UNIVERSITATEA TEHNICĂ A MOLDOVEI

Cu titlu de manuscris

C. Z. U: 004.93

RUSU MARIANA

ANALIZĂ ŞI RECUNOAŞTERE DE FORME PENTRU APLICAŢII CU IMAGINI DIGITALE

122.03 – MODELARE, METODE MATEMATICE, PRODUSE PROGRAM

Teză de doctor în informatică

Conducător ştiinţific UTM:

conf.univ., dr.**Vasile MORARU** _____

Conducător ştiinţific UT "Gh.Asachi":

acad., prof.univ., dr. **Horia-Nicolai TEODORESCU**[1] _____

Autorul: Mariana RUSU _____

CHIŞINĂU, 2018

1. Exclusiv pe durata a două stagii de pregatire susţinute prin burse « E. Ionescu » oferite de Guvernul României, sub egida Agenţiei Universitare a Francofoniei (AUF)

MULŢUMIRI

Studiile doctorale au fost realizate în cotutelă, la Universitatea Tehnică a Moldovei şi Universitatea Tehnică „Gh. Asachi” din Iaşi, având 2 coordonatori ştiinţifici şi beneficiind de mai multe burse: Bursa "Eugen Ionescu" oferită de Guvernul României, sub egida Agenţiei Universitare a Francofoniei (AUF), anul de studii 2012-2013 (5 luni), Bursa de excelenţă a Guvernului pentru doctoranzi pe anul 2013 (10 luni) şi Bursa "Eugen Ionescu" oferită de Guvernul României, anul de studii 2013-2014 pe o perioadă de 4 luni. Vreau să mulţumesc instituţiilor enumerate mai sus – persoanelor direct implicate – pentru şansa acordată, pentru ocazia de a realiza activitatea de cercetare, pentru încrederea lor şi stimulul oferit la momentul oportun.

Vreau să mulţumesc coordonatorului ştiinţific domnului **Vasile Moraru**, conf., dr. la Universitatea Tehnică a Moldovei, pentru ghidarea pe parcursul anilor de studii doctorale şi implicarea dumnealui în procesul de cercetare şi perfectare a tezei.

De asemenea vreau să mulţumesc şi coordonatorului ştiinţific de la Universitatea Tehnică „Gh. Asachi” din Iaşi domnului **Horia-Nicolai Teodorescu**, profesor şi director fondator al centrului de excelenţă. Vreau să-i mulţumesc în primul rând pentru gazdă, în al doilea rând pentru încrederea care mi-a acordat-o acceptând acordul de cotutelă. Ca urmare o parte din teză reprezintă descrierea activităţilor de cercetare realizate în timpul studiilor doctorale în incinta Universităţii Tehnice „Gh. Asachi” din Iaşi, mai exact în laboratorul de Inginerie Biomedicală şi Sisteme Inteligente de la Facultatea Electronică, Telecomunicaţii şi Tehnologia Informaţiei a acestei universităţi. Doresc să-mi exprim recunoştinţa pentru disponibilitatea dumnealui, sfaturile şi recomandările privind domeniul cercetării, pentru bunătatea şi răbdarea de care dumnealui a dat dovadă.

Mulţumiri călduroase colegilor de facultate pentru primele remarci şi sfaturi - domnilor **Mihail Kulev**, dr., conf. univ., **Sergiu Răilean**, dr., conf. univ., **Valentin Negură**, dr., conf. univ., şi **Nicolae Secrieru**, dr., conf. univ., **Dumitru Ciorbă**, dr. conf. univ., şi **Viorel Rusu**, lector universitar.

CUPRINS

ADNOTARE

la teza „Analiză şi recunoaştere de forme pentru aplicaţii cu imagini digitale”, prezentată de către Rusu Mariana pentru conferirea gradului ştiinţific de doctor în informatică, Chişinău 2018

Teza de doctor este structurată pe 3 capitole, urmate de bibliografie din 188 titluri şi 4 anexe. Lucrarea conţine 60 figuri, 25 tabele, text de bază pe 116 pagini. Rezultatele obţinute sunt publicate în 10 lucrări ştiinţifice.

Cuvinte cheie: procesare imagini, GMM, GUMM, recunoaştere forme, coeficient de corelaţie, template matching, clasificare automată.

Domeniul de cercetare: analiza şi recunoaşterea formelor în aplicaţii cu imagini digitale.

Scopul lucrării constă în elaborarea unor metode, algoritmi ce ar permite recunoaşterea şi clasificarea automată a formelor/obiectelor din imagini digitale.

Obiectivele lucrării: implementarea şi elaborarea algoritmilor de procesare a imaginilor în scopul recunoaşterii şi clasificării formele în aplicaţii cu imagini digitale şi realizarea unei comparaţii obiective a algoritmilor implementaţi.

Noutatea şi originalitatea ştiinţifică: Au fost identificate şi argumentate: un algoritm euristic de segmentare, o metodă de separare liniară a două mulţimi de date şi o metodă hibridă de recunoaştere şi clasificare a formelor.

Problema ştiinţifică şi de cercetare soluţionată constă în elaborarea unei metode de segmentare bazată pe histogramă ce nu necesită indicarea numărului de praguri. S-a descris un model matematic de separare liniară a două mulţimi de date. S-a elaborat un sistem de clasificare automată care uneşte diverşi algoritmi din inteligenţa artificială (algoritm genetic, sistem fuzzy) şi defineşte combinaţia particulară care poate oferi o mai bună soluţie pentru clasificarea automată a formelor/obiectelor.

Semnificaţia teoretică şi valoarea aplicativă a lucrării constă în elaborarea algoritmului de segmentare bazat pe GUMM, a modelului de separare liniară, a metodei de clasificare automată bazat pe logica fuzzy şi algoritmi de identificare/recunoaştere a imaginilor bazaţi pe coeficientul de corelaţie. Algoritmul de segmentare propus poate fi implementat şi la alte tipuri de imagine, la fel şi metoda hibridă formată din CCN şi AG. Metodele de recunoaştere automată propuse pot fi readaptate şi la alte seturi de forme.

ABSTRACT

to thesis „Analysis and pattern recognition for applications with digital images", presented by Rusu Mariana for conferring a PhD Degree in Computer Science, Chisinau, 2018.

The thesis is divided into three chapters, followed by bibliography of 188 titles and 4 appendices. The paper contains 60 figures, 25 tables, 116 pages of basic text. The number of published papers on the topic of thesis is 10.

Keywords: image processing, GMM, G-U-MM, forms recognition, correlation, template matching, automatic classification.

Field of research is image processing in order to pattern/objects recognition and detection in the image.

The purpose of this paper is to develop methods, algorithms that would allow the recognition and automatic classification of forms / objects in digital images.

The objectives are to analyze, implement and develop image processing algorithms for recognition of forms for digital images applications and make an objective comparison of implemented algorithms.

Scientific novelty and originality of the results: there were identified and justified an heuristic segmentation algorithm, a linear separation method for two data sets and a hybrid method of automatic classification of forms / objects.

The theoretical importance consists in the development of a histogram-based segmentation method that does not require to indicate the number of thresholds. A mathematical model of linear separation of two sets data has been described. An automatic classification system has been developed that combines different algorithms from artificial intelligence (genetic algorithm, fuzzy system) and defines the particular combination that can provide a better solution for templates / objects classification.

The applied value of the thesis: it demonstrated the practical effectiveness of of the proposed segmentation algorithm and the advantage of the hybrid method CCN + AG against the application of individual CCN. Also has been shown effectiveness and advantages of automatic sorting system.

The research results can be applied to automatic classification of waste, this area was chosen to emphasize that them correctly sorting would be a solution for reducing pollution. The system can be adapted and developed to classify other objects.

АННОТАЦИЯ

диссертации на соискание ученой степени доктора наук в информатики

„Анализ и распознавание форм для приложений с цифровыми изображениями”,

автор: Русу Мариана, Кишинэу, 2018.

Диссертация состоит из 3 глав, а также последующей биографией содержащей 188 названий и 4 приложения. Диссертация содержит 60 фигур, 25 таблиц, главное текстовое содержание – 116 страниц. Количество опубликованных работ на данную тему – 10.

Ключевые слова: обработка изображений, сегментация изображений, распознавание форм, идентификация подписи, извлечение ключевых точек, сопоставление шаблонов, GMM, G-U-MM, автоматическая классификация.

Область исследования – обработка изображения с целью распознавания форм.

Цель работы состоит в разработке способов и алгоритмов для распознавания и автоматической классификации предметов/форм из изображения.

Задачи работы: анализ, проектирование и разработка алгоритмов обработки изображений с целью распознавания форм для приложений с цифровыми изображениями.

Инновационность и оригинальность работы: были идентифицированы и аргументированы эвристический алгоритм сегментации, метод линейного разделения двух наборов данных, а также гибридный метод автоматической классификации форм/ объектов.

Теоретическая значимость заключается в разработке метода сегментации на основе гистограммы, который не требует указания количества порогов и гибридный метод автоматической системы классификации, который объединяет различные алгоритмы из искусственного интеллекта (генетический алгоритм, нечеткая система) и определяет конкретную комбинацию которая может обеспечить лучшее решение для автоматической классификации форм / объектов.

Практическая значимость работы было доказано эффективность предлагаемого алгоритма сегментации, преимущество гибридного метода CCN+AG в сравнении с индивидуальным применением CCN. Также было продемонстрирована эффективность и преимущества системы автоматической классификации.

Результаты исследования могут быть использованы для автоматической сортировки отходов. Таким образом, правильное применение результатов исследования для сортировки отходов может снизить загрязнение среды. Разработанная система может быть приспособлена и для классификации других объектов.

LISTA ABREVIERILOR

AG – Algoritm Genetic

ASIFT – Affine Scale-Invariant Feature

CCN –Coeficient de Corelaţie Normalizat

d.s.p. – densitate spectrală de putere

DU – Distribuţie Uniformă

ECG – ElectroCardioGrama

EEG – ElectroEncefaloGrama

FCM – Fuzzy C-Means

FN – False Negative (Fals Negativ)

FNR – False Negative Rate (Rata Fals Negativ)

FP – False Positive (Fals Pozitiv)

FPM – Fuzzy Pattern Matching

FPR – False Positive Rate (Rata Fals Pozitiv)

GMM – Gaussian Mixture Models (MMG –Modele de Mixturi Gaussiene)

G-U-MM – Gaussian and Uniform Mixed Models (MMGU–Modele de Mixturi Gaussiene şi Uniforme)

IR – InfraRoşu

k-NN – k-Nearest Neighbor (k- cei mai apropiaţi vecini)

LSB – Least Significant Bit

MLP – Multi-Layer Perceptron (multistrat)

MSB – Most Significant Bit

NLP – Natural Language Processing (Procesarea Limbajului Natural)

OCR – Optical Character Recognition (Recunoaşterea optică a caracterelor)

PCA – Principal Component Analysis (Analiza Componentelor Principale)

PDF – Funcţie Densitate de Probabilitate

PET – PolyEthylene Terephthalate (PoliEtilen Tereftalat)

PVC – PolyVinyl Chloride (PoliClorura de Vinil)

RdF – Recunoaştere de Forme

RGB – Red, Green, Blue (Roşu, Verde, Albastru)

RNA – Reţele Neuronale Artificiale

RNR – Reţea Neuronală Recurentă

SIFT – Scale-Invariant Feature Transform

SNR – Signal Nose Rapport (Raportul semnal zgomot)

SSD – Sisteme Suport de Decizie

SVM – Support Vector Machine

TF/IDF – Term Frequency / Inverse Document Frequency (frecvenţa unui termen (în document)/ invers frecvenţa documentului)

TN – True Negative (Adevărat Negativ)

TNR – True Negative Rate (Rata Adevărat Negativ)

TP – True Positive (Adevărat Pozitiv)

TPR – True Positive Rate (Rata Adevărat Pozitiv)

UV – UltraViolet

INTRODUCERE

Actualitatea temei este confirmată de numărul crescător de lucrări, proiecte, aplicaţii în domeniul procesării de imagini cu scopul interpretării automate a acestora: asistarea unui diagnostic medical; recunoaşterea amprentelor, vocii, retinei şi a persoanelor (pentru securitate); dirijări şi evitări de rachete, submarine, roboţi etc. în servicii militare; observarea şi preîntâmpinarea cataclismelor naturale bazate pe imaginile captate de sateliţi; recunoaşterea scrisului de mână la serviciile poştale pentru gruparea scrisorilor etc.

Tendinţa spre robotizare a apărut din necesitatea suplinirii activităţii umane sau chiar înlocuirea ei. Un sistem dotat cu vedere artificială are multe avantaje:

- automatizează procesele intensive de muncă;
- facilitează luarea unor decizii, de ex. stabilirea unui diagnostic, sortarea obiectelor după numeroase criterii, etc. influenţând pozitiv dinamica procesului decizional;
- reduce implicarea umană;
- oferă rapiditate, obiectivitate, eficienţă, rentabilitate;
- activează non-invaziv şi non-destructiv;
- lucrează în spectre IR şi UV;
- lucrează în medii nefavorabile/neprielnice sau nocive.

Astăzi, un inginer ştie să doteze o maşină cu capacităţi de învăţare. Pentru a putea crea softuri complexe, care ar ”percepe” mediul înconjurător asemeni omului, noi trebuie să înţelegem şi să descifrăm cum are loc acest proces la nivel neuronal şi sub ce formă este stocată informaţia în memoria noastră. Se încearcă simularea acestui proces, dar ţinând cont de faptul că spaţiile de stocare ale calculatorului devin din ce în ce mai mari şi viteza de prelucrare a datelor creşte şi ea, nu este elaborată încă o metodă care să includă recunoaşterea mediului înconjurător cu toată complexitatea sa. Fiecare aplicaţie tratează domenii înguste.

Se cunosc aplicaţii bazate pe reţele neuronale artificiale, care sunt dotate cu capacitatea de a învăţa şi a sintetiza informaţiile astfel încât să poată da răspunsuri corecte pentru intrări diferite de cele cu care au fost antrenate (dar din acelaşi domeniu). Cu toate astea nu se cunoaşte încă o metodă care ar permite sistemului de vedere artificială să se adapteze la mediul înconjurător şi să studieze independent ceva nou.

Elaborarea unui sistem de recunoaştere a imaginilor implică cunoştinţe din matematică, inteligenţă artificială, bioinginerie, informatică etc. şi rămâne ca problemă deschisă în continuare.

Pentru aplicarea în practică a algoritmilor de recunoaştere s-a ales domeniul trierii deşeurilor. În Republica Moldova nu se respectă sortarea deşeurilor menajere şi dacă ar apărea investiţii referitor la incinerarea deşeurilor pentru producerea de energie nu ar risca nimeni, căci la noi se aruncă şi produse toxice, baterii, uleiuri uzate, care trebuie să fie puse în centrul de reciclare în recipiente adecvate, deci deşeurile trebuie să fie sortate în prealabil. Se mai poate obţine compost din deşeuri organice ce ar fi util şi benefic pentru mediu. Se cere de găsit o soluţie de sortare efectivă a deşeurilor.

Nu este aplicată până în prezent sortarea prin recunoaşterea obiectelor, dar sunt deşeuri care nu sunt colectate corect şi ar trebui eliminate de pe banda de sortare: containerele care au conţinut pesticide; containerele de uleiuri şi vopsele, solvenţi; containerele de ulei de motor, baterii, containere ce conţin uleiuri alimentare arse sau vechi etc. Aceste deşeuri sunt toxice pentru mediu şi sănătate, deci trebuiesc stocate separat. Ar fi bine să poată fi eliminate din grămada de deşeuri pentru reciclare specială.

S-a propus de a investiga şi a prezenta rezultatele actuale ale aplicării metodelor de identificare şi clasificare automată ale ambalajelor.

Scopul lucrării constă în elaborarea unor metode, algoritmi ce ar permite recunoaşterea formelor/obiectelor din imagine şi clasificarea lor automată, scop atins prin următoarele **obiective**:

- analiza etapelor de procesare a imaginilor în vederea extragerii informaţiei, recunoaşterii şi clasificării obiectelor din imagini;
- elaborarea unui algoritm de segmentare bazat pe G-U-MM;
- analiza principalelor metode de recunoaştere a formelor din scene mono-obiect;
- analiza metodelor fundamentale de clasificare automată;
- elaborarea unei metode de separare liniară a două mulţimi de date;
- elaborarea unui sistem de recunoaştere şi clasificare automată a formelor bazat pe logica fuzzy. Sistem ce include 3 paşi esenţiali:
    1) *recunoaşterea simbolurilor de pericol* prin aplicarea metodei hibride elaborate: corelaţie şi algoritm genetic. Aplicarea algoritmului genetic permite căutarea de noi poziţionări folosite de coeficientul de corelaţie la potrivirea imaginii cu şabloanele din baza de date;
    2) clasificarea formelor/obiectelor în 3 clase prin aplicarea logicii fuzzy;
    3) confirmarea deciziei luate la pasul anterior prin identificarea semnăturii de contur prin corelaţie.

– compararea rezultatelor obţinute la fiecare etapă a procesării imaginilor cu metodele de referinţă respective (segmentare, recunoaştere şi clasificare a formelor); concluzionarea eficacităţii metodelor elaborate.

Algoritmii propuşi în lucrare pot fi aplicaţi individual la o etapă de procesare a imaginilor sau ca sistem.

Noutatea ştiinţifică: Au fost identificate şi argumentate

– o metodă nouă de segmentare bazată pe modele uniforme şi gaussiene;
– un model matematic de separare liniară a două mulţimi de date;
– o metodă hibridă de recunoaştere (CNN+GA) şi clasificare a formelor (bazată pe logica fuzzy).

Problema ştiinţifică soluţionată constă în:

– elaborarea unei metode de segmentare bazată pe histogramă ce nu necesită indicarea numărului de praguri;
– propunerea unei metode de separare liniară a două mulţimi de date (descrierea modelului matematic);
– elaborarea unui sistem de clasificare automată elaborat ce uneşte diverşi algoritmi din inteligenţa artificială (algoritm genetic, sistem fuzzy) şi defineşte combinaţia particulară care poate oferi o mai bună soluţie pentru clasificarea automată a formelor/obiectelor.

Semnificaţia teoretică. Lucrarea dată conţine o analiză comparativă a metodelor fundamentale de clasificare automată şi de recunoaştere a formelor bazate pe descriptori locali; o descriere detaliată a algoritmului elaborat bazat pe G-U-MM, o evaluare obiectivă nesupervizată pe bază de metrici a rezultatelor obţinute la segmentare; o descriere a sistemului hibrid de recunoaştere şi o evaluare a rezultatelor obţinute la recunoaşterea imaginilor.

Valoarea aplicativă a lucrării. La etapa de extragere a informaţiei s-a propus un algoritm de segmentare bazat pe distribuţii Gaussiene și uniforme. Acest algoritm poate fi aplicat ca extragere de informaţie: perimetru, suprafaţă, valoare medie a pixelilor etc. pentru alţi algoritmi de recunoaştere bazaţi pe caracteristicile date de exemplu un sistem fuzzy de decizie. De asemenea, algoritmul este util la separarea obiectului de fundal.

Pentru clasificarea a două mulţimi de date s-a propus un algoritm de separare liniară. S-a prezentat o procedură efectivă de reducere a problemei considerate la rezolvarea unui sistem de ecuaţii prin reformularea condiţiilor de optimalitate Karush-Kunh-Tucker şi utilizând metoda Newton.

Pentru recunoaşterea formelor s-a elaborat un sistem bazat pe logica fuzzy cu un pas preliminar de recunoaştere a simbolului de pericol prin algoritm hibrid ce combină coeficientul de corelaţie cu algoritmul genetic şi un pas de verificare a corectitudinii clasificării prin identificarea semnăturii obiectului. Coeficientul de corelaţie determină potrivirea sau nu a două imagini, iar algoritmul genetic generează noi dimensiuni şi unghiuri de rotire ale imaginii căutate. Este un algoritm rapid şi poate fi aplicat şi pentru alte tipuri de imagine, având un randament de recunoaştere înalt. Identificarea semnăturii obiectului e bazată pe funcţia diferenţă. Sistemul de decizie fuzzy e bazat pe 3 atribute: formă, mărime şi simetrie. Sistemul fuzzy poate fi adaptat pentru clasificarea automată şi a altor obiecte.

Implementarea rezultatelor ştiinţifice. Algoritmul de segmentare elaborat este utilizat la separarea obiectelor de fundal, ca o etapă preliminară pentru recunoaşterea de forme într-un sistem de decizie bazat pe caracteristici.

Algoritmul hibrid bazat pe combinarea CCN şi GA a fost implementat la trierea deşeurilor, dar poate fi utilizat şi la alte tipuri de obiecte pentru recunoaştere, de ex. la găsirea logourilor de pe documente.

Metoda matematică de separare liniară propusă poate fi utilizată la clasificarea automată a obiectelor în două grupuri.

În dependenţă de obiectele ce necesită sortate, se pot schimba atributele ce sunt semnificative la clasificare şi sistemul propus bazat pe logica fuzzy se adaptează uşor la noi cerinţe.

Rezultatele ştiinţifice înaintate spre susţinere:

1. Algoritmul de segmentare bazat pe G-U-MM.
2. Metoda matematică de separare liniară a două mulţimi de date.
3. Metoda hibridă bazată pe coeficientul de corelaţie normalizat şi algoritm genetic.
4. Sistemul automat de clasificare bazat pe logica fuzzy.

Aprobarea rezultatelor cercetărilor. Analiza metodelor implementate, descrierea algoritmului elaborat şi unele rezultate prezentate în teză au fost publicate în reviste internaţionale şi naţionale:

- Computer Science Journal of Moldova, Vol.20, Nr.2(59), 2012, Vol.25, Nr.1(73), 2017;
- Proceedings of the Romanian Academy, Series A, Vol.14, No.1, 2013;
- Meridian Ingineresc (2 lucrări), No.2, 2013;
- Romanian Journal of Information Science and Technology, Vol.16, No.1, 2013.

De asemenea, rezultatele obţinute pe parcursul anilor de cercetare au fost prezentate la Conferinţe Internaţionale şi publicate în volumele Conferinţelor (două dintre care în BDI – IEEE):

- 4th International Conference Telecommunications, Electronics and Informatics, UTM, 2012.
- 2nd International Conference on Nanotechnologies and Biomedical Engineering, Chisinau, Republic of Moldova, 2013.
- 7th International Conference Electronics, Computers and Artificial Intelligence, IEEE Conference Publications, Bucharest – Romania, Vol.7, No.2, 2015.
- International Conference on E-Health and Bioengineering, IEEE Conference Publications, 2015.

Rezultatele finale au fost prezentate la simpozionul IIVA 2016 (Information in Image and Video Analysis Theory and Applications), Academia Română, Iasi – Romania şi publicate în revista:

- Computer Science Journal of Moldova, Vol.25, Nr.1(73), 2017.

Publicaţii la tema tezei. Sunt publicate 10 lucrări (dintre care 2 lucrări ISI şi 2 BDI) la tema tezei, aceste articole sunt parafrazate în conţinutul tezei, observându-se astfel importanţa teoretică şi valoarea aplicativă a lucrării.

Structura şi volumul lucrării.

Teza de doctor este structurată pe 3 capitole, urmate de bibliografie din 188 titluri şi 3 anexe. Lucrarea conţine 60 figuri, 25 tabele, text de bază – 116 pagini. Numărul de lucrări publicate la tema tezei este 10, la două dintre care sunt singur autor.

Cuvinte cheie: procesare imagini, segmentare imagini, GMM, G-U-MM, recunoaştere forme, keypoints matching, template matching, clasificare automată, sistem fuzzy, identificare semnătură.

Domeniul de cercetare este procesarea imaginilor în scopul recunoaşterii şi clasificării formelor/obiectelor din imagine.

Conţinutul tezei:

Lucrarea dată prezintă o descriere succintă a etapelor de procesare a imaginilor. În Capitolul 1 „ANALIZA SITUAŢIEI ÎN DOMENIUL PROCESĂRII IMAGINILOR DIGITALE CU SCOPUL RECUNOAŞTERII ŞI CLASIFICĂRII AUTOMATE” este prezentată o privire de ansamblu asupra domeniului de recunoaştere de forme şi clasificare automată a etapelor de analiză şi procesare a imaginilor în vederea clasificării automate: preprocesarea imaginilor,

extragerea atributelor/descriptorilor de imagine, analiza şi interpretarea rezultatelor. Din metodele de clasificare automată a imaginilor s-au tratat: clusterizarea automată a datelor, clasificatorul Bayes, SVM şi reţelele neuronale artificiale. S-a prezentat importanţa indicilor de performanţă în aprecierea metodelor de clasificare.

În Capitolul 2 „SEGMENTAREA IMAGINILOR FOLOSIND MODELE DE MIXTURI GAUSSIENE ŞI MODELE DE MIXTURI UNIFORME-GAUSSIENE" se descrie cadrul experimental dezvoltat pentru segmentarea imaginilor bazată pe modele de mixturi Gaussiene şi pe modele de mixturi uniforme – Gaussiene, cât şi avantajul implementării a ultimului model menţionat. Compararea cantitativă a rezultatelor e bazată pe implementarea a cinci metode de evaluare a segmentării.

Capitolul 3 „ALGORITMI DE RECUNOAŞTERE ŞI DE CLASIFICARE AUTOMATĂ A FORMELOR/OBIECTELOR ÎN APLICAŢII CU IMAGINI DIGITALE" conţine descrierea unor metode elaborate de recunoaştere a formelor şi rezultatele implementării lor.

A fost propusă şi implementată metoda hibridă ce uneşte coeficientul de corelaţie şi algoritmul genetic pentru identificarea unei imagini cu un şablon. Au fost testate metodele ce realizează corespondenţa între imagine şi şablon prin potrivirea punctelor de interes: SIFT, Affine SIFT şi au fost comparate rezultatele cu metoda hibridă (CCN+AG) elaborată.

Este descris şi implementat sistemul fuzzy propus pentru clasificare automată ce utilizează atributele: formă, mărime şi simetrie pentru a clasifica ambalajele în reciclabile, periculoase şi nedeterminate. Pentru confirmarea deciziei clasificării s-a utilizat metoda de identificare a semnăturii.

În „CONCLUZII GENERALE ŞI RECOMANDĂRI" se prezintă principalele contribuţii şi unele idei de continuare a cercetării.

1. ANALIZA SITUAŢIEI ÎN DOMENIUL PROCESĂRII IMAGINILOR DIGITALE CU SCOPUL RECUNOAŞTERII ŞI CLASIFICĂRII AUTOMATE

1.1 Privire de ansamblu asupra domeniului de recunoaştere de forme şi clasificare automată

Sistemele de calcul în prezent permit prelucrarea unui volum colosal de date ceea ce face posibilă analiza de imagini în timp real şi automatizarea unor procese tehnologice complexe.

Aplicaţiile ce includ analiză de imagine şi *Recunoaştere de Forme* (RdF) pot fi grupate în funcţie de problema ce trebuie rezolvată. Printre domeniile cele mai populare de aplicare a RdF se enumără serviciile militare, medicina, robotica, industria. Aplicaţiile sunt de diferită complexitate: de la controlul calităţii produselor până la aplicaţii militare şi de securitate:

1. Recunoaşterea semnelor de circulaţie [1-3];
2. Recunoaşterea documentelor, actelor de identitate, semnăturilor [4-6];
3. Recunoaşterea amprentelor digitale [7,8];
4. Detectarea feţelor [9,10], persoanelor [11-13], vehiculelor [14,15] etc.;
5. Imagistica medicală în scopul facilitării elaborării unui diagnostic [16-18];
6. Cartografia [19,20];
7. Analiza resurselor terestre (imagini captate de sateliţi) [21-23];
8. Aplicaţii în astronomie [24,25];
9. Gestionarea traficului aerian [26-28];
10. Aplicaţii militare: ghidare rachete [29,30], recunoaştere (aeriană [31-33], marină [34], submarină [35-38] etc.) detectarea mişcării [39];
11. Securitate (detectarea mişcării [40,41], recunoaşterea gesturilor [42-44], emoţiilor [44-46] etc.)
12. Controlul calităţii în industrie [47-50];
13. Sortarea produselor [51];
14. ș.a.

Printre cele mai uzuale aplicaţii de RdF se enumără recunoaşterea scrisului [52,53] şi a vocii [54]. Optical Character Recognition (OCR) reprezintă recunoaşterea caracterelor din textul imprimat. Chiar şi cele mai performante sisteme OCR necesită o calitate bună a documentelor imprimate/scanate. Se cunosc sisteme de recunoaştere a scrisului manual utilizate la serviciile poştale pentru a citi adresa destinatarului sau la serviciile sociale unde e necesar de elaborat o statistică pe baza rezultatelor din formularele completate etc.

La sistemele de prelucrare a vocii cel mai des întâlnim aplicaţii adaptate pentru o persoană anume (un singur interlocutor), dar e posibil de adaptat şi pentru un public mai larg (cu rezultate mai slabe). E posibil de tratat fiecare cuvânt pronunţat separat sau o discuţie continuă (va fi necesară segmentarea datelor în cuvinte). Sistemele de securitate utilizează vocea pentru a identifica proprietarul. Alte date biometrice precum amprentele digitale, retina, palma sau faţa sunt de asemenea utilizate în sistemele de securitate.

Detectarea mişcării braţelor sau a emoţiilor persoanelor ce ar indica un comportament agitat sunt utile la păstrarea ordinii publice în locuri precum aeroporturi, bănci etc.

Analiza imaginilor biomedicale (digitale) se utilizează la tomografia computerizată, analiza de cromozomi, numărarea particulelor de sânge şi altele.

Automatizarea industrială include aplicaţii de identificare şi sortare a obiectelor, de proiectare a şabloanelor, de testare a cablajelor de circuit imprimat etc.

Alte domenii importante precum: *robotica* – include interpretarea scenei sau ghidarea spaţială prin tehnici de vedere artificială (dirijarea unei mişcări prin feed-back vizual). *Cartografia* – urmăreşte întocmirea hărţilor şi a planurilor topografice pe baza unor fotografii. *Alte servicii* precum securitatea unor obiective, prelucrarea imaginilor radar ce permit detectarea automată a ţintelor sau ghidarea avioanelor la coborâre.

Sistemele de recunoaştere a formelor includ toate etapele de procesare a imaginii: de la achiziţionarea datelor brute până la înţelegerea acestora. Datele iniţiale sunt supuse între timp mai multor transformări.

Prin urmare, RdF implică şi discipline conexe, cum ar fi procesarea de semnal şi de imagine, inteligenţa artificială sau procesarea limbajului natural (NLP). De exemplu citirea imaginii de texte tipărite: OCR nu se rezumă doar la recunoaşterea caracterelor, dar implică şi segmentarea imaginii, selectarea segmentelor text, tăierea acestora în linii, cuvinte, apoi în caractere. Aceste analize implică şi prelucrarea semnalului. După recunoaştere în sine, rezultatele pot fi îmbunătăţite prin utilizarea modelelor de limbaje naturale.

1.2 Etapele de analiză şi procesare a imaginilor în vederea recunoaşterii şi clasificării automate

Analiza imaginilor cu scopul RdF este de a realiza sisteme informatice care simulează activitatea umană de percepţie, recunoaştere şi de înţelegere: recunoaşterea scrisului, interpretarea scenelor, robotica, recunoaşterea semnalelor medicale EEG (electroencefalograma), ECG (electrocardiograma). Aceasta implică cunoştinţe pluridisciplinare pentru a înţelege aspectul fizic al captorilor, aspectele matematice de clasificare şi aplicarea în informatică.

Metodele implementate în sistemele de RdF se bazează pe mai multe domenii: analiza numerică, statistică, optimizare combinatorică, cercetare operaţională, analiză sintactică (parsing), teoria grafurilor, inteligenţa artificială, ş.a. Nu există o modelare exactă a fenomenului de recunoaştere a mediului înconjurător de către om, deoarece nu sunt cunoscuţi toţi factorii ce duc la luarea unei astfel de decizii şi nu este cunoscut nici numărul de parametri implicaţi la proces (se estimează de a fi prea mare pentru a fi modelat cu exactitate).

Sistemele de recunoaştere a formelor cuprind toate etapele de procesare începând cu achiziţia datelor până la determinarea apartenenţei obiectului (formei) unei clase. O schemă a sistemului de achiziţie şi analiză a imaginilor în scopul recunoaşterii de forme este reprezentată în figura 1.1.

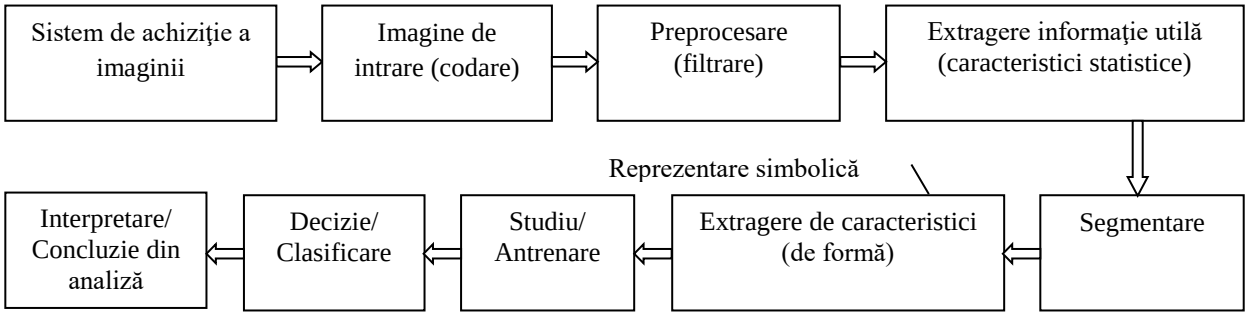


Fig. 1.1. Sistem de achiziţie şi analiză a imaginilor în scopul recunoaşterii de forme

Operaţiile de preprocesare au rolul de a elimina zgomotul sau informaţia inutilă din imagine sau de a restaura imaginea. Astfel de prelucrări sunt necesare pentru îmbunătăţirea rezultatelor algoritmilor de recunoaştere şi clasificare.

Extragerea caracteristicilor imaginii presupune separarea caracteristicilor spaţiale (de ex. de amplitudine), celor de formă şi textură, caracteristicilor statistice (entropia, dispersia, media pătratică etc.), separarea muchiilor şi a contururilor etc.

Clasificarea formelor (obiectelor) dintr-o imagine include gruparea acestora, măsurarea similarităţii între imagini, crearea unei statistici şi a arborilor decizionali etc.

Interpretarea rezultatelor constă în luarea deciziei pe baza analizei caracteristicilor obţinute.

1.2.1 Preprocesarea imaginilor

Preprocesarea imaginii constă în îmbunătăţirea calităţii imaginii – reducerea zgomotului şi a altor defecte ce pot fi prezente în imagine (distorsiune, umbre, reflecţii care apar în timpul achiziţiei), evidenţierea unor zone de interes prin modificarea luminozităţii, a contrastului, accentuarea muchiilor, etc.

Cei mai utilizaţi algoritmi de preprocesare sunt:

– algoritmi de amplificare a contrastului prin modificarea locală sau globală a histogramei, metoda Gordon, Beghdadi etc.;
– algoritmi de atenuare a zgomotului (noise reduction);

Histograma unei imagini $h(x)$ este funcţia care asociază unei valori de intensitate $x \in [0,255]$ numărul de pixeli din imagine care au această valoare.

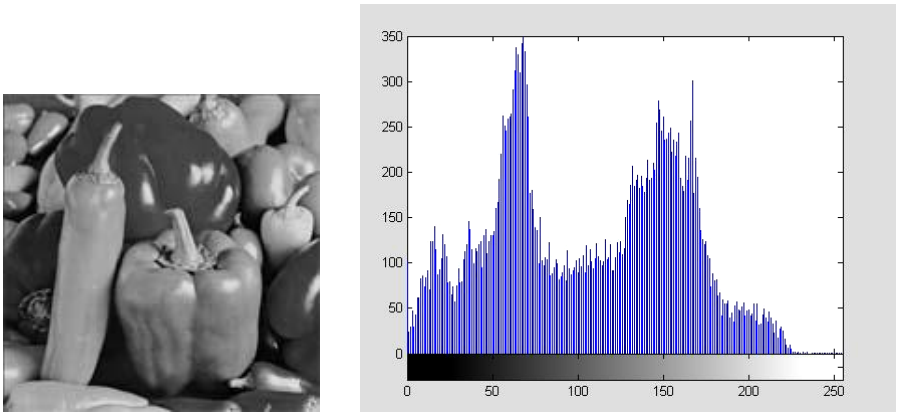


Fig. 1.2. Imaginea „peppers.jpg” şi histograma corespunzătoare acestei imagini.

Pentru o imagine color histograma conţine 3 componente, câte una pentru fiecare culoare de bază (RGB – roşu, verde, albastru).

Histograma poate fi propoţionată pentru a face o estimare a densităţii de probabilitate a pixelilor ce reprezintă raportul dintre numărului de pixeli de o anumită intensitate şi numărul total de pixeli din imagine :

$$p(x) = h(x) / n \times m$$

unde $n \times m$ - dimensiunile imaginii.

Histograma propoţionată la numărul de pixeli din imagine se numeşte funcţia densităţii de probabilitate (PDF*)* a nivelelor de intensitate.

