini într-o modalitate ușor de asimilat şi procesat de către o mașină de calcul;
- determinarea sensului şi caracteristicilor logice ale unei imagini. În această zonă trebuie găsit un echilibru între asocieri corecte şi eronate, pentru că așa cum am arătat anterior, atât creierul, cât şi mașina de calcul pot eșua în a cataloga corect o imagine;
- pentru sistemele care trebuie sa rezolve probleme în timp real există restricții majore în ceea ce privește resursele alocate, respectiv timpul de calcul.

În aceste condiții, este evidentă necesitatea includerii unor **sisteme inteligente** în toate etapele procesării de imagini.

Indiferent de tehnicile folosite, niciun algoritm de până acum nu a obținut rezultate de 100%, chiar în contextul zonei pentru care a fost proiectat. De obicei, schimbarea setului de date produce scăderi majore de acuratețe. În aceste condiții, soluția descrisă mai sus propune extragerea mai multor seturi de descriptori şi ponderarea lor în cadrul determinării rezultatului final printr-un modul de vot majoritar după ce au fost clasificați anterior de către o serie de rețele neuronale. În momentul de faţă implementarea este completă, dar sunt în continuare probleme în partea de ajustare a rețelelor neuronale.

Ca şi categorie aparte în cadrul procesării de imagini, analiza documentelor ridică o altă serie de probleme. În general, acestea sunt cauzate de starea documentelor în momentul înregistrării lor, respectiv de poziția, sau iluminarea paginii. Majoritatea cercetătorilor activează preponderent în zona OCR, a binarizării, sau a segmentării paginii în blocuri de text, coloane, propoziții, cuvinte şi litere. Algoritmii de segmentare a imaginilor dintr-un document sunt foarte puțini.

Algoritmii de binarizare sunt foarte importanți, deoarece stau la baza oricărei alte implementări ulterioare, ca şi etapă de preprocesare a documentului. Există o mare varietate de algoritmi de binarizare, meniți să funcționeze în diferite ipostaze, în funcție de tipul de documente analizat. Cei mai noi algoritmi introduc elemente de inteligența artificială, atât din zona învăţării supervizate, cât şi din zona învăţării nesupervizate.

Lucrarea de faţă este interesată de segmentarea imaginilor; în această zonă există foarte puține soluții disponibile. Algoritmul CVSEG filtrează textul din document, iar ulterior elimină din zonele invalide prin calcularea parametrilor de conectivitate. Rezultatele obţinute sunt superioare algoritmului lui Bloomberg. În această zonă se dorește îmbunătăţirea etapei de filtrare a textului şi eventual a etapei de *cluster*-izare, datorită problemelor întâlnite în cadrul procesării documentelor de calitate scăzută. O altă posibilă îmbunătăţire poate viza timpul de execuție, care în momentul de faţă este inferior algoritmului lui Bloomberg.

5.2 Contribuții

Toate motoarele de CBIR implementate până în prezent au fost construite vizând clasificarea imaginilor provenite dintr-o zonă asemănătoare cu cea din care au provenit datele de antrenare. Aceste abordări sunt de multe ori orientate către anumite zone din contextul CBIR şi nu țintesc implementarea un sistem de clasificare general, sau polivalent. În acest context, imaginile folosite pentru antrenare şi testare provin din domenii de nişă, iar modalitatea de a extrage descriptori este de obicei orientată către problemele specifice fiecărei zone de interes. Toți acești factori conduc la scăderi drastice ale performanţei sistemelor atunci când sunt expuse (incluzând etapa de antrenare) unor imagini noi, din alte zone de interes.

Lucrarea de faţă descrie principiile de funcționare, implementarea, respectiv rezultatele obţinute ale unui sistem mixt, capabil să clasifice atât imagini tradiționale, cât şi imagini extrase din documente. În cele ce urmează, vom descrie sub-problemele întâlnite, modul de rezolvare al lor şi contribuțiile în zonele CBIR, respectiv DAR.

În **zona CBIR**, sistemul urmărește principiile clasice de funcționare ale unui modul de învăţare supervizată şi folosește ca element de bază rețele neuronale de tip *feed-forward*, cu *back-propagation*. Structura sistemului însă este particulară şi inovativă, deoarece se folosește de mai mulți clasificatori, care sunt ulterior analizați de două elemente de vot majoritar. Modulul este antrenat cu ajutorul unor imagini tradiționale, color, pentru ca ulterior să fie testat atât pe imagini similare, cât şi pe imagini extrase din documente, atât color, cât şi *gray scale*. Nu am mai întâlnit acest tip de clasificare în zona CBIR, atât în ceea ce privește diferența nivelului de culoare dintre imaginile de antrenare şi cele de test, cât şi referitor la zona de provenienţă a imaginilor.

În cadrul etapei de antrenare, în faza incipientă s-a încercat o implementare tradiţională, cu un singur clasificator; rezultatele au fost similare cu cele ale altor cercetători, dar așa cum am menționat anterior, performanţele au scăzut considerabil odată cu schimbarea imaginilor de test, ceea ce a condus la adăugarea de noi elemente, pentru a creşte performanţa şi robustețea sistemului. Prin urmare, arhitectura finală include vectori de caracteristici extrași din 7 zone de interes, pentru a rafina procesul de clasificare. În faza incipientă sunt extrase 4 seturi de caracteristici din spațiul culorilor - RGB tradițional, histograme RGB, c1c2c3 şi l1l2l3.

Relevanţa acestor caracteristici este ponderată în funcție de relevanţă, pentru ca apoi rezultatele să fie expuse unui modul de vot majoritar. În acest stadiu, performanţele sistemului de clasificare au crescut în ceea ce privește recunoașterea imaginilor din setul de

date incipient, dar nu au apărut îmbunătăţiri substanțiale atunci când au fost folosite imagini noi. Prin urmare, ne-am îndreptat atenția către alte spaţii de descriptori, respectiv de textură şi locali. Au fost introduși 3 clasificatori noi, ale căror rezultate, împreună cu cele din spațiul de culoare au fost ponderate de un alt bloc de vot majoritar, care oferă clasificarea finală. În aceste condiții, performanţa a crescut la 92%, pe imagini tradiționale, color. Considerăm această **arhitectură** ca fiind o contribuție în zona CBIR, deoarece în afara structurii unice produce rezultate deosebite atât în ceea ce privește testarea pe imagini similare, cât şi pe imagini noi.