Suma tuturor probabilităţilor este o unitate $\sum p(x) = 1.$

Contrastul este o unitate de măsură a diferenţei în luminozitate dintre zonele de intensitate mare (nuanţe deschise) şi cele joase (întunecate). Histogramele ce cuprind întreaga gamă de nuanţe reprezintă o scenă cu contrast semnificativ, pe când cele „înguste” prezintă mai puţin contrast. Observăm că imaginea „peppers.jpg” din Fig.1.2 conţine puţini pixeli de nuanţă albă şi cei prezenţi se datorează reflecţiei de lumină. Pentru a mări contrastul este necesară „împrăştierea histogramei”, numită şi extindere dinamică. Scopul său este de a da unei imagini

cu contrast mic dinamica completă (de ex. 256 nivele pe 8 biţi) printr-o dilatare artificială, obţinându-se de obicei un contrast mai mare.

Ajustarea contrastului unei imagini prin rescalarea intensităţii fiecărui pixel se obţine aplicând formula:

$$x' = \frac{V_{\min} - V_{\max}}{V_1 - V_0} * x + \frac{V_{\min} * V_1 - V_{\max} * V_0}{V_1 - V_0}$$

unde $V_{\min}$ şi $V_{\max}$ sunt valorile extreme ale intervalului maximal de valori (în general $V_{\min} = 0$ iar $V_{\max} = 255$);

V_0 şi V_1 reprezintă limitele zonei nenule a histogramei imaginii I.

$$\forall x \in [V_0, V_1], x \xrightarrow{T} x' \in [V_{\min}, V_{\max}]$$

Figura următoare ilustrează această transformare:

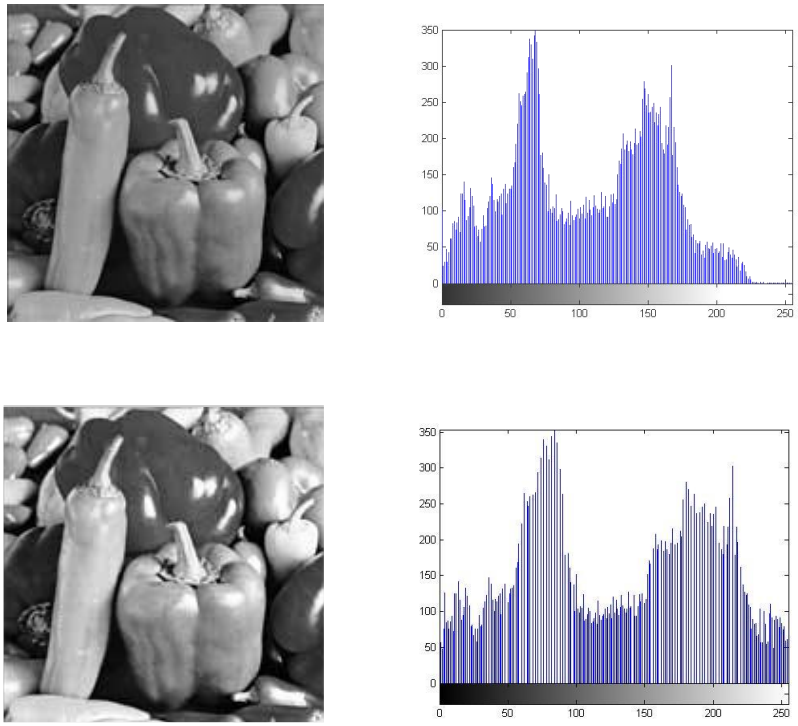


Fig. 1.3. Imaginea şi histograma iniţială şi după ajustarea contrastului (a doua linie)

O altă posibilitate de amplificare a contrastului este egalizarea histogramei. Principiul egalizării este următorul: histograma imaginii este transformată într-o histogramă plată cu un tabel de transcodare care este strict monoton pentru a păstra ordinea contrastelor din imagine. Fie exemplul unei codificări pe 8 biţi, dinamica originală [min, max] este extinsă la [0,255]. Este de

dorit să se atribuie acelaşi număr de pixeli la fiecare nivel de gri, de aceea utilizăm histograma proporţionată.

Pentru a efectua egalizarea histogramei avem nevoie de calcularea histogramei imaginii, a histogramei proporţionate şi celei cumulative.

```
//initializarea histogramei
for(i=0; i<256; i++) H[i]=0;

/* calcularea histogramei */
for (i=0; i<n; i++){
  for (j=0; j<m; j++){
        H[matr [n][m]]++;
  }
}

/* calcularea histogramei proportionate */
for (i=0; i<255; i++) H_proportionata=H[i]/(n*m);

/* calcularea histogramei cumulative */
H_cumulativa=0;
for (i=0; i<255; i++)
   H_cumulativa +=H_proportionata[i];

/* egalizarea histogramei */
for (i=0; i<n; i++){
  for (j=0; j<m; j++){
        nivel_initial = matr [n][m];
        matr_egalizata[n][m]=255* H_cumulativa[nivel_initial]
  }
}
```

Figura următoare ilustrează egalizarea histogramei şi amplificarea contrastului imaginii după egalizare:

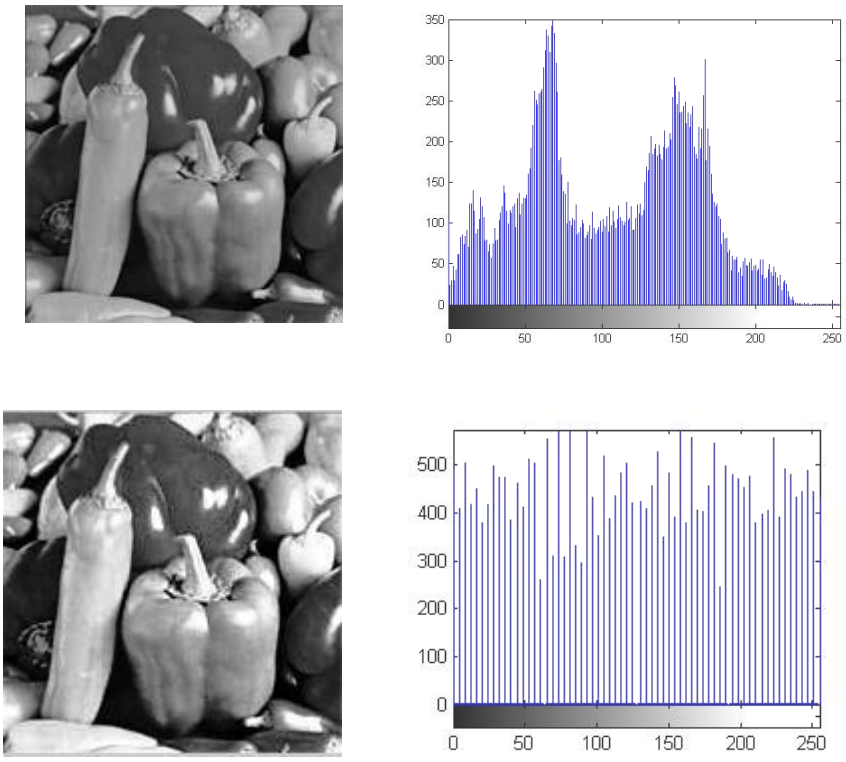


Fig. 1.4. Imaginea şi histograma iniţială şi după egalizarea histogramei (a doua linie)

Preprocesarea şi extragerea de atribute implică operaţii asupra pixelilor vecini, cel mai des se aplică filtrarea liniară, în cazul semnalelor multidimensionale – operaţii elementare de convoluţie. În general, se poate considera că majoritatea algoritmilor de preprocesare şi de extragere a atributelor aplică una sau mai multe convoluţii elementare cu o mască 3×3 sau 5×5. Aplicarea filtrelor pentru metode de detectare a contururilor sunt descrise în capitolul 2, în capitolul dat voi aborda doar filtrele median şi de mediere pentru îmbunătăţirea calităţii imaginii – suprimarea zgomotului. Aplicarea acestor două filtre pentru netezirea histogramei este descrisă în capitolul 2.

Un filtru este definit prin:

- o fereastră pătrată de dimensiune impară (3 sau 5), care se deplasează pe imagine;
- o matrice a coeficienţilor de aceeaşi mărime (3 sau 5).

Aplicarea filtrului asupra imaginii constă în mutarea ferestrei şi înlocuirea valorii fiecărui pixel cu rezultatul operaţiei asupra vecinilor săi. O nouă imagine (îmbunătăţită) este astfel generată.

Medierea pe o vecinătate de 3×3 : $I= h=1/9\begin{bmatrix}1 & 1 & 1\\ 1 & 1 & 1\\ 1 & 1 & 1\end{bmatrix}\times$

86	93	90	98	93
76	84	80	88	88
77	87	80	86	89
75	88	78	80	86
63	83	73	73	83

Primul pas constă în calcularea mediei celor 9 elemente din fereastră: (1x86 + 1x93 + 1x90 + 1x76 + 1x84 + 1x80 + 1x77 + 1x87 + 1x80)/9 şi atribuirii valorii obţinute elementului de pe poziţia centrală. La al doilea pas glisăm fereastra pe imagine cu o poziţie spre dreapta şi procedăm la fel ca la pasul 1. Prima linie şi prima coloană rămân, de obicei, neschimbate sau pot fi copiate rezultatele ca în oglindă (primei linii i se atribuie valorile liniei 2 după schimbare, primei coloane – valorile de pe a doua coloană după schimbare) similar se procedează şi cu ultima linie şi ultima coloană.

La prelucrarea semnalelor este important raportul dintre semnalul util şi nivelul de zgomot (Signal Noise Rapport). Nu putem înlătura total zgomotul, putem doar să-l diminuăm, astfel maximizând raportul SNR.

O primă abordare se bazează pe redundanţa informaţiilor. Noua valoare a unui pixel este calculată prin medierea valorilor din vecinătate. Această operaţie liniară poate fi văzută ca convoluţie discretă a imaginii printr-o mască de unităţi.

$$I'(i,j)=\sum_{(m,n\in v)}k(m,n)I(i-m,j-n),\ \sum_{(m,n\in v)}k(m,n)=1,$$

unde I este intensitatea imaginii originale, I' – intensitatea imaginii filtrate, v – vecinătatea utilizată şi k – masca de convoluţie.

Medierea est un filtru trece-jos ce elimină degradările locale de dimensiuni mici utilă atunci când obiectele din imagine sunt mai mari ca defectele prezente. Trebuie de ţinut cont că acest filtru estompează contururile, nu este indicat de aplicat ca etapă anterioară a segmentării prin determinarea contururilor. Pentru evidenţierea efectului filtrului de mediere s-a adăugat zgomot alb gaussian imaginii „peppers.jpg”.

Notă: Zgomotul alb este caracterizat de o densitatea spectrală de putere (d.s.p.) constantă într-o bandă infinită, de unde rezultă ca ar avea o putere infinită ceea ce e imposibil de generat. Conceptul de zgomot alb este unul pur teoretic. În practică avem zgomot de d.s.p. constantă într-o bandă limitată, numit zgomot alb filtrat. Cel mai cunoscut model este Additive White Gaussian Noise (zgomot aditiv alb şi gaussian). Zgomotul alb nu este în mod necesar şi gaussian, iar un

zgomot gaussian nu este în mod necesar şi alb, deoarece distribuţia probabilistică a amplitudinilor nu asigură şi o d.s.p. constantă.

Rezultatul aplicării filtrului de mediere (http://www.mathworks.com/help/coder/examples/averaging-filter.html) asupra imaginii afectate de zgomot alb gaussian cu medie şi varianţă constantă cu o mască de 3x3 se observă în tabelul de mai jos.

Tabelul 1.1 Rezultatul aplicării filtrului de mediere

			
Imaginea originală	Imagine cu zgomot alb gaussian*	Imaginea filtrată cu o mască de 3x3	Imaginea filtrată de 2 ori cu aceeaşi mască de 3x3

*Note: Add noise to image http://www.mathworks.com/help/images/ref/imnoise.html

Filtrul median constă în parcurgerea imaginii cu o fereastră glisantă 3x3 sau 5x5 şi înlocuieşte valoarea pixelului central cu valoarea mediană (de pe poziţia centrală) a nivelelor de gri ale pixelilor din această fereastră (nivelele de gri sunt sortate în ordine crescătoare).

86 93 90 98 93 76 84 80 88 88 77 87 80 86 89 75 88 78 80 86 63 83 73 73 83	76 77 80 80 84 86 87 90 93
Imaginea originală	Primul pas – sortarea elementelor din fereastră

Fig. 1.5. Exemplu de implementare a filtrului median

Filtru median este un filtru neliniar, elimină zgomotul de tip impuls, nu şterge marginile doar contururi foarte fine. O fereastră de o mărime adecvată poate limita acest efect. Filtru median poate fi aplicat iterativ.

Notă: Zgomotul de tip impuls (cunoscut şi sub numele de *salt and pepper*) constă în impulsuri aleatorii de energie de o amplitudine aleatorie şi conţinut spectral.

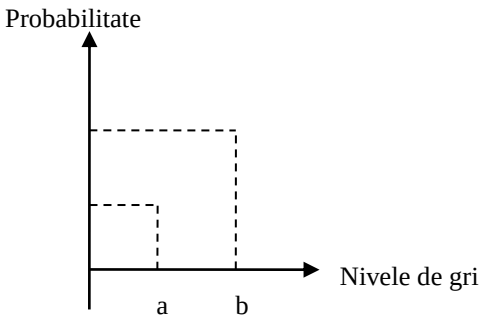
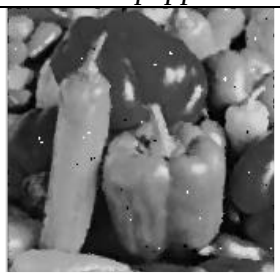


Fig. 1.6. Graficul densităţii de probabilitate a zgomotului de impuls

În modelul de zgomot de tip *salt & pepper* există doar două valori posibile. Pentru o imagine codificată pe 8 biţi, valoarea de intensitate tipică pentru zgomotul *pepper* este ≈ 0 şi pentru zgomotul *salt* este în jur de 255.

Rezultatul aplicării filtrului median (http://www.mathworks.com/help/images/ref/medfilt2.html) asupra imaginii afectate de zgomot *salt and pepper* se observă în tabelul de mai jos. Imaginea mai grav afectată de zgomot necesită filtrare repetată.

Tabelul 1.2 Rezultatul aplicării filtrului median

		
Imaginea originală	Imagine cu zgomot* *salt and pepper* d=0.02	Imaginea filtrată cu filtru median
		
Imagine cu zgomot* *salt and pepper* d=0.2	Imaginea filtrată cu filtru median	Imaginea filtrată de 2 ori cu filtru median

*Note: Add noise to image http://www.mathworks.com/help/images/ref/imnoise.html
d - densitatea zgomotului.

În practică se pot utiliza măşti şi de alte dimensiuni decât 3x3 şi 5x5, cât şi măşti de diferite ponderi. În exemplu următor pentru înlăturarea zgomotului uniform s-a aplicat un filtru ponderat: W=[1 2 1; 2 0 2; 1 2 1].

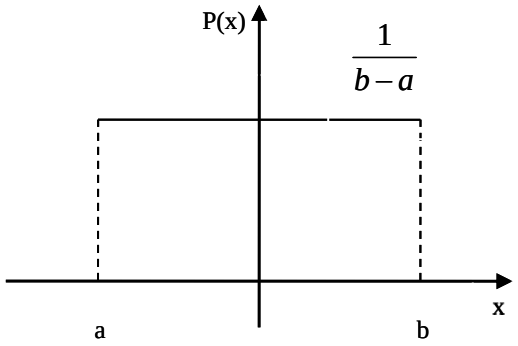
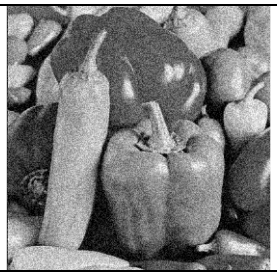


Fig. 1.7. Graficul densităţii de probabilitate uniforme

Pentru $a \le x < b$ probabilitate P(x) este $1/(b-a)$ şi 0 în rest.

În tabelul de mai jos se observă rezultatul aplicării diminuării zgomotului uniform.

Tabelul 1.3 Rezultatul aplicării filtrului ponderat

		
Imaginea originală	Imagine cu zgomot uniform	Imaginea filtrată cu filtru ponderat

După preprocesare obţinem o imagine de o mai bună calitate, cu unele detalii mai bine conturate. Mărimea ferestrei şi repetarea filtrării sau utilizarea mai multor tipuri de filtre consecutiv se determină experimental.

1.2.2 Extragerea atributelor/descriptorilor de imagine

Extragerea caracteristicilor (feature extraction) constă în aplicarea unor algoritmi specializaţi, cum ar fi:

- algoritmi morfologici: dilatare, eroziune, umplere, scheletizare;

- algoritmi de segmentare a imaginii: detectarea contururilor, a unor discontinuităţi – puncte, linii, muchii, conectarea segmentelor (edge linking), segmentarea bazată pe histogramă, segmentarea bazată pe regiuni;
- algoritmi de reprezentare şi descriere a formelor: descrierea contururilor, calcularea momentelor statistice invariante, descriptori Fourier, analiza texturilor;

În capitolul 2 sunt descrise metode uzuale de segmentare a imaginilor şi unele metode de reprezentare a formelor/obiectelor prin regiuni: reprezentarea prin schelet, prelucrări şi transformări morfologice, reprezentări sintactice, descriptori de formă. În capitolul 3 sunt descrise unele metode de recunoaştere bazate pe descriptori locali, inclusiv şi determinarea lor.

Sunt multe posibilităţi de selectare şi reprezentare a caracteristicilor:

- de tip statistic, prin extragerea unui vector de caracteristici X format din diverse măsuri de tip numeric extrase în mod sistematic pe baza formelor. Un astfel de vector poate fi scris:

  $$X = [x_1, x_2, ..., x_k]^T$$

  unde $x_1, x_2, ..., x_k$ – caracteristicile / atributele formei reprezentate prin X .
- de tip structural: se urmăreşte descompunerea formei în constituenţi elementari numiţi primitive. Reprezentarea este apoi ordonată sub forma unui graf sau arbore.

Caracteristicile trebuie să permită distingerea diferitor clase de forme între ele. Experienţa şi intuiţia sunt necesare la alegerea caracteristicilor, iar numărul lor este determinat în funcţie de mărimea setului de antrenare. Pentru obiectele reprezentate bidimensional se pot cita:

- caracteristicile direcţionale: histograme de gradienţi şi caracteristici SIFT (Scale Invariante FeatureTransform) invariante la scalare după cum sugerează şi numele;
- caracteristicile de formă (în limba engleză: shape context) la fiecare punct de contur se calculează distanţele şi unghiurile de la acest puncte la toate celelalte puncte de contur. Aceste valori sunt apoi cuantificate şi reprezentate de o histogramă bidirecţională (distanţă, unghi);
- descriptori Fourier: calcule pe baza punctelor de contur. Aceste caracteristici sunt invariante la deformări cauzate de translaţie, rotaţie şi scalare;
- caracteristici de regiuni (în limba engleză: grid features): se înconjoară forma cu un dreptunghi care este decupat în regiuni şi în fiecare regiune, se calculează procentul de pixeli negri;
- momentele;

– pentru cuvintele unui text care aparţine unui corpus de documente: măsura TF – IDF (Term Frequency – Inverse Document Frequency) şi variantele sale.

Extragerea informaţiei metrice din imagini este o etapă importantă în procesarea imaginilor, de calitatea ei depinde rezultatul algoritmilor de recunoaştere. Se urmăreşte reducerea vectorului de caracteristici, deoarece de ex. caracteristicile de bază: formă, culoare şi textură formează un număr foarte mare de combinări, ceea ce face anevoioasă aplicarea metodelor teoretice de decizie. Pentru reuşita acestei etape se tinde spre micşorarea numărului de forme şi mărimi, dacă nu e principial se ignoră caracteristica de culoare şi textură. Rezultatul final al acestei etape este un vector de n atribute/caracteristici extrase.

Exemplu de distanţe (măsurători) utilizate în metode statistice asupra vectorilor: Euclidiană, absolută, Chebychev ş.a.

În [55-57] găsim patru abordări (metodologii, modelări matematice) importante ale sistemelor de recunoaştere a obiectelor: modelul statistic, modelul sintactic, modelul potrivirii cu şablon (Template Matching Model), model de reţele neuronale.

Pentru măsurarea atributelor sau descriptorilor (feature/pattern measurement) se alege o metodă de măsurare, evaluare şi comparare corespunzătoare modelului matematic ales [55]:

– modelarea prin metode statistice – metode statistice de minimizare a riscului (conditional average risk statistical equation);
– modelare sintactică-structurală – gramatici şi reguli de derivare sintactică, arbori de derivare (analiză) sintactică, automate finite de recunoaştere;
– pentru recunoaşterea prin metoda potrivirii cu un şablon (Template Matching) se utilizează algoritmi de determinare a potrivirii: clasificarea bazată pe distanţa minimă (minimum distance classifier), potrivirea prin corelaţie (matching by correlation);
– modelare prin reţele neuronale – metode de antrenare, reţele neuronale multistrat, algoritmi de învăţare.

În capitolul 3 este descrisă şi implementată metoda de potrivire bazată pe indicele de corelaţie, apoi este propusă o versiune ameliorată a acesteia utilizând şi algoritmul genetic pentru generarea de noi date cu scalarea şi rotirea imaginii pentru o nouă potrivire cu şablonul.

1.2.3 Analiza şi interpretarea rezultatelor

Analiza şi interpretarea rezultatelor reprezintă etapa finală a procesării imaginilor cu scopul recunoaşterii obiectelor. La etapa dată se stabileşte (manual sau automat) apartenenţa

obiectului la o clasă pe baza caracteristicilor extrase şi măsurate anterior. Este vorba despre determinarea, după o perioadă de studiu/antrenare, a unei funcţii de clasificare care permite trecerea de la reprezentare la etichetare.

Uneori este necesar să se introducă o "clasă" nedeterminată, notată ω_0, la care sunt atribuiţi vectorii x a căror formă este ambiguă. Clasa ω_0 nu se potriveşte cu adevărat unei forme. Trebuie să se considere ca o decizie suplimentară în procesul de decizie. Când se decide atribuirea unei forme la clasa nedeterminată, urmează, de obicei, un proces mai fin şi mai costisitor: se apelează un clasificator mai complex, folosind mai multe caracteristici, sau se ia decizia manual făcând apel către utilizator.

Randamentul clasificării reprezintă raportul dintre numărul de determinări corecte şi numărul total de imagini analizate, iar rata de eroare este raportul dintre numărul de determinări false şi numărul total de imagini.

Fie Ω spaţiul de forme, adică ansamblul tuturor formelor posibile a obiectelor propuse spre analiză. Fiecare formă $\omega \in \Omega$ constituie un eveniment elementar. Fie R, spaţiul de reprezentare, ansamblul de caracteristici sau reprezentări extrase ale formelor. Fie C, spaţiul claselor, adică ansamblul etichetelor. Acest ansamblu este presupus finit, de ordinul k.

Exemplu: recunoaşterea cifrelor izolate din imagini. Ω este ansamblul tuturor imaginilor posibile ce reprezintă cifre de la 0 la 9, C este ansamblu de etichete C = {1, 2, 3, 4, 5, 6, 7, 8, 9, 0}, numărul claselor este k = 10.

Se cunosc mai multe metode de clasificare: arbori de decizie, algoritmi de clusterizare de exemplu: k-NN (K-nearest neighbor – k-cei mai apropiaţi vecini), clusterizare ierarhică, clasificatorul Bayes, reţele neuronale artificiale, Support Vector Machine (SVM) etc.

Alegerea algoritmilor de clasificare depinde de tipul caracteristicilor extrase. Unele metode de clasificare vor fi analizate în continuare.

1.3 Analiza metodelor de clasificare automată a imaginilor

Metodele de clasificare folosesc exclusiv un set de măsuri şi/sau cunoştinţe euristice ale funcţionării unui proces pentru a trece de la spaţiu de observare la spaţiu de decizie numit şi spaţiu de reprezentare. Rezultatele clasificării depind atunci de metoda în sine, de cunoştinţele a priori, de parametrii ce caracterizează sistemul şi de calitatea datelor.

Problema RdF este de a caracteriza modelul şi eticheta fiecărei clase asociate unei forme. Acest lucru necesită utilizarea de tehnici de clasificare pentru gruparea formelor similare. Clasele pot fi statice sau dinamice. Clasele statice sunt bazate pe date staţionare, ce înseamnă că

parametrii modelului de clasificare nu vor fi schimbaţi în procesul de recunoaştere. Există multe metode de clasificare a datelor printre care se pot cita K-nearest neighbor [58,59], SVM [60,61], clasificatorul Bayes [62-65]. Metoda de analiză a componentelor principale (PCA – Principal Component Analysis) [66-69], metoda Fuzzy Pattern Matching (FPM) [70-72], metoda Fuzzy C-Means (FCM) [73] precum şi alte numeroase versiuni ale acestor metode sunt implementate la procesarea datelor cu scopul clasificării automate. Sistemele sunt în continuă dezvoltare, de exemplu sistemele evolutive [74-76], pentru care este necesară utilizarea metodelor de clasificare dinamică.

În cazul claselor şi/sau formelor dinamice majoritatea datelor se schimbă datorită evoluţiei. Această dinamicitate se exprimă prin mărirea, eliminarea, fuziunea, dividerea etc. formelor, respectiv claselor. Printre aplicaţiile reale ce necesită o clasificare dinamică se poate enumera supravegherea video, diagnosticul industrial (dezvoltarea modului de funcţionare), diagnosticul medical (expansiunea unei suprafeţe afectate de o maladie) etc. Se cunosc metode RdF pentru prelucrarea datelor dinamice printre care se pot cita: Dynamic Fuzzy Pattern Matching [77,78], Dynamic Fuzzy K- nearest Neighbors [79-81], metode de clasificare pentru forme dinamice [82,83], metode mixte de clasificare pentru date dinamice [84-87].

Formele pot fi statice sau dinamice; parametrii extraşi din forme sunt statistici, structurali şi micşti; metodele statistice, structurale şi mixte, iar modul de identificare şi clasificare poate fi asistat, neasistat sau combinat.

1.3.1 Clusterizarea automată a datelor. Problema clasificării automate

Clasificarea este un mod de organizare a datelor. Scopul clasificării este de a identifica clasele cât mai omogene posibil pe baza trăsăturile descriptive (atribute, caracteristici etc.). Există două tipuri de clasificare: supervizată (asistată) şi nesupervizată (neasistată).

Scopul clasificării neasistate este de a determina într-un ansamblu de entităţi grupuri (clustere, clase) de date omogene – similare între ele şi distincte faţă de cele din alte grupuri. În cazul clasificării automate nu avem informaţii cu privire la apartenenţa datelor la o clasă sau alta, ci doar caracteristici, observaţii specifice unei clase sau alteia. Această apartenenţă este, în general, dedusă din distribuţia spaţială a punctelor (se grupează observaţiile care sunt "aproape" unele de altele în spaţiul de reprezentare (spaţiul de variabile)) explicată de Campedel şi Moulines [88].

Clasificarea include:

i. regruparea consecutivă a observaţiilor celor mai 'similare' (metoda ierarhică) sau regruparea în k grupe a tuturor observaţiilor simultan;

ii. aplicarea criteriul de 'similaritate' între două observaţii;

iii. aplicarea criteriul de 'similaritate' între două grupuri sau între o observaţie şi un grup.

Indiferent de tipul de clasificare automat sau manual, alegerea metodei se face întotdeauna pentru o anumită problemă. Într-adevăr, nu există o abordare optimă pentru orice tip de date. În plus, odată ce abordarea este adoptată, rezultatele depind în mare măsură şi de parametrizarea metodei. În cazul asistat, datele de instruire (de învăţare) nu reprezintă întotdeauna perfect realitatea datelor în cauză, ceea ce poate degrada rezultatele clasificării, în special dacă clasele selectate pentru învăţare în cazul clasificării asistate nu sunt relevante. Se poate întâmpla ca unele clase să fie uitate, sau ca etichetarea să fie făcută în condiţii proaste ce nu permite distincţia claselor interesate, obţinându-se astfel o clasificare ce depinde de calificarea persoanei ce a făcut etichetarea. În clasificarea neasistată se determină automat clasele, deci aceasta poate completa o clasificare asistată, pentru a confirma sau a nega alegerea claselor iniţiale. O soluţie posibilă ar fi fuzionarea rezultatelor date de clasificarea asistată şi neasistată pentru a beneficia de rezultatele ambelor metode. Prin urmare, combinaţia oferă utilizatorului un compromis între cele două metode. În literatură găsim lucrări referitoare la realizarea fuziunii unui tip de clasificare: de exemplu fuziunea clasificărilor automate (propuse de Forestier et al. [89], Gançarski et al. [90], Wemmert şi Gançarski [91], Masson şi Denœux [92, 93]); sau asistate (propuse de Xu et al [94], Chen et al [95]). Abordarea fuziunii clasificărilor automate este mai dificilă, având în vedere lipsa de informaţii cu privire la etichetele claselor. Se observă faptul că fuziunea dintre metodele automată şi asistată nu este realizată cu scopul direct de clasificare, ele sunt complementare, în sensul că metoda automată se aplică la învăţarea asistată [96-99].

Un criteriu de omogenitate între entităţi poate fi o similaritate, o disociere sau o distanţă. O *similaritate* (asemănare) poate fi exprimată prin coeficientul de corelaţie de exemplu, iar o *disociere* (depărtare, diferenţiere) prin distanţe, de exemplu distanţa euclidiană. Măsura de similaritate utilizată variază în funcţie de tipul de date, adică de tipul de variabile ale matricei: cantitative, calitative sau mixte.

Se consideră un set $\Omega = \{1,...,i,...,k\}$ de k obiecte descrise de c variabile $x_1,...,x_k$ într-o matrice X cu k linii şi c coloane:

$$X = (x_{i,j})_{k \times c} = \begin{matrix} 1 \\ \vdots \\ k \end{matrix} \underbrace{\begin{bmatrix} \vdots & \\ \cdots & x_{i,j} \in R \quad \cdots \\ \vdots & \end{bmatrix}}_{1 \cdots c}$$

Unde $1 \geq j \leq c$ şi $1 \geq i \leq k$.

O entitate $i \in \Omega$ este descrisă de vectorul $x_i \in R^c$ (linia i a matricei X). O pondere Ω_i este asociată fiecărei entităţi i. Pentru datele ce rezultă dintr-o extragere aleatoare cu o probabilitate uniformă $\omega_i = 1/k$ pentru fiecare i. În unele cazuri este util de a lucra cu ponderi neuniforme.

Dacă datele sunt cantitative distanţele clasice aplicate la clasificare sunt:

- distanţa euclidiană simplă:

  $$d^2(x_i, x_i') = (x_i - x_i')^T (x_i - x_i')$$

- distanţa Manhattan:

  $$d(i, i^i) = \sum_{j=1...c} |x_{ij} - x_{i\,j'}|$$

- distanţa Chebychev, sau distanţa maximului:

  $$d(i, i^i) = \max_{j=1...c} |x_{ij} - x_{i\,j'}|$$

Mai sunt utilizate şi alte distanţe precum distanţa Mahalanobis, distanţa euclidiană normalizată sau valoarea absolută.

Alegerea unei distanţe se face în dependenţă de problema concretă efectuând mai multe încercări alegând rezultatele optimale şi se face în trei paşi:

- Normalizarea – această etapă permite evitarea efectului de scară: variabilele au astfel o contribuţie echivalentă la calcularea distanţelor, indiferent de care a fost valoarea lor iniţială;
- Selectarea variabilelor – se bazează pe analiza descriptivă a datelor, cât şi pe cunoştinţele a priori pe care le avem referitoare la problemă;
- Alegerea ponderilor – se determină experimental ponderile fiecărei variabile, dacă se doreşte ca anumite variabile sa aibă o influenţă mai mare la calculul distanţelor.

Alegerea distanţei trebuie să reflecte o analiză a rezultatelor experimentale, a statisticii, nicidecum să nu fie întâmplătoare sau bazată pe faptul că în literatură găsim de exemplu mai multe lucrări cu utilizarea distanţei euclidiene normalizate şi alegem distanţa cea mai populară.

Clasificarea ascendentă ierarhică. Iniţializarea acestui algoritm constă în calcularea tabloului de distanţe (sau de diferenţiere) între elementele care urmează să fie clasificate. Algoritmul porneşte apoi partiţia a n singletons (fiecare element reprezintă o clasă) şi se caută la fiecare pas, formarea claselor prin reuniunea a două elemente situate cel mai aproape (distanţa cea mai mică) la etapa anterioară, construind progresiv un arbore sau o dendogramă, regrupând în final toate elemente într-o singură clasă, la rădăcină. La fiecare pas al algoritmului este necesar să se actualizeze tabelul de distanţe (sau disimilarităţi). După fiecare reuniune a două elemente (entităţi, obiecte), a două clase sau a unui element la o clasă, distanţa dintre noul element şi altele sunt calculate înlocuind în matrice distanţele la elementele care au fost reunite (incluse în clasă). Diferite abordări sunt posibile la acest nivel, care rezultă în diferite clasificări.

Primul pas este gruparea a două elemente. Fie A şi B două elemente ale unei partiţii date, cu ω_A şi ω_B ponderile lor, problema este de a defini $d(A,B)$, distanţa între aceste elemente ale unei partiţii din Ω. Se cunosc 3 posibilităţi: "single linkage" ("cel mai apropiat vecin"), metoda "complete linkage" ("vecinul cel mai depărtat") şi metode intermediare ("average linkage").

Pasul doi: Fie A,B şi C trei grupuri/clase, Presupunem că am reunit A şi B. Se va calcula similaritatea $A+B$ cu C în felul următor:

1. $d(A+B,C)=\max[d(A,C),d(B,C)]$ dacă optăm pentru "single linkage";
2. $d(A+B,C)=\min[d(A,C),d(B,C)]$ dacă optăm pentru "complete linkage";
3. $d(A+B,C)=\frac{n_A}{n_A+n_B}*d(A,C)+\frac{n_B}{n_A+n_B}*d(B,C)$ dacă aplicăm metoda "average linkage", unde n_A : numărul de observaţii ce constituie clasa A.

Pasul doi se repetă până la obţinerea unei singure clase, având loc astfel agregări succesive reunindu-se toate elementele.

Numărul de clase este determinat a posteriori, obţinându-se o dendrogramă care ilustrează grafic aceste agregări succesive. Înălţimea unei ramuri este proporţională cu indicele de disociere sau distanţa dintre cele două obiecte.

Dendrograma este o reprezentare grafică, în formă de arbore binar. Un exemplu de dendogramă este ilustrat în figura de mai jos, aplicând codul din sursa

http://www.mathworks.com/help/stats/hierarchical-clustering.html. Funcţia pdist(X) calculează implicit distanţa Euclideană, pentru calcularea altei distanţe se specifică cea dorită, de exemplu pdist(X, 'chebychev') http://www.mathworks.com/help/stats/pdist.html.

Pentru X = [1 2; 2.5 4.5; 3 4; 5 7; 4 3.5; 5.5 6.5; 8 9.5];

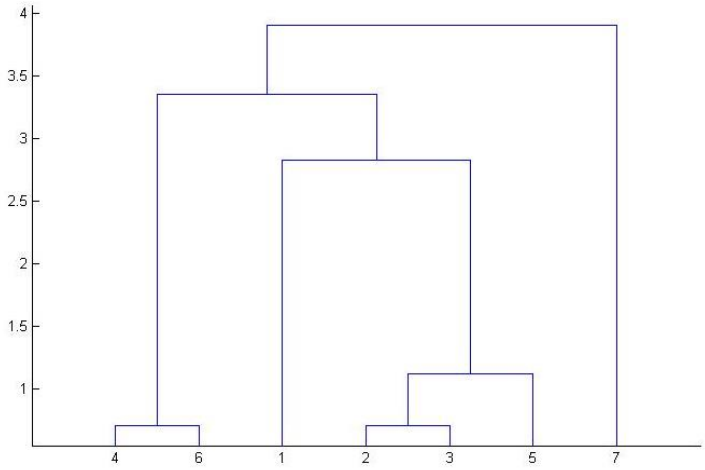


Fig. 1.8. Dendogramă obţinută aplicând clasificarea ascendentă ierarhică

Corelaţia cophenetică este un indicator de calitate a clasificării sau a unei dendrograme ierarhice şi provine de la distanţa cophenetică. Această distanţă este definită între două puncte reprezentate într-un arbore de înălţimea ramurilor care în cele din urmă se reunesc în acelaşi grup. Aceasta este, de asemenea, distanţa dintre cele două grupuri conţinând aceste observaţii înainte de a fi regrupate şi ele. În mod evident, cu cat această valoare este mai aproape de 1, cu atât este mai bună clasificarea.

Pentru exemplul prezentat mai sus c = 0.7541.

1.3.2 Clasificatorul Bayes

Modelarea bayesiană se bazează pe teoria statistică a lui Bayes. Acest tip de modelare este cel mai potrivit pentru gruparea datelor incerte. Modelarea bayesiană nu oferă rezultate exacte, doar probabilităţi. Într-un sistem probabilistic, se tinde spre minimizarea probabilităţii de eroare. Aceasta este echivalentă cu alegerea clasei cele mai probabile.