Ulterior, sistemului i-a fost adăugat un modul de procesare de documente, pentru a analiza care sunt performanţele clasificatorului descris anterior atunci când este expus unor astfel de date. **Domeniul DAR** ridică o altă serie de sub-probleme, în special în zona segmentării de imagini – binarizare, segmentare propriu zisă, respectiv încadrarea rezultatelor.

În cadrul etapei de **binarizare**, am testat şi evaluat performanţa celor mai folosiți algoritmi moderni, dar şi a algoritmilor clasici, de complexitate redusă. Setul de date folosit a inclus copii ale unor documente problematice, deteriorate, dar şi conversii de calitate excelentă ale unor documente PDF. A fost analizat impactul fiecărui algoritm asupra clasificatorului şi au fost aleși 2 din setul inițial de 6 algoritmi pentru a valida ulterior algoritmul de segmentare. Considerăm acest proces de **validare a algoritmilor de binarizare** ca fiind o contribuție pentru domeniile CBIR şi DAR.

Ulterior am implementat şi validat un **algoritm de segmentare de imagini** nou, bazat pe criterii de varianța, *clustering* şi reconstrucție, intitulat **CVSEG**, cu rezultate de peste 84%. Testele efectuate pe acest algoritm confirmă robustețea lui, în diferite scenarii, care implică variația imaginilor de test, expunerea la toate problemele specifice DAR, respectiv variația algoritmilor de binarizare, analizându-se ulterior impactul asupra procesului de clasificare final, descris anterior.

O altă problemă din cadrul etapei de segmentare de imagini este reprezentată de încadrarea rezultatelor; această dificultate este determinată de numărul scăzut de algoritmi. Prin urmare, am implementat şi validat un **algoritm responsabil de analiza şi compararea** atât a rezultatelor segmentării de imagini, cât şi ale segmentării de text. Acest algoritm a fost validat prin teste directe, pe imagini variate, dar şi prin teste negative, pentru a determina robustețea algoritmului. Performanţele obţinute sunt de peste 95% pentru imagine, respectiv de peste 97% pentru text. Ulterior, procesul de validare a algoritmului CVSEG a fost definitivat după cum urmează:

- performanţe cu 10% mai bune decât algoritmul de segmentare de imagini al lui Bloomberg, unul dintre puţinii algoritmi de acest gen;
- prin suprapunerea rezultatelor cu cele ale unui algoritm de segmentare de text, prin aplicarea algoritmului de comparare. Performanţele au fost în medie de peste 85%, cu maxime de peste 91%.

Considerăm atât acești doi algoritmi, cât şi procesul de validare în sine ca fiind contribuții în cadrul domeniului DAR, datorită caracterului inovativ al abordării.

După finalizarea etapei de segmentare de imagini am continuat procesul de testare al clasificatorului, prin introducerea de documente. Performanţele au scăzut, dar nu drastic, ajungând la o medie de peste 89%. Au fost analizate circumstanțele care au determinat această scădere de performanţă în cadrul procesului de clasificare; concluzia la care am ajuns este că zona cea mai sensibilă este reprezentată de binarizarea documentului, în special determinarea pragului corect. În ceea ce privește segmentarea documentului, aceasta poate omite anumite zone dintr-o imagine, sau adăuga zone adiacente, dar rezultatul final al procesului de clasificare nu este afectat în majoritatea cazurilor.

5.3 Direcții viitoare de cercetare

În viitor ne vom îndrepta atenția în aceeași zonă, în vederea continuării cercetării întreprinse până acum. Sistemul implementat până acum poate fi îmbunătăţit în mai multe

zone, pornind de la interacțiunea cu utilizatorul, până la valorile *precision* şi *recall*, sau în ceea ce privește timpii de răspuns. Vom detalia în cele ce urmează problemele existente, respectiv îmbunătăţirile care pot fi aduse:

1. Interacțiunea cu utilizatorul
   - în faza incipientă, *design*-ul sistemului per ansamblu presupunea existenţa unei entităţi responsabile de partea de ***user management***. Acest lucru este crucial nu numai în ceea ce privește problemele de securitate, ci mai ales pentru înregistrarea unui anumit profil asociat utilizatorului. În acest fel, sistemul poate oferi rezultate mai apropiate de criteriile de căutare ale fiecărui utilizator;
   - **obținerea unui scor din partea utilizatorului** şi stocarea lui în baza de date, pentru a putea pondera pe viitor rezultatele oferite de clasificator. Desigur, această îmbunătăţire ar putea fi corelată foarte bine cu elementul de *profiling* descris anterior;
   - **sincronizarea bazelor de date** de la diferiți utilizatori – în acest fel putem centraliza, analiza şi extrage descrieri statistice ale opțiunilor utilizatorilor din diferite zone de interes. Acest proces este foarte important în rezolvarea problemei factorului uman;
   - rezolvarea unor limitări existente în momentul curent în interfața grafică (şi în majoritatea interfețelor grafice din zona CBIR, în general). În momentul de faţă, utilizatorului i se oferă doar posibilitatea de a interoga modulul cu o imagine, urmând a primi ca răspuns un set de imagini corespunzătoare. Eventualele îmbunătăţiri pot include:
     - **adăugarea posibilității de a efectua interogări textuale** – această ţintă este foarte greu de atins, datorită componentei afective a interogării. În faza incipientă vom corela cuvintele care compun interogarea cu denumirile claselor şi cu textul extras din imagini după rularea unor algoritmi OCR (în special în cazul imaginilor de tip document);
     - **adăugarea posibilității de a descrie o imagine**. De exemplu, în lipsa unei imagini potrivite, utilizatorul va primi rezultate nesatisfăcătoare. Acest sub-modul trebuie să ofere o interfață grafică minimală prin care utilizatorul să poată creiona elementele care i se par reprezentative şi de a indica una sau mai multe clase ţintă (o zonă cu albastru+o zonă cu galben+clasa pisici ar putea avea ca rezultat setul de imagini care conțin pisici aflate pe plaja). Trebuie să menționăm că atât în acest caz, cât şi în cel anterior, răspunsurile oferite de sistem vor fi mai mult orientate către căutare în spațiul text, respectiv către procesarea elementară de imagini şi vor fi mai puțin legate de inteligenţa artificială;
   - desigur, **interfața grafică** folosită în acest moment a fost implementată doar pentru a servi unor scopuri minimale, menite testării şi evaluării sistemului. Există două modalităţi prin care aceasta poate fi îmbunătăţită:
     - pornind de la cea existentă, prin adăugarea de noi funcţionalităţi – cu dezavantajul că fiecare utilizator va depinde de modulul de client, respectiv de bibliotecile aferente;
     - renunțarea la interfața curentă şi implementarea uneia web, accesibilă prin *browser* – cu dezavantajele evidente ale problemelor de compatibilitate;
2. Îmbunătăţirea preciziei sistemului
   - adăugarea unui **modul de detecție a umbrelor şi a zonelor de highlight**, pentru a pondera dinamic rezultatele clasificării în spațiul culorilor. În momentul de faţă, sistemul favorizează spatiile c1c2c3 şi I1I2I3, chiar dacă în unele imagini ar trebui folosit spațiul tradițional, RGB. După cum am arătat anterior, aceste transformări au tendința de a aplatiza anumite zone, atunci când nu sunt folosite corespunzător;

- adăugarea **descriptorilor de formă** şi investigarea rezultatelor obţinute. În forma lui actuală, sistemul folosește descriptori din spațiul culorilor, spațiul texturilor, respectiv două forme de descriptori locali. S-a evitat folosirea descriptorilor de formă, deoarece sunt foarte sensibili la anumite modificări ale obiectului. Cu toate acestea, urmărim cu deosebit interes rezultatele studiului domnului profesor Jan Koenderink, angrenat în dezvoltarea unor descriptori invarianți;
- **despărțirea obiectelor ţintă în elemente reprezentative** şi adăugarea unor sub-module care analizează conectivitatea lor. De exemplu, în cazul unei persoane, se pot detecta membrele, trunchiul şi capul, pentru ca apoi să se calculeze un meta-descriptor pentru modul în care sunt conectate. Unul din avantajele secundare ale acestei abordări este obținerea de detalii suplimentare, legate de starea obiectului (static, în mișcare, faţă, profil etc.);
- introducerea de module fuzzy în locul blocurilor de vot majoritar. Desigur, acest lucru nu garantează o creștere de performanţă, mai ales în condițiile în care ponderile pot fi modificate dinamic, dar nu ne putem pronunţa în lipsa experimentelor;
- una din problemele majore ale rețelelor neuronale este reprezentată de incapacitatea acestora de a sesiza modificările de scală. Ca efect secundar, dezvoltatorul are două opțiuni pentru ca rezultatele unui asemenea clasificator să aibă performanţe ridicate:
  - **expandarea setului de date de antrenare** în așa fel încât să includă exemple (imagini) cu obiectul ţintă scalat diferit;
  - **adăugarea unor tehnici de căutare în imagine**, de genul *sliding window*;
- în starea lui curentă, submodulul care analizează imaginea într-un oarecare spațiu al culorilor este compus din 3 rețele neuronale, asociate celor 3 canale de culoare. Pe viitor, intenţionăm să studiem cum sunt afectate rezultatele clasificării după ce adăugăm elemente de preprocesare, care să ţină cont de relevanţa pixelului în contextul întregii imagini. Una din posibilităţi ar putea fi **normalizarea imaginii pe fiecare canal de culoare**;
- investigarea performanţelor oferite de alți **descriptori locali** - ne referim aici în primul rând la LESH şi SURF, două abordări noi, care par să ofere rezultate superioare tehnicilor bazate pe HOG şi SIFT;

3. Îmbunătăţiri aduse modulului de procesare de documente
   - algoritmul de segmentare poate fi îmbunătăţit în **zona filtrării preliminare a textului**, prin folosirea unei tehnici mai precise, deoarece proiectarea pe axe poate fi sensibilă la deviații de unghi;
   - **invalidarea *cluster*-elor** poate fi îmbunătăţită de asemenea, prin adăugarea unui modul OCR, care să elimine caracterele de dimensiuni atipice;
   - **folosirea unui modul OCR** care să extragă cuvinte cheie din documente. Aceste cuvinte pot fi asociate documentului şi categoriei determinate în urma procesului de clasificare;
4. Îmbunătăţiri legate de performanţa timpilor de calcul
   - în momentul de faţă, sistemul complet include un număr foarte mare de elemente care prezintă o complexitate ridicată de calcul (15 rețele neuronale, calcul trigonometric, calcul diferențial, interacțiuni cu baza de date etc.). Din acest motiv, timpii de antrenare a clasificatorului depășesc frecvent 3 zile (pe o mașină de calcul cu procesor Xeon quad-core, 12GB RAM şi SCSI); clasificarea propriu zisă este efectuată rapid, deocamdată. Desigur, una din variantele imediate de îmbunătăţire a performanţelor poate fi achiziționarea unei **mașini de calcul mai performante**;
   - o altă soluție la problema complexității de calcul ar fi folosirea unor **sisteme distribuite** în rețea;

- **aproximarea funcţiilor trigonometrice** cu altele mai simple – de obicei aceste abordări duc la scăderi de performanţă;
- pentru o imagine color de 640x480 pixeli există 3 canale de culoare, ceea ce duce la un total de 921600 de valori, care trebuie procesate, numai pentru un singur spațiu de culoare. Pentru imagini cu conținut variat, în același context, numărul de descriptori SIFT poate depăși suma de 100000, fiecare din ei conținând peste 130 de valori reale. În aceste condiții, este evidentă introducerea unui modul de **reducere a dimensionalităţii datelor** (literatura ştiinţifică recomandă abordări de tip VP);
- nu în ultimul rând, întregul sistem a fost dezvoltat în python, care introduce cu siguranţă un *overhead*; acesta ar putea fi eliminat prin **folosirea unor limbaje de programare mai eficiente** (C, sau C++). De asemenea, în momentul de faţă, aplicația folosește un motor de stocare a datelor învechit, deoarece am considerat că numărul de intrări în baza de date, respectiv structura sa nu necesită abordări complexe. Este posibil ca odată cu creșterea setului de date de antrenare şi trecerea la versiuni mai noi să înlocuim abordarea curentă cu una tranzacțională, mai eficientă;

5. Adăugarea de alte scenarii de folosire
   - clasificarea de documente, ca fiind colecții de imagini şi cuvinte cheie. Acest tip de scenariu ar permite utilizatorului să acceseze rapid toate documentele asemănătoare cu un criteriu anume. Exemple de astfel de interogări ar putea fi:
     - *afișează documentele care sunt legate de pisici* - în acest caz, sistemul ar putea căuta în tabela de cuvinte cheie asociate documentelor procesate anterior şi în eventuala tabelă corespunzătoare clasei (dacă există);
     - *afișează documentele care sunt corespund acestei imagini* - în acest caz, imaginea ar fi clasificată, iar răspunsul ar cuprinde toate documentele care au eticheta clasei descoperite;
     - *afișează documentele care sunt asemănătoare cu acest document* - această situație le include de fapt pe cele două menționate anterior, iar criteriul de stabilire a unei ierarhii ar putea fi bazat pe minimizarea erorii pătratice, de exemplu.

Listă lucrări publicate şi prezentate

1. **Mihai ILIE**, Luminita DUMITRIU, Daniel ONOSE, 2011, A survey on content based image retrieval approaches, presented at The First PhD Student Symposium 7th-8th December 2011, ”Dunarea de Jos” University of Galati;
2. Daniel ONOSE, Luminita DUMITRIU, **Mihai ILIE**, 2011, Current approaches in the cloud computing field, presented at The First PhD Student Symposium 7th-8th December 2011 ”Dunarea de Jos” University of Galati;
3. **Mihai ILIE**, Daniel ONOSE, 2012, Document segmentation for content based image retrieval, presented at The Second PhD Student Symposium 13th-14th December 2012, "Dunărea de Jos" University of Galati;
4. Daniel ONOSE, **Mihai ILIE**, 2012, Adaptive virtual machine provisioning techniques in cloud computing, presented at The Second PhD Student Symposium 13th-14th December 2012, "Dunărea de Jos" University of Galati;
5. **Mihai ILIE**, 2012, Content based image retrieval in document characterization, presented at the International Computer Vision Summer School, Sicily, Italy;
6. **Mihai ILIE**, 2013, A survey on image processing techniques in the context of content based image retrieval, Annals of Dunarea de Jos - EEAI, ”Dunarea de Jos” University of Galati;
7. **Mihai ILIE**, 2014, A content Based Image Retrieval Approach Based on Document Queries, The 2014 International Conference on Image Processing, Computer Vision, & Pattern Recognition (IPCV), Las Vegas, Nevada, USA;
8. **Mihai ILIE**, 2014, Document image segmentation through clustering and connectivity analysis, The 9th International Conference on "Multimedia & Network Information Systems" (MISSI), Wroclaw, Poland;
9. **Mihai ILIE**, 2014, A content Based Image Retrieval Approach Based on Document Queries (extended version), Emerging Trends in Image Processing, Computer Vision, and Pattern Recognition, Elsevier Inc., 2014