În [100-102] este implementată Teorema lui Bayes pentru clasificarea datelor. Dacă C_i , pentru $i = 1,2,....,k$, reprezintă cele k clase, şi $X = [x_1, x_2, ..., x_n]$ este vectorul de caracteristici extras pentru forma a cărei apartenenţă de clasă trebuie găsită, atunci probabilitatea că forma aparţine unei anumite clase C_i este dată de probabilitatea a posteri $P = (C_i \mid X)$ a acelei clase C_i dată de caracteristica vector X (folosind Teorema lui Bayes):

$$P = (C_i \mid X) = \frac{P(X \mid C_i)P(C_i)}{P(X)}$$

unde $P(C_i)$ este probabilitatea a priori a clasei C_i; Dacă nu se ştie valoarea a priori a probabilităţilor claselor, se presupune că sunt egale $P(C1) = P(C_2) = ...P(C_k)$. $P(X \mid C_i)$ este probabilitatea vectorului caracteristică X, având în vedere că forma face parte din clasa C_i;

$P(X)$ este probabilitatea totală a vectorului caracteristică *X*, adică, $\sum_i^k P(X \mid C_i)P(C_i)$.

Un clasificator Bayesian calculează mai întâi $P(C_i \mid X)$, folosind ecuaţia de mai sus. Apoi, clasificatorul atribuie eticheta C_m la un anumit vector caracteristică X_o dacă $P(C_m \mid X_o)$ este maximă, adică, $C_m = \arg\max_i \{P(C_i \mid X)\}$. Probabilităţile a priori $P(C_i)$, $P(X)$ şi probabilitatea condiţionată $P(X \mid C_i)$ sunt calculate din imaginile etichetate.

În [103] este implementat algoritmul de clasificare Gaussian Naive Bayes care are o acurateţe a rezultatelor de 97%-100%. Probabilitatea caracteristicilor se presupune a fi Gaussiană:

$$P(x_i \mid C_i) = \frac{1}{\sqrt{2\pi\sigma_{C_i}^2}} \exp\left(-\frac{x_i - \mu_{C_i}}{2\sigma_{C_i}^2} \right)$$

Parametrii $\sigma_{C_i}^2$ şi μ_{C_i} sunt estimaţi folosind probabilitatea maximă.

În [104] este dată o comparare a algoritmilor Fuzzy Naive Bayes şi Gaussian Naive Bayes. Clasificatorii sunt antrenaţi cu numere diferite de exemple de instruire şi numere diferite de atribute. În cele din urmă, autorii constată că abordarea Fuzzy Naive Bayes oferă rezultate foarte promiţătoare în domeniul tratat de ei.

Una din versiunile algoritmului clasificatorului Naive Bayes este Bernoulli Naive Bayes [105] ce foloseşte distribuţii cu variabile multiple Bernoulli. Regula de decizie se bazează pe :

$$P(x_i \mid C_i) = P(i \mid C_i)x_i + (1 - P(i \mid C_i))(1 - x_i)$$

care diferă de la regula multinominală Naïve Bayes prin faptul că penalizează în mod explicit lipsa caracteristicii i care este un indicator pentru clasa C_i.

Importanţa clasificatorului Naive Bayes este bine argumentată în [106]. Autorii lucrării testează diverse combinări ale clasificatorului cu alte tehnici pentru a îmbunătăţi acurateţea rezultatelor. Concluzia lor este că tehnica Naive Bayes oferă o precizie de clasificare superioară

atunci când e completată cu alte tehnici, de exemplu cu Support Vector Machine au obţinut o acurateţe de ≈90%.

1.4 Metode de recunoaştere a formelor şi interpretare a imaginilor

Pentru aplicarea în practică a algoritmilor de recunoaştere am ales *domeniul trierii deşeurilor*. În Republica Moldova nu se respectă sortarea deşeurilor menajere şi dacă ar apărea investiţii referitor la incinerarea deşeurilor pentru producerea de energie nu ar risca nimeni căci la noi se aruncă şi produse toxice, baterii, uleiuri uzate, care trebuie să fie puse în centru de reciclare în recipiente adecvate, deci deşeurile trebuie să fie sortate în prealabil. Se mai poate obţine compost din deşeuri organice. Ar fi util şi benefic pentru mediu de a găsi o soluţie de sortare efectivă a deşeurilor.

Elaborarea „Strategiei naţionale de gestionare a deşeurilor pentru perioada 2013-2027", aprobată prin HG nr. 248 din10.04.2013 are ca obiectiv gestionarea deşeurilor în R. Moldova care rămâne a fi o problemă dificilă. „Gestionarea deşeurilor în Republica Moldova rămâne a fi o problemă dificilă şi nerezolvată „atât din punct de vedere organizatoric cât şi legislativ". Cu toate că domeniul protecţiei mediului este reglementat de circa 35 de acte legislative şi peste 50 de Hotărâri de Guvern, aspectul legal al gestionării deşeurilor lasă mult de dorit, fiind necesară atât restructurarea cadrului legal şi instituţional, cât şi crearea unui sistem integru de reglementare tehnică şi ecologică în domeniile de colectare selectivă pentru reciclarea, valorificarea, eliminarea şi depozitarea deşeurilor. [107]"

Pentru implementarea practică s-a ales sistemul existent până la 1 ianuarie 2015 care conţine 15 categorii de pericol ale substanţelor şi amestecurilor: explozive, oxidanţi, extrem de inflamabile, uşor inflamabile, inflamabile, foarte toxice, toxice, nocive, corozive, iritanţi, sensibilizatori, cancerigene, mutagene, toxice pentru reproducere, periculoase pentru mediu.

Definiţiile pentru diferitele categorii de pericol sunt incluse în Codul Muncii.

Gestionarea necorespunzătoare a deşeurilor afectează grav mediul nu numai prin poluarea locală ci şi contribuind la emisiile globale de gaze cu efect de seră.

Se cunoaşte sortarea mecanică [108]:

- magnetică: metalele feroase (de ex. oţel) sunt extrase prin simpla magnetizare;
- maşini cu curenţi turbionari (Foucault): metalele neferoase, cum ar fi aluminiu, sunt extrase din fluxul de deşeuri;
- platforme vibratoare, rotative, etc.;

- sortare aeraulică: deşeurile (toate categoriile) sunt separate prin suflare de aer în funcţie de greutatea şi densitatea lor.
- alte procese de sortare ce pot fi aplicate, ar fi spectroscopia în infraroşu şi detectarea cu raze X pentru plastic, e posibilă chiar şi diferenţierea tipurilor (PVC, PET, etc.).

Cea mai de încredere sortare în prezent este sortarea manuală. Personalul de sortare poate separa sticle de diferită culoare, hârtie de diferită calitate, deşeuri periculoase etc. Această metodă este costisitoare şi trebuie de mărit randamentul de selectare cu ajutorul utilajelor speciale, de aici a apărut ideea de sortare optică prin aplicarea algoritmilor de recunoaştere a imaginilor digitale.

Nu este aplicată până în prezent sortarea prin recunoaşterea obiectelor, dar sunt deşeuri care nu sunt colectate corect şi ar trebui eliminate de pe banda de sortare: fiole medicale şi sticle de parfum (de pe banda cu sticlă); containerele care au conţinut pesticide; containerele de uleiuri şi vopsele, solvenţi; containerele de ulei de motor (de pe banda cu plastic), baterii, containere ce conţin uleiuri alimentare arse sau vechi, ambalaje de hârtie acoperite cu folie de plastic etc. (pe banda de deşeuri alimentare). Aceste deşeuri sunt toxice pentru mediu şi sănătate, deci trebuiesc stocate separat. Ar fi bine să poată fi eliminate din grămada de deşeuri pentru reciclare specială.

Scopul este de a investiga şi a prezenta rezultatele actuale ale aplicării metodelor de identificare automată şi clasificarea substanţelor chimice etichetate.

Utilizarea substanţelor chimice este inevitabilă: îngrăşăminte, pesticide, aditivi alimentari, produse farmaceutice, materiale de curăţare, aparate, combustibili, etc. Ei ne ajută, dar utilizarea lor ar trebui să fie controlată. Primul pas care duce la utilizarea în siguranţă a substanţelor chimice este cunoaşterea identităţii lor, pericolul lor asupra sănătăţii şi mediului inclusiv şi mijloacele de a le controla. Nu numai utilizarea, ci şi sortarea, depozitarea ambalajelor după utilizare a substanţelor chimice este importantă, deoarece acestea afectează mediul.

Propun spre implementare inspecţia vizuală automată a tuturor ambalajelor, este o opţiune fezabilă în prezent şi ar reduce la minimum daunele produselor chimice asupra mediului şi deci asupra sănătăţii noastre. Sistemele de vedere computerizate permit inspecţia fără contact direct cu deşeurile (încă un avantaj): o imagine a ambalajului este obţinută şi stocată în format digital [109].

1.4.1 Algoritmi de detecţie a punctelor de interes SIFT şi ASIFT

1) ***Scale-Invariant Feature Transform (SIFT).*** O aplicaţie tipică de potrivire bazată pe puncte de interes are următoarele etape : în primul rând, un set de puncte de interes sunt detectate pe ambele imagini. De multe ori se ajunge la un număr mare: sute sau chiar mii (depinde de complexitatea obiectelor din imagine). Punctele de interes (keypoints) trebuie să fie invariante la scalare, rotire, schimbarea punctului (unghiului) de vedere şi la unele modificări în iluminare. După aplicarea algoritmului de potrivire se stabilesc corespondenţele.

David G. Lowe [110] afirmă că SIFT este "o metodă de extragere a caracteristicilor distinctive invariante din imagini care pot fi utilizate pentru a efectua o potrivire sigură între puncte de vedere diferite ale unui obiect sau scenă. Caracteristicile sunt invariante la scalarea şi rotirea imaginii, şi sunt afişate pentru a oferi o potrivire robustă pentru o gamă substanţială de modificări: distorsiune afină, schimbare punct de vedere 3D, adăugare de zgomot, şi schimbare în iluminare."

Autorul propune utilizarea acestor caracteristici pentru recunoaşterea obiectelor. Recunoaşterea prin potrivirea caracteristicilor individuale dintr-o bază de date de caracteristici ale obiectelor cunoscute include implementarea unui algoritm complex – utilizarea iniţială a principiului – cel mai apropiat vecin, urmat de transformata Hough pentru a identifica grupuri care aparţin unui singur obiect, şi în cele din urmă verificarea soluţiei prin intermediul celor mai mici pătrate pentru reprezentarea parametrilor consistenţi. Această abordare a recunoaşterii poate identifica obiecte aproape în timp real.

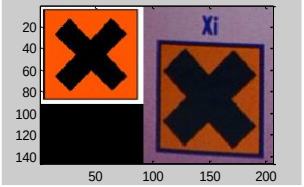
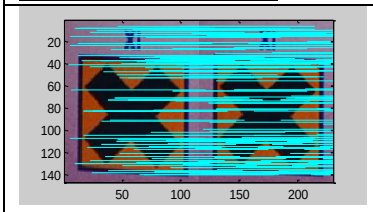
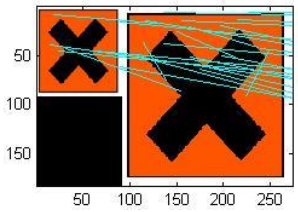
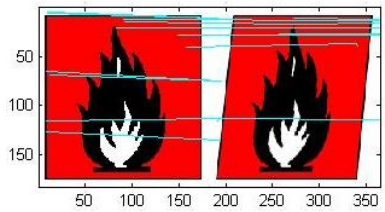
Mikotajczyk şi Schmid au evaluat o varietate de abordări şi au identificat algoritmul SIFT [110] ca fiind cel mai rezistent la deformări comune de imagine. Fiind stabil la detectarea şi reprezentarea trăsăturilor locale, SIFT este o componentă fundamentală a multor înregistrări de imagini şi algoritmi de recunoaştere a obiectelor.

Algoritmul citeşte două imagini, determină caracteristicile lor locale şi afişează linii de corespondenţă între punctele de interes comune. O potrivire este acceptată numai în cazul în care distanţa sa este mai mică de distRatio (distRatio = 0,6;) Se returnează numărul de potriviri afişate.

Algoritmul dezvoltat de D. Lowe [111] dă rezultate relativ bune numai atunci când este scanat simbolul de pericol, nu şi eticheta produsului în întregime. Analizând rezultatele prezentate mai jos se poate constata că algoritmul e sensibil la scalare, intensitate şi nu e aşa de robust la rotire, distorsiune după cum susţine autorul. Încă un dezavantaj e că contează care imagine e şablon, obţinându-se rezultate diferite dacă se schimbă ordinea celor două imagini.

2) ***Affine SIFT.*** Combinând simularea şi normalizarea metodei SIFT funcţionează mai bine decât alte metode. În timp ce SIFT este invariant la scalare, rotaţie şi translaţie, metoda ASIFT îşi propune să trateze şi unghiurile ce definesc orientarea axei camerei. Astfel metoda permite identificarea în mod fiabil a caracteristicilor care au fost supuse distorsiunilor afine foarte mari măsurate de un nou parametru, înclinarea de tranziţie [112].

Tabel 1.4 Exemple de potrivire a punctelor de interes aplicând SIFT

Finding keypoints... 155 keypoints found. Finding keypoints... 9564 keypoints found. Found 0 matches.	Finding keypoints... 380 keypoints found. Finding keypoints... 9564 keypoints found. Found 0 matches.	Finding keypoints... 155 keypoints found. Finding keypoints... 380 keypoints found. Found 0 matches.
		
Finding keypoints... 380 keypoints found. Finding keypoints... 380 keypoints found. Found 380 matches.	Finding keypoints... 155 keypoints found. Finding keypoints... 230 keypoints found. Found 30 matches.	Finding keypoints... 232 keypoints found. Finding keypoints... 167 keypoints found. Found 27 matches.
		

Exemplul 1 de implementare a algoritmului ASIFT:

Computing keypoints on the two images...
3636 ASIFT keypoints are detected.
15127 ASIFT keypoints are detected.
Keypoints computation accomplished in 5 seconds.
Matching the keypoints...
The two images match! **17 matchings are identified**. log(nfa)=-10.2588.

Fig. 1.9. Exemplu de aplicare a algoritmului ASIFT [113] cu răspuns pozitiv

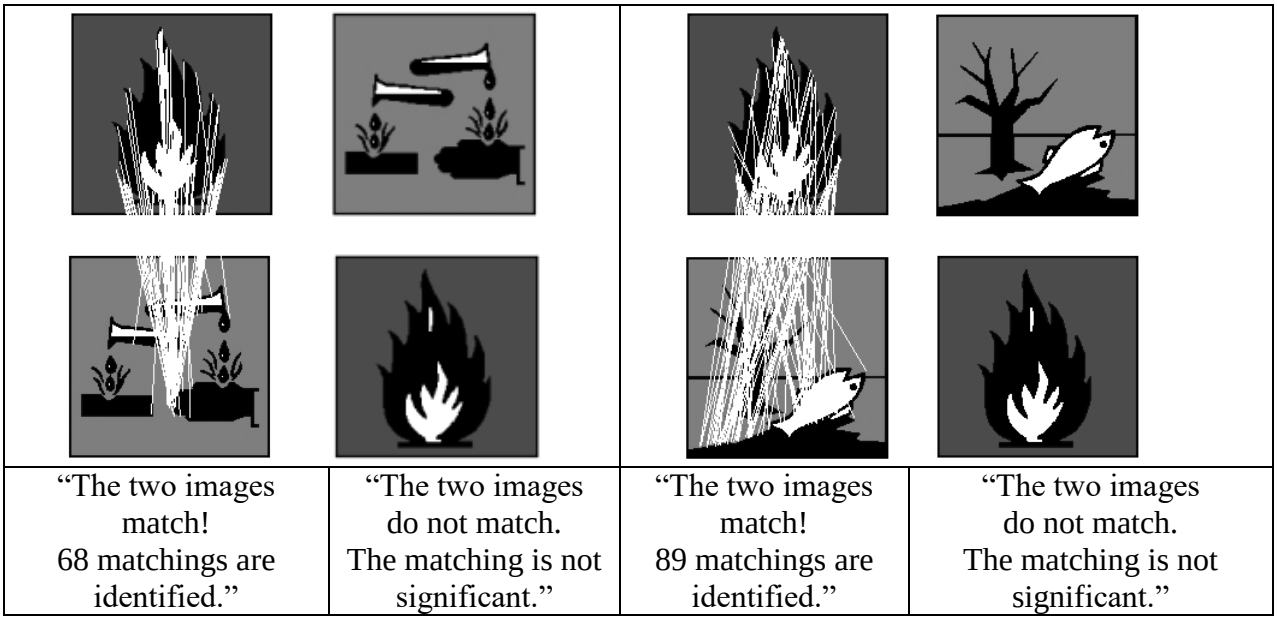
Exemplul 2 de implementare a algoritmului ASIFT:

```
Computing keypoints on the two images...
5460 ASIFT keypoints are detected.
29476 ASIFT keypoints are detected.
Keypoints computation accomplished in 5 seconds.
Matching the keypoints...
The two images do not match. The matching is not significant: log(nfa)=0.184707.
Keypoints matching accomplished in 0 seconds.
```

Fig. 1.10. Exemplu de aplicare a algoritmului ASIFT [113] cu răspuns negativ

În unele cazuri SIFT eşuează în timp ce ASIFT oferă rezultate mai bune, aşa cum putem vedea în exemplele de mai sus.

ASIFT are şi dezavantaje, rezultatele nu sunt stabile, dupa cum se poate vedea din Fig.1.11 contează care imagine este şablon si care referinţă.



"The two images match! 68 matchings are identified."	"The two images do not match. The matching is not significant."	"The two images match! 89 matchings are identified."	"The two images do not match. The matching is not significant."

Fig.1.11. Determinarea imaginilor similare prin aplicarea algoritmului ASIFT

1.4.2 Reţele neuronale artificiale

Primele încercări de modelare a creierului datează din 1943, McCulloch (neurofiziolog) şi Pitts (specialist în logica-matematică) au propus primele noţiuni de neuron artificial. Primul model operaţional al unei reţele neuronale artificiale (RNA) cu scopul de a imita funcţia retinei şi a simula recunoaşterea obiectelor a fost elaborat de Rosenblatt în 1958 şi conţinea un strat de intrare (informaţia de prelucrat) şi unul de ieşire (rezultatul prelucrării).

Neuronul artificial este elementul de bază al unei reţele neuronale artificiale. Funcţionarea lui poate fi modelată astfel [114]:

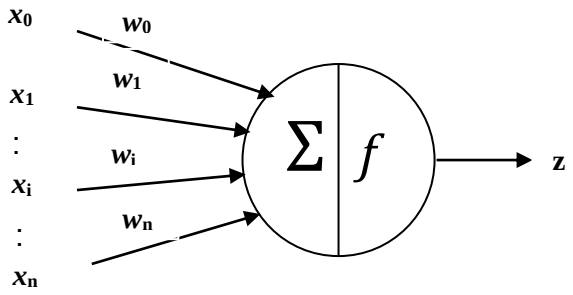


Fig. 1.12. Reprezentarea unui neuron artificial

Valorile de intrare $x_0, x_1, ..., x_i, ..., x_n$ reprezintă conexiunile neuronului. Ponderea sinaptică atribuită fiecărei conexiuni măsoară importanţă relativă: ponderea este proporţională cu indicele de importanţă a conexiunii. Neuronul descris mai sus are o ieşire cu valoarea $y = x_0 w_0 + x_1 w_1 + ... + x_i w_i + ... + x_n w_n$, iar z în figura de mai sus reprezintă rezultatul funcţiei de activare a neuronului $z = f(y)$.

Dacă ponderea w_i este o valoare pozitivă, atunci are loc o acţiune excitatoare, dacă este negativă – inhibatore.

Funcţia de activare reprezintă funcţia de transfer intrare-ieşire a neuronului. Ea poate fi de tipul:

- liniară (de identitate) $f(x) = \alpha x$;
- binară (0 sau 1) $f(x) = \begin{cases} 1, x > 0 \\ 0, x \le 0 \end{cases}$;
- funcţie cu prag $f(x) = \begin{cases} \alpha, x \ge \theta \\ \beta, x < \theta \end{cases}$;
- rampă $f(x) = \begin{cases} 1, x \ge 1 \\ x, |x| < 1 \\ -1, x \le -1 \end{cases}$;
- sigmoidă $f(x) = 1/(1+e^{-\beta x})$;
- funcţie Gaussiană $f(x) = \sqrt{\frac{1}{2\pi\sigma}} e^{-\frac{x-\mu}{2\sigma^2}}, \sigma > 0$, unde μ reprezintă media, iar σ dispersia.

Perceptronul (Fig. 1.13) presupune un strat de intrare constituit din *n* neuroni elementari a căror funcţie de activare este liniară şi un strat de ieşire constituit din unul sau mai mulţi neuroni a căror funcţie de activare poate fi totul sau nimic (0 sau 1).

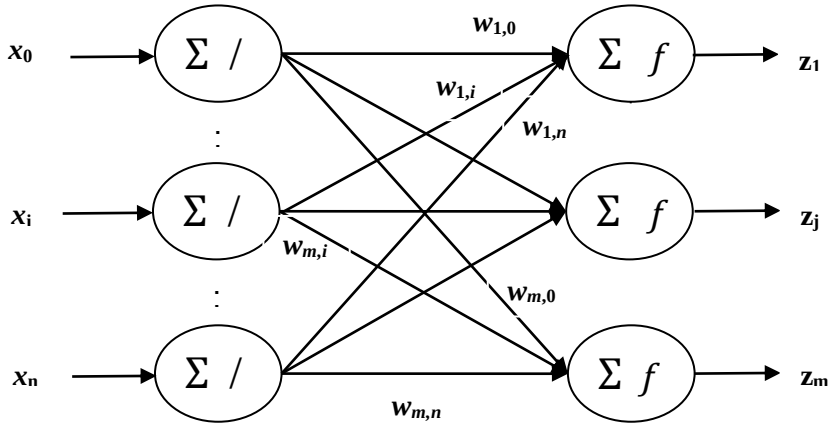


Fig. 1.13 Structura unui perceptron [114]

O reţea neuronală este asocierea sub forma unui graf a neuronilor artificiali. Reţelele se disting prin: arhitectura lor (organizarea lor în graf - monostrat, în straturi ş.a.), nivelul de complexitate (număr de neuroni, prezenţa sau nu a buclelor de reacţie în reţea), tipurile de

neuroni (funcţiile lor de tranziţie sau de activare) şi obiectivul lor: de învăţare asistată sau nu, de optimizare sau sisteme dinamice.

Haykin S. [115] descrie reţeaua neuronală ca un procesor masiv distribuit paralel elaborat din unităţi de procesare simple, care are tendinţa naturală de a stoca cunoştinţele experimentale şi a le face valabile pentru utilizare. Autorul consideră că RNA se aseamănă cu creierul în două aspecte: 1) RNA achiziţionează cunoştinţe din mediul exterior prin intermediul procesului de studiu şi 2) Conexiunile inter-neuronale, cunoscute ca ponderi sinaptice, sunt utilizate la stocarea cunoştinţelor.

După sensul în care informaţia parcurge reţeaua se cunosc reţele neuronale *feed-forward* (într-o singură direcţie) şi reţele *feed-back (cu reacţie)*. Reţeaua care conţine cel puţin o conexiune de feed-back se numeşte reţea neuronală recurentă (RNR), activările ei sunt incluse în buclă. Arhitectura RNR poate avea mai multe forme. Cea mai comună formă constă dintr-un standard de Multi-Layer Perceptron (MLP), plus bucle adăugate.

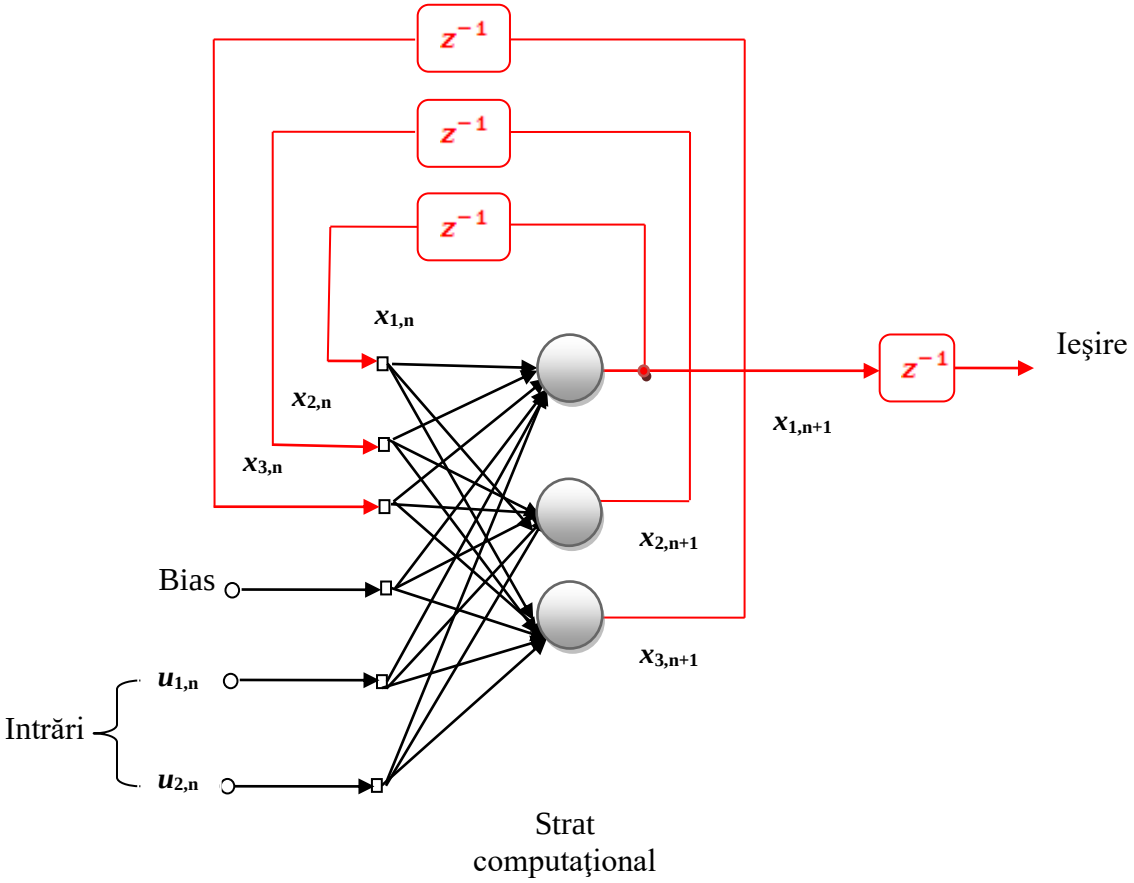


Fig.1.14. Reţea recurentă cu două intrări, doi neuroni ascunşi, şi un neuron de ieşire. Conexiunile de feedback sunt afişate în roşu pentru a sublinia rolul lor la nivel global.

Învăţarea presupune codificarea informaţiei astfel încât sistemul să poată „percepe” datele. În cazul reţelelor neuronale, învăţarea înseamnă modificarea ponderilor conexiunilor dintre neuronii reţelei prin analiza statistică a vectorilor de intrare în scopul clasificării informaţiei. Instruirea reţelelor neuronale reprezintă un proces iterativ şi se realizează prin tehnici de învăţare supervizată, sau automată.

Algoritmul de instruire include următorii paşi :

1. iniţializarea ponderilor;
2. calcularea rezultatului la ieşire;
3. calcularea corecţiei Δw cu care se modifică fiecare pondere;
4. modificarea ponderilor;
5. se revine la pasul 2 – recalcularea ieşirilor – cât modificările ponderilor sunt semnificative.

În procesul de învățare supervizată, rezultatele corecte (adică valorile care reţeaua ar trebui să le obţină la ieşire) sunt furnizate reţelei, astfel încât să poată fi ajustate ponderile pentru obţinerea lor. După instruire, reţeaua este testată doar pe valori de intrare, fără rezultatele dorite, observând astfel dacă rezultatul la ieşire este corect sau nu.

În procesul de învățare nesupravegheată, nu oferim reţelei valorile pe care reţeaua ar trebui să le obţină la ieşire. Reţeaua evoluează în mod liber până se stabilizează.

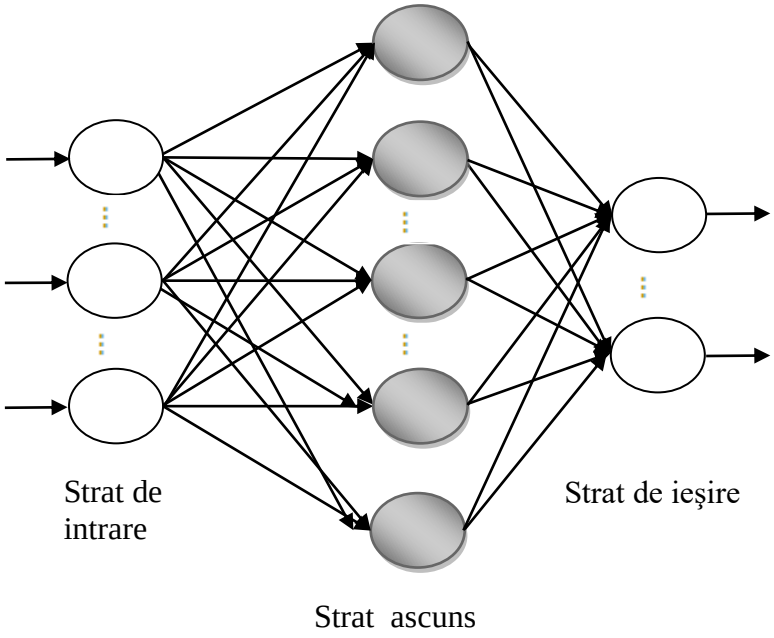


Fig. 1.15. Arhitectura unei reţele neuronale multistrat

RNA acţionează ca un separator neliniar şi poate fi utilizată pentru clasificare, prelucrare de imaginii sau suport decizional. Principala deosebire a reţelelor neuronale faţă de alte sisteme de prelucrarea a informaţiilor este capacitatea de învăţare în urma interacţiunii cu mediul.

1.4.3 Sistem cu logica fuzzy

Logica fuzzy este destinată prelucrării datelor incerte. Spre deosebire de logica binară ($f : R \to \{0,1\}$) logica fuzzy permite unui element/obiect să facă parte din mai multe mulţimi/clase cu diferite grade de apartenenţă, 1 reprezentând apartenenţa totală şi 0 reprezentând neapartenenţa obiectului la o anumită clasă.

Primul pas în elaborarea unui sistem fuzzy (SF) este transformarea valorilor lingvistice (de ex. mare, mediu, mic) în valori numerice – etapă numită *fuzzificare*. Apoi se stabilesc regulile care sunt exprimate prin propoziţii de forma DACĂ ... ŞI/SAU ... ATUNCI ... . *Motorul de inferenţă* aplică seturile de reguli pentru a obţine soluţia căutată. Transformarea reciprocă din seturi fuzzy în valori numerice este numită *defuzificare*.

Schema unui SF este reprezentată în figura 1.16.

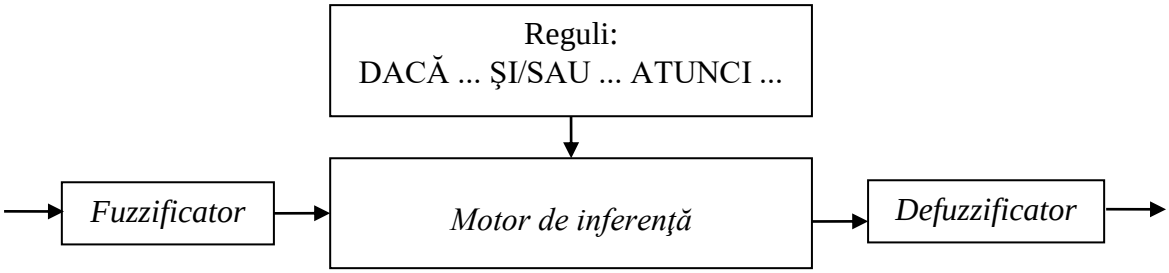


Figura 1.16 Sistem bazat pe logica Fuzzy

Un exemplu de set de reguli este reprezentat în figura 1.17. Se observă că regulile din acest set au diferite ponderi. O descriere mai detaliată a unui sistem de sortare forme bazat pe reguli fuzzy este descris în capitolul 3.

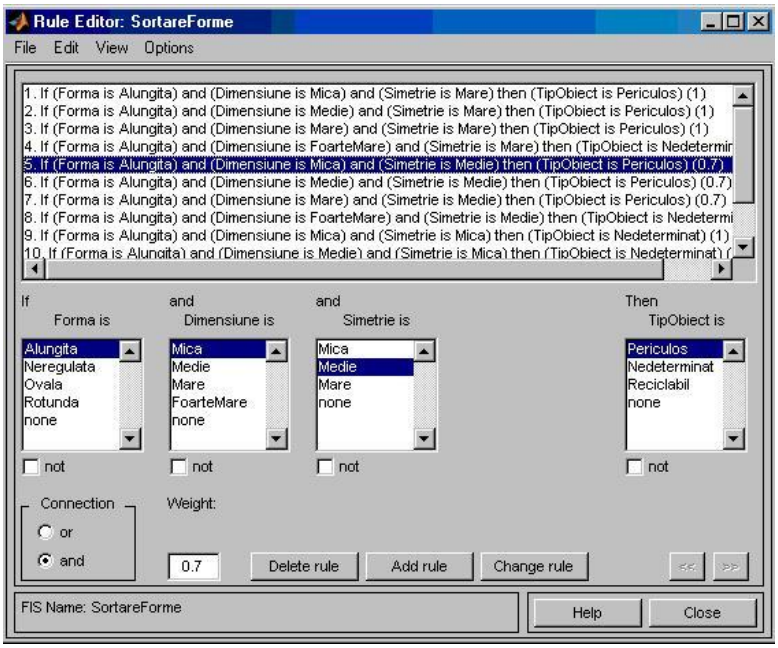


Fig. 1.17. Set de reguli pentru clasificare forme

1.5 Indici de performanţă în aprecierea metodelor de recunoaştere forme şi de clasificare

Pentru a evalua performanţa algoritmilor de clasificare au fost definite următoarele noţiuni şi formule [http://www.cs.rpi.edu/~leen/misc-publications/SomeStatDefs.html#Power]:

1. Adevărat Pozitiv (True Positive - TP) – numărul de obiecte din clasa de interes prezise ca fiind din clasa dată.
2. Adevărat Negativ (True Negative - TN) – numărul de obiecte din altă clasă prezise ca fiind din altă clasă decât cea de interes.
3. Fals Pozitiv (False Positive - FP) – numărul de obiecte din clasa de interes prezise ca fiind din altă clasă decât cea de interes.
4. Fals Negativ (False Negative - FN) – numărul de obiecte din altă clasă prezise ca fiind din clasa de interes.

   Rata Adevărat Pozitiv (True Positive Rate - TPR) – proporţia de date din clasa de interes care sunt etichetate drept clasa de interes şi de către clasificator TPR=TP/(TP+FN). TPR mai poarta numele de *sensitivity* (sau *recall*).
5. Rata Fals Pozitiv (False Positive Rate - FPR)– proporţia de date care nu aparţin clasei de interes dar care sunt etichetate drept această clasă FPR=FP/(FP+TN).

   Rata Adevărat Negativ (True Negative Rate - TNR) – proporţia de date din altă clasă decât cea de interes care sunt etichetate drept altă clasa şi de către clasificator TNR=TN/(TN+FP). TNR se mai numeste şi *specificity*.

6. Rata Fals Negativ (False Negative Rate - FNR) – proporţia de date din altă clasă decăt cea de interes care sunt etichetate drept clasa de interes FNR=FN/(FN+TP).

Tabelul 1.5 Matricea de confuzie

		Clasa prezisă	
		Clasa=1	Clasa=0
Clasa reală	Clasa=1	True Positive (TP or n_{11})	False Negative (FN or n_{01})
	Clasa=0	False Positive (FP or n_{10})	True Negative (n_{00})

Note: n_{ij} este numărul de obiecte din clasa i prezise ca fiind din clasa j.

Numărul total de ***predicţii corecte*** este $n_{11} + n_{00} =$ True Positive + True Negative.

Numărul total de ***predicţii incorecte*** este $n_{01} + n_{10} =$ False Positive + False Negative.

Pentru k clase se obţine o matrice de $k \times k$, clasa de interes este numită pozitivă.