Bibliografie

1. **Sebe, N.** Feature extraction & content description - DELOS - MUSCLE Summer School on Multimedia digital libraries, Machine learning and cross-modal technologies for access and retrieval. *www.videolectures.net.* [Interactiv] 25 02 2007. www.videolectures.net/dmss06_sebe_fecd/.
2. *Fuzzy color histogram-based video segmentation .* **Onur Küçüktunç, Ugur Güdükbay, Özgür Ulusoy.** 2010, Computer Vision and Image Understanding .
3. **Hui Yu, Mingjing Li, Hong-Jiang Zhang, Jufu Feng.** *Color Texture Moments for Content-Based Image Retrieval.* s.l. : Proc. IEEE Intl Conf. on Image Processing, 2002. pg. 929-932.
4. **Vassilieva, Natalia.** RuSSIR - Russian Summer School in Information Retrieval . [Interactiv] 2012. http://videolectures.net/russir08_vassilieva_cbir/.
5. **Siddique, Sharmin.** *A Wavelet Based Technique for Analysis and Classification of Texture Images.* 2002.
6. *A survey of document image classification: problem statement, classifier architecture and performance evaluation.* **Nawei Chen, Dorothea Blostein.** 2007, International Journal of Document Analysis and Recognition (IJDAR), p. Volume 10.
7. **A. Antonacopoulos, D. Bridson, C. Papadopoulos.** ICDAR 2007 Page Segmentation Competition. *ICDAR.* [Interactiv] 2007. http://www.primaresearch.org/ICDAR2007_competition/.
8. **Wikipedia.** F1 score. *www.wikipedia.org.* [Interactiv] 15 03 2013. http://en.wikipedia.org/wiki/F1_score.
9. *Multiresolution morphological approach to document image analysis.* **Bloomberg, D. S.** Proc. Int. Conf. Document Analysis and Recognition.
10. **D.S., Bloomberg.** libleptonica. *http://www.ubuntuupdates.org/package/core/precise/universe/base/libleptonica.* 06 03 2012.
11. **Google.** Ocropus. *http://code.google.com/p/ocropus/.* 21 04 2013.
12. **Homepage, Computer Vision.** Computer Vision Homepage. *Computer Vision Homepage.* [Interactiv] 30 June 2005. http://www.cs.cmu.edu/afs/cs/project/cil/www/v-images.html.
13. *Histograms of Oriented Gradients for Human Detection.* **Navneet Dalal, Bill Triggs.** 2005, International Conference on Computer Vision \& Pattern Recognition, pg. 886-893.
14. **Krizhevsky, Alex.** *Learning Multiple Layers of Features from Tiny Images.* 2009.
15. **MLCOMP.** http://mlcomp.org/datasets. *MLcomp.* [Interactiv] 2014. http://mlcomp.org/datasets.
16. **Irvine, UC.** http://archive.ics.uci.edu/ml/. *UCI Machine Laerning Repository.* [Interactiv] 2014. http://archive.ics.uci.edu/ml/.
17. *Image Search Engines - An Overview.* **Gevers, Th.** 2004, S. B. G. Medioni, Emerging Topics in Computer Vision. Prentice Hall.
18. **Lotto, Beau.** Optical illusions show how we see. *www.ted.com.* [Interactiv] 7 2009. http://www.ted.com/talks/beau_lotto_optical_illusions_show_how_we_see.html.
19. **Sebe, Nicu.** Feature extraction & content description - DELOS - MUSCLE Summer School on Multimedia digital libraries, Machine learning and cross-modal technologies for access and retrieval . *www.videolectures.net.* [Interactiv] 25 02 2007. http://videolectures.net/dmss06_sebe_fecd/.
20. *Document image analysis: A primer.* **Rangachar Kasturi, Lawrence O'Gorman, Venu Govindaraju.**
21. *Improved Document Image Segmentation Algorithm using Multiresolution Morphology.* **Syed Saqib Bukharia, Faisal Shafaitb, Thomas M. Breuela.**
22. *Segmentation of historical machine-printed documents using Adaptive Run Length Smoothing and skeleton segmentation paths.* **Nikos Nikolaou, Michael Makridis, Basilis Gatos, Nikolaos Stamatopoulos, Nikos Papamarkos.**

23. *Recursive X-Y Cut using Bounding Boxes of Connected Components.* **Jaekyu Ha, Robert M. Haralick.**
24. *A neuro-fuzzy technique for document binarisation.* **Papamarkos, Nikos.** 2003, Springer-Verlag.
25. *Adaptive document image binarization.* **J. Sauvola, M. Pietikainen.**
26. *Rule-based document structure understanding with a fuzzy combination of layout and textual features.* **Stefan Klink, Thomas Kieninger.** 2001, IJDAR.
27. *Document seal detection using GHT and character proximity graphs.* **Partha Pratim Roya, Corresponding author contact informationr, Umapada Pal, Josep Lladós.** 2011, Pattern Recognition, pg. 1282-1295.
28. *An adaptive local binarization method for document images based on a novel thresholding method and dynamic windows.* **Bilal Bataineh, Siti Norul Huda Sheikh Abdullah, Khairuddin Omar.** 2011, Pattern Recognition Letters, pg. 1805–1813.
29. **Wikipedia.** Histogram of oriented gradients. *Wikipedia.* [Interactiv] 2013. http://en.wikipedia.org/wiki/Histogram_of_oriented_gradients.
30. —. Scale invariant feature transform. *Wikipedia.* [Interactiv] 2013. http://en.wikipedia.org/wiki/Scale-invariant_feature_transform.
31. —. *Wikipedia.* [Interactiv] 2013. http://en.wikipedia.org/wiki/LESH.
32. *Head Pose Estimation in Face Recognition across Pose Scenarios.* **Sarfraz, Saquib.** Madeira, Portugal : s.n., 2008. Proceedings of VISAPP 2008, Int. conference on Computer Vision Theory and Applications. pg. 235-242.
33. *Survey over image thresholding techniques and quantitative performance evaluation.* **Mehmet Sezgin, Bulent Sankur.** January 2004, Journal of Electronic Imaging, pg. 146–165.
34. *An Evaluation Technique for Binarization Algorithms.* **Pavlos Stathis, Ergina Kavallieratou, Nikos Papamarkos.** 2008, Journal of Universal Computer Science,, pg. 3011-3030.
35. **Abderrahmane Kefali, Toufik Sari, Mokhtar Sellami.** *Evaluation of several binarization techniques for old Arabic documents images.* 2010.
36. *Adaptive thresholding methods for documents image binarization.* **Bilal Bataineh, Siti N. H. S. Abdullah, K. Omar, M. Faidzul.** Mexico City : s.n., 2011. MCPR'11 Proceedings of the Third Mexican conference on Pattern recognition . pg. 230-239.
37. **Niblack, W.** *An Introduction to Digitall Image Processing.* Englewood Cliffs : Prentice Hall, 1986.
38. *Adaptive Document Binarization.* **J. Sauvola, T. Seppanen, S.Haapakoski, M. Pietikainen.** Ulm, Germany : s.n., 1997. International Conference On Document Analysis and Recognition. pg. 147-152.
39. *Comparison of Niblack inspired binarization methods for ancient documents.* **Khurshid, K., Siddiqi, I., Faure, C., Vincent, N.** 2009. DRR. pg. 1-10.
40. *An intelligent character recognizer for Telugu scripts using multiresolution .* **Arun K. Pujaria, C. Dhanunjaya Naidub, M. Sreenivasa Raoc, B.C. Jinagad.** 2004, Image and Vision Computing, pg. 1221–1227.
41. *A multi-scale framework for adaptive binarization of degraded document images.* **Reza Farrahi Moghaddam, Mohamed Cheriet.** 2010, Pattern Recognition, pg. 2186-2198.
42. *Automatic extraction of titles from general documents using machine learning.* **Yunhua Hua, Hang Lib, Yunbo Caob, Li Tengc, Dmitriy Meyerzond, Qinghua Zhenga.** 2006, Information Processing & Management, pg. 1276-1293.
43. *Towards an omnilingual word retrieval system for ancient manuscripts.* **Yann Leydiera, Asma Oujia, Frank LeBourgeoisa, Hubert Emptoza.** 2009, Pattern Recognition, pg. 2089-2105.
44. *Similarity-based training set acquisition for continuous handwriting recognition.* **Jerzy Sas, Urszula Markowska-Kaczmar.** 2012, Information Sciences, pg. 226-244.
45. *Contextual Swarm-Based Multi-layered Lattices: A New Architecture for Contextual Pattern Recognition.* **David G. Elliman, Sherin M. Youssef.** 2004, Document Analysis Systems, pg. 496-507.

46. *An investigation of the modified direction feature for cursive character recognition.* **Michael Blumenstein, Xin Yu Liu, Brijesh Verma.** 2007, Pattern Recognition, pg. 376-388.
47. *A model-based recognition engine for sketched diagrams.* **Florian Brieler, Mark Minas.** 2010, Journal of Visual Languages & Computing, pg. 81-97.
48. *Dynamically constructed bayes nets for multi-domain sketch understandin.* **C. Alvarado, R. Davis.** 2005, IJCAI, pg. 1407-1412.
49. *Cali: An online scribble recognizer for calligraphic interface.* **M. J. Fonseca, C. Pimentel, J. A. Jorge.** 2002, AAAI Sprin Symposium - Sketch Understanding, pg. 51-58.
50. *Hmm-based efficient sketch recognition.* **M. Sezgin, R. Davis.** s.l. : ACM Press, 2005. Proceedings of the International Conference on Intelligent User Interfaces.
51. *A multi-scale framework for adaptive binarization of degraded document images.* **Reza Farrahi Moghaddam, Mohamed Cheriet.** 2010, Pattern Recognition, pg. 2186-2198.
52. *A novel approach for structural feature extraction: Contour vs. direction.* **Brijesh Verma, Michael Blumenstein, Moumita Ghosh.** 2004, Pattern Recognition Letters, pg. 975–988.
53. *A pictorial dictionary for printed Farsi subwords.* **Afshin Ebrahimi, Ehsanollah Kabir.** 2008, Pattern Recognition Letters, pg. 656-663.
54. *A focused crawler with document segmentation.* **Yang, JY, Kang, JB și Choi, JM.** 2005. INTELLIGENT DATA ENGINEERING AND AUTOMATED LEARNING IDEAL. pg. 94-101.
55. *An unsupervised method for joint information extraction and feature mining across different Web.* **Tak-Lam Wong, Wai Lam.** 2009, Data & Knowledge Engineering, pg. 107-125.
56. *An automatic approach for efficient text segmentation.* **Cai, KK, și alții.** KNOWLEDGE-BASED INTELLIGENT INFORMATION AND ENGINEERING SYSTEMS, pg. 417-424.
57. **Wikipedia.** Entropy (information theory). *Wikipedia.* [Interactiv] 5 July 2013. http://en.wikipedia.org/wiki/Entropy_%28information_theory%29.
58. *Extraction and segmentation of tables from Chinese ink documents based on a matrix model.* **Xi-wen Zhanga, Michael R. Lyu, Guo-zhong Dai.** 2007, Pattern Recognition, pg. 1855-1867.
59. *Segmentation of mixed Chinese/English documents based on Chinese radicals recognition and complexity analysis in local segment pattern.* **Xia, Y, și alții.** 2006, INTELLIGENT COMPUTING IN SIGNAL PROCESSING AND PATTERN RECOGNITION, pg. 497-506.
60. *Use of aggregation pheromone density for image segmentation.* **Susmita Ghosh, Megha Kothari, Anindya Halder, Ashish Ghosh.** 2009, Pattern Recognition Letters, pg. 939-949.
61. *Intelligent region-based thresholding for color document images with highlighted regions.* **Tsai, Chun-Ming.** 2012, Pattern Recognition, pg. 1341-1362.
62. *Adaptive thresholding algorithm: Efficient computation technique based on intelligent block detection for degraded document images.* **Yu-Ting Pai, Yi-Fan Chang, Shanq-Jang Ruan.** 2010, Pattern Recognition, pg. 3177-3187.
63. *Parallel Implementation of Souvola's Binarization Approach on GPU.* **Singh, Brij Mohan.** 2011, International journal of computer applications, p. 28.
64. *Preprocessing of Low-Quality Handwritten Documents Using Markov Random Fields .* **Cao, H and Govindaraju, V.** 2009, IEEE TRANSACTIONS ON PATTERN ANALYSIS AND MACHINE INTELLIGENCE, pg. 1184-1194.
65. *Analysis and recognition of highly degraded printed characters.* **Anna Tonazzini, Stefano Vezzosi, Luigi Bedini.** 2003, Document Analysis and Recognition, pg. 236-247.
66. **Wang, Shengjiu.** *A Robust CBIR Approach Using Local Color Histograms.* Department of Computer Science, University of Alberta, Edmonton, Alberta, Canada : s.n., 2001.
67. **Wikipedia.** Local binary patterns. *www.wikipedia.org.* [Interactiv] 08 11 2011. http://en.wikipedia.org/wiki/Local_binary_patterns.
68. **Coelho, Luis Pedro.** Mahotas: Image Processing in Python. *www.luispedro.org.* [Interactiv] 2010. http://luispedro.org/software/mahotas.
69. **Wikipedia.** Kurtosis. *Wikipedia.* [Interactiv] 2012. http://en.wikipedia.org/wiki/Kurtosis.