Pe baza matricii de confuzie se pot determina unele metrici de performanţă:

- **Acurateţea**

$$Acuratetea = \frac{Numarul\ de\ predictii\ corecte}{Numarul\ total\ de\ predictii} = \frac{n_{11} + n_{00}}{n_{11} + n_{10} + n_{01} + n_{00}}$$

- **Rata de eroare** reprezintă diferenţă dintre numărul total de predicţii şi acurateţe: $1 - Acuratetea$:

$$Rata\ de\ eroare = \frac{Numarul\ de\ predictii\ incorecte}{Numarul\ total\ de\ predictii} = \frac{n_{10} + n_{01}}{n_{11} + n_{10} + n_{01} + n_{00}}$$

- **Precizia**

$$P = \frac{True\ Positive}{True\ Positive + False\ Positive} = \frac{n_{11}}{n_{11} + n_{01}}$$

- **Recall**

$$R = \frac{True\ Positive}{True\ Positive + False\ Negative} = \frac{n_{11}}{n_{11} + n_{10}}$$

- **F-score**

$$F - score = \frac{2 * P * R}{P + R}$$

La evaluarea acurateţei clasificării (Capitolul 3) a fost calculată acurateţea algoritmului, rata de eroare, precizia, recall şi Fscore.

La evaluarea segmentării (Capitolul 2) s-au folosit alţi indici: criteriul de uniformitate, criteriul definit de Borsotti, metoda de evaluare bazată pe entropie ş.a.

1.6 Concluzii la capitolul 1

Datorită multitudinii tipurilor de imagini (naturale, prin satelit, medicale) şi a factorilor (iluminare neuniformă, zgomot) ce pot influenţa reprezentarea obiectelor în scenă nu a fost elaborată încă o metodă unică de recunoaştere a conţinutului (obiectelor/formelor) care să permită obţinerea rezultatelor satisfăcătoare pentru orice tip de imagine.

Etapa de preprocesare are ca scop îmbunătăţirea calităţii imaginilor: mărirea contrastului, eliminarea zgomotului, dar se preferă captarea calitativă a imaginilor (calibrarea camerei, iluminare suficientă etc.) deoarece nici o metodă de preprocesare nu poate reda o parte lipsă a obiectului/formei.

Metoda de extragere şi evaluare a caracteristicilor se alege în dependenţă de tipul imaginii şi complexitatea ei (numărul de obiecte reprezentate în imagini, rezoluţia, importanţa exactităţii rezultatelor sau acceptarea unei aproximări etc.). Metoda de clasificare depinde de tipul şi numărul de caracteristici extrase.

Analiza efectuată în domeniul studiat – procesarea imaginilor – a evidenţiat importanţa obţinerii rezultatelor corecte la fiecare etapă de procesare şi criticitatea calităţii segmentarii ca etapă anterioară recunoaşterii imaginii şi clasificării automate.

2. SEGMENTAREA IMAGINILOR FOLOSIND MODELE DE MIXTURI GAUSSIENE ŞI MODELE DE MIXTURI UNIFORME-GAUSSIENE

2.1 Descrierea metodelor uzuale de segmentare a imaginilor

Segmentarea reprezintă divizarea imaginii în regiuni uniforme, după un anumit criteriu. Această etapă este importantă în procesarea imaginilor şi urmăreşte, de obicei, extragerea, identificarea sau recunoaşterea obiectelor. Regiunile imaginii astfel formate se numesc segmente şi ele constituie obiectele separate de fundal.

Performanţa segmentării este influenţată de calitatea imaginii şi de complexitatea scenei. O bună segmentare se obţine atunci când obiectele din imagine au contururile bine definite şi nu sunt prezente umbre sau reflecţii de lumină. Aceste efecte duc la rezultate proaste la segmentarea imaginilor, îndeosebi la cele reprezentate în nivele de gri. Imaginile color au avantajul de a include ca criteriu de segmentare şi factorul de culoare.

Separarea unui obiect de fundal poate fi anevoioasă atunci când fundalul însuşi reprezintă o scenă complexă, adică pot fi contopite/confundate obiectele din prim-plan cu cele din fundal. Sunt cazuri când în scenă unele obiecte acoperă parţial alte obiecte. O persoană poate să spună cert, de exemplu, că în scenă este o fructieră în care se găsesc mere, pere, banane chiar dacă fructele sunt suprapuse şi se vede parţial o banană. Pentru un algoritm de detectare a formelor cazul dat este dificil.

În dependenţă de calitatea imaginii şi complexitatea ei se alege un anumit algoritm de segmentare. Ar putea fi nevoie în prealabil de o îmbunătăţire a calităţii pentru a obţine un contrast mai mare şi o accentuare mai bună a contururilor şi doar apoi de aplicat algoritmul de segmentare.

În literatura de specialitate găsim multe tipuri de tehnici de segmentare a imaginilor color (sau nivele de gri), care pot fi grupate în patru categorii principale:

1. Segmentare bazată pe pixel – o regiune este definită ca un set de pixeli ce au aproximativ aceeaşi intensitate luminoasă/culoare.
   - tehnici bazate pe histogramă [116-118];
   - tehnici de clusterizare [119];
   - tehnici fuzzy de clusterizare [120-122].
2. Segmentare bazată pe regiuni. Metodele de detectare a regiunilor folosesc similitudinea şi proximitatea spaţială între pixeli pentru determinarea regiunilor. Se cunosc:

- tehnici de creştere a regiunilor, se alege poziţia unui pixel şi se caută în cele 8 direcţii dacă pixelii vecini corespund unui criteriu de similaritate, formând astfel regiuni omogene [123, 124];
- algoritmi de dividere şi fuziune a regiunilor (Splitting and Merging) [125]. Scopul este de a împărţi imaginea în regiuni, fiecare dintre care este omogenă în un anumit sens, dar unirea a două regiuni adiacente nu mai este omogenă în acelaşi sens.

3. Segmentare bazată pe contur [126, 127] – cazul în care o regiune este definită ca un set de pixeli, delimitate de un contur de culoare. Pentru determinarea contururilor este importantă rata de variaţie a nivelelor de gri (a valorilor pixelilor de culoare).
4. Tehnici hibride de segmentare [128-130] – îmbunătăţesc rezultatele de segmentare prin completarea şi/sau combinarea metodelor de mai sus.

Mai sunt întâlnite în literatură şi alte tehnici, precum cele bazate pe grafuri [131] – algoritmi speciali ce au fost adaptaţi pentru segmentare, tehnici ce utilizează reţele neuronale [132], modele Markov, domeniul wavelet, algoritmi bazaţi pe textură, culoare şi altele.

Aplicarea unei metode sau alta depinde de natura imaginii: mod de captare, rezoluţie, iluminare, nivelul zgomotului, tipul informaţiei utile conţinute (obiecte uniform colorate sau texturi, text, etc.) şi evident de scopul segmentării: localizare obiect, recunoaştere forme, interpretare, control calitate, diagnosticare etc. Caracteristicile ce trebuiesc extrase influenţează şi ele alegerea unei metode de segmentare: contururi, regiuni, forme, texturi etc.

O iluminare neuniformă afectează negativ rezultatele segmentării, îndeosebi la metodele bazate pe histogramă. De asemenea şi prezenţa zgomotului *sare şi piper* afectează negativ rezultatele segmentării şi sunt necesare unele preprocesări înainte de a aplica metodele de segmentare.

Pentru imagini gri este indicată utilizarea metodelor de segmentare bazate pe histogramă, pentru cele color – cele orientate pe regiuni. Metoda de detectare a contururilor poate fi implementată atât pentru imagini pe nivele de gri, cât şi color.

2.1.1 Metode de segmentare bazate pe clasificarea pixelilor

În literatură se întâlnesc mai multe clasificări ale metodelor de segmentare. În urma unei ample analize a articolelor din domeniu şi ţinând cont de principiu de bază al algoritmului implementat am decis următoarea grupare a metodelor de segmentare:

- bazate pe clasificarea pixelilor;
- de detectare a contururilor;

– bazate pe regiuni ş.a.

2.1.1.1 Metode bazate pe histogramă

Tehnicile de segmentare bazate pe histogramă au ca principiu de bază calcularea frecvenţei de apariţie a valorilor pixelilor ce formează imaginea. Aceste tehnici sunt bazate pe prăguirea ("tresholding") histogramelor şi sunt eficiente atunci când există o separare relativ clară a valorilor între obiectele analizate, astfel încât un interval dat de culoare să reprezinte o clasă unică de obiecte.

Metoda de tresholding constă în alegerea unui N număr de praguri: Th_1, Th_2, ... Th_N şi crearea unei imagini de etichete, pe baza imaginii iniţiale, astfel:

if $g(i,j) < Th_1$

then $g(i,j) \in segment_1$

if $g(i,j) >= Th_1$ && $g(i,j) <= Th_2$

then $g(i,j) \in segment_2$

...

if $g(i,j) > Th_{N-1}$ && $g(i,j) <= Th_N$

then $g(i,j) \in segment_{N-1}$

if $g(i,j) > Th_N$

then $g(i,j) \in segment_N$

unde *g(i,j)* – valoarea unui pixel, $segment_1$ - $segment_{N-1}$ – obiectele din imagine.

În general, *pragurile* sunt bine selectate în cazul în care vârfurile histogramei sunt înalte, înguste, simetrice, şi separate de văi adânci.

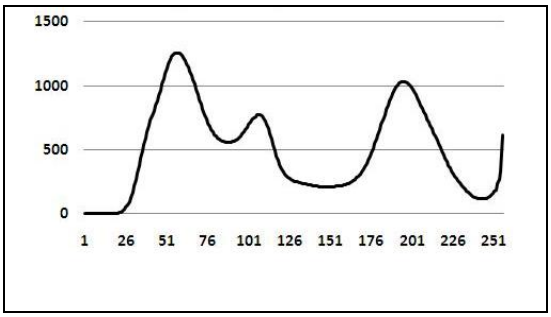


Fig. 2.1. Histograma netezită a imaginii *butterfly* [133]

Pe această histogramă se observă 3 vârfuri bine determinate, prin urmare avem 2 praguri. Prin metoda Otsu descrisă mai jos am obţinut Th1=69 şi Th2=141.

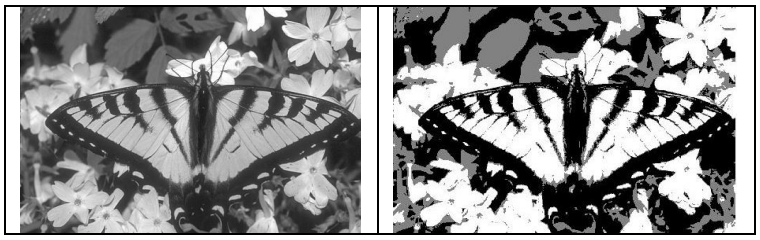


Fig. 2.2. Imaginea originală *butterfly* şi imaginea segmentată (3 segmente)

Una din metodele de referinţă a metodelor de segmentare bazate pe histogramă este **metoda Otsu**. Această metodă are ca scop minimizarea variaţiei în interiorul claselor. Implicit, metoda este creată pentru binarizarea imaginii, deci obţinerea a două clase (obiect şi fundal), dar poate fi adaptată şi la obţinerea mai multor clase.

Regiunile cu o omogenitate înaltă au o variaţie joasă. Pentru fiecare prag *t* (de la 1 la 255) se calculează:

$$\sigma_w^2(t) = q_1(t)\sigma_1^2(t) + q_2(t)\sigma_2^2(t),$$

unde t – pragul determinat, iar σ_i^2 - variaţia claselor respective, $\sigma_w^2(t)$ - suma variaţiilor claselor (within-class variance) ;

$q_1(t)$ şi $q_2(t)$ sunt probabilităţile celor două clase separate de pragul t şi se determină ca suma probabilităţilor ca pixelii unei clase să fie de o anumită intensitate (nivel de gri) pe intervalul indicat (de la 1 la t pentru prima clasă şi de la t până la intensitatea maximă – pentru a doua):

$$q_1(t) = \sum_{i=1}^{t} P(i) \text{ şi } q_2(t) = \sum_{i=t+1}^{I} P(i),$$

Mediile claselor (fundal şi respectiv obiect) se calculează:

$$\mu_1(t) = \sum_{i=1}^{t} \frac{iP(i)}{q_1(t)} \text{ şi } \mu_2(t) = \sum_{i=t+1}^{I} \frac{iP(i)}{q_2(t)}.$$

Variaţia claselor:

$$\sigma_1^2(t) = \sum_{i=1}^{t} [i - \mu_1(t)]^2 \frac{P(i)}{q_1(t)} \text{ şi } \sigma_2^2(t) = \sum_{i=t+1}^{I} [i - \mu_2(t)]^2 \frac{P(i)}{q_2(t)}.$$

Variaţia imaginii este:

$$\sigma^2 = \sum_{i=1}^{I} [i - \mu]^2 P(i).$$

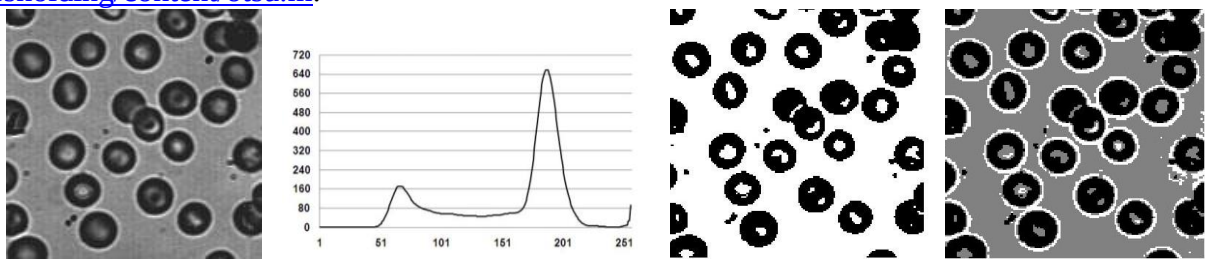
Se poate de demonstrat că variaţia imaginii poate fi calculată şi astfel :

$$\sigma^2 = q_1(t)\sigma_1^2(t) + q_2(t)\sigma_2^2(t) + q_1(t)(\mu_1(t) - \mu)^2 + q_2(t)(\mu_2(t) - \mu)^2 = \sigma_w^2(t) + \sigma_b^2(t)$$

unde $\sigma_b^2(t)$- variaţia între clase (between-class variance). Deoarece variaţia imaginii σ nu depinde de pragul t, putem conclude că în cazul când $\sigma_w^2(t)$ este minim atunci $\sigma_b^2(t)$ este maxim. Pragul t este calculat pe intervalul 1-255, suma minimă $\sigma_w^2(t)$ şi respectiv $\sigma_b^2(t)$ maxim corespunde pragului optimal.

Regiunile cu intensitate uniformă sunt reprezentate în histogramă cu vârfuri foarte ascuţite, bine definite. Segmentarea cu mai mult de un prag e mai dificilă, dar posibilă. De exemplu se aleg 2 praguri, atunci se determină 3 regiuni cu o variaţie de intensitate mică. O implementare a metodei Otsu este disponibilă la adresa

http://www.mathworks.com/matlabcentral/fileexchange/26532-image-segmentation-using-otsu-thresholding/content/otsu.m.

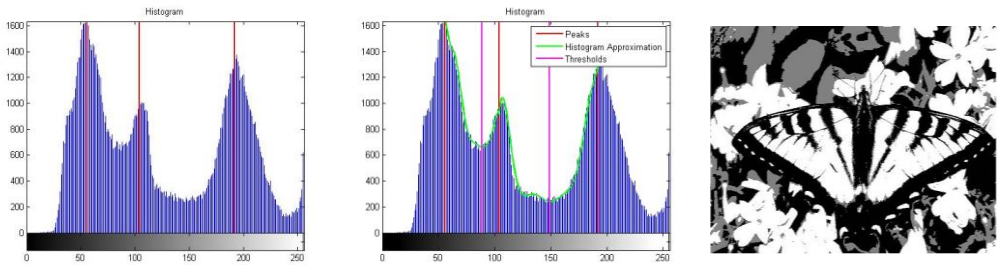


t=85 t1=71, t2=136

a. b. c. d.

Figura 2.3 a) Imaginea test *blood cells* [134]; b) histograma netezită; c) imaginea binarizată (segmentată cu un prag); d) imaginea segmentată cu 2 praguri (3 segmente)

Multe din metodele bazate pe histogramă propuse în literatura de specialitate se referă la binarizarea imaginii, adică determinarea unui singur prag. Evident că pentru imaginile ce conţin mai multe obiecte aceste metode nu sunt eficiente, e necesară separarea imaginii în atâtea segmente câte obiecte sunt în imagine, deci a *n* praguri. N. Papamarkos and B. Gatos în lucrarea "A new approach for multithreshold selection" (1994) [135] au propus un algoritm ce efectuează segmentarea cu mai multe praguri (Multithresholding). Metoda propusă determină vârfurile pe histogramă (valorile maxime), apoi găseşte valorile minime ce se află între 2 maxime.



a. b. c.

Fig. 2.4 a) determinarea maximelor pe histogramă; b) determinarea pragurilor; c) imaginea segmentată

La o iluminare neuniformă vârfurile pe histogramă nu mai sunt ascuţite şi ar putea să nu fie separate prin „văi”, ar putea să arate ca în figura de mai jos.

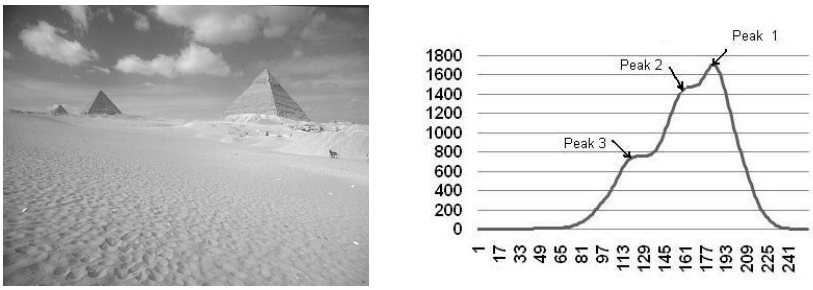


Fig. 2.5. Imaginea test #260058 [136] şi histograma netezită corespunzătoare

Observăm în Fig. 2.5 că histograma conţine 3 gaussiene, iar un algoritm de segmentare bazat pe prăguire ar putea detecta doar una singură (vârfurile nu sunt ascuţite, iar „văile” lipsesc).

În aceste cazuri e recomandat un model de mixturi gaussiene (GMM). Acest model este descris detaliat de D. Reynolds and C.Rose în lucrarea “Robust Text-Independent Speaker Identification Using Gaussian Mixture Speaker Models” [137]. În lucrarea dată autorii se referă la modelul destinat semnalelor acustice. Modelul a fost preluat şi implementat apoi pentru segmentarea imaginilor, de exemplu [138-140].

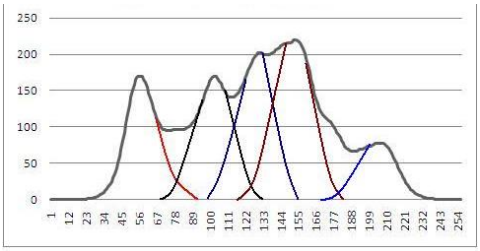


Fig. 2.6. Model de Mixturi Gaussiene (5 Gaussiene)

Un **model de mixturi gaussiene** [141] este o funcţie a densităţii de probabilitate reprezentată ca o sumă ponderată de M componente cu densităţi gaussiene şi poate fi scris:

$$p(x \mid \lambda) = \sum_i^M w_i g(x \mid \mu_i \sum_i)$$

unde x – vector aleator cu dimensiune D, w_i, $i = 1,...,M$ – ponderile mixturii şi $g(x \mid \mu_i \sum_i)$ – densităţile componentelor.

Densităţile componentelor sunt funcţii D-variate şi pot fi exprimate astfel:

$$g(x \mid \mu_i \Sigma_i) = \frac{1}{(2\pi)^{D/2} \mid \sum_i \mid^{1/2}} \exp\left\{ -\frac{1}{2}(x-\mu_i)^{'}(\sum_i)^{-1}(x-\mu_i) \right\}$$

cu media distribuţiei μ_i şi matricea de covarianţă $\sum_i$. Ponderile mixturilor trebuie să satisfacă condiţia $\sum_i^M w_i = 1$.

Un model de mixturi gaussiene se consideră complet dacă este caracterizat de medii de distribuţii, matrice de covarianţă şi ponderi ale tuturor componentelor. Acest model poate fi exprimat ca o funcţie a parametrilor enumeraţi mai sus:

$$\lambda = \{w_i, \mu_i, \sum_i\} \quad i = 1,...,M.$$

GMM este slab implementat la segmentarea imaginilor. Z. K. Huang şi K. W. Chau [138] au elaborat un algoritm ce poate fi aplicat doar pentru histograme bimodale. H.Tang et al. [139] consideră că modelul de mixturi Gaussiene, bazat doar pe distribuţii de intensitate, este insuficient în cazul în care imaginea e afectată de zgomot. Pentru a rezolva această problemă ei propun un model care include şi ponderea de vecinătate (neighborhood weighted Gaussian mixture model). Experimentele efectuate de autori au demonstrat că metoda propusă de ei obţine un rezultat mai bun la clasificare şi este mai puţin afectată de zgomot.

Completarea algoritmilor clasici (de bază) duce la obţinerea unor rezultate mai bune.

2.1.1.2 Tehnici de clusterizare

Tehnicile de clusterizare reprezintă proceduri iterative de partiţie a unei imagini într-un număr de seturi sau clustere disjuncte. Metodele de clusterizare precum K means, fuzzy C-means (FCM) şi versiunile lor completate, îmbunătăţite sunt propuse în literatură [122].

Algoritmul FCM şi versiunea ameliorată (improved FCM) sunt caracterizate de o flexibilitate mare, pixelii aparţin mai multor clase cu grade de apartenenţă diferite. Metoda de clusterizare K-means este una din cele mai des aplicate din motivul simplităţii şi eficienţei computaţionale. Numărul iteraţiilor este redus în versiunea imbunătăţită a metodei clasice (improved K-means). Metoda de segmentare *K-means* reprezintă o procedură iterativă de obţinere a centrelor claselor pentru un set de date de intrare.

Se urmăreşte identificarea a K clase în setul de date de intrare (imaginea de segmentat) astfel încât datele din fiecare clasă să fie suficient de similare pe când datele din clase diferite să fie suficient de diferite.

Paşii algoritmului:

1. Numărul de clase se stabileşte de la început (k-clase) ;

2. Se iniţializează aleatoriu centrele fiecărei clase (k-centroizi) ;
3. În mod iterativ se execută:
   - Atribuirea datelor la clase folosind criteriul distanţei minime (distanţa Euclidiană);
   - Recalcularea centrelor ca fiind media elementelor asignate fiecărei clase;
4. Se termină procesul atunci când nici o clasă nu se mai modifică.

Matlab are predefinită funcţia kmeans, pentru a o apela se scrie:

```
>> I = double(imread('peppers.jpg'));
>> J = reshape(kmeans(I(:),3),size(I));
>> imshow(J,[]);
```

3 clusters 5 clusters

Fig. 2.7. Imaginea segmentată „peppers” [142] segmentată cu algoritmul K-means

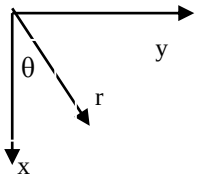
unde kmeans(I(:),3) stabileşte numărul de segmente şi anume 3.

Clusterizarea este cu succes utilizată la segmentarea, clasificarea imaginilor, datorită simplităţii algoritmului şi rezultatelor satisfăcătoare obţinute. O acurateţe la segmentare prin metoda clusterizării se poate obţine şi cu ajutorul algoritmilor genetici [143].

2.1.2 Metode de detectare a contururilor

Contururile (muchiile) sunt frontierele obiectelor din imagine şi sunt determinate de adiacenţa regiunilor uniforme. Pentru caracterizarea contururilor este importantă rata de variaţie a nivelelor de gri (a valorii pixelilor de culoare).

Fie imaginea f modelată printr-o funcţie continuă de două variabile $f(x,y)$. Derivata funcţiei imagine după o direcţie *r* oarecare este:



$$\frac{\partial f(x,y)}{\partial r} = \frac{\partial f(x,y)}{\partial x}\frac{\partial x}{\partial r} + \frac{\partial f(x,y)}{\partial y}\frac{\partial y}{\partial r};$$

$$\frac{\partial f(x,y)}{\partial r} = \frac{\partial f(x,y)}{\partial x}\cos\theta + \frac{\partial f(x,y)}{\partial y}\sin\theta;$$

$$\frac{\partial f(x,y)}{\partial r} = \frac{\partial f(x,y)}{\partial x}\frac{\partial x}{\partial r} + \frac{\partial f(x,y)}{\partial y}\frac{\partial y}{\partial r};$$

$$\frac{\partial f(x,y)}{\partial r} = f_x \cos\theta + f_y \sin\theta = F(\theta).$$

Tehnica de extragere a contururilor este bazată pe calculul gradientului imaginii $f(x,y)$ de-a lungul lui *r*, orientat pe direcţia θ, de aici şi numele – tehnica gradientului.

Pentru determinarea conturului este important de determinat direcţia *r* după care derivata este maximă (pe ce direcţie este tranziţia cea mai puternică) şi valoarea maximă a acestei derivate (cât de semnificativă este tranziţia).

Deci direcţia de variaţie maximă a funcţiei imagine este:

$$\theta_0(x,y) = \arctan\frac{f_x(x,y)}{f_y(x,y)}.$$

iar modulul gradientului pe direcţia determinată poate fi calculat după formula:

$$\left(\frac{\partial f(x,y)}{\partial r}\right)_{\max} = \sqrt{f_x^2(x,y) + f_y^2(x,y)}.$$

Algoritmul de detectare a contururilor (tehnica gradient) conţine următorii paşi de bază:

Pasul 1: Calculul derivatei pe verticală/orizontală pentru fiecare pixel.

Pasul 2: Calculul variaţiei maxime pentru fiecare pixel.

Pasul 3: Dacă variaţia maximă a unui pixel este suficient de mare, pixelul respectiv este considerat pixel de contur.

Pasul 4: Pentru pixelii de contur se calculează orientarea.

Majoritatea algoritmilor utilizaţi pentru detectarea marginei/muchiei procesează matricea imagine pe ferestre de 3x3.

P(i-1,j-1)	P(i-1,j)	P(i-1,j+1)
P(i,j-1)	**P(i,j)**	P(i,j+1)
P(i+1,j-1)	P(i+1,j)	P(i+1,j+1)

Derivarea este liniară, deci se va implementa printr-un filtru linear:

0	0	0
-1	**0**	1
0	0	0

0	0	0
1	**0**	-1
0	0	0

W_y - derivate pe direcţie orizontală

0	1	0
0	**0**	0
0	-1	0

0	-1	0
0	**0**	0
0	1	0

W_x - derivate pe direcţie verticală

Un dezavantaj ar fi că derivata amplifică zgomotul, deci ea trebuie să fie precedată de o filtrare de netezire. Această filtrare trebuie să fie orientată pe direcţia perpendiculară direcţiei de derivare.

0	1	0
0	**c**	0
0	1	0

$\oplus$

0	0	0
-1	**0**	1
0	0	0

=

-1	0	1
-c	**0**	c
-1	0	1

0	0	0
1	**c**	1
0	0	0

$\oplus$

0	-1	0
0	**0**	0
0	1	0

=

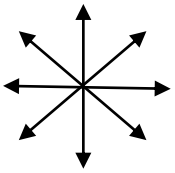
-1	-c	-1
0	**0**	0
1	c	1

Valori particulare pentru constante de pondere *c:*

	W_y	W_x
c=1 gradient **Prewitt**	-1 0 1 -1 **0** 1 -1 0 1	-1 -1 -1 0 **0** 0 1 1 1
$c=\sqrt{2}$ gradient **izotrop**	-1 0 1 $-\sqrt{2}$ **0** $\sqrt{2}$ -1 0 1	-1 $-\sqrt{2}$ -1 0 **0** 0 1 $\sqrt{2}$ 1
c=2 gradient **Sobel**	-1 0 1 -2 **0** 2 -1 0 1	-1 -2 -1 0 **0** 0 1 2 1

Pornind de la o vecinătate de bază restul vecinătăţilor se obţin prin deplasări circulare ale frontierei vecinătăţii cu o poziţie.

N=0

V=6  E=2

S=4

-1	0	1
-2	**0**	2
-1	0	1

Sobel E

-2	-1	0
-1	**0**	1
0	1	2

Sobel SE

-1	-2	-1
0	**0**	0
1	2	1

Sobel S

Am testat algoritmul Sobel Edge Detection disponibil la adresa http://pages.cs.wisc.edu/~dyer/cs534-spring10/slides/matlab-tutorial/sobel.m pe imaginea *peppers.jpg* şi am obţinut următoarele contururi:

Fig. 2.8. Detectarea contururilor utilizând tehnica de gradient Sobel

Contururile pot fi multiple sau pot avea puncte lipsă, marginile pot reprezenta şi linii întrerupte. Sunt cunoscute şi extractoare mai optimale ale contururilor – filtrele Canny şi Deriche, ele păstrează poziţia spaţială a tranziţiei.

John Canny în lucrarea sa „A computational approach to Edge Detection" (1986) [144] menţionează că sunt trei criterii de performanţă ale algoritmului de extragere a contururilor:

1. detecţie bună;
2. localizare bună;
3. doar un răspuns la un singur contur.

Primul criteriul definit de Canny [Canny, 1986] este raportul semnal/zgomot ca coeficient al ambelor răspunsuri.

$$SNR = \frac{\left| \int_{-W}^{+W} G(-x) f(x) dx) \right|}{n_0 \sqrt{\int_{-W}^{+W} f^2(x) dx}}$$

şi localizarea este definită ca:

$$Localization = \frac{|\int_{-W}^{+W} G(-x) f'(x) dx)|}{n_0 \sqrt{\int_{-W}^{+W} f'^2(x) dx}}$$

Se tinde spre maximizarea ambelor (*SNR* şi *Localization*) în mod simultan. Utilizând inegalitatea lui Schwarz pentru integrale se poate demonstra că SNR este mărginit sus astfel:

$$n_0^{-1} \sqrt{\int_{-W}^{+W} G^2(x) dx}$$

şi localizarea este:

$$n_0^{-1}\sqrt{\int_{-W}^{+W} G'^2(x)dx}$$

produsul SNR şi *Localization* este maxim când $f(x) = G(-x)$ pe intervalul [-W W].

Pentru a obţine un contur bine definit a fost propusă restricţia ca funcţia *f* să nu aibă „prea multe" răspunsuri la o singură muchie în vecinătatea acestei muchii. Deci muchiile sunt marcate ca maxime locale în răspunsul filtrului linear aplicat pe imagine.

Metoda descrisă a fost implementată pentru a găsi operatori optimali pentru muchii de forma unei creste/culmi, cât şi pentru forme ascuţite (unghiuri ascuţite), dar e posibilă implementarea pentru a găsi operatori optimali pentru diverse tipuri de margini (muchii).

Canny [144] a propus convoluţia imaginii cu operatorul G_n care este derivata întâi a Gaussienei bidimensionale G în direcţia n, adică:

$$G_n = \exp\left(-\frac{x^2 - y^2}{2\sigma^2}\right)$$

Am implementat algoritmul Canny de detectare a conturului de pe adresa http://robotics.eecs.berkeley.edu/~sastry/ee20/cacode.html pe imaginea peppers.gif şi am obţinut rezultatele prezentate în figura 2.9. Analizând imaginile segmentate după aplicarea tehnicei de gradient Canny pot conclude că au fost detectate foarte bine contururile pe axe, datorită derivatelor aplicate pe direcţiile orizontală şi verticală.

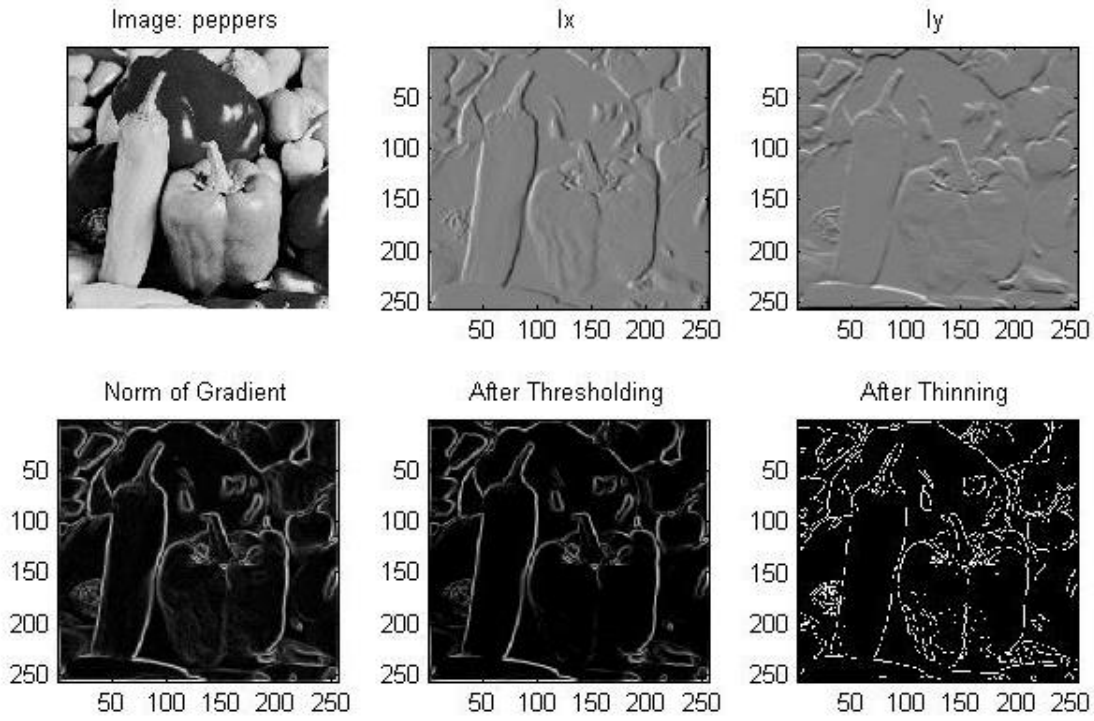


Fig. 2.9. Detectarea contururilor utilizând tehnica de gradient Canny

Metoda detecţiei contururilor prin filtrare liniară se poate exprima ca elaborarea unui filtru a cărui răspuns să marcheze cu maxime bine definite poziţiile tranziţiilor din semnalul de intrare. Filtrul proiectat trebuie să fie optimal, pentru a oferi o bună performanţă, pentru orice poziţie şi intensitate a tranziţiei şi orice nivel de zgomot aditiv. Performanţa de detecţie a contururilor este caracterizată de detecţia şi localizarea bună a tranziţiei, precum şi de minimizarea probabilităţii de detecţie falsă prin maxime posibile din cauza zgomotului.

Deseori contururile reprezintă linii întrerupte, pentru a le uni sub forma de linii continue se utilizează transformata Hough [29]. Această metodă permite estimarea parametrilor unei forme având punctele sale de frontieră.

Exemplu de cod în Matlab pentru determinarea dreptelor se găseşte la adresa următoare: [http://www.mathworks.com/help/images/ref/hough.html]

```
>> BW = edge(I,'canny');
>>[H,T,R] = hough(BW,'RhoResolution',0.5,'Theta',-90:0.5:89.5);
```

Domeniul de variaţie al parametrilor pentru acest exemplu este $[-90^0, +90^0]$ pentru θ. Acurateţea parametrilor este 0.5 pixeli pentru ρ, şi 0.5 grade pentru θ. Valoarea maximă pentru ρ este diagonala imaginii.

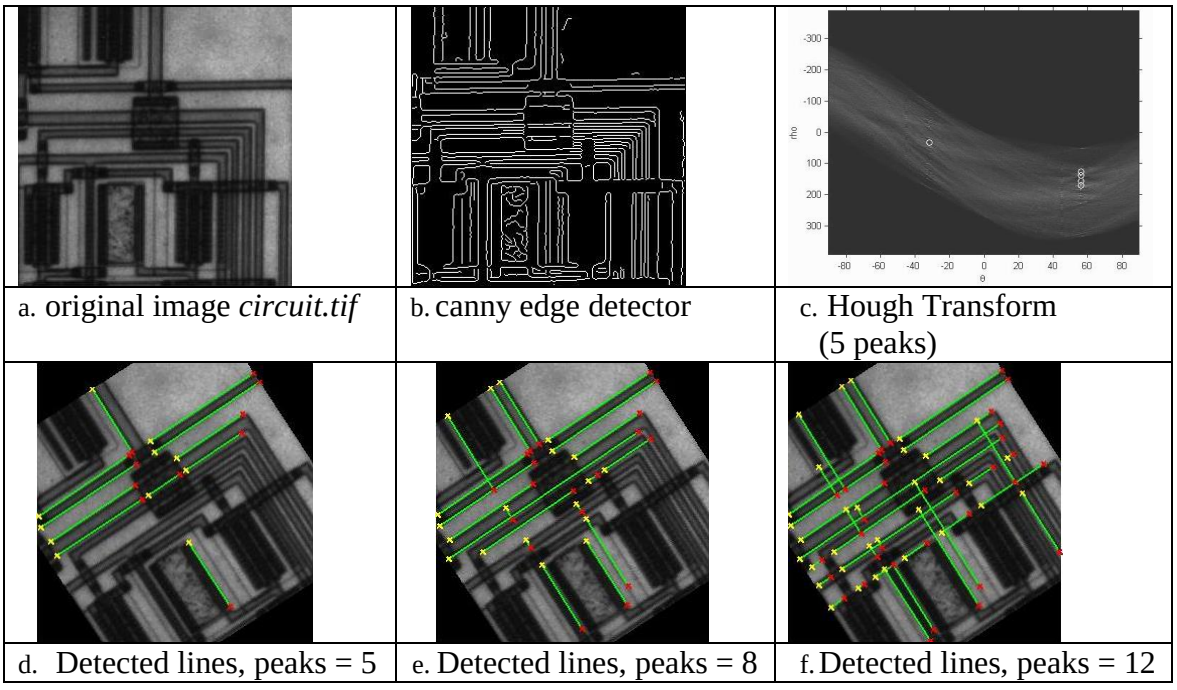


Fig. 2.10. Detectarea liniilor drepte cu Hough Transform [145]

Deşi transformata Hough se foloseşte cel mai des pentru detecţia dreptelor şi cercurilor, ea poate fi extinsă şi pentru detecţia curbelor mai complexe, atâta timp cât o parametrizare adecvată este disponibilă.

2.1.3 Metode de segmentare bazate pe regiuni

Obiectivele metodelor de segmentare bazate pe regiuni sunt obţinerea regiunilor omogene cât mai largi posibile (adică obţinerea a cât mai puţine regiuni) şi cu un nivel de divergenţă între ele. Tehnicile de segmentare orientate pe detecţia regiunilor omogene se pot clasifica în:

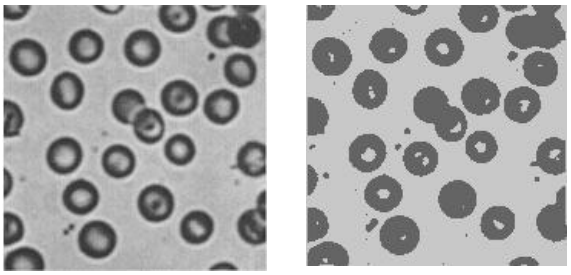
1) metode bazate pe etichetarea grupurilor conexe de pixeli cu caracteristici similare:
   a) Etichetarea componentelor;
   b) Metoda arborelui cuaternar.
2) tehnici de creştere şi fuziune a regiunilor:
   c) Creşterea regiunilor (Region growing);
   d) Dividerea şi fuziunea regiunilor (Splitting and Merging);

a) Etichetarea componentelor – după obţinerea imaginii binare, se începe baleierea linie cu linie din colţul stânga sus al matricei cu o fereastră de 3x3 centrată pe pixelul curent. Primul pixel negru găsit se notează cu 1. Un nou pixel negru care nu are vecini cu 1 se notează cu 2, şi aşa mai departe, adică un nou pixel negru care nu are vecini din clasele 1,2,..,k este notat cu k+1. Un nou pixel negru care are vecini din clasa q va fi de tip q.

Pentru a evita notarea cu doi indici diferiţi a doi pixeli vecini (cel mai des marginile regiunilor sunt considerate obiecte diferite, deoarece la baleiere nu întotdeauna au pixeli vecini din aceeaşi clasă) parcurgem încă o dată matricea, dar din colţul drept jos şi comasăm pixelii vecini cu indici diferiţi, etichetându-i cu valoarea cea mai mică.

De regulă, numărul obţinut de segmente este considerat numărul de obiecte din imagine, însă imaginile pot conţine obiecte alipite şi evident – zgomot.

Exemplu de aplicare a acestui algoritm pentru numărarea celulelor de sânge.



a. Imaginea originală b. Imaginea segmentată

Fig. 2.11. Imaginea originală [134] şi cea segmentată [146]

Pentru a obţine numărul real al celulelor calculăm aria (numărul de pixeli) pentru fiecare obiect din imagine. Orice segment depistat sub un prag este considerat zgomot, iar dacă obţinem o suprafaţă prea mare considerăm ca sunt celule lipite.

Pragul a fost ales $media \pm 2\sigma$, unde *media* este suprafaţa medie a unui segment din imagine, iar σ reprezintă deviaţia standard $\sigma = \sqrt{\sum_{i=1}^{N}(x_i - \bar{x})/N}$.

Numărul de celule obţinut în mod automat a coincis cu numărul de celule calculate manual, ceea ce demonstrează eficacitatea algoritmului descris.

b) Metoda arborelui cuaternar – se bazează pe împărţirea recursivă a imaginii în câte 4 regiuni până la obţinerea de regiuni uniforme sau regiuni formate dintr-un singur pixel. Evident imaginea trebuie să fie pătrată – lungimea egală cu lăţimea. Acestor împărţiri se poate asocia un arbore cuaternar în care fiecare nod terminal are patru descendenţi. Împărţirea imaginii se repetă până când se obţin sferturi uniforme. Apoi se concatenează zonele cu caracteristici similare, rezultând obiectele.

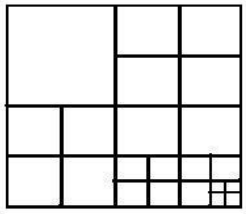


Fig. 2.12. Exemplu de divizare a imaginii în regiuni omogene

c) Creşterea regiunilor (Region growing) – se alege un pixel (sau un set mic de pixeli) şi se compară cu alt pixel adiacent, dacă sunt similari se unesc într-o singură regiune. Dacă două regiuni sunt suficient de similare se contopesc. Similaritatea e bazată pe compararea statisticilor fiecărei regiuni în parte. Regiunile se contopesc până când nu mai satisfac condiţia de alipire.

Un exemplu de creştere a regiunii pornind de la un pixel poate fi consultat la adresa http://www.mathworks.com/matlabcentral/fileexchange/19084-region-growing elaborat de profesorul D. Kroon. Pentru implementare se citeşte o imagine color (*I*), se indică punctul de pornire (centroidul pe axa *x,y*) şi intensitatea maximă a distanţei (implicit e 0.2) :

J = regiongrowing(I,x,y,0.2);

Am aplicat acest algoritm pe imaginea color „peppers.jpg” [22] şi am obţinut rezultatul prezentat în figura de mai jos.

Fig. 2.13. Imaginea *peppers* originală şi imaginea segmentată utilizând metoda *region growing*

d) Dividerea şi fuziunea regiunilor (Split and Merge) – se începe de la conceptul că imaginea întreagă reprezintă o regiune. Dacă această regiune este coerentă, adică toţi pixelii din regiune au o suficientă similaritate, nu se modifică. Dacă regiunea nu e suficient de coerentă, se divide în patru şi se aplică acest pas fiecărei noi regiuni obţinute.

În aşa mod pot fi obţinute regiuni similare. Pasul doi constă în verificarea similarităţii de jos în sus. Se unesc două regiuni adiacente ce satisfac condiţia de similaritate. Deoarece algoritmul porneşte de la regiune mai mare decât un pixel, metoda dată e mai eficientă decât cea precedentă (computaţional mai rapidă).

O metodă de unire a regiunilor este descrisă şi implementată în Matlab de către Sylvain Boltz şi e disponibilă la adresa: http://www.mathworks.com/matlabcentral/fileexchange/25619-image-segmentation-using-statistical-region-merging/content/srm.m. Contururile mai slab definite reprezintă limitele regiunilor iniţiale, cele mai pronunţate – ale regiunilor finale obţinute prin unirea mai multor regiuni iniţiale.

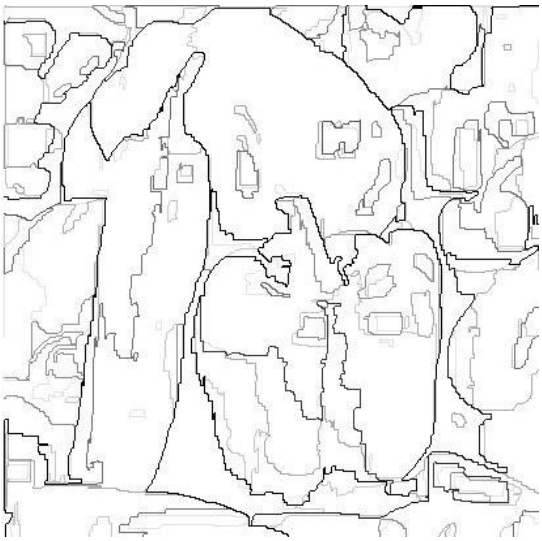


Fig. 2.14. Imaginea *peppers* segmentată utilizând metoda de unire a regiunilor

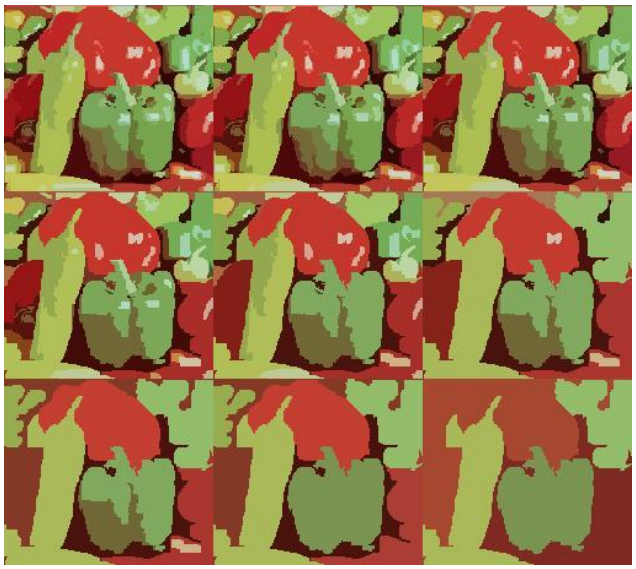


Fig. 2.15. Imagini consecutive a rezultatelor segmentării bazate pe metoda de unire a regiunilor

Un alt algoritm de segmentare, mult mai complex, bazat pe divizare şi unire a regiunilor poate fi consultat la http://www.mathworks.com/matlabcentral/fileexchange/22711-free-split-and-merge-expectation-maximization-for-multivariate-gaussian-mixture.

Poate fi extinsă performanţa segmentării utilizând ambele metode: de detecţie contur şi bazată pe regiuni. De exemplu, putem să adăugăm la decizia de a uni două regiuni care era bazată iniţial doar pe similaritatea pixelilor şi contururile detectate.

După ce am identificat două regiuni adiacente care se propun spre alipire, examinăm şi contururile între ele. Dacă conturul e suficient de definit (magnitudinea gradientului, numărul de pixeli corespunzător unui operator etc.) păstrăm regiunile separat, dacă conturul e slab definit – regiunile se unesc.

2.2 Algoritm de segmentare bazat pe modele de Mixturi Gaussiene (GMM) şi modele de Mixturi Uniforme-Gaussiene (G-U-MM)

Algoritmul propus de segmentare face parte din metodele bazate pe histogramă. Ideea constă în determinarea pragurilor ce separă intervalele cu distribuţie Gaussiană şi uniformă. Se porneşte de la presupunerea ca fiecare obiect din imagine are un anumit tip de distribuţie, deci este reprezentat de o anumită intensitate a pixelilor cu o uşoară variaţie.

Prima versiune a algoritmului de segmentare bazat pe Modelul de Mixturi Gaussiene (GMM) a fost descrisă în articolele “A Method for Image Segmentation based on Histograms – Preliminary Results”, prezentat la Conferinţa Internaţională „Telecomunicaţii, Electronică şi Informatică” [146] şi “Yet Another Method for Image Segmentation based on Histograms and Heuristics”, publicată în Computer Science Journal of Moldova [147].

Metodele de segmentare bazate pe histogramă, de obicei, determină pragurile ca fiind valorile minime de pe histogramă. Noi, profesorul HN. Teodorescu şi eu, propunem o altă metodă de determinare a pragurilor [146-147], schema bloc a căreia este reprezentată mai jos.

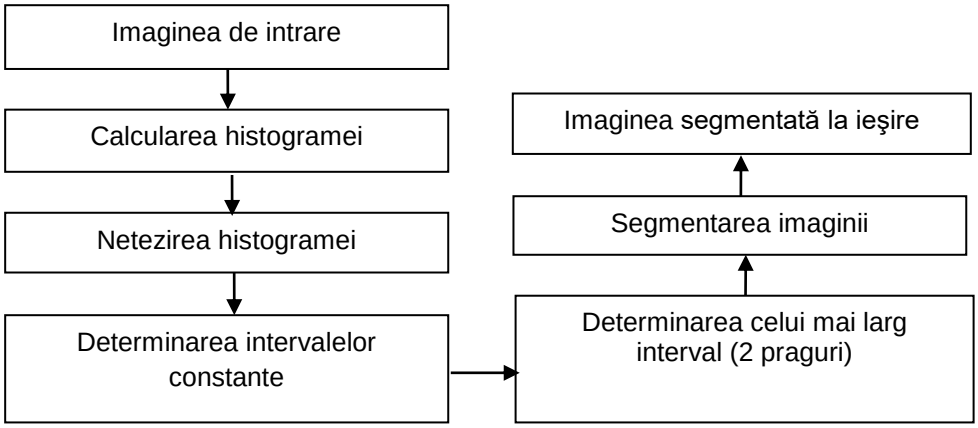


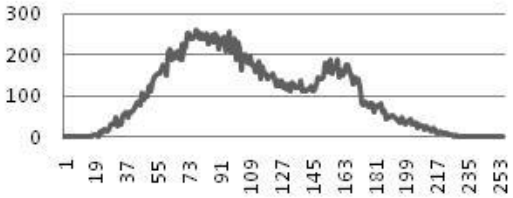
Fig. 2.16 Schema bloc a metodei de segmentare propuse iniţial

Primul pas al metodei îl constituie calcularea distribuirii nivelelor de gri în imagine – histograma, apoi netezirea histogramei utilizând filtrele median şi de mediere. În final se determină cel mai constant interval pe histogramă, iar marginile intervalului sunt considerate praguri.

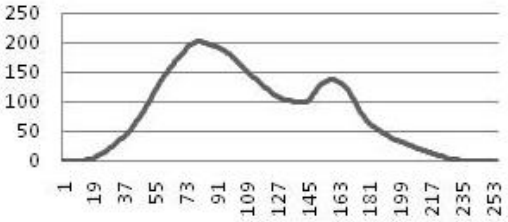
Prin aplicarea filtrelor pe histogramă nu se doreşte decât „netezirea" graficului acesteia cu scopul de a avea mai puţine valori de minim local şi de a identifica mai uşor pragul de segmentare. La metodele de segmentare automată se pot aplica diferite tehnici de aproximare a histogramei fie cu gaussiene, fie de liniarizare pe porţiuni cu segmente de dreaptă, fie de aplicare a unui FTJ (filtru trece-jos) scopul acestora fiind acela de a permite găsirea automată a pragului de segmentare.

Netezirea histogramei (en.: „*histogram smoothing"*) ca etapă de preprocesare pentru calcule ulterioare permite obţinerea rezultatelor mai exacte, mai satisfăcătoare. Pentru a obţine o histogramă netezită se aplică filtrul de mediere de două ori şi cel median o singura dată, utilizând o fereastră de 11 elemente şi respectiv 5.

a.



b.



c.

Fig. 2.17. Imaginea iniţială *giraffe* [148] b.Histograma iniţială, c. Histograma netezită

Cel mai mare interval constant a fost determinat cu o fereastră glisantă de 24 elemente luând în calcul următoarele :

(a) dacă $\frac{V_{\max} - V_{\min}}{Nw} < p$ atunci intervalul este considerat constant, unde $V_{\max}$ şi $V_{\min}$ sunt valorile maxim şi minim ale ferestrei şi Nw este numărul de elemente din fereastră, iar p este indicile de comparare care e direct proporţional cu rezoluţia imaginii (pentru imagini cu rezoluţii mai mari se alege un indice de comparare mai mare). Pentru imaginile test de o rezoluţie 170x170, p=1.5. Dacă două ferestre consecutive au această proprietate atunci se defineşte intervalul fiind egal cu lungimea a două ferestre.

(b) altfel intervalul nu este considerat constant.

Dacă se obţin mai multe intervale constante, pentru cazul cu două praguri, se selectează cel mai mare şi limitele lui sunt considerate praguri. Ideea este de a obţine mai puţine segmente, de a determina obiectele principale din imagine. Nu se urmăreşte separarea fiecărui detaliu al imaginii, care ar putea fi în unele cazuri, o parte umbrită sau puternic iluminată a obiectului.

Segmentarea se face în conformitate cu exemplul de pseudocod, unde Th1 şi Th2 sunt limitele celui mai mare interval semi-constant şi g(i,j) este valoarea pixelului din imagine:

if $g(i,j) < \text{Th1}$ then $g(i,j) \in \text{segment1}$
if $g(i,j) \geq \text{Th1}$ && $g(i,j) \leq \text{Th2}$ then $g(i,j) \in \text{segment2}$
if $g(i,j) > \text{Th2}$ then $g(i,j) \in \text{segment3}$
Dacă utilizăm n praguri, atunci avem $n+1$ segmente.

Programul elaborat în colaborare cu prof. HN. Teodorescu poate fi consultat pe adresa: http://www.etti.tuiasi.ro/cercetare/cerfs/index.php?option=com_content&view=article&id=98%3Acolectiesoftdate&catid=14%3Asectiuneapublicatii&Itemid=21&lang=ro

Exemplu de segmentare după metoda propusă a imaginii *giraffe*. S-au obţinut mai multe intervale semiconstante: 0-23, 72-95, 126-149, 198-254, dar s-a ales intervalul cel mai mare – 126-149, obţinându-se astfel segmentele [0-126), [126-149] şi (149-254].

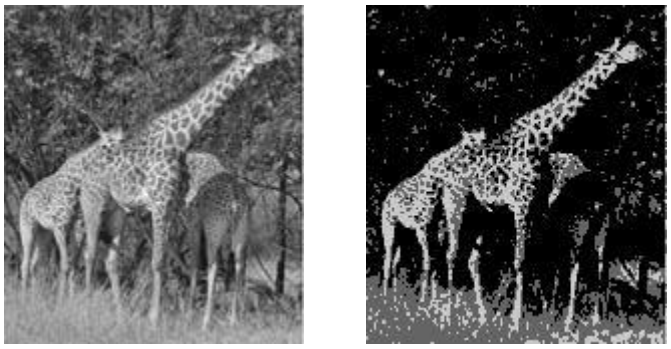


Figura 2.18 Imaginea originală *giraffe* şi cea segmentată

Pentru alegerea optimală a lungimii ferestrei glisante, a numărului de elemente pentru filtrul de mediere a histogramei şi a indicelui de comparaţie s-au făcut mai multe teste.

Tabelul 2.1 Intervale semiconstante obţinute cu diferite valori ale indicilor de comparaţie

Valoarea *p*	**Intervale semi-constante pentru imaginea *giraffe***
1.1	0-23, 72-95, 126-149, 198-254
1.2	0-23, 72-95, 126-149, 198-254
1.3	0-23, 72-95, 126-149, 198-254
1.4	0-23, 72-95, 126-149, 198-254
1.5	0-23, 72-95, 126-167, 198-254
1.6	0-23, 72-95, 126-167, 180-254
1.7	0-23, 72-95, 126-167, 180-254
1.8	0-23, 72-95, 126-167, 180-254

Tabelul 2.2 Ariile segmentelor obţinute utilizând diferite valori ale indicilor de comparaţie

p=1.1-1.4 Aria segmentelor	**Intervalele semi-constante**	**p=1.5 Aria segmentelor**	**Intervalele semi-constante**	**p=1.6-1.8 Aria segmentelor**	**Intervalele semi-constante**
S1=69	[0-23]	S1=69	[0-23]	S1=69	[0-23]
S2=5500	(23-72)	S2=5500	(23-72)	S2=5500	(23-72)
S3=5755	[72-95]	S3=5755	[72-95]	S3=5755	[72-95]
S4=5264	(95-126)	S4=5264	(95-126)	S4=5264	(95-126)
S5=2976	[126-149]	S5=5944	[126-167]	S5=5944	[126-167]
S6=5196	(149-198)	S6=2228	(167-198)	S6=1236	(167-180)
S7=590	[198-254]	S7=590	[198-254]	S7=1582	[180-254]

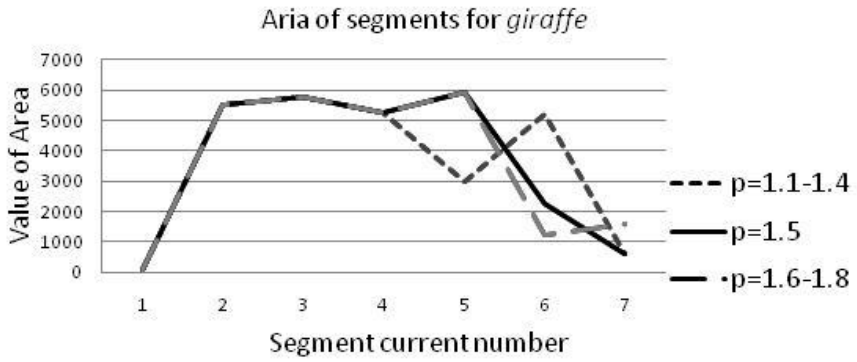


Fig. 2.19. Reprezentarea grafică a ariilor segmentelor obţinute cu diferite valori ale indicilor de comparaţie *p* pentru imaginea *giraffe*

Tabelul 2.3 Intervale semiconstante obţinute cu diferite valori ale ferestrei glisante

Intervalele	**Pragurile pentru *Giraffe*, lăţimea ferestrei**			
	l=20	**l=24**	**l=28**	**l=36**
J1	0	0	0	0
J2	19	23	27	35
J3	70	72	66	120
J4	89	95	93	155
J5	126	126	132	180
J6	145	149	159	255
J7	196	198	198	
J8	254	254	254	

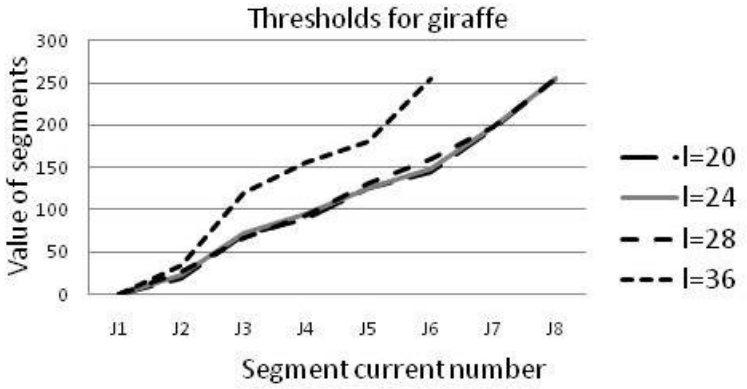
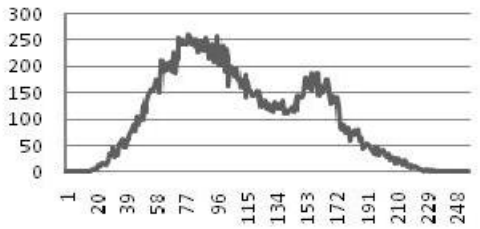
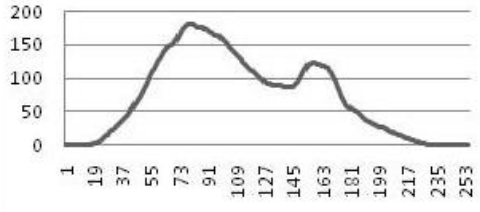
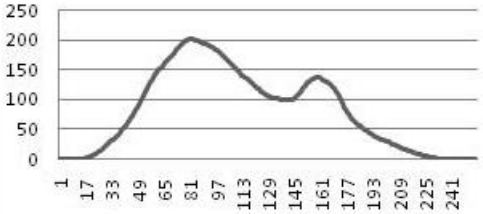
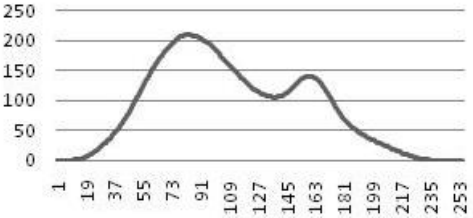


Fig. 2.20. Reprezentarea grafică a valorilor segmentelor obţinute pentru diferite lungimi de ferestre pentru imaginea *giraffe*

Tabelul 2.4 Intervale semiconstante obţinute cu număr diferit de valori pentru filtrul de mediere

Valoarea filtrului de mediere	**Intervalele semi-constante (*giraffe*)**
W[7]	0-23, 72-95, 126-149, 180-254
W[9]	0-23, 72-95, 126-149, 180-254
W[11]	0-23, 72-95, 126-167, 198-254
W[15]	0-23, 72-95, 126-167, 198-254

Tabelul 2.5 Histograme nivelate cu diferite dimensiuni ale filtrului de mediere pentru *giraffe*

	
Histograma originală	Histograma netezită cu W[7]
	
Histograma netezită cu W[11]	Histograma netezită cu W[15]

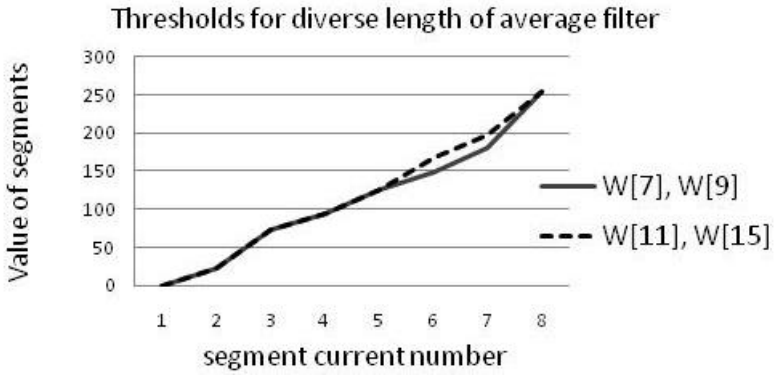


Fig. 2.21. Reprezentarea grafică a segmentelor semiconstante obţinute cu dimensiuni diferite ale filtrului de mediere

Similar, cu indici diferiţi, a fost testată metoda de segmentare şi pe alte imagini. Rezultatele preliminare au fost prezentate la a 4-a Conferinţă Internaţională „Telecomunicaţii, Electronică şi Informatică", Chişinău 2012 [146]. La această etapă au fost stabilite lăţimea ferestrei glisante şi mărimea măştilor pentru netezirea histogramei. O analiză mai amplă a metodei şi a rezultatelor obţinute a fost prezentată în lucrarea "Image Segmentation Based on G-UN-MMs and Heuristics - Theoretical Background and Results –" [149].

Metoda propusă constă în determinarea intervalelor semiconstante care ar reprezenta intervalele cu distribuţie uniformă, ce ar separa intervalele cu distribuţie Gaussiană. De facto, nu în toate cazurile intervalele uniforme reprezintă distribuţii uniforme şi separă Gaussiene. În exemplul de mai jos, interval constant [126-149] reprezintă vârful plat al reuniunii a mai multor Gaussiene.

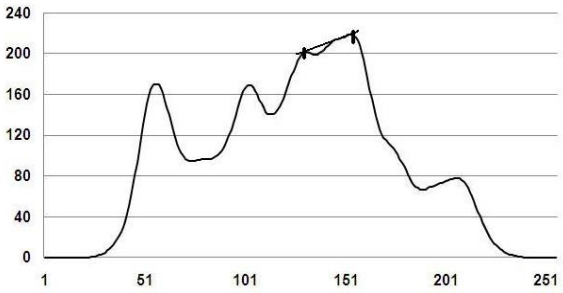


Fig. 2.22. Exemplu de determinare eronată a intervalului de distribuţie uniformă

A fost necesară îmbunătăţirea algoritmului pentru a evita astfel de cazuri. Descrierea amplă a noilor paşi de verificare a intervalelor pentru evitarea clasificării eronate este prezentată în articolul "Improved Heterogeneous Gaussian and Uniform Mixed Models (G-U-MM) and Their Use in Image Segmentation" [150]. Secţiunea următoare este o parafrazare a conţinutului acestui articol.

După cum a mai fost menţionat, rolul principal al segmentării este găsirea în imagini a elementelor semnificative, importante pentru interpretatorul uman şi obţinerea unei imagini simplificate, cu puţine nivele de gri, care reduce elementele esenţiale ale imaginii iniţiale. Pentru observatorii umani segmentarea pare a fi o etapă naturală a perceperii imaginii, totuşi nu e clar câte caracteristici de recunoaştere a formelor sunt implicate la segmentarea imaginilor de către om. Prin urmare, încercarea de a elabora o metodă tehnică de segmentare bazată numai pe tehnici de prelucrarea a imaginii, fără cunoştinţe de nivel superior asupra proprietăţilor imaginii şi fără aplicarea răspunsului despre potenţialele forme ce au fost descoperite ar putea fi imposibilă atunci când segmentele identificate necesită conectarea la semnificaţii.

Nu există nici o legătură evidentă între segmentele din imagine şi distribuţia nivelelor de gri a pixelilor, de fapt, legătura nu este directă. Prin urmare, utilizarea de MMG sau a altor modele similare la segmentarea imaginilor poate părea o tehnică greşită spre utilizare. Cu toate acestea, unele ipoteze de bază, adesea verificate experimental, aşa cum am aflat în mod empiric în numeroase experimente, conectează utilizarea de MMG şi MMGU pentru segmentarea imaginilor. În primul rând, imaginile care nu reprezintă o multitudine de obiecte, elementele cu semnificaţie semantică sunt adesea destul de bine aproximate de componentele Gaussiene. Prin urmare, identificarea componentelor Gaussiene ajută în mod corect la obţinerea segmentelor. În al doilea rând, o mare parte a fundalului precum şi mai multe elemente cu semnificaţie semantică din imagine au nivelul de gri adesea aproape uniform, ceea ce înseamnă că distribuţia lor este uniformă. Prin urmare, statisticile locale, care se referă la distribuţiile cum ar fi Gauss, sau uniform, tind să fie semnificative pentru procesul de segmentare. Prezenţa acestor regiuni (figurile 2.23 şi 2.25) arată că numai modelele de mixturi Gaussiene nu sunt suficiente, pentru că ele omit de fapt, regiunile cu distribuţii uniforme ale nivelelor de gri. Modelarea pe baza distribuţiei uniforme cu MMG, dacă este posibilă, reprezintă totuşi o metodă de aproximare slabă pentru că necesită un număr mare de Gaussiene. Dovezile empirice sugerează că, astfel, un amestec de Gaussiene şi distribuţii uniforme ar permite o mai simplă metodă de calcul şi mai puţin costisitoare, şi, cel mai important, mult mai uşor de interpretat descompunerea imaginii în segmente semnificative.

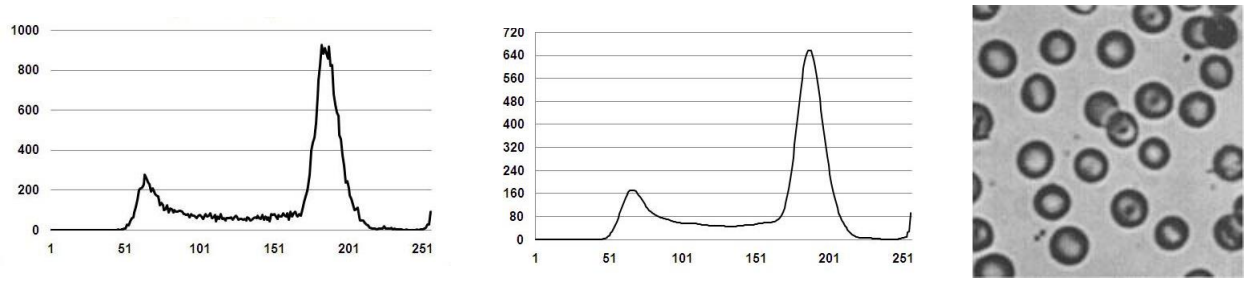


Fig. 2.23. Histograma tipică (a imaginii test *Blood cells* [134] 170x170 pixeli) care poate fi descompusă într-un interval mare constant în mijloc şi două funcţii Gauss

Prima distribuţie Gaussiană din histograma imaginii *Blood cells* din Fig. 2.13 pare suprapusă cu distribuţia uniformă (DU), dar nu şi cea de a doua.

Perspectiva descompunerii imaginii prezentată mai sus arată că i) nu există nici un motiv să credem că metoda MMG este o modalitate potrivită de a detecta segmente într-o imagine, ii) de fapt, este evidentă dovada că MMG ar putea fi o metodă de calcul intensivă şi neoptimală pentru mai multe clase de imagini; iii) nu ne putem aştepta să obţinem rezultate similare segmentării manuale umane, cu metode bazate exclusiv pe funcţii densitate de probabilitate, atunci când textura joacă un rol în definirea segmentelor.

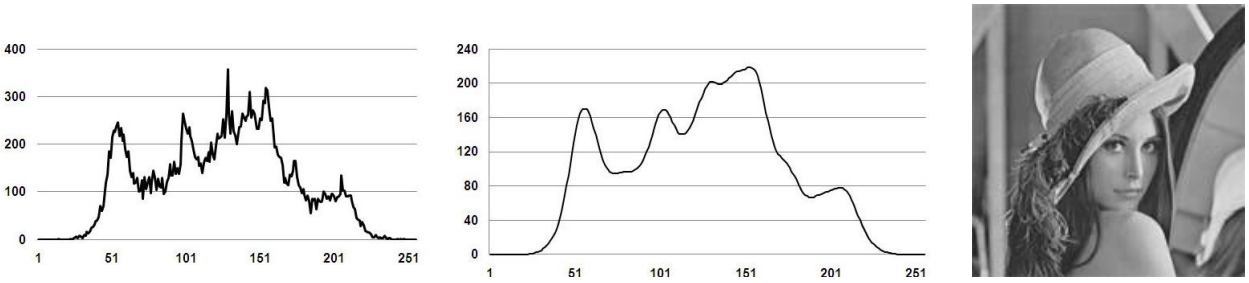


Fig. 2.24. Histogramă tipică (a imaginii test *Lena* [151]), cel mai bine modelată cu MMG. Histograma netezită este percepută în mod clar ca fiind compusă dintr-un amestec de Gaussiene

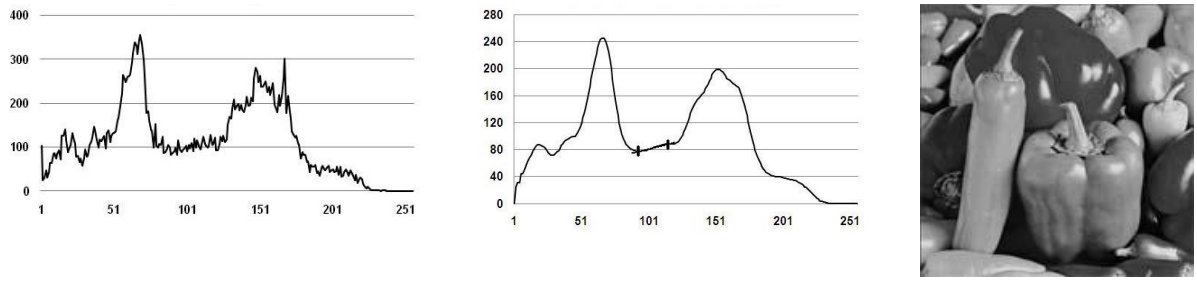


Figura 2.25 Histogramă (a imaginii test *peppers* [142]) a versiunii cu nivele de gri. Intervalul central al histogramei are distribuţia aproape constantă

2.2.1 Optimizarea segmentării bazate pe funcţii de distribuţii

Subiectul general al segmentării este abordat în acest subparagraf prin următoarea ipoteză:

Segmentarea imaginilor bazată pe descompunerea liniară a p.d.f. în conformitate cu un set de funcţii specificate este corectă (acceptabil pentru orice scop.)

Aceasta este doar o ipoteză de lucru. Aşa cum am subliniat deja, ipoteza este discutabilă ca fiind optimală şi falsă în cel mai rău caz. Notăm cu *f(g)*, p.d.f. a imaginii, unde *g* este nivelul de gri şi f este o funcţie continuă $f : R \to R$. considerăm că segmentarea se bazează pe o descompunere a lui f în continuare într-un set Ω de funcţii. Presupunem că Ω este o familie completă de funcţii pentru spaţiul funcţiilor continue cu valori reale pe R, care este, pentru orice *f*, un (posibil infinit) subset a lui Ω, $\Omega_f \subset \Omega$ şi un set de numere reale, a_k, astfel încât $f(g) = \sum_k a_k \omega_k(g)$, $\omega_k \in \Omega_f$. Pentru moment, se presupune că setul Ω este specificat. O reprezentare finită aproximativă a lui f este o sumă a unui număr finit de funcţii ω_k, $\tilde{f}(g) = \sum_{j=1...n} a_j \omega_j(g)$. În cadrul sarcinii de segmentare, fiecare funcţie $a_j \omega_j$ în descompunere finită reprezintă un segment. Nu se pune întrebarea despre adecvarea atribuirii semnificaţiei de "segment" funcţiilor $a_j \omega_j$.

Funcţia $\tilde{f}$ este o aproximare, cu oarecare eroare, a lui *f*. Se urmăreşte o aproximare globală a erorii pătratice. Acest criteriu de optimalitate nu are nimic în comun cu realizarea segmentării specificate. Eroarea pătratică ε^2 cu o aproximare globală este definită ca:

$$\varepsilon^2(f, \tilde{f}) = \int_{g1}^{g2} (f(g) - \tilde{f}(g))^2 dg.$$

Unde g_1, g_2 sunt limitele intervalului de nivele gri. Computaţional, soluţia optimă a aproximării este realizarea unei aproximări cu o eroare mai mică decât o valoare specificată, ε_0 Utilizarea expresiei descompunerii finite pentru un set specificat a *n* indici, $j_1, j_2, ..., j_n$:

$$\varepsilon^2(f, \tilde{f}) = \int_{g1}^{g2} (f(g) - \sum_{j_1, ..., j_n} a_k \omega_k(g))^2 dg.$$

Problema devine: Caută q astfel încât să fie cel mai mic număr al funcţiilor ω_k şi să comită o eroare mai mică de ε_0. Formal condiţia de mai sus este simplă, aplicarea acesteia s-ar putea să nu fie la fel. Acesta este unul din motivele pentru care preferăm o soluţie euristică.

Problema este simplă numai atunci când se poate determina cu uşurinţă principalul component al descompunerii funcţiei *f*.

2.2.2 Model aditiv format din mixturi de distribuţii Gaussiene şi uniforme

Modelul G-U-MM propus se încadrează în categoria bine-cunoscută a modelelor aditive [152, 153].

$$f(x) = f_1(x) + f_2(x) + ... + f_n(x),$$

care este potrivit pentru modele de semnale în forma $X = S + N$, unde *S* este un semnal într-un spaţiu de dimensiuni mici *E* şi *N* este un zgomot Gaussian independent cu medie nulă şi matrice de covarianţă; în cazul unui singur component Gaussian în zgomot, matricea se reduce la dispersia Gaussiană [152].

Aceasta ne aduce la trei probleme diferite: prima se referă la filtrarea imaginii (eliminarea zgomotului), a doua la netezirea histogramei, iar a treia la segmentare.

Referitor la prima problemă, (deoarece se presupune valoarea zgomotului aditiv aproximativ egală cu zero pentru imaginile analizate) sugerăm o filtrare repetată cu o fereastră mică (de 3x3) a filtrului de mediere pentru reducerea semnificativă a zgomotului *N*. Când este prezent zgomotul *sare şi piper*, un amestec de filtre de mediere repetate şi median pot curăţi eficient semnalul. Detalii cu privire la această procedură preliminară de filtrare sunt furnizate în [149]. Prin urmare, vom considera ulterior că singurul proces pe care avem de realizat este segmentarea.

Imaginile totuşi pot include "zgomote semantice", care reprezintă obiecte mici care apar în imagine, dar nu au nici un sens pentru analizator. Un astfel de "zgomot semantic" este eliminat după netezirea histogramei. Cu toate acestea, "zgomotul semantic" poate include umbre şi detalii de fundal care sunt lipsite de interes, sau detalii de fundal şi umbre care se suprapun cu părţi ale semnalului util şi care au nivele de gri identice sau aproximative (sau culori) cu nivelele de gri ale obiectelor de interes din imagine.

În ceea ce priveşte modelul aditiv referitor la segmente, vom face presupunerea că fiecare dintre funcţiile f_k, a acestui model aditiv este diferită de zero pe un interval în care celelalte funcţii, $f_{j \neq k}$, sunt nule (care este descompunerea pe porţiuni). Numai ultimul sens al modelului aditiv este urmărit în continuare.

2.3 Algoritmul euristic G-U-MM. Limitele şi îmbunătăţirile ale G-U-MM de bază

Unele dintre ipotezele de bază expuse mai departe nu sunt verificate. În primul rând se presupune că ochiul recunoaşte imaginile folosind atât informaţia despre culoare – nivelurile de gri cât şi textură. Textura poate fi de obicei caracterizată prin statistici de ordinul trei legate cu momente de ordinul al doilea, în timp ce la elaborarea algoritmului am luat în considerare numai statistica de bază (p.d.f).

Modelul aditiv $f(x) = f_1(x) + f_2(x) + ... + f_n(x),$ este acelaşi, fie în cazul în care funcţiile compun acelaşi domeniu de definire, sau în cazul în care reprezintă extensii la un domeniu comun de definire a funcţiilor cu domenii disjuncte de definire. Cu toate că în mod formal această distincţie este lipsită de importanţă, simplificări semnificative algoritmice pot apărea în al doilea caz, în care diferenţa semantică la segmentare poate fi demnă de remarcat.

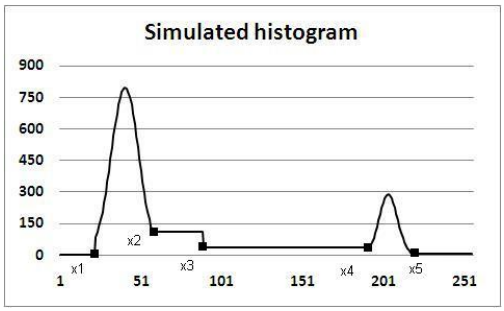


Fig. 2.26. Histogramă simulată compusă din două distribuţii uniforme şi două Gaussiene

Un exemplu realist de histogramă poate arata ca în Fig.2.26. Histograma este compusă de o sumă de patru componente (fără a lua în calcul intervalele mai mici de x1 şi mai mari de x5). Al doilea şi al treilea sunt două distribuţii uniforme suprapuse:

$$h(x) = u3(x) + u4(x),$$

unde distribuţiile au forma $u3(x) = a3,$ pentru $x \in [x2, x3],$ 0 altfel; $u4(x) = a4,$ pentru $x \in [x3, x4], 0$ altfel.

Prima şi a patra componentă sunt Gaussiene. Pe intervalele rămase ale nivelelor de gri, unde *h* nu este constant (nu are o distribuţie uniformă), *h* este modelat ca mixturi Gaussiene. Rezultatul final este compus din setul de intervale care corespund distribuţiilor uniforme şi Gaussiene. Pentru fiecare astfel de interval, o valoare de gri se atribuie segmentului, în cazul distribuţiei uniforme valoarea de segment este valoarea medie a nivelelor de gri ale intervalului. Aşa cum este prezentat în lucrările preliminare [147], [149], preprocesarea şi etapele de analiză de bază sunt efectuate cu o fereastră glisantă (până la 24 valori de gri). La această etapă a

analizei de mixturi, intervalele mai mici decât două ferestre, au fost împărţite în două şi au fost conectate la intervalele alăturate.

Pentru evitarea clasificării vârfurilor de Gaussiene plate, cu o distribuţie uniformă locală, a fost folosită o metodă semi-empirică şi algoritmul aferent descris în această secţiune. Metoda este semi-empirică pentru că scopul nu este de a stabili cu adevărat (exact) descompunerea în Gaussiene pentru intervalul dat. În schimb, a fost testat dacă o aproximare rigidă a unei Gaussiene pe intervalul dat nu este suficient de bună, se extinde la un interval mai mare aproximarea pentru a vedea dacă obţinem rezultate mai bune decât pentru distribuţia uniformă. În cazul în care rezultatul este mai bun pentru aproximarea rigidă a Gaussienei, s-a decis că intervalul corespunde unei secţiuni de Gaussiană pe histogramă.

S-a efectuat o aproximare rigidă a Gaussienelor pe toate intervalele cu distribuţie uniformă (UD). Şi anume, pentru fiecare interval cu DU obţinut de la testul DU explicat în lucrările anterioare [147], [149], se consideră mijlocul intervalului respectiv şi se determină distribuţia conform ecuaţiei ce se referă la valoarea Gaussienei, eroarea acceptată şi lăţimea intervalului,

$$A(1-e^{-(x-m)^2/2\sigma^2}) \le \varepsilon, \qquad (2.1)$$

unde ε este eroarea utilizată în criteriu, m este mijlocul intervalului determinat ca DU, σ este distribuţia Gaussiană, iar x este una dintre extremităţile intervalului. Factorul A este valoare de vârf a intervalului, pentru a elimina în continuare efectul zgomotului, se determină media celor mai mari trei valori a intervalului. Se rezolvă cazul egalităţii pentru (2.1) şi se determină σ ca:

$$1-\varepsilon / A = e^{-(x-m)^2/2\sigma^2}.$$

După rescrierea logaritmică, obţinem:

$$\sigma^2 = -(x-m)^2 / (2\ln(1-\varepsilon / A))$$

Două valori ale lui σ^2 se obţin câte una pentru fiecare extremitate a intervalului fals DU, notată aici prin [x1, x2]. Anume, cele două valori sunt:

$$\sigma_{1,2}^2 = -(x_{1,2}-m)^2 / (2\ln(1-\varepsilon / A)).$$

Se presupune că valoarea m corespunde vârfului histogramei în intervalul dat, $m = \arg\max_{(k)}(h_k)$, unde h_k denotă valoarea k a histogramei. (Când estimarea este corectă, $m = \arg\max_{(k)}(h_k)$ este într-adevăr valoarea medie conform distribuţiei Gaussiene.) Se calculează media $(\sigma_1+\sigma_2)/2$ ca valoare pentru σ. Apoi, se calculează eroarea medie pătratică

cu aproximare pentru intervalul Gaussian obţinut. Dacă această eroare este mai mică decât eroarea medie pătrată determinată în intervalul respectiv folosind valoare constantă de aproximare, se decide că intervalul dat nu este uniform, ci parte a unui Gaussiene. În acest caz, segmentul fals DU este alipit la segmentul adiacent.

Algoritmul de determinare dacă un interval aparent cu DU este de fapt „capacul" unei Gaussiene cu distribuţie largă este descris în continuare. Presupunând că Gaussiana (în cazul în care este efectiv prezentă) nu este prea afectată de zgomot, atunci:

1. Se alege un interval cu DU $[k_1, k_2]$ şi se determină valoarea maximă în interiorul acestuia $\max_{(k)} h_{(k)}$, şi valoarea corespunzătoare lui k, k_m. Dacă

   $\left| k_m - \frac{(k_1 + k_2)}{2} \right| < \varepsilon(k_2 - k_1)$, atunci se decide că k_m este centrul funcţiei Gauss. Dacă nu, se determină cele mai mari trei valori în intervalul $[k_1, k_2]$ şi se calculează media lor. Se aplică testul de centralitate asupra mediei celor 3 valori.
2. În cazul în care testul de centralitate nu este satisfăcut, intervalul $[k_1, k_2]$ este un interval cu DU, sau este de tip necunoscut, dar se forţează aproximarea la o distribuţie uniformă.
3. Dacă cel puţin un test de centralitate este satisfăcut, apoi se determină σ_1, σ_2 şi σ aşa cum se explică mai sus.
4. Calculăm eroarea pătratică pentru funcţia presupusă Gauss,

   $\varepsilon_G^2 = \sum_{k=k_1}^{k_2} (G_k - h_k)^2$

   unde G_k denotă valoarea funcţiei Gauss în punctul k; calcularea erorii pătratice pentru DU presupusă

   $\varepsilon_U^2 = \sum_{k=k_1}^{k_2} (C - h_k)^2$

   unde C este o constantă reprezentată de media valorilor din interval, iar ε_U^2 este deviaţia standard a intervalului.
5. Dacă $\varepsilon_G^2 < \varepsilon_U^2 (1 + \varepsilon_{G-U})$, unde ε_{G-U} este un număr mic (de exemplu 0.5), provizoriu se decide că intervalul corespunde unei Gaussiene.
6. Dacă intervalul corespunde provizoriu unei Gaussiene, se extinde intervalul cu un nivel de gri spre stânga. Se calculează din nou pe noul interval valorile pentru ε_G^2 şi

pentru ε_U^2. Dacă $\varepsilon_G^2 < \varepsilon_U^2(1-\varepsilon_{G-U})$, se decide că intervalul iniţial aparţine unei Gaussiene. Observaţi semnul "−" în condiţia de mai sus, în $1-\varepsilon_{G-U}$. Apoi, se extinde intervalul la dreapta cu un eşantion. Procedura se opreşte atunci când $\varepsilon_G^2 \geq \varepsilon_U^2$, caz în care aproximarea Gaussienei produce o eroare mai mică comparativ cu aproximarea distribuţiei uniforme.

7. Procedura de mai sus se aplică o dată la fiecare interval cu DU.

2.3.1 Descrierea algoritmului euristic de segmentare G-U-MM

În această secţiune voi prezenta paşii algoritmului aproximării cu comentarii. Două versiuni ale aproximării sunt date, realizarea celei dea doua metode atinge aproximări mai bune. Se observă că algoritmul nu are ca scop să determine o aproximare foarte bună, ceea ce ar implica numeroase funcţii în modelul G-U-MM, în schimb, se păstrează numărul scăzut de componente la aproximare, deoarece una dintre calităţile unei bune segmentări este numărul mic de segmente. Amintesc că ideea principală a algoritmului este de a obţine o aproximare rezonabilă care ajută separarea regiunilor imaginii în funcţie de statistica specifică în speranţa că distribuţia de probabilitate este legată, în mecanismul uman de vedere, de segmentare. De aceea, se permit aproximări slabe, atunci când acestea păstrează numărul de segmente mic şi nu afectează semnificativ regiunile atribuite segmentelor, în imagini. Unele dintre etapele algoritmului prezentate ulterior sunt bazate pe Fig. 2.27.

Algoritmul prezentat, ulterior, constă numai în noua secţiune a algoritmului de ansamblu şi se adaugă la secţiunea de bază deja prezentată la începutul capitolului. Secţiunea de bază a întregului algoritm determină două tipuri de intervale, numite "uniforme" şi (de tip) "nedeterminat". Aceste intervale sunt prelucrate în continuare, după cum urmează.

1. Se presupune intervalul $[x_1, x_2]$ clasificat ca "de tip nedeterminat" (Fig. 2.27). Se determină nivelul de gri în centrul intervalului: $h(x_1 + x_2)/2$, în cazul în care h este funcţia histogramă.

   IF $h(x_1 + x_2)/2 > h(x_1)$ AND $h(x_1 + x_2)/2 > h(x_2)$ THEN mergem la pasul 2, ELSE trecem la pasul 9.

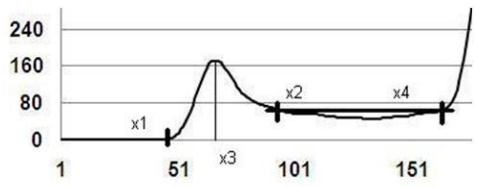


Fig.2.27 Fragment al histogramei

2. Găsim $x_3 = \arg\max(h(x)), x \in [x_1.x_2]$.
3. Se presupune că x_3 este vârf de Gaussiană, atunci aproximarea locală (notată cu *h*, ca histograma) $h(x) = A \cdot \exp(-(x - x_3)^2 / b)$. Prin urmare, $A = h(x_3)$.

   Comentariu. Presupunerile făcute în paşii 1-3 conduc la o rigidă identificare a vârfului funcţiei Gauss, şi, astfel, o aproximare rigidă. Este posibil ca condiţia dublă (conectat cu AND) la pasul 1 să fie falsă, şi intervalul corespunde unui vârf Gauss. Pasul 9 are menirea de a corecta aceste situaţii.
4. Se determină b_1 optimal pentru partea stângă a intervalului:

   $for\, b = 3\, to\, 7200$,

   se determină eroarea aproximării pentru distribuţia Gaussiană:

   $\varepsilon^2_{(b)} = \sum_{x_i}^{x_3} (h(x) - Ae^{(-(x_i - x_3)^2 / b)})$.

   Se găseşte b optimal corespunzător $\arg\min \varepsilon^2_{(b)}$.

   Comentarii: creşterea în buclă este 1; dispersia trebuie să nu fie un număr întreg; valorile 3 şi 7200 s-au constatat empiric convenabil pentru imaginile prelucrate, dar alte limite pot fi necesare pentru alte imagini. Deoarece suma este de la x_1 la x_3, valoarea optimă a lui *b* este astfel determinată, de fapt, optimizată pentru partea stângă, care reprezintă b_1 optimal.
5. Se determină eroarea şi b_2 optimal pentru partea dreaptă.
6. Se determină b optimal ca media $b = (b_1 + b_2)/2$ şi eroarea totală a intervalului ca $error = error_1 + error_2$, unde $error_1$ şi $error_2$ sunt determinate la paşii 4 şi 5.

   Comentarii: A doua metodă de găsire a valorii lui *b* optimal este următoarea:

   Utilizând iar $g(x) = A \cdot \exp(-(x - (x_1 + x_2)/2)^2 / b)$, unde x_1 şi x_2 sunt limitele intervalului "de tip nedeterminat", se găseşte pentru partea stângă a funcţiei de aproximare $g(x_1) = A \cdot \exp(-(x_1 - center)^2 / b_1) = h[x_1]$, unde *center* este x_3. Atunci $-(x_1 - center)^2 / b_1 = \log(h[x_1] / A)$ sau $b_1 = -(x_1 - center)^2 / \log(h[x_1] / A)$ unde b_1 este determinat pentru $x = x_1$.

   Similar b_2 este determinat pentru $x = x_2$: $-(x_2 - center)^2 / b_2 = \log(h[x_2] / A)$ sau

   $b_2 = -(x_2 - center)^2 / \log(h[x_2] / A)$.

În final, se consideră $b = (b_1 + b_2)/2$.

7. Se calculează eroarea $\varepsilon^2_{(b)} = \sum_{x_i}^{x_3}(h(x) - Ae^{(-(x_i - center)^2/b)})$ $i = 1,2$, similar paşilor anteriori 4-6.
8. Se compară *b* optim determinat la pasul 6 de prima metodă cu cel stabilit pe baza erorii calculate de a doua metodă la pasul 7. Se alege b care corespunde erorii mai mici (pentru fiecare b optim se calculează erorile la etapele 6 şi 7).

   Comentariu. Paşii de mai sus determină şi aproximează numai partea de sus (vârfurile) a Gaussienelor, porţiuni de Gaussiene care sunt detectate eronat ca „intervale uniforme". Pentru segmentele ascendente şi, respectiv descrescătoare a Gaussienelor care sunt lăsate ca „nedeterminate" la detectarea intervalelor cu DU, sunt utilizate următoarele etape.
9. Pentru intervalele declarate uniforme în prima etapă a algoritmului, atunci când acestea nu satisfac condiţia de la pasul 2, se procedează după cum urmează. Folosind o fereastră cu suprapunere a câte 24 nivele de gri pe histogramă, IF media a 6 nivele de gri de la centru este mai mare decât media a 6 nivele de gri de pe stânga AND media a 6 nivele de gri de la centru este mai mare decât media a 6 nivele de gri din dreapta, THEN intervalul este Gaussian, ELSE, intervalul este considerat constant (DU). Pentru intervale noi Gaussiane, se trece la pasul 4, se determină *b* optim şi se calculează eroarea în funcţie de paşii 4-6.

   Comentariu. Următorul pas în algoritm decide dacă pixelii de la limitele intervalelor Gaussiene, aparţin intervalelor cu DU, dacă ar trebui sau nu ar trebui să fie unite la intervale G. Decizia se face pe baza erorii minime de aproximare.
10. Se incrementează cu 1 intervalul Gaussian la stânga; se calculează eroarea pentru Gaussiană şi distribuţie uniformă. Dacă eroarea de Gaussiană este mai mică, atunci se ataşează valoarea nivelului de gri pentru intervalul Gaussian, dacă nu, rămâne în intervalul cu DU şi nu se mai încearcă extinderea intervalului la stânga. Apoi, se verifică în acelaşi mod la dreapta intervalului. În cazul în care unitatea de extindere a intervalului a fost acceptată la stânga sau la dreapta, atunci se obţin şi se păstrează noile Gaussiene şi intervale cu DU, care sunt determinate de noile praguri de segmentare.

Mixturile hibride pot fi exprimate ca amestecuri de distribuţii pe intervale disjuncte care formează o partajare a intervalului de nivele de gri [0-255]. Alternativ, ele pot fi mixturi aditive, cu distribuţii suprapuse pe anumite intervale. Modelele de Mixturi Gaussiene presupun amestecuri aditive cu distribuţii suprapuse pe întregul interval [0, 255]. Evident, presupunând o

suprapunere totală se simplifică abordarea matematică la aproximări, dar produce convoluţia funcţiilor de distribuţie de probabilitate. Noi, coordonatorul ştiinţific HN.Teodorescu şi eu, nu cunoaştem vreo lucrare care tratează oportunitatea unei astfel de abordări din punct a vedere al observatorului uman. Mai mult decât atât, am observat lipsa de dovezi pentru a susţine caracterul adecvat al MMG (model de suprapunere Gauss) pentru imagini, dincolo de confortul tehnic pur. Luând în considerare de exemplu, un singur obiect cu nivele de gri uniforme în intervale mici, pe fond uniform distribuit. Cele două părţi ale imaginii, obiect şi fundal, nu sunt în nici un fel corelate, de altfel nici o valoare a nivelul de gri a unei părţi a imaginii nu este prezentă în altă parte (alt obiect), în acest caz, nu are loc nici o suprapunere. Un alt exemplu de imagine în care distribuţiile pot fi considerate nesuprapuse, Gaussiene truncate sau niveluri de zero pe intervale extinse pe histogramă este *rabbit toy* [154].

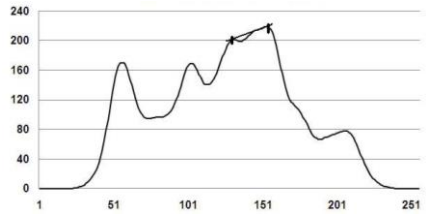
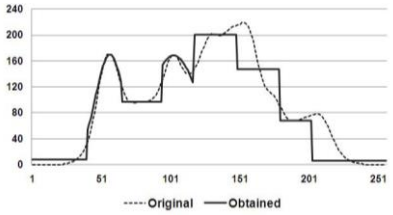
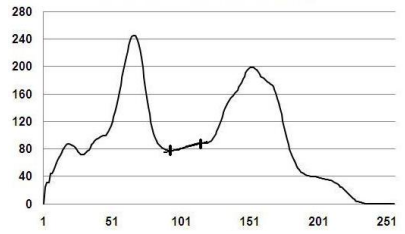
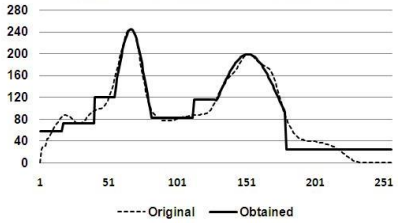
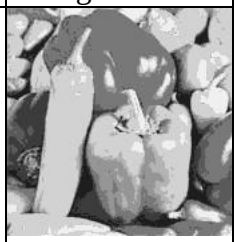
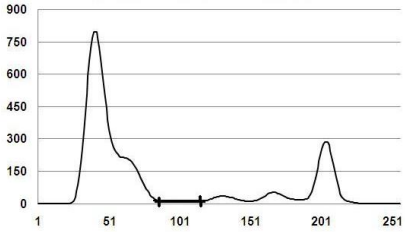
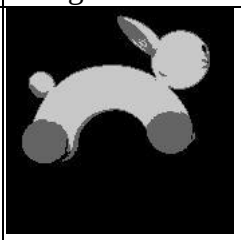
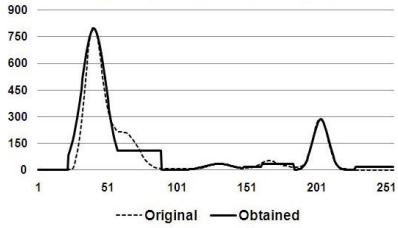
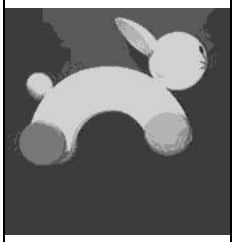
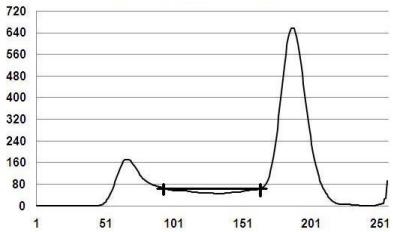
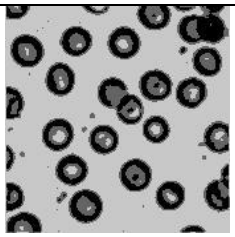
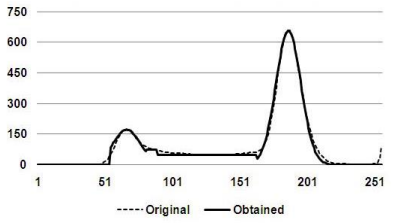
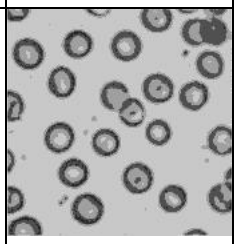
2.3.2 Prezentarea rezultatelor obţinute

În secţiunile anterioare au fost descrise metoda şi algoritmul pentru reprezentarea histogramei ca o funcţie pe intervale. Problema cheie a fost determinarea intervalelor G şi DU pentru reprezentarea pe porţiuni. Ca urmare, aceste intervale se presupune că definesc segmentele de nivele de gri în imagine, care reprezintă de fapt ideea de bază a algoritmului. Rezultatele parţiale şi preliminare prezentate în capitolul dat au fost prezentate în [155].

Tabelul 2.6 sintetizează un set de rezultate, inclusiv histograme nivelate, imaginile aferente, aproximările histogramelor, şi imaginile segmentate corespunzătoare. Este frapant că pentru imaginea *Lena* care este cea mai prost aproximată, din cauza complicaţiilor sale, se obţine o segmentare destul de bună, comparativ cu segmentarea prin metoda lui Otsu. De asemenea, se observă că pe histograma pozei *Lena* o parte a Mixturii Gaussiene este aproximată cu o funcţie sub formă de scară. Aceeaşi situaţie se întâmplă pentru alte histograme, mai ales atunci când două funcţii Gauss se suprapun în mare măsură.

Rezultatele obţinute de segmentare cu algoritmul GUMM îmbunătăţit prezentat în acest subcapitol sunt vizibil mai bune decât cele obţinute cu versiunea anterioară a algoritmului. Cu toate acestea am făcut un compromis între complexitatea modelului, care este reprezentat de numărul de distribuţii Gaussiene şi uniforme în mixtură şi sarcina de calcul. De aceea, după cum arată histogramele aproximate, aproximarea este imperfectă, mai ales în anumite intervale a nivelurilor de gri. De exemplu, pentru imagini "Lena", "pepper", şi "rabbit", sub-intervalele care fac parte din regiunile cu distribuţii Gaussiene sunt aproximate cu distribuţii uniforme.

Tabelul 2.6 Imagini segmentate cu ambele metode

Histograma originală netezită, cu pragurile stabilite de metoda iniţială (3 segmente)	**Imagine segmentată cu metoda iniţială**	**Histograma originală şi cea obţinută prin aproximarea distribuţiilor**	**Imagine segmentată cu noua metodă**
1	2	3	4
	 3 segmente		 8 segmente
	 3 segmente		 8 segmente
	 3 segmente		 8 segmente
	 3 segmente		 6 segmente

2.4 Compararea obiectivă a rezultatelor obţinute în urma aplicării algoritmului de segmentare propus cu rezultatele altor algoritmi de referinţă

Evaluarea segmentării se face fie manual, de către experţi, fie cu ajutorul maşinii de calcul. Evaluarea supervizată (cu implicarea factorului uman) constă în delimitarea regiunilor de către experţi şi compararea rezultatelor obţinute cu ajutorul unui/unor algoritm/i. Site-ul http://www.eecs.berkeley.edu/Research/Projects/CS/vision/bsds/ conţine imagini segmentate

manual ce pot fi folosite la evaluarea calităţii segmentării. De exemplu poza cu doi elefanţi http://www.eecs.berkeley.edu/Research/Projects/CS/vision/bsds/bench/gray/ren_nips2012_gray/296059.html.

Metoda dată este migăloasă şi necesită mult timp, de aceea se tinde spre utilizarea unor indici de evaluare a calităţii ce ar permite evaluarea non-supervizată (fără implicarea experţilor).

În cazul evaluării nesupervizate sunt propuse mai multe metrici de evaluare care ar determina omogenitatea regiunilor, diferenţa valorilor medii dintre regiuni, contrastul dintre regiunea obiect şi fundal, numărul prea mare sau prea mic de segmente obţinute şi altele [156-158]. Un factor important la aprecierea calităţii segmentării ar fi şi evaluarea texturii. Imaginile test pentru această lucrare nu conţin texturi, dar segmentarea bazată pe texturi este larg utilizată [159] iar Sharma M. şi Singh S. analizează caracteristicile texturii pentru o determinare precisă a regiunilor [160].

O metodă bună de evaluare a tehnicilor de segmentare trebuie să fie independentă de conţinutul şi tipul imaginilor. Este necesar să se determine cât mai exact performanţa segmentării şi să se minimizeze implicarea umană.

În scopul de a face o comparaţie a rezultatelor obţinute folosind metoda propusă cu alte metode din literatură, luăm în considerare în primul rând numărul de segmente obţinute [161]. Pentru fiecare metodă numărul de segmente obţinute ar trebui să fie acelaşi pentru o comparaţie obiectivă.

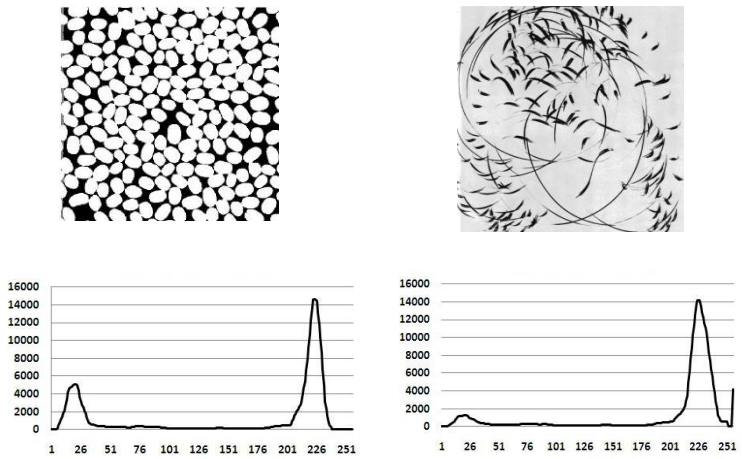


Fig. 2.28. Imaginile sintetice *D75* şi *D45* [162] şi histogramele lor nivelate

Putem observa că histograma este compusă din două Gaussiane separate de o distribuţie uniformă (3 segmente). Avem nevoie deci de două praguri.

Metoda Multithresh [164] recomandă alt număr de praguri şi nu putem face o comparaţie obiectivă între metoda noastră (sau Otsu [163]) şi această metodă deoarece numărul de segmente obţinute este diferit.

Tabelul 2.8 Pragurile imaginilor sintetice *D75*.jpg şi *D45*.jpg

	Pragurile imaginii D75.jpg
Multithresh	0, 17, 73, 113, 153, 231, 255*****
Metoda Otsu	0, 25, 70, 255
Metoda propusă	0, 54, 185, 255
	Pragurile imaginii D45.jpg
Multithresh	0, 68, 81, 113, 152, 255******
Metoda Otsu	0, 32, 76, 255
Metoda propusă	0, 36, 185, 255

* Numărul recomandat de praguri este 6
** Numărul recomandat de praguri este 5

În literatura de specialitate sunt propuse mai multe metode cantitative obiective de evaluare [157-159], inclusiv:

- F, propusă de Liu şi Yang;
- F′ şi Q, indici propuşi de Borsotti, Campadelli şi Schettini ca o îmbunătăţire a funcţiei F;
- Criteriul de uniformitate elaborat de Levine şi Nazif;
- E – indice bazat pe analiză empirică, propus de Zhang et al.

Indicii care sunt implementați la evaluarea obiectivă a rezultatelor sunt prezentaţi pe scurt mai jos, parafrazând literatura de specialitate [156-158].

1) ***Funcţia de evaluare propusă de Liu and Yang:***

$$F = \sqrt{N} \sum_{j=1}^{N} \frac{e_j^2}{\sqrt{S_j}}$$

unde N este numărul regiunilor obţinute după segmentare, S_j – aria regiunii j şi e_j^2 – eroarea pătratică a culorii (sau nivelului de gri) care se calculează:

$$e_j^2 = \sum_{k \in S_j} (x_k - \bar{x})^2$$

unde x_k este nivelul de gri al pixelului, şi $\bar{x}$ media nivelelor de gri a regiunii.

Putem observa că F nu este obiectivă faţă de un număr redus de segmente sau un număr mare de segmente mici. Funcţia F este 0 atunci când eroarea de culoare este zero pentru toate segmentele, care apare atunci când fiecare pixel este propria sa regiune, numărul mare de regiuni din imaginea segmentată este sancţionat doar de măsura la nivel global – $\sqrt{N}$.

2) *Funcţia F' propusă de Borsotti, Campadelli şi Schettini:*

$$F' = \frac{1}{1000 \cdot S_I} \sqrt{\sum_{a=1}^{MaxArea} [N(a)]^{1+1/a}} \sum_{j=1}^{N} \frac{e_j^2}{\sqrt{S_j}}$$

unde S_I – suprafaţa imaginii;

$N(a)$ – denotă numărul regiunilor din imaginea segmentată care au o arie de mărimea exact *a;*

MaxArea – aria celei mai mari regiuni.

F' este mai bună decât F când imaginea segmentată reprezintă o mulţime de regiuni cu un număr mic de pixeli.

3) *Criteriul definit de Borsotti*

$$Q = \frac{1}{10000 \cdot S_I} \sqrt{N} \sum_{j=1}^{N} \left(\frac{e_j^2}{1 + \log S_j} + \left(\frac{N(S_j)}{S_j} \right)^2 \right)$$

unde $N(S_j)$ – indică numărul de regiuni din imaginea segmentată ce au o suprafaţă exact S_j.

Segmentarea cu un număr mare de regiuni nu este penalizată aşa puternic. Putem concluziona că criteriul este:

- foarte obiectiv faţă de regiunile cu suprafeţe mari. Nu sunt penalizate insă regiunile mari ce conţin mici variaţii de culoare, adică cele uniforme.
- conţine al doilea termen care reprezintă o însumare ce are de obicei o valoare foarte mică în comparaţie cu primul termen, deci are un efect neglijabil asupra rezultatelor evaluării.

4) *Criteriul de uniformitate definit de Levine şi Nazif* [157]:

$$Lev = \sum_j \sum_{x \in R_j} \left(f(x) - \frac{1}{S_j} \sum_{x \in R_j} f(s) \right)^2 = \sum_j \frac{\sigma_j^2}{C}$$

$f(x)$ – intensitatea pixelului x

C – coeficientul normalizat, egal cu varianţa maximă posibilă

$$C = \frac{(f_{\max} - f_{\min})^2}{2}$$

Acest criteriu calculează suma raporturilor dintre deviaţia standard normalizată a fiecărei regiuni şi contrastul acelei regiuni dat de coeficientul normalizat C.

5) ***Metoda de evaluare bazată pe entropie*** [156]

După cum spun autorii entropia este o măsură a 'dezordinii' dintr-o regiune şi este o caracteristică naturală inclusă într-o metodă de evaluare a segmentării.

Entropia pentru regiunea *j* este definită ca:

$$H_v(R_j) = -\frac{L_j(m)}{S_j}\log\frac{L_j(m)}{S_j}$$

unde $L_j(m)/S_j$ reprezintă probabilitatea ca un pixel din regiunea R_j să aibă luminanţa de valoarea m.

Notaţia $H_v(R_j)$ a fost redusă la forma $H(R_j)$ cu caracteristici implicite, v fiind luminanţa. Zhang H. et al. definesc entropia regiunilor (expected region entropy) imaginii *I* ca:

$$H_r(I) = \sum_{j=1}^{N}\left(\frac{S_j}{S_I}\right)H(R_j),$$

şi entropia structurii (the layout entropy) :

$$H_l(I) = -\sum_{j=1}^{N}\left(\frac{S_j}{S_I}\right)\log\frac{S_j}{S_I}.$$

Autorii propun să combine atât entropia structurii (the layout entropy) cât şi entropia regiunilor (the expected region entropy) pentru a măsura eficienţa unei metode de segmentare:

$$E = H_l(I) + H_r(I).$$

Pentru imagini naturale (imagini standard de testare), se determină pragurile, care reprezintă limitele Gaussienelor şi a intervalelor uniforme, dar nu au fost obţinute la fel de bune rezultate ca şi pentru imaginile sintetice.

Tabelul 2.9 Evaluarea cantitativă a imaginii segmentate *butterfly* [133] folosind diferite metode

Indici	*Multithresh*	*Metoda Otsu*	*Metoda propusă*
F	*333665*	*189539*	*278709*
F′	*0.0026*	*0.0015*	*0.0021*
Q	*0.0109*	*0.0040*	*0.0080*
Lev	*1.37*	*1.23*	*1.21*
E	*7.5*	*6.96*	*7.24*

Mai multe detalii sunt în [165]. Vizual este dificil de evaluat care rezultate sunt mai bune, respectiv care metodă de segmentare este mai eficientă. Parametrii cantitativi facilitează evaluarea metodelor, permit elaborarea unei concluzii obiective.

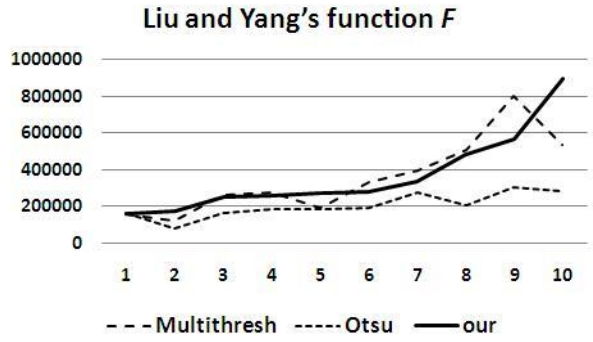


Fig. 2.29. Reprezentarea rezultatelor utilizând funcţia de evaluare propusă de Liu şi Yang

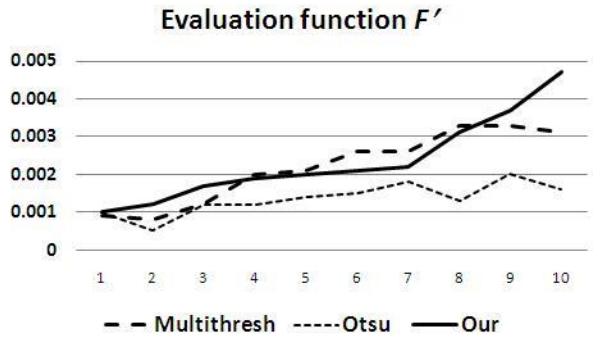


Fig. 2.30. Reprezentarea rezultatelor utilizând funcţia de evaluare $\boldsymbol{F'}$

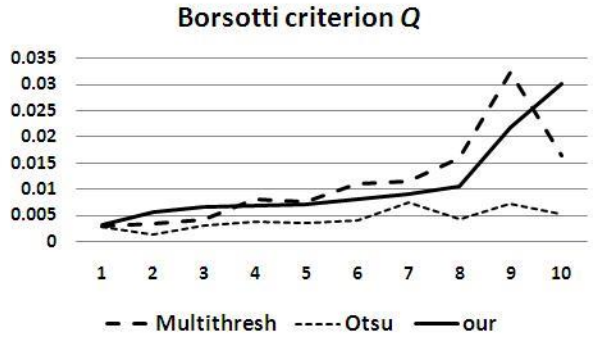


Fig. 2.31. Reprezentarea rezultatelor utilizând criteriul propus de Borsotti

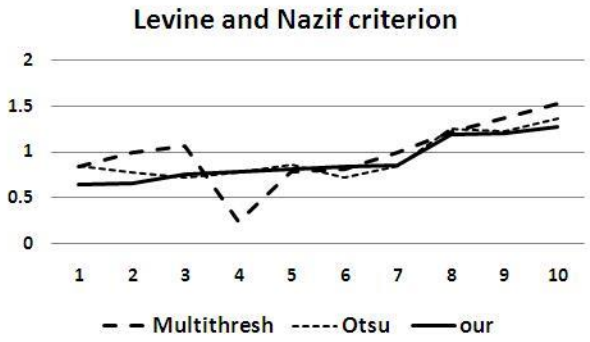


Fig. 2.32. Reprezentarea rezultatelor utilizând criteriul lui Levine şi Nazif

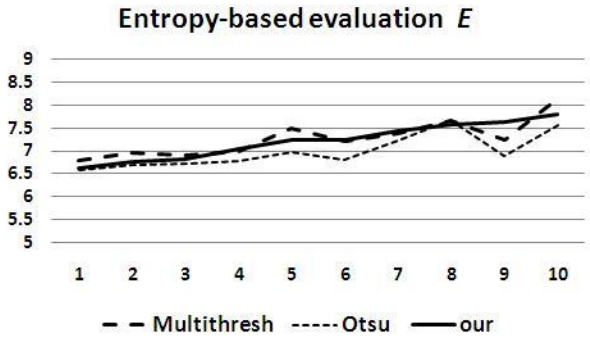


Fig. 2.33. Reprezentarea rezultatelor utilizând metoda de evaluare bazată pe entropie

Conform criteriului de eficienţă, metoda propusă este una simplă, cu consum minim de resurse şi computaţional rapidă. Rezultatele obţinute prezentate sunt numeric apropiate de cele obţinute cu cele ale metodelor de referinţă. Prin urmare, se poate confirma că metoda este eficientă şi satisfăcătoare.

Calitatea rezultatelor validează în mod indirect utilizarea Modelului de Mixturi Uniforme şi Gaussiene (G-U-MM) propuse în documentele noastre anterioare [147, 149].

2.5 Concluzii la capitolul 2

Segmentarea este o etapă esenţială în procesarea imaginilor, de rezultatele obţinute în urma acestei etape depinde calitatea interpretării scenei de către calculator (metoda nesupervizată). Datorită multitudinii tipurilor de imagini (naturale, prin satelit, medicale) şi a factorilor (iluminare neuniformă, zgomot) ce pot influenţa reprezentarea obiectelor în scenă nu a fost elaborată încă o metodă unică de segmentare care să permită obţinerea rezultatelor satisfăcătoare pentru orice tip de imagine.

După preprocesare se alege metoda de segmentare: bazată pe clasificarea pixelilor, de determinare a contururilor sau orientată pe regiuni în dependenţă de caracteristicile necesare de

extras: forme, contururi, regiuni, texturi, text etc. La alegerea dată mai contează şi tipul imaginii, şi spectrul de culoare. De exemplu, imaginile ultrasonore sunt procesate cu metode mai complexe utilizând domeniul wavelet. Pentru imaginile test gri cele mai eficiente (complexitate redusă şi rapiditate computaţională) sunt metodele bazate pe clasificarea pixelilor – prăguirea şi clusterizarea, pentru cele color – metodele orientate pe regiuni. Metodele de determinare a contururilor sunt deopotrivă utilizate, deoarece ele sunt bazate pe determinarea tranziţiei valorilor de culoare (nivele de gri) ceea ce e efectiv pentru ambele cazuri.

Zgomotul impulsional (sare şi piper) influenţează într-o mai mare măsură metodele bazate pe histogramă. Toţi pixelii cu valoarea 0 (negru) sunt atribuiţi unei zone, iar cei cu valoare maximă (255- alb) altei zone. Un alt dezavantaj al metodelor bazate pe histogramă este faptul că segmentele obţinute nu reprezintă regiuni omogene adiacente.

La aplicarea metodelor de determinare a contururilor/muchiilor, deseori se obţin contururi multiple sau contururi false. Din cauza unei iluminări neuniforme (umbre, pete de lumină) muchiile obiectelor din scenă ar putea fi reprezentate cu linii întrerupte, ceea ce duce la o interpretare eronată a scenei.

Metodele de segmentare orientate pe regiuni sunt consumatoare de timp (fiecare pixel din imagine este comparat cu un pixel ales aleatoriu (centroid, germene) şi se verifică similaritatea pentru a forma zone omogene). Este mai eficientă pentru imaginilor color.

Complexitatea algoritmului de segmentare reprezintă un compromis între timpul consumat la implementare şi acurateţea necesară a rezultatelor.

S-a sugerat că utilizarea mixturilor cu două tipuri de distribuţii elementare pentru modelarea distribuţiei nivelurilor de gri sau de culoare într-o imagine, este o modalitate mai simplă de aproximare a histogramei şi mai mult, permite obţinerea de rezultate mai bune la segmentare. În plus, rezultatele de segmentare, uneori, sunt mai aproape de conţinutul semantic din imagine. Distribuţia mixtă specifică propusă în contextul de segmentare a imaginilor este G-U-MM. Alegerea sa s-a bazat pe analiza empirică a unui set de histograme a imaginilor test.

A fost prezentată o explicaţie detaliată pentru raţiunea modelelor G-U-MM propuse ca un fundament al procesului de segmentare. Procedura de segmentare propusă se bazează direct pe clasa de modele G-U-MM. O aproximare pe porţiuni a fost utilizată pentru histograme; reprezentarea pe porţiuni conectează direct la segmentare.

Algoritmii de calcul sunt computaţional eficienţi. Algoritmul de bază este semi-euristic deoarece îşi are rădăcinile în funcţia de aproximare. La un anumit nivel de rafinare a aproximării se foloseşte euristica pentru a reduce sarcina de calcul. Comparativ cu metoda propusă în

lucrările noastre anterioare pe această temă, s-a demonstrat corectitudinea procedurii şi a fost îmbunătăţită pentru a elimina confuzia între vârful plat al unei distribuţii Gaussiene şi a unei distribuţii uniforme.

Analiza rezultatelor de segmentare a fost efectuată utilizând indici de calitate [155]. Compararea dintre valorile acestor indici, obţinute cu metoda propusă şi obţinute cu segmentarea Otsu arată că segmentarea este îmbunătăţită prin utilizarea metodei G-U-MM pentru unele imagini tipice, dar nu pentru toate.

3. ALGORITMI DE RECUNOAŞTERE A FORMELOR ŞI CLASIFICARE AUTOMATĂ A IMAGINILOR ÎN APLICAŢII CU IMAGINI DIGITALE

3.1 Clasificarea automată a datelor bazată pe separare liniară

Secţiunea următoare este o parafrazare a articolului „Algorithm for linear pattern separation" [166], publicat în „Meridianul Ingineresc" No. 2, 2013.

Problema separării liniare a mulţimilor este un concept important în analiza datelor. Această teorie este utilizată pe scară largă şi este aplicată în multe domenii, cum ar fi: recunoaşterea formelor, luarea deciziilor, diagnosticarea bolilor, biometrie, tratarea automată a documentelor ş.a.

Fie date două mulţimi de obiecte (atribute):

$A = \{a^1, a^2, ..., a^m\}$

$B = \{b^1, b^2, ..., b^k\}$

unde $a^i, b^j \in R^n$ pentru $\forall i = 1, 2, ..., m,\ \forall j = 1, 2, ..., k$ şi $A \bigcap B = \varnothing$.

Scopul separării liniare este de a construi funcţia de decizie de forma:

$$f(x) = w^T x - w_0$$

care împarte spaţiul R^n în două submulţimi astfel încât

$f(a^i) < 0$ şi $f(b^j) > 0$

pentru $\forall i \in A$ şi $\forall j \in B$.

Aici w este un vector coloană din R^n, iar w_0 este un scalar: $w_0 \in R$. Simbolul „T" semnifică operaţia de transpunere, astfel toţi vectorii sunt vectori coloană.

Separarea (clasificarea) poate fi formulată ca o problemă de programare pătratică. În această secţiune se propune abordarea a trei modele ale separării liniare. Pentru unul din aceste modele se propune o procedură efectivă de rezolvare numerică.

3.1.1 Modelul de separare maximă [167]

Se caută un hiperplan

$$w^T x - w_0 = 0 \qquad (1)$$

pentru care se maximizează distanţa minimă până la orice punct din mulţimile A şi B. Distanţa de la punctul $z \in R^n$ până la hiperplanul (1) este egală cu

$$d(z) = \frac{f(z)}{\| w \|} = \frac{w^T z - w_0}{\| w \|}.$$

Aici şi în continuare $\|.\|$ este norma euclidiană. Astfel

$$d(a^i) = \frac{w_0 - w^T a^i}{\| w \|},\ d(b^i) = \frac{w^T b^i - w_0}{\| w \|},$$

$d_{\min} = \min\left\{d(a^1), d(a^2),..., d(a^m), d(b^1),..., d(b^k)\right\}$ Problema revine la maximizarea mărimii $d_{\min}$ ceea ce este echivalentă cu următoarea problemă :

$$\left.\begin{array}{l} \delta \to \min, \\ referitorla: \\ w_0 - w^T a^i \geq \delta \| w \|, i = 1,2,..., m \\ w^T b^i - w_0 \geq \delta \| w \|, i = 1,2,..., k. \end{array}\right\} \quad (2)$$

Problema (2) este o problemă neliniară (neconvexă) faţă de necunoscutele $\delta \in R$, $w_0 \in R$, $w \in R^n$.

Se va impune următoarea restricţie: $\| w \| = 1$. Atunci, introducând variabilele

$$u = \frac{w}{\delta} \in R^n \text{ şi } v = \frac{w_0}{\delta} \in R,$$

problema (2) este echivalentă următoarei probleme:

$$\left.\begin{array}{l} \frac{1}{2} u^T u \to \min \\ referitorla: \\ [b^i]^T u - v \geq 1, i = 1,2,..., k, \\ -[a^i]^T u + \delta \geq 1, i = 1,2,..., m, \\ u \in R^n, v \in R. \end{array}\right\} \quad (3)$$

Problema (3) este o problemă convexă de programare pătratică cu $(n+1)$ variabile şi $(m+k)$ restricţii liniare. Dacă $u^* \in R^n$ şi $v^* \in R$ este soluţie optimă a problemei (3), atunci soluţia problemei (2) este:

$$w^* = \frac{u^*}{\| u^* \|}, w_0^* = \frac{v^*}{\| u^* \|}, \delta^* = \frac{1}{\| u^* \|}.$$

Vectorul w^* este perpendicular pe hiperplanul considerat şi are lungimea egală cu 1, iar mărimea w_0^* este distanţa cea mai mică dintre hiperplan şi originea sistemului de coordonate.

3.1.2 Support Vector Machines (SVM) [168]

În această metodă elementele din mulţimea *A* sunt etichetate cu $t = -1$, iar elementele din mulţimea *B* cu $t = 1$. Cu alte cuvinte

$$t(x) = \begin{cases} -1, \textit{dacă } f(x) < 0, \\ +1, \textit{dacă}(x) > 0, \end{cases}$$

adică
$$\left.\begin{array}{l} w^T x - w_0 < 0, \textit{dacă } t(x) = -1, \\ w^T x - w_0 > 0, \textit{dacă } t(x) = +1 \end{array}\right\} \qquad (4)$$

Se observă că hiperplanul (1) nu se va schimba, dacă w şi w_0 este înmulţit cu aceeaşi constantă pozitivă. Este comod de ales această constantă astfel încât

$$\left.\begin{array}{l} w^T a^i - w_0 = -1, i = 1,2,..., m, \\ w^T b^i - w_0 = +1, i = 1,2,..., k. \end{array}\right\}$$

Astfel luând în consideraţie (4) se poate scrie

$$t(x^i)(w^T x^i - w_0) \geq 1, \forall x^i \in A \bigcup B.$$

Determinarea hiperplanului optim de separare se reduce la rezolvarea problemei:

$$\left.\begin{array}{l} \frac{1}{2} w^T w \to \min \\ \textit{referitorla}: \\ t(x^i)(w^T x^i - w_0) \geq 1, \forall x^i \in A \bigcup B. \end{array}\right\} \qquad (5)$$

Problema (5) este asemănătoare problemei (3). Constrângerile problemei (5) asigură faptul că în soluţia optimă w^*, w_0^* avem:

$$f(x^i) = \begin{cases} +1, \textit{pentru } t(x^i) = 1, \\ -1, \textit{pentru } t(x^i) = -1. \end{cases}$$

3.1.3 Reformularea în termenii programării pătratice convexe

Fie învelişurile convexe ale mulţimilor *A* şi *B*:

$$conv(A) = \left\{ x : x = \sum_{i=1}^{m} \alpha_i a^i , \sum_{i=1}^{m} \alpha_i = 1, \alpha_i \geq 0, i = 1,2,...,m \right\},$$

$$conv(B)=\left\{y: y=\sum_{i=1}^{k}\beta_i b^i, \sum_{i=1}^{k}\beta_i=1, \beta_i\geq 0, i=1,2,...,k\right\}$$

Atunci problema separării optime a mulţimilor *A* şi *B* poate fi formulată astfel:

$$\left.\begin{array}{l}\frac{1}{2}\|x-y\|^2=\frac{1}{2}(x-y)^T(x-y)\to\min\\ referitorla:\\ \sum_{i=1}^{m}\alpha_i=1, \alpha_i\geq 0, i=1,2,...,m,\\ \sum_{i=1}^{k}\beta_i=1, \beta_i\geq 0, i=1,2,...,k.\end{array}\right\} \quad (6)$$

Fie matricele $U_{m\times m}, V_{k\times k}$ şi $Z_{m\times k}$ definite în felul următor:

$$U_{m\times m}=\begin{pmatrix}(a^1,a^1) & (a^1,a^2) & ... & (a^1,a^m)\\ (a^2,a^1) & (a^2,a^2) & ... & (a^2,a^m)\\ ... & ... & ... & ...\\ (a^m,a^1) & (a^m,a^2) & ... & (a^m,a^m)\end{pmatrix}$$

$$V_{k\times k}=\begin{pmatrix}(b^1,b^1) & (b^1,b^2) & ... & (b^1,b^k)\\ (b^2,b^1) & (b^2,b^2) & ... & (b^2,b^k)\\ ... & ... & ... & ...\\ (b^k,b^1) & (b^k,b^2) & ... & (b^k,b^k)\end{pmatrix}$$

$$Z_{m\times k}=\begin{pmatrix}(a^1,b^1) & (a^1,b^2) & ... & (a^1,b^k)\\ (a^2,b^1) & (a^2,b^2) & ... & (a^2,b^k)\\ ... & ... & ... & ...\\ (a^m,b^1) & (a^m,b^2) & ... & (a^m,b^k)\end{pmatrix}$$

Notăm:

$\gamma=(\alpha_1,\alpha_2,...,\alpha_m,\beta_1,\beta_2,...,\beta_k)^T\in R^{m+k}$,

$$Q=\begin{pmatrix}U_{m\times m} & -Z_{m\times k}\\ -Z_{k\times m}^T & V_{k\times k}\end{pmatrix},\quad B=\begin{pmatrix}\underbrace{1\ 1\ ...\ 1}_{m\ ori} & \underbrace{0\ 0\ ...\ 0}_{k\ ori}\\ \underbrace{0\ 0\ ...\ 0}_{m\ ori} & \underbrace{1\ 1\ ...\ 1}_{k\ ori}\end{pmatrix},$$

$$e=\begin{pmatrix}1 & 1^T\end{pmatrix}.$$

Matricele $U_{m\times m}$, $V_{k\times k}$ şi Q sunt pozitiv semidefinite.

Cu aceste notaţii problema (6) devine:

$$\left.\begin{array}{l}\frac{1}{2}\gamma^T Q\gamma \to \min \\ referitorla: \\ B\gamma = e, \\ \gamma \geq 0.\end{array}\right\} \qquad (7)$$

Se menţionează că funcţia scop depinde doar de produsul scalar între vectorii a^i şi b^j. Problema (7) este o problemă de programare convexă şi deci are o soluţie globală optimă.

3.1.4 Reducerea SVM la o problemă de rezolvare a unui sistem de ecuaţii

Utilizând teorema Kunh-Tucker, se demonstrează că duala problemei (5) este:

$$\left.\begin{array}{l}\frac{1}{2}\gamma^T Q\gamma - \sum_{i=1}^{m+k}\gamma_i \to \min \\ referitorla: \\ \sum_{i=1}^{m+k} t(x_i)\gamma_i = 0, \\ \gamma_i \geq 0, \forall i = 1,2,\ldots,m+k.\end{array}\right\} \qquad (8)$$

Vectorii a^i, b^i pentru care $\gamma_i > 0$ se numesc vectori suport.

Se constată că problemele (7) şi (8) dau unul şi acelaşi rezultat. În cele ce urmează se va arăta cum problema (7), care este echivalentă cu problema (8), poate fi redusă la rezolvarea unui sistem de ecuaţii pătratice.

Dacă $\gamma^* \in R^{m+k}$ este o soluţie optimă a problemei (7) atunci există $\lambda^* \in R^2$ astfel încât:

$$\left.\begin{array}{l}B\gamma^* = e, \\ \Gamma^*\left(Q\gamma^* + B^T\lambda^*\right) = 0, \\ \gamma_i^* = 0 \Rightarrow \left[Q\gamma^* + B^T\lambda^*\right]_i \geq 0, \forall i.\end{array}\right\}$$

unde $\Gamma = Diag(\gamma)$ matricea diagonală:

$$\Gamma = \begin{pmatrix}\gamma_1 & \ldots & 0 \\ \vdots & \ddots & \vdots \\ 0 & \ldots & \gamma_{m+k}\end{pmatrix},$$

iar notaţia $[c]_i$ aici şi în continuare semnifică componenta i a vectorului c.

Notăm $G = Diag\left(Q\gamma + B^T\lambda\right)$ şi

$$F(\gamma,\lambda) = \begin{pmatrix}\Gamma\left(Q\gamma + B^T\lambda\right) \\ B\gamma - e\end{pmatrix}.$$

Astfel problema (7) se reduce la rezolvarea sistemului de ecuaţii şi inecuaţii:

$$\left.\begin{array}{l} F(\gamma,\lambda)=0, \\ \gamma\geq 0, G\geq 0. \end{array}\right\}$$

Teorema. Pentru $\forall\gamma\in D=\{\gamma : B\gamma=e, \gamma\geq 0\}$ matricea Jacobiană

$$\nabla F=\begin{pmatrix} G+\Gamma Q & \Gamma B^T \\ B & 0 \end{pmatrix}$$

este nesingulară.

Demonstrarea acestei teoreme este analoagă demonstraţiei **Teoremei 1** din [169].

Definim acum funcţiile $p,q:R\to R_+$

$$p(x)=x^2\max(0,x)=\frac{1}{2}\left(x^3+|x|x^2\right),\ q(x)=-x^2\min(0,x)=-\frac{1}{2}\left(x^3-|x|x^2\right).$$

Funcţiile $p(x)$ şi $q(x)=p(-x)$ sunt de două ori continuu diferenţiabile:

$$p'(x)=\frac{3}{2}\left(x^2+|x|x\right),\ p''(x)=3(x+|x|),$$

$$q'(x)=\frac{3}{2}\left(-x^2+|x|x\right),\ q''(x)=3(-x+|x|).$$

Se constată cu uşurinţă că:

$$\begin{cases} p(x)q(x)=0,\ p'(x)q'(x)=0, \\ p''(x)q''(x)=0,\ \forall x\in R. \end{cases}$$

$$\begin{cases} p(x)+q(x)>0,\ p'(x)+q'(x)\neq 0, \\ p''(x)+q''(x)>0,\ \forall x\neq 0. \end{cases}.$$

$$\begin{cases} p(x)=0 \textit{ dacă şi numai dacă } p'(x)=0, \\ q(x)=0 \textit{ dacă şi numai dacă } q'(x)=0. \end{cases}$$

Luând acestea în consideraţie şi introducând variabilele auxiliare $\eta_1,\eta_2,\ldots,\eta_m$, $\mu_1,\mu_2,\ldots,\mu_k$, λ_1 şi λ_2 problema (7) poate fi redusă la rezolvarea unui sistem din $2(m+k+1)$ ecuaţii cu tot atâtea necunoscute [169]:

$$\left.\begin{array}{l} p(\eta_i)=\alpha_i .i=1,2,\ldots,m, \\ q(\eta_i)=\left[U_{m\times m}\alpha - Z_{m\times k}\beta+\lambda^1\right]_i, \\ i=1,2,\ldots,m, \\ p(\mu_i)=\beta_i, i=1,2,\ldots,k, \\ q(\mu_i)=\left[-Z_{m\times k}\alpha + V_{k\times k}\beta+\lambda^2\right]_i, \\ i=1,2,\ldots,k, \\ B\gamma=e. \end{array}\right\} \qquad (9)$$

Aici s-a notat

$\lambda^1=(\lambda_1,\lambda_1,\ldots,\lambda_1)^T\in R^m,\ \lambda^2=(\lambda_2,\lambda_2,\ldots,\lambda_2)^T\in R^k$.

Sistemul (9) poate fi redus doar la $(m+n+2)$ ecuaţii cu $(m+n+2)$ necunoscute, înlocuind vectorii α şi β cu ajutorul funcţiilor p şi q:

$\alpha=(p(\eta_1),p(\eta_2),\ldots,p(\eta_m))^T$, $\beta=(p(\mu_1),p(\mu_2),\ldots,p(\mu_k))^T$.

Metoda cea mai cunoscută pentru rezolvarea sistemelor neliniare de ecuaţii (9) este metoda Newton [170]. Metoda lui Newton are proprietăţi teoretice şi practice foarte atractive, datorită convergenţei rapide a acesteia: sub nonsingularitatea matricei Jacobiane aceasta va converge la nivel local superlinear.

Fie

$\alpha^*=\left(\alpha_1^*,\alpha_2^*,\ldots,\alpha_m^*\right)^T$ şi $\beta^*=\left(\beta_1^*,\beta_2^*,\ldots,\beta_k^*\right)^T$

soluţia optimă a problemei (8). Atunci funcţia de decizie este dată de:

$$f(x)=\frac{2}{\left\|x^*-y^*\right\|^2}\left[\left(x^*-y^*\right)x-\left\|x^*\right\|^2+\left\|y^*\right\|^2\right],$$

unde

$$x^*=\sum_{i=1}^{m}\alpha_i^* a^i,$$

$$y^*=\sum_{i=1}^{k}\beta_i^* b^i.$$

Secţiunea 3.1.4 conţine o viziune generală matematică a problemei de separare a două mulţimi de date. Metodele de clasificare se bazează pe căutarea unui hiperplan care separă optim mulţimile considerate. Un loc aparte îl ocupă SVM introdus în anul 1995 de către Vapnik V. şi discutat în literatura de specialitate de mulţi cercetători [171].

S-a propus reformularea condiţiilor de optimalitate Kunh-Tucker într-un sistem echivalent de ecuaţii neliniare (cubice) netede. Sistemul de ecuaţii poate fi rezolvat eficient cu ajutorul metodei Newton. În vecinătatea soluţiei optime γ^* viteza de convergenţă a şirului Newton este supraliniară. Exemplele numerice arată că metoda propusă este promiţătoare.

Această metodă permite clasificarea obiectelor doar în două clase, de ex: ambalaje reciclabile sau nereciclabile. Pentru o clasificare mai detaliată s-a creat un sistem hibrid de clasificare ce permite o clasificare mai detaliată.

3.2 Un sistem de clasificare automată bazat pe reguli fuzzy

Metoda de clasificare automată propusă în această lucrare include patru paşi de bază. La primul pas se identifică, dacă există, simbolul de pericol pe etichetă cu ajutorul *Coeficientului de Corelaţie Normalizat* (CCN) şi a unui *Algoritm Genetic* (AG). Dacă eticheta conţine unul din simbolurile de pericol ambalajul este considerat periculos şi este scos de pe banda ambalajelor pentru reciclare. Pasul 2 constă în extragerea caracteristicilor ambalajelor şi include un algoritm de urmărire a conturului şi un algoritm de extragere a semnăturii (distanţa dintre centroid şi contur). La această etapă se aplică operatorii morfologici: dilatarea şi erodarea pentru a obţine o formă completă (fără găuri) şi se calculează perimetrul şi aria formei. Un alt parametru al sistemului fuzzy: simetria formei este determinată pe baza semnăturii. Pasul 3 include un sistem de decizie cu logica fuzzy, care pe baza variabilelor de intrare fuzzy: formă, dimensiune, simetrie grupează ambalajele în trei clase: reciclabil, periculos şi nedeterminat. La pasul 4 pe baza funcţiei diferenţă a semnăturilor se confirmă sau se infirmă decizia luată la pasul anterior.

Baza de date utilizată de sistemul de clasificare automată elaborat (ca date de intrare) reprezintă imagini captate ale ambalajelor din plastic cu diferit conţinut (inclusiv periculos pentru mediu şi sănătatea omului), baterii, PET-uri, flacoane sub presiune etc. ce necesită colectare specială şi nu pot fi aruncate cu deşeurile menajere.

Se cunoaşte sortarea mecanică, aeraulică, spectroscopia în infraroşu şi detectarea cu raze X pentru plastic (e posibilă chiar şi diferenţierea tipurilor PVC, PET, etc.), dar cea mai eficientă sortare în prezent este sortarea manuală. Personalul de sortare poate separa sticle de diferită culoare, hârtie de diferită calitate, deşeuri periculoase etc. Această metodă este costisitoare atât ca cost cât şi ca timp. O posibilitate de mărire a randamentului de selectare este utilizarea Sistemelor Suport de Decizie semiautomate [172-174]. Se cunosc implementări de sortare optică prin aplicarea algoritmilor de recunoaştere a imaginilor digitale [175-176].

Obiectivul principal al SSD elaborate pentru sortarea deşeurilor depinde de problemele critice cu care se confruntă o anumită societate, ţară. Mulţi cercetători au aplicat posibilităţile de gestionare a deşeurilor: valorificarea materialelor prin reciclare/ reutilizare (hârtie, plastic, metale etc.), incinerarea, compostarea şi depozitarea deşeurilor periculoase.

3.2.1 Metodă hibridă bazată pe coeficientul de corelaţie normalizat şi algoritm genetic

Potrivirea şablonului cu imaginea de referinţă este o tehnică în prelucrarea digitală a imaginii şi reprezintă identificarea potrivirii unei mici părţi a imaginii (imaginea şablon) cu imaginea de referinţă. Ea se bazează pe:

1. calcularea corelării din domeniul spaţial sau de frecvenţă, în funcţie de dimensiunea imaginilor. Utilizarea corelării pentru potrivirea şablonului se motivează prin distanţa euclidiană la pătrat [177].

   $d^2_{f,t}(u,v) = \sum_{x,y}[f(x,y) - t(x-u, y-v)]^2$

   unde f este imaginea şi suma depăşeşte x, y sub fereastra conţinând caracteristica t poziţionată pe u, v.

2. Calcularea sumelor locale prin calculul preventiv al sumelor ce rulează. Dacă se extinde formula d^2 şi apoi se reduc termenii constanţi, rămân termenii de corelaţie:

   $c(u,v) = \sum_{x,y} f(x,y) - t(x-u, y-v)$

   $c(u,v)$ reprezintă măsura de similaritate între imagine caracteristică.

3. utilizarea sumelor locale pentru a normaliza corelaţia şi pentru a obţine coeficienţii de corelaţie. Coeficientul de corelaţie depăşeşte dependenţa de dimensiune a caracteristicii prin normalizarea imaginii şi vectorilor de caracteristici la unitatea de lungime, rezultând un coeficient de corelaţie cum ar fi:

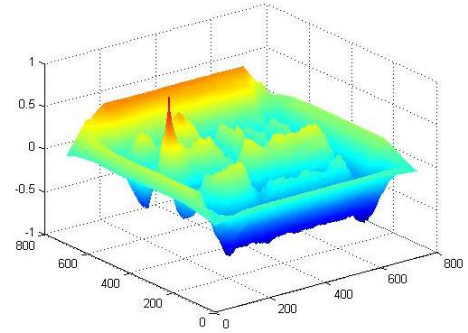
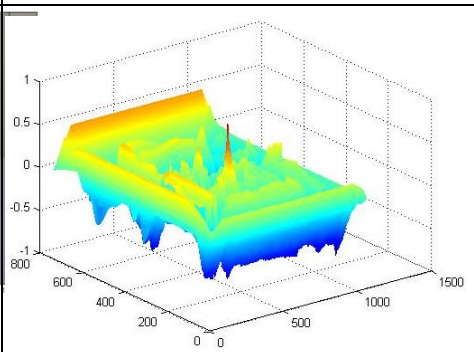
   $$\gamma(u,v) = \frac{\sum_{x,y}[f(x,y) - \overline{f}_{u,v}][t(x-u, y-v) - \bar{t}]}{\left(\sum_{x,y}[f(x,y) - \overline{f}_{u,v}]^2 \sum_{x,y}[t(x-u, y-v) - \bar{t}]^2\right)^{0.5}}$$

   unde $\bar{t}$ este media şablonului, $\overline{f}_{u,v}$ este media $f(x,y)$ din regiunea în conformitate cu şablonul.

   $\gamma(u,v)$ reprezintă auto-corelaţia normalizată – *normalized cross-correlation (NCC).*

NCC nu este variabilă în conformitate cu scalarea, rotirea şi distorsionarea imaginilor. Una dintre soluţii este determinarea maximului de auto-corelaţie şi presupunerea lui ca centru al suprapunerii şablonului cu imaginea.

Tabelul 3.1 Identificarea semnelor de pericol de pe ambalaje utilizând NCC

Şablonul şi imaginea	**Alegerea corelaţiei**	**Suprapunerea şablonului pe imagine**
		
		

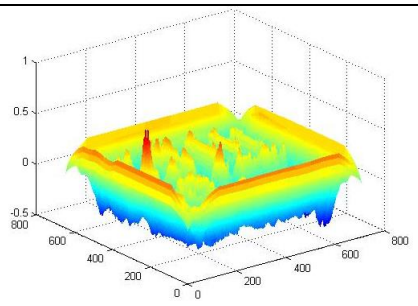
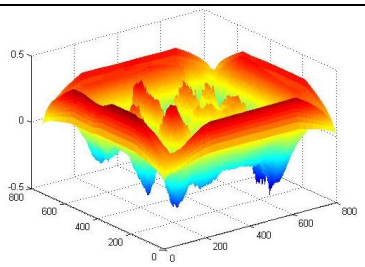
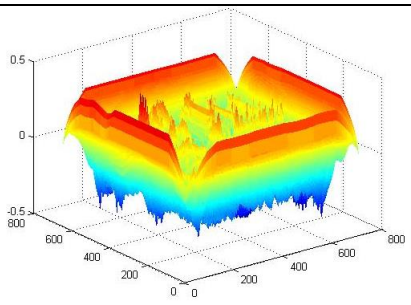
În ambele exemple de mai sus şablonul este un fragment din imagine şi poate fi observată o potrivire maximă. In exemplul de mai jos şablonul [preluat de pe http://www.ilo.org/legacy/english/protection/safework/cis/products/safetytm/clasann1.htm] are o scară diferită cu imaginea, când scara este aproape de cea a imaginii (până la 10%) se observă o potrivire mai slabă, dar suficientă pentru a determina eticheta.

Valoarea de corelaţie este între -1 şi 1, valorile mai mari reprezentând o mai bună potrivire între cele două imagini (şablon şi fragmentul din imagine), dacă nu am stabili un prag de corelaţie (de exemplu, > 0,6) riscăm obţinerea unor rezultate false. Pentru un coeficient de corelaţie de valoare mai mică decât 0.6 putem considera gradul de potrivire între şablon şi imaginea de referinţă ca fiind nesemnificativ şi se pot efectua verificări suplimentare.

În cazul în care dimensiunea şablonului este mult diferită decât cea a imaginii pe care se suprapune se obţin rezultate greşite.

Pentru a obţine rezultate satisfăcătoare folosind această metodă trebuie să luăm în considerare pragul de corelaţie şi scara imaginilor.

Tabelul 3.2 Dificultăţi la identificarea semnelor de pericol de pe ambalaje utilizând NCC

Şablonul şi imaginea	**Alegerea corelaţiei**	**Suprapunerea şablonului pe imagine**
		
		
		

Metoda de potrivire a şablonului cu imaginea scenă este îmbunătăţită cu ajutorul unui algoritm genetic pentru a testa în mod automat potrivirea mai multor poziţii (rotirea şablonului) şi ajustarea dimensiunii (scalarea) şablonului. Algoritmul reprezintă o potrivire a şablonului bazată pe un proces în doi paşi, corelaţie şi algoritm genetic [178]. Pentru îmbunătăţirea performanţei potrivirii, coeficientul de corelaţie normalizat este combinat cu algoritmul genetic. Coeficientul normalizat de corelaţie calculează poziţia probabilă a şablonului în imaginea scenă. Algoritmul genetic calculează scalarea şi rotirea şablonului în imaginea scenă. Rezultatele

experimentale arată că algoritmul este invariant la scalare şi rotire cu o precizie satisfăcătoare şi rezultatele sunt mult mai bune decât la potrivirea şablonului doar prin corelaţie.

În literatură găsim lucrări despre aplicarea GA la procesări de imagini [179,180], inclusiv la identificări de şabloane [181, 183].

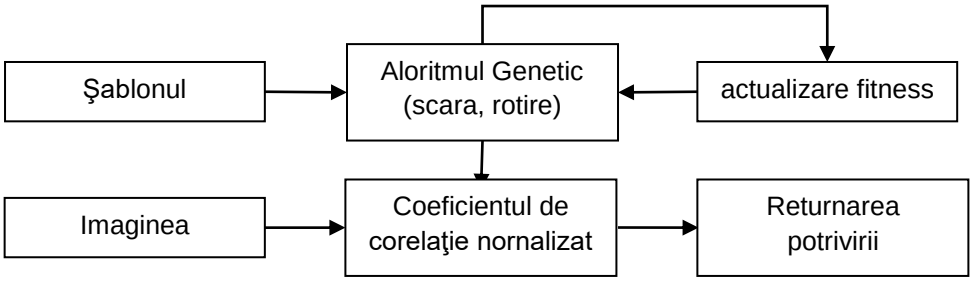


Fig. 3.1 Schema bloc a algoritmului de potrivire a şablonului utilizând algoritm genetic

Descrierea algoritmului genetic implementat, unele rezultate obţinute, cât şi o comparare a rezultatelor cu alte metode sunt incluse în lucrarea "Automated identification of objects based on Normalized Cross-Correlation and Genetic Algorithm" [178] prezentată la a 5-a Conferinţă Internaţională IEEE - EHB 2015. Următoarea secvenţă este o parafrazare a acestui articol.

Algoritmul genetic pentru identificarea automată a şabloanelor are următorul pseudo-cod:

Pasul 1: Iniţializarea populaţiei;

Pasul 2: Calculul fitness populaţie şi ordonarea indivizilor în funcţie de performanţe;

Pasul 3: Se păstrează cel mai bun individ. Dacă acesta nu s-a schimbat timp de un număr predefinit de generaţii atunci algoritmul se opreşte (Pasul 7);

Pasul 4: Aplicarea operatorilor genetici (mutaţie şi încrucişare / crossover);

Pasul 5: Înlocuirea indivizilor din populaţia curentă cu cei nou generaţi;

Pasul 6: Dacă s-a ajuns la un număr maxim de iteraţii / generaţii algoritmul se opreşte, altfel se revine la Pasul 2;

Pasul 7. Se afişează cea mai bună potrivire a şablonului dată de cel mai bun individ.

Informaţia pe care o conţine fiecare individ din populaţie se referă la scalarea şi rotaţia aplicată şablonului. Se consideră că scalarea şablonului are limitele impuse între [1/3, 3] ceea ce implică faptul că în procesul de potrivire şablonul poate fi redimensionat micşorând respectiv mărind dimensiunea lui iniţială de 3 ori. Gradul de rotaţie este în intervalul [0, 359]. Informaţia se codifică pe un număr de biţi dat de rezoluţia dorită (în cazul simulărilor efectuate aceasta este de 256). Din acest motiv codificarea informaţiilor privind scalarea, respectiv rotirea imaginii şablon va avea un număr de 8 biţi (2^8=256) şi fiecare individ din populaţie un număr total de 16

gene (fiecărei gene corespunzându-i un bit). Pentru o rezoluţie mai mare de 65.536 codarea informaţiei s-ar fi realizat pe un număr de 32 biţi, dar am considerat în urma rezultatelor obţinute prin simulări că este suficientă o rezoluţie de 256, rotirea imaginii fiind realizată în acest caz cu un pas se 360/256 ≈ 1.4 grade.

Populaţia are un număr de indivizi între 30 şi 50 (în majoritatea simulărilor s-au utilizat 30 de indivizi). Numărul maxim de generaţii / epoci de antrenare după care algoritmul se opreşte a fost setat la 100. Am impus că dacă după 20 de generaţii nu se schimbă cea mai bună soluţie (best individ) se consideră că algoritmul genetic a convers către o soluţie finală. În funcţie de necesităţi se pot schimba numărul de indivizi şi numărul de generaţii. În urma simulărilor efectuate pentru imaginile de test utilizate am considerat a fi suficiente aceste valori.

S-a considerat că 3% din populaţie reprezintă elita şi că aceasta va fi păstrată la trecerea de la o generaţie la alta. Pentru generarea de indivizi noi s-au aplicat operatorii genetici de încrucişare/crossover a unui segment de 60% din populaţie, respectiv operatorul de mutaţie restul populaţiei (cca. 37-39% procentul fiind influenţat de dimensiunea populaţiei şi a elitei).

Operatorul de crossover se aplică pentru doi indivizi (părinţi), rezultând un număr de doi indivizi noi (copiii). S-a optat pentru o încrucişare într-un singur punct, poziţia de tăiere fiind între a doua genă, respectiv penultima.

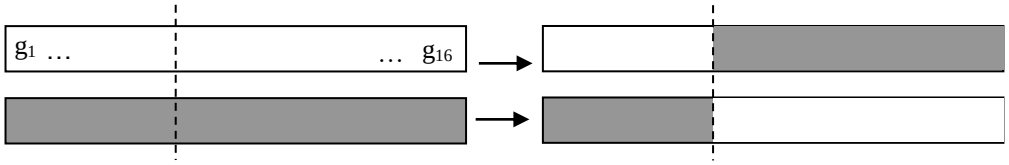


Fig. 3.2 Exemplu de încrucişare într-un singur punct

Iniţial s-a utilizat un mecanism de selecţie a indivizilor de tip ruletă în care în funcţia de fitness creştea probabilitatea unui individ de a fi ales, dar s-a renunţat la acest tip de selecţie deoarece s-a constatat că populaţia migra rapid către cel mai bun individ (care poate fi un optim local în spaţiul soluţiilor) şi s-a optat pentru o selecţie random (în care fiecare individ are aceeaşi probabilitate de a fi ales). O soluţie de compromis ar fi cea în care mecanismul de selecţie ce favorizează indivizii cu un fitness mai mare să fie aplicată după un număr de generaţii când se consideră că cea mai bună soluţie este apropiată de optimul global.

Operatorul modifică una din gene ca valoare binară din 1 logic în 0 sau invers. Dacă modificarea are loc mai aproape de LSB (Least Significant Bit) al unuia din cei doi cromozomi de rotaţie sau scalare, atunci valoarea nou obţinută va fi apropiată de cea iniţială şi invers. Dacă mutaţia genei are loc mai aproape de MSB (Most Significant Bit) atunci salturile făcute de noul individ în spaţiul soluţiilor sunt mari raportate la poziţia individului iniţial.

În imaginile de mai jos este reprezentată variaţia poziţiei celui mai bun individ pe parcursul mai multor iteraţii. Pe parcursul procesului de potrivire şablonul căutat a fost suprapus şi peste un al doilea şablon existent pe etichetă, în final realizându-se o potrivire corectă.

Tabelul 3.3 Implementarea algoritmului bazat pe Template Matching şi GA

		
Imaginea şablon şi imaginea scenă	Primul pas al algoritmului: Fitness: 0.404293 Scalare: 1.64 Rotire 178.80	Iteraţia 6: Fitness: 0.423420 Scalare: 1.54 Rotire 270.31
		
Iteraţia 8: Fitness: 0.543253 Scalare: 0.96 Rotire 178.80	Iteraţia 13: Fitness: 0.690040 Scalare: 0.55 Rotire 0.00	Resultatul final: Fitness: 0.690040 Scalare: 0.55 Rotire 0.00

Tabelul 3.4 Rezultatele obţinute aplicând metoda potrivirii utilizând GA

Imaginile şablon şi de referinţă	**Rezultatele obţinute**
1	**2**
	 Fitness:0.924660 Scalare:1.01 Rotire:0.00
	 Fitness:0.878400 Scalare:1.33 Rotire:0.00

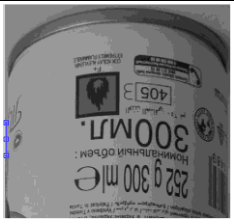
Continuare Tabel 3.4

1	2
	 Fitness:0.731329 Scalare:0.73 Rotire:0.00
	 Fitness:0.681835 Scalare:0.71 Rotire:180.20
	 Fitness:0.747223 Scalare:1.27 Rotire:0.00
	 Fitness:0.616609 Scalare:0.53 Rotire:0.00

În urma testelor s-a constatat că ar fi bine de plasat obiectele a căror rezultate au valoarea fitnessului între 0.6-0.7 într-o clasă nedeterminată pentru o analiză ulterioară manuală.

Se observă îmbunătăţirea metodei bazate pe coeficientul de corelaţie datorită aplicării GA. Pentru reducerea timpului de rulare s-ar putea crea condiţia ca algoritmul să ruleze doar pentru imagini cu acelaşi unghi de captare a imaginii sau cu o rotire de 180 de grade. Deoarece rotirea la diferite unghiuri induce apariţia de pixeli negri sau albi ce ar putea afecta negativ rezultatele.

Tabelul 3.5 Rezultatele obţinute aplicând CCN şi metoda bazată pe CCN şi GA

Imaginile şablon şi de referinţă	**Potrivirea prin CCN**	**Potrivirea prin metoda bazată pe CCN şi GA**
		 Fitness:0.874111 Scale:1. 26 Rotate:0.00
		 Fitness:0.845151 Scale:0.61 Rotate:0.00
		 Fitness:0.731329 Scale:0.73 Rotate:0.00
		 Fitness:0.739534 Scale:0.71 Rotate:180.20
		 Fitness:0.616609 Scale:0.53 Rotate:0.00

3.2.2 Extragerea semnăturii şi a descriptorilor de formă

Procesul de extragere a caracteristicilor formelor cuprinde următoarele etape:

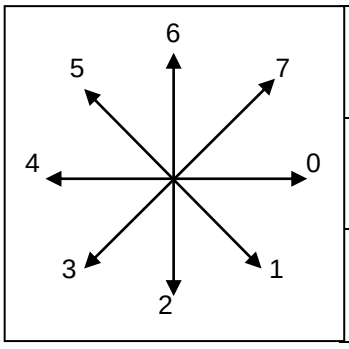
- binarizare imagine, segmentare folosind un prag pe histogramă;
- aplicare operatori morfologici: dilatare, erodare;
- implementarea unui algoritm de urmărire contur;
- implementarea unui algoritm de extragere semnătură;
- determinare formă, dimensiune, simetrie.

1) Determinare contur

Pentru binarizarea imaginii se aplică algoritmul de segmentare [146] sau [163] cu un singur prag. Având imaginea binară se urmăreşte selecţia pixelilor aparţinând conturului, de aici şi denumirea algoritmului – algoritm de urmărire a conturului [185].

Algoritmul de urmărire contur (mersul orbului) baleiază imaginea din colţul stânga sus până când se găseşte un pixel (diferit de background) care aparţine obiectului, acest pixel este notat cu P_s şi reprezintă pixelul de start al conturului. El devine pixelul curent P_c.

Se alege un sens de parcurgere a conturului. Se defineşte o variabilă *dir* care reţine direcţia mutării anterioare de-a lungul conturului de la elementul anterior spre cel curent. Se iniţializează $dir = 0$ şi brat = (dir+5) modulo 8.



$x_{i-1,j-1}$	$x_{i-1,j}$	$x_{i-1,j+1}$
$x_{i,j-1}$	$x_{i,j}$	$x_{i,j+1}$
$x_{i+1,j-1}$	$x_{i+1,j}$	$x_{i+1,j+1}$

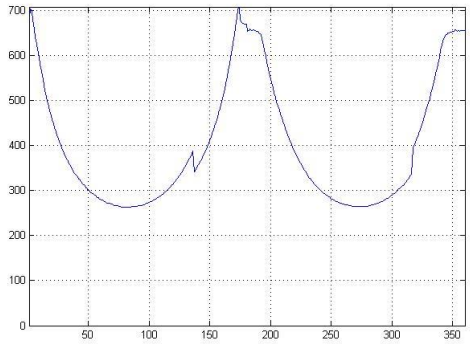
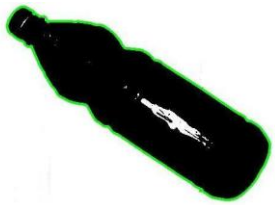
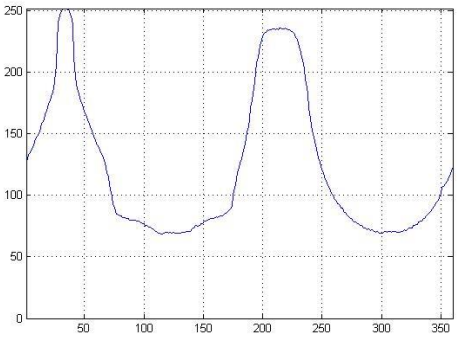
Fig. 3.3 Reprezentarea direcţiei şi vecinătatea de 3x3 a pixelului curent

Se baleiază vecinii pixelului curent, în sensul de parcurgere ales, plecând de la poziţia braţ, până la găsirea unui nou pixel obiect, care se marchează şi devine pixel curent Pc. Variabila direcţie dir este reactualizată în funcţie de pixelul curent găsit, iar variabila brat se recalculează în funcţie de *dir*.

Se repetă pasul anterior până la închiderea conturului, adică până când pixelul de start devine iar pixel curent.

Exemple de determinarea conturului sunt reprezentate în tabelul de mai jos, coloana 1.

Tabelul 3.6 Exemple de semnături

Determinare contur	**Semnătura extrasă**
	
	

2) <u>Extragerea semnăturii formei</u>

O semnătură este o reprezentare simplificată a unei suprafeţe. S-a optat pentru definirea semnăturii ca distanţa de la un punct de referinţă – centrul de greutate al formei (g_x, g_y) – la fiecare punct de pe contur [186]. Ideea este de a reduce reprezentarea conturului la o funcţie unidimensională, care este în principiu mai uşor de descris decât conturul original bidimensional.

$$d(n) = \sqrt{(x(n) - g_x)^2 + (y(n) - g_y)^2}$$

Semnătura formei este o reprezentare reversibilă – cunoscând semnătura se poate reconstrui conturul obiectului. Dimensiunea vectorului semnăturii este de 360 de valori, câte o valoare obţinută pentru fiecare valoare a unghiului semidreptei ce porneşte din punctul (g_x,g_y) şi se intersectează cu conturul imaginii. Dacă intersecţia conturului cu semidreapta ce scanează forma se realizează în mai multe puncte, atunci se păstrează valoarea cea mai îndepărtată.

3) Determinare formă, dimensiune, simetrie

Lucrarea [187] reprezintă o revizuire a tehnicilor de descriere şi reprezentare a formelor în scopul recunoaşterii obiectelor sau a clasificării automate a lor. Numărul de caracteristici distinctive este direct proporţional cu corectitudinea rezultatelor obţinute de un algoritm de recunoaştere a formelor şi indirect cu timpul de calcul. De ex., dacă culoarea nu reprezintă o trăsătură distinctivă nu se recomandă de inclus în algoritm.

Pentru descrierea ambalajelor s-au ales trei caracteristici distinctive: forma, dimensiunea şi simetria. Înainte de calcularea ariei obiectului s-au aplicat operatorii morfologici: dilatare pentru a completa „găurile" care s-au obţinut la segmentare din cauza reflectării luminii şi apoi erodare pentru a reveni la forma iniţială (se şterg pixelii adăugaţi la dilatare). Aria (A) este considerată a fi numărul total de pixeli ce reprezintă obiectul. Perimetrul (P) este reprezentat de pixelii conturului.

Forma şi dimensiunea sunt calculate conform formulelor de mai jos:

$$Forma = \frac{4\pi A}{P^2}$$
$$Dimensiune = \frac{A\_forma}{A\_imagine} \quad (1)$$

S-a notat cu A_forma aria obiectului ce se doreşte a fi identificat şi cu A_imagine – numărul total de pixeli din imagine.

După segmentare conturul formei nu reprezintă o linie netedă, ci conţine denivelări, de aceea la aplicarea formulei pentru formă, obţinem rezultate aproximative, cu o marjă de eroare. Din acest motiv se face o normalizare astfel încât valorile să fie în intervalul [0,1].

Simetria se determină după semnătură. Vom presupune că formele simetrice (după axa orizontală) reprezintă ambalaje periculoase – flacoane de spray sub presiune, baterii etc. Valorile pentru simetrie se determină pe baza corelaţiei dintre semnătura normalizată a obiectului pe intervalele [0,90]°, [90,180]°, [180,270]°, [270,360]°.

3.2.3 Un sistem de logică fuzzy pentru clasificare automată

Logica fuzzy se deosebeşte de logica binară bazată pe variabile 0 sau 1 ("adevărat" sau "fals") prin valori intermediare, adică "adevărat" sau "fals" cu un grad de apartenenţă, 0 reprezentând neapartenenţa, iar 1 – apartenenţa totală.

Sistemul fuzzy transformă exprimările lingvistice „dimensiune foarte mare", „formă alungită", „simetrie mică" care includ imprecizie / ambiguitate în valori numerice reprezentând gradele de apartenenţă în fiecare mulţime fuzzy în intervalul [0, 1], corespunzătoare adevărului

parţial cuprins între „complet adevărat” şi complet fals”. Se pot folosi şi modificatori / intensificatori lingvistici „oarecum mare”, „puţin”, „destul de puţin”, „extrem de” etc. cu valori numerice pentru intrări în sistem. Logica fuzzy redă valorile.

1) <u>Definirea mulţimilor fuzzy pentru variabile de intrare</u>

Fuzificarea intrărilor constă în determinarea gradului în care datele de intrare aparţin mulţimilor fuzzy, prin funcţia de apartenenţă. Cele mai uzuale funcţii de apartenenţă sunt de formă triunghiulară şi trapezoidală. La definirea mulţimilor am ales funcţia de apartenenţă trapezoidală, pentru că acestea spre deosebire de funcţiile de apartenenţă triunghiulare au un interval în care gradul de apartenenţă este 1, dar alegerea e subiectivă.

1.1 Definirea mulţimilor fuzzy pentru variabila de intrare *Dimensiune*

Variabila de intrare *Dimensiune* se defineşte prin 4 mulţimi fuzzy trapezoidale: Mica, Medie, Mare şi FoarteMare), având ca univers de discurs intervalul [0. 0.5]. Se consideră că obiectele de dimensiune foarte mare pot ocupa maxim 50% din spaţiul de vizualizare, de unde şi limita superioară a universului de discurs. Un obiect de dimensiune mare ocupă circa 10%-15% din spaţiul de vizualizare. Conform Fig.3.4 atunci când un obiect ocupă între 15% si 20% din spaţiul de vizualizare atunci acesta poate fi considerat ca aparţinând şi mulţimii fuzzy „mare” şi mulţimii fuzzy „foarte mare” dar cu grade de apartenenţă diferite.

Cele 4 mulţimi fuzzy pentru variabila de intrare *Dimensiune* sunt reprezentate în Fig. 3.4.

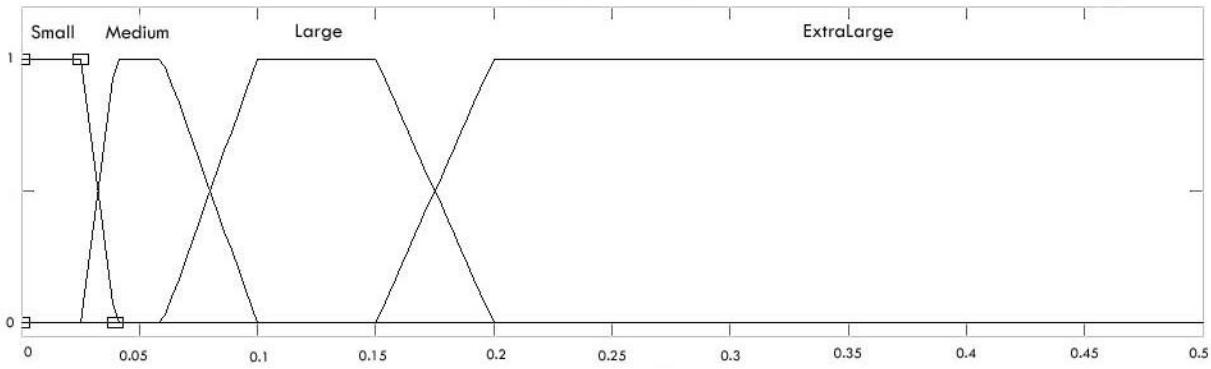


Fig. 3.4 Mulţimile fuzzy pentru variabila de intrare *Dimensiune*

1.2 Definirea mulţimilor fuzzy pentru variabila de intrare *Forma*

Variabila de intrare *Forma* se defineşte prin 4 mulţimi fuzzy trapezoidale: Alungita, Neregulata, Ovala şi Rotunda. Conform formulei (1) valorile obţinute pentru formă pot fi în intervalul [0, 1], 0 corespunzând unui obiect alungit asemănător cu un segment de dreaptă şi 1 unui cerc.

Mulţimile fuzzy pentru variabila de intrare *Forma* sunt reprezentate în Fig. 3.5

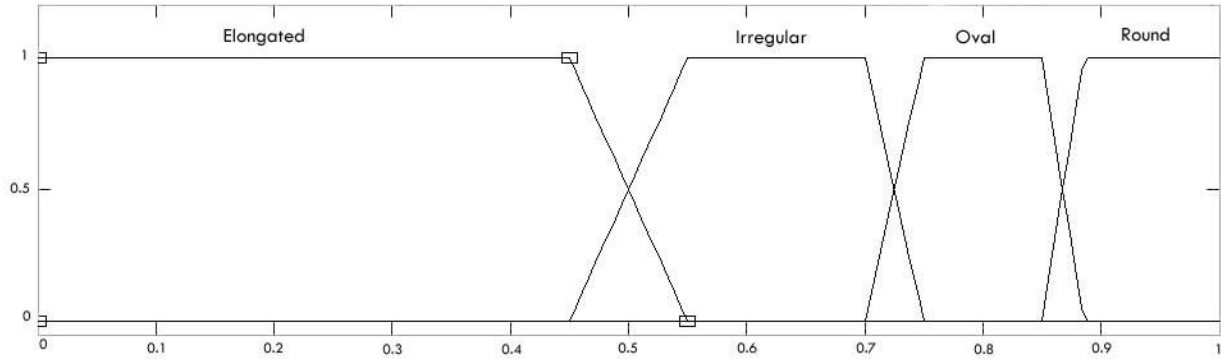


Fig.3.5 Mulţimile fuzzy pentru variabila de intrare *Forma*

1.3 Definirea mulţimilor fuzzy pentru variabila de intrare *Simetrie*

Variabila de intrare *Simetrie* se defineşte prin 3 mulţimi fuzzy trapezoidale: Mica, Medie şi Mare. Este exprimată pe baza corelaţiei dintre semnătura normalizată a obiectului pe intervalele [0,90]°, [90,180]°, [180,270]°, [270,360]°.

Definirea mulţimilor fuzzy pentru *Simetrie* s-a bazat pe propria observare: flacoanele sub presiune (de la diverse spray-uri, lacuri), bateriile sunt simetrice şi faţă de axa orizontală. Aceste ambalaje/obiecte sunt periculoase şi nu pot fi reciclate, de aceea apartenenţa la mulţimea *Simetrie* are o pondere importantă la luarea deciziei de către sistem.

Mulţimile fuzzy pentru variabila de intrare *Simetrie* sunt reprezentate în Fig. 3.6

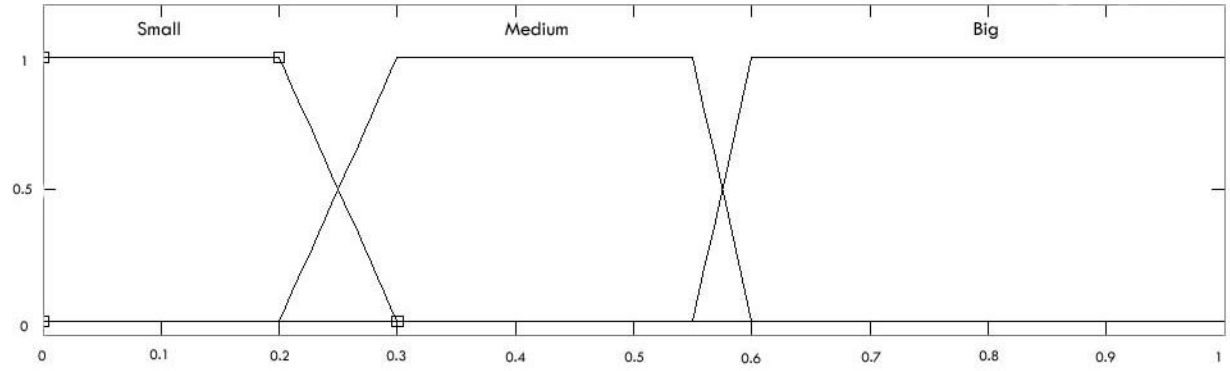


Fig. 3.6 Mulţimile fuzzy pentru variabila de intrare *Simetrie*

2) <u>Definirea mulţimilor fuzzy pentru variabile de ieşire</u>

Sistemul fuzzy clasifică deşeurile în trei clase: *periculoase* (toxice, inflamabile, dăunătoare mediului), *reciclabile* şi *nedeterminate*.

Variabila de ieşire *TipObiect* se defineşte prin mulţimi fuzzy ca în Fig.3.7, creând astfel un sistem de tip Mamdani. Funcţiile de apartenenţă triunghiulare, au baza triunghiului de valoare mică (simbolică), din acest motiv sistemul poate fi considerat mai apropiat de un sistem fuzzy de tip Sugeno. Alegerea tipului sistem Mamdani şi nu Sugeno este dată de funcţia de defuzificare de

tip MoM (Moment of Maximum). Valorile bazei celor 3 mulţimi fuzzy se aleg astfel: in jur de valoarea 0 pentru Periculos, ~ 0.5 pentru Nedeterminat şi ~1 pentru Reciclabil.

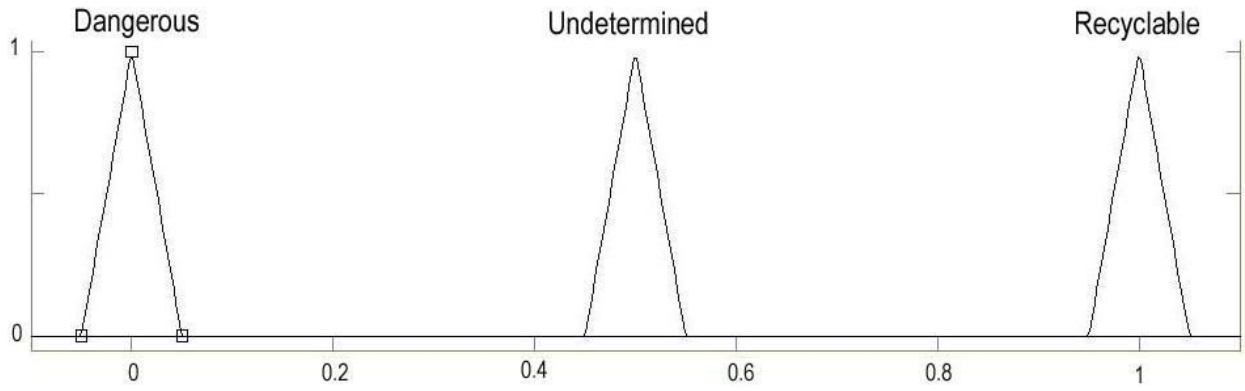


Fig. 3.7 Reprezentare ieşiri fuzzy

3) Baza de reguli a sistemului cu logică fuzzy

Regulile de inferenţă sunt acele reguli care leagă variabilele fuzzy de intrare a unui sistem cu variabile fuzzy de ieşire a aceluiaşi sistem. Aceste reguli sunt prezentate sub forma: DACĂ condiţie/premisa1 ŞI/SAU condiţie/premisa2 (ŞI/SAU ...), ATUNCI acţiune. Totalitatea regulilor formează baza de reguli, numită şi baza de cunoştinţe.

Regulile sunt afirmaţii calitative deduse din analizele realizate în procesul de reciclare a ambalajelor. Pe această bază, o enunţare intuitivă a regulilor fuzzy pentru aplicaţia prezentată a fost definită, se observă diferite ponderi asociate regulilor. Aceste ponderi se mai numesc şi coeficienţi de certitudine.

Exemple de reguli de inferenţă aplicate:

IF *Forma* alungita and *Dimensiune* Mica and *Simetrie* Mare then Tip Obiect is Periculos (1)
IF *Forma* alungita and *Dimensiune* Medie and *Simetrie* Mare then TipObiect is Periculos (1)
IF *Forma* alungita and *Dimensiune* Mica and *Simetrie* Medie then TipObiect is Periculos (0.7)
IF *Forma* alungita and *Dimensiune* Medie and *Simetrie* Mica then TipObiect is Nedeterminat (1)
IF *Forma* alungita and *Dimensiune* Mica and *Simetrie* Mica then TipObiect is Nedeterminat (1)
IF *Forma* alungita and *Dimensiune* FoarteMare and *Simetrie* Mare then TipObiect is Nedeterminat (1) etc.

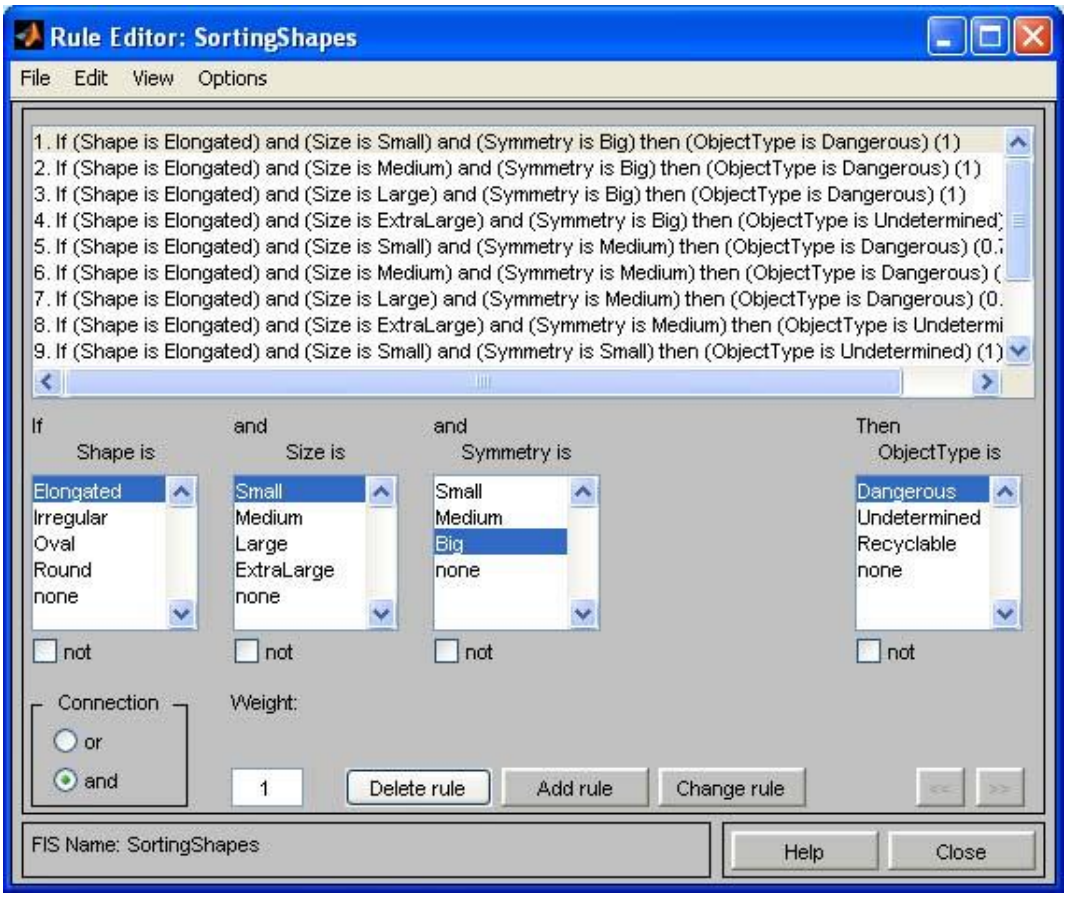


Fig. 3.8 Reguli de inferenţă a SF aplicat

Intersecţia a două mulţimi fuzzy este echivalentă cu operaţia "ŞI logic", reprezentând minimul dintre două grade de apartenenţă. Între premisele regulilor se aplică operatorul „min", valoare rezultată fiind folosită la trunchierea variabilei de ieşire. Atunci când avem mai multe reguli active se aplică o reuniune a mulţimilor fuzzy trunchiate rezultate la aplicarea fiecărei reguli.

4) <u>Operaţia de defuzzificare</u>

Pentru ca sistemul elaborat să clasifice *TipObiect* e necesar ca mulţimile fuzzy de la ieşire să fie transformate în date crisp. Transformarea dată poartă denumirea de defuzzificare. Cele mai uzuale metode de defuzificare sunt:

- Centroid, COA, COG;
- SOM (smallest of maximum);
- MOM (mean of maximum);
- LOM (largest of maximum method).

Pentru sistemul elaborat s-a ales metoda MoM – media maximelor.

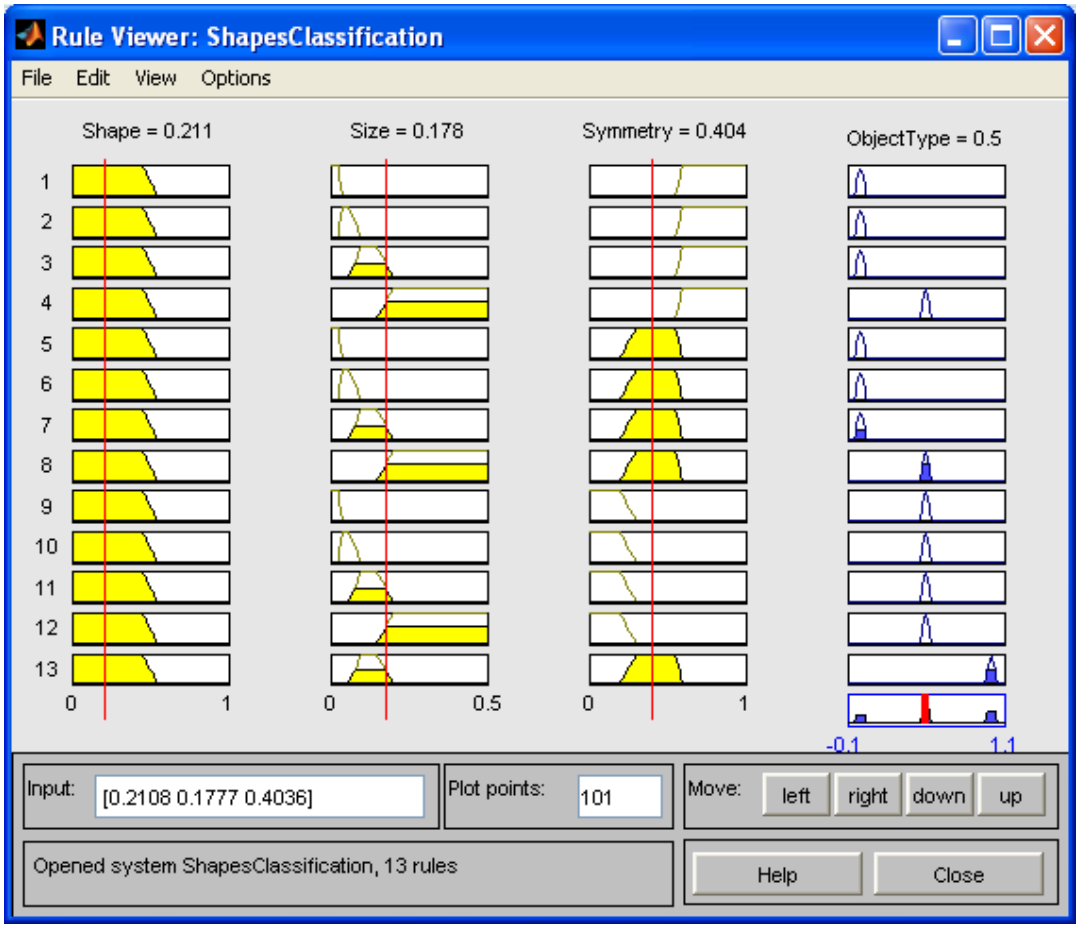


Fig. 3.9 Exemplu de defuzzificare

Sistemul fuzzy de detecţie a tipului de ambalaj aplică la defuzificare operatorul de MOM, selectează din mulţimile fuzzy active doar pe cea cu grad de apartenenţă maxim.

Randamentul acestui sistem este de 95%. Unele ambalaje din grupa *Nedeterminate* au fost identificate ca *Periculoase.* Un procent din ambalajele *Periculoase* au fost clasificate *Nedeterminate* şi ~1.5% din *Reciclabile* a fost clasificat greşit la *Nedeterminate* Important este ca aceste ambalaje să nu fie clasificate drept *Reciclabile*, dar pentru asigurarea unei clasificări automate corecte, obiectele din clasa *Reciclabile* sunt verificate la etapa următoare prin identificarea semnăturii formei.

Complexitatea algoritmului de clasificare fuzzy este dată de timpul necesar pentru calcule cumulat pentru cele 3 etape: 1.fuzzificare 2. reguli fuzzy şi 3. defuzzificare. Sistemele fuzzy sunt sisteme de control care lucrează în timp real. Multe dintre ele sunt incluse în sisteme cu microcontrolere şi pornind de la valorile citite de la senzori sunt capabile să asigure în timp real controlul unui sistem în buclă de reacţie cu performanţe mai mari decât sistemele clasice de control. Ţinând cont că frecvenţa de lucru a unui microcontroler este de ordin MHz (nu GHz ca

la un procesor) şi că ocupă puţină memorie este clar că sistemele fuzzy nu sunt algoritmi care să necesite putere mare de procesare.

Pentru fuzzificare se calculează gradele de apartenenţă pentru variabilele de intrare. Deci complexitarea ar fi $O(n_v \times n_a)$, unde n_v – numărul de variabile n_a – numărul de atribute lingvistice.

Dacă sunt alese convenabil funcţiile de apartenenţă e suficient să se calculeze un singur grad de apartenenţă şi celălalt se obţine ca fiind negatul primului. Complexitatea se reduce astfel la $O(n_v)$.

Pentru reguli fuzzy complexitatea este tot $O(n_v \times n_a)$. Acesta fiind numărul total de reguli – în cadrul regulii se aplică o simplă operaţie de minim între gradele de apartenenţă ale premiselor. Dacă e optimizat scade şi aici complexitatea pentru că sunt luate în calcul doar regulile care au toate premisele cu grade de apartenenţă diferite de zero. Numărul regulilor aplicate de sistemul fuzzy fiind practic mult mai mic.

Dacă complexitatea primelor etape era $O(n_v \times n_a)$ şi prin optimizări era diminuată la $O(n_v)$ pentru fuzzificare şi