70. —. High order moments. *Wikipedia.* [Interactiv] 2012. http://en.wikipedia.org/wiki/Moment_%28mathematics%29#Higher_moments.
71. **Th. Gevers, A.W.M. Smeulders.** Image Search Engines - An Overview. [autorul cărții] S. B. Kang G. Medioni. *Emerging Topics in Computer Vision.* s.l. : Prentice Hall, 2004.
72. **Wikipedia.** Cumulative histogram. [Interactiv] 2012. http://en.wikipedia.org/wiki/Histogram#Cumulative_histogram.
73. **Hall-Beyer, Mryka.** The GLCM Tutorial Home Page. *GLCM Texture Tutorial.* [Interactiv] 2007. http://www.fp.ucalgary.ca/mhallbey/tutorial.htm.
74. **Barbeau Jerome, Vignes-Lebbe Regine, and Stamon Georges.** *A Signature based on Delaunay Graph and Co-occurrence Matrix.* Laboratoire Informatique et Systematique, University of Paris, Paris, France : s.n., 2002.
75. **Centre for Image Analysis, Uppsala, Sweden.** "Texture," class notes for Computerized Image Analysis MN2. [Interactiv] 2002. http://www.cb.uu.se/~ingela/Teaching/ImageAnalysis/Texture2002.pdf.
76. **Robert M. Haralick, K Shanmugam and Its'Hak Dinstein.** *Textural Features for Image Classification.* s.l. : IEEE Transactions on Systems, Man, and Cybernetics, 1979.
77. *Comparison Study of Textural Descriptors for Training Neural Network Classifiers.* **G. D. Magoulas, S. A. Karkanis, D. A. Karras and M. N. Vrahatis.** s.l. : Proceedings of the 3rd IEEE-IMACS World Multi-conference on Circuits, Systems, Communications and Computers, Athens, Greece, 1999.
78. **Techasith, Pravi.** *Image Search Engine.* Imperial College, London, UK : s.n., 2002.
79. **Johansson, Bjorn.** QBIC (Query By Image Content). *QBIC (Query By Image Content).* [Interactiv] 2002. http://www.isy.liu.se/cvl/Projects/VISIT-bjojo/survey/surveyonCBIR/node26.html.
80. **Wikipedia.** Wavelets. *Wikipedia.* [Interactiv] 2012. http://en.wikipedia.org/wiki/Wavelet.
81. —. Gabor Filter. *Wikipedia.* [Interactiv] 2012. http://en.wikipedia.org/wiki/Gabor_filter.
82. —. Independent component analysis. *Wikipedia.* [Interactiv] 2012. http://en.wikipedia.org/wiki/Independent_component_analysis.
83. *Representation of images for classification with independent features.* **H. Borgne, A. Guerin-Dugue, A. Antoniadis.** 2004, Pattern Recognition Letters.
84. **Paul Over, George Awad, Martial Michel, Jonathan Fiscus, Wessel Kraaij, Alan F. Smeaton, Georges Quéenot.** *TRECVID 2011 -- An Overview of the Goals, Tasks, Data, Evaluation Mechanisms and Metrics.* 2012.
85. **ImageCLEF.** ImageCLEF. *ImageCLEF.* [Interactiv] 2009. http://www.imageclef.org/publications.
86. *Robust texture features for still image retrieva.* **P. Howarth, S. Rüger.** 2006, Proc. IEE Vis. Image Signal Processing.
87. *Independent Component Analysis of Textures in Angiography Images. Computational Imaging and Vision.* **Snitkowska, E. Kasprzak, W.** 2006, pg. 367-372.
88. **Gui-Song Xia, Julie Delon, Yann Gousseau.** Shape-based Invariant Texture Indexing. *Shape-based Invariant Texture Indexing.* [Interactiv] 2008. http://perso.telecom-paristech.fr/~xia/texture.html.
89. **Z., Yang G.** *Computer Vision Lecture Notes: Fourier Methods.*
90. *Shape measures for content based image retrieval: a comparison. Inf. Processing and Management.* **Mehtre B. M., Kankanhalli M. S., Lee W. F.** 1997.
91. **Wikipedia.** Canny edge detector. *Wikipedia.* [Interactiv] 2012. http://en.wikipedia.org/wiki/Canny_edge_detector.
92. —. Thresholding. *Wikipedia.* [Interactiv] 2012. http://en.wikipedia.org/wiki/Thresholding_%28image_processing%29.
93. —. Otsu's method. *Wikipedia.* [Interactiv] 2012. http://en.wikipedia.org/wiki/Otsu%27s_Method.
94. **Scholarpedia.** Scale Invariant Feature Transform. *Scholarpedia.* [Interactiv] 2012. http://www.scholarpedia.org/article/Scale_Invariant_Feature_Transform.
95. **Wikipedia.** Scale-invariant feature transform . *Wikipedia.* [Interactiv] 2012. http://en.wikipedia.org/wiki/Scale-invariant_feature_transform#David_Lowe.27s_method.

96. —. SURF. *Wikipedia.* [Interactiv] 2012. http://en.wikipedia.org/wiki/SURF.
97. —. Summed area table . *Wikipedia.* [Interactiv] 2012. http://en.wikipedia.org/wiki/Integral_image.
98. **BoofCV.** Performance: SURF. *BoofCV.* [Interactiv] 2012. http://boofcv.org/index.php?title=Performance:SURF.
99. **Krystian Mikolajczyk, Cordelia Schmid.** A Performance Evaluation of Local Descriptors. *IEEE transactions on pattern analysis and machine intelligence.* 2005.
100. **Herbert Bay, Tinne Tuytelaars, Luc Van Gool.** *Speeded-Up Robust Features (SURF).* Zurich, Leuven, Belgia : s.n., 2008.
101. **Wikipedia.** Histogram of oriented gradients . *Wikipedia.* [Interactiv] 2012. http://en.wikipedia.org/wiki/Histogram_of_oriented_gradients.
102. —. LESH. *Wikipedia.* [Interactiv] 2010. http://en.wikipedia.org/wiki/LESH.
103. *Head Pose Estimation in Face Recognition across Pose Scenarios.* **Saquib Sarfraz, Olaf Hellwich.** Madeira, Portugal : s.n., 2008. Proceedings of VISAPP 2008, Int. conference on Computer Vision Theory and Applications. pg. 235-242.
104. **Herbert Bay, Andreas Ess, Tinne Tuytelaars, Luc Van Gool.** *Speeded-Up Robust Features (SURF).* 2006.
105. **Wikipedia.** R-tree. *wikipedia.org.* [Interactiv] 01 March 2012. http://en.wikipedia.org/wiki/R-tree.
106. —. Quadtree. *www.wikipedia.org.* [Interactiv] 23 03 2012. http://en.wikipedia.org/wiki/Quadtree.
107. —. k-d tree. *www.wikipedia.org.* [Interactiv] 07 03 2012. http://en.wikipedia.org/wiki/K-d_tree.
108. —. vp-tree. *www.wikipedia.org.* [Interactiv] 17 01 2012. http://en.wikipedia.org/wiki/Vp-tree.
109. *VP-tree: Content-Based Image Indexing.* **Markov, Il'ya.** Moscow, Saint-Petersburg : s.n., 2007. The Fourth Spring Young Researchers Colloquium on Databases and Information Systems.
110. *Dynamic vp-tree indexing for n-nearest neighbor search.* **Ada Wai-chee Fu, Polly Mei-shuen Chan, Yin-Ling Cheung, Yiu Sang Moon.** 2000, The VLDB Journal, pg. 154–173.
111. **Wikipedia.** Locality-sensitive hashing . *www.wikipedia.org.* [Interactiv] 13 03 2012. http://en.wikipedia.org/wiki/Locality-sensitive_hashing.
112. —. Hash function. *www.wikipedia.org.* [Interactiv] 27 03 2012. http://en.wikipedia.org/wiki/Hash_Function.
113. *Human Detection and Identification by Robots Using Thermal and Visual Information in Domestic Environments.* **Mauricio Correa, Gabriel Hermosilla, Rodrigo Verschae, Javier Ruiz-del-Solar.** 2011, Springer Science+Business Media.
114. *Learning Novel Objects for Extended Mobile Manipulation.* **Tomoaki Nakamura, Komei Sugiura, Takayuki Nagai, Naoto Iwahashi, Tomoki Toda, Hiroyuki Okada, Takashi Omori.** 2012, Springer Science+Business Media B.V.
115. *Johnny: An Autonomous Service Robot for Domestic Environments.* **Thomas Breuer, Geovanny R. Giorgana Macedo, Ronny Hartanto, Nico Hochgeschwender, Dirk Holz, Frederik Hegger, Zha Jin, Christian Müller, Jan Paulus, Michael Reckhaus, José Antonio Álvarez Ruiz, Paul G. Plöger, Gerhard K. Kraetzschmar.** 2012, Springer Science+Business Media B.V.
116. *Commute time guided transformation for feature extraction.* **Yue Deng, Qionghai Dai, Ruiping Wang, Zengke Zhang.** 2012, Computer Vision and Image Understanding.
117. *Range map superresolution-inpainting, and reconstruction from sparse data.* **Arnav V. Bhavsar, Ambasamudram N. Rajagopalan.** 2012, Computer Vision and Image Understanding.
118. *Illumination invariant extraction for face recognition using neighboring wavelet coefficients.* **X. Cao, W. Shen, L.G. Yu, Y.L. Wang, J.Y. Yang, Z.W. Zhang.** 2012, Pattern Recognition.
119. *Indoor Mobile Robotics at Grima, PUC .* **Luis Caro, Javier Correa, Pablo Espinace, Daniel Langdon, Daniel Maturana, Ruben Mitnik, Sebastian Montabone, Stefan**

Pszczółkowski, Anita Araneda, Domingo Mery, Miguel Torres, Alvaro Soto. 2012, Journal of Intelligent & Robotic Systems.
120. *Content-based image retrieval approach for biometric security using colour, texture and shape features controlled by fuzzy heuristics.* **Kashif Iqbal, Michael O. Odetayo, Anne James.** 2012, Journal of Computer and System Sciences.
121. *Local and global features based image retrieval system using orthogonal radial moments.* **Chandan Singh, Pooja.** 2012, Optics and Lasers in Engineering.
122. *Cytoplasm and nucleus segmentation in cervical smear images using Radiating GVF Snake.* **Kuan Li, Zhi Lu, Wenyin Liu, Jianping Yin.** 2012, Pattern Recognition.
123. *Local maximum edge binary patterns: A new descriptor for image retrieval and object tracking.* **M. Subrahmanyam, R.P. Maheshwari, R. Balasubramanian.** 2012, Signal Processing.
124. *Pill-ID: Matching and retrieval of drug pill images.* **Young-Beom Lee, Unsang Park, Anil K. Jain, Seong-Whan Lee.** 2012, Pattern Recognition Letters.
125. *Discriminative compact pyramids for object and scene recognition.* **Noha M. Elfiky, Fahad Shahbaz Khan, Joost van de Weijer, Jordi Gonzalez.** 2012, Pattern Recognition.
126. *Feature fusion within local region using localized maximum-margin learning for scene categorization.* **Jianzhao Qin, Nelson H.C. Yung.** 2012, Pattern Recognition.
127. *Discriminative features for image classification and retrieval.* **Shang Liu, Xiao BaiCorresponding author contact information.** 2012, Pattern Recognition Letters.
128. *An application of swarm intelligence to distributed image retrieval.* **David Picard, Arnaud Revel, Matthieu Cord.** 2012, Information Sciences.
129. **80 million tiny images. *http://groups.csail.mit.edu/vision/TinyImages/.* [Interactiv] [Citat:] http://groups.csail.mit.edu/vision/TinyImages/.**
130. Google, Inc. Tesseract. *http://en.wikipedia.org/wiki/Tesseract_%28software%29.* 2013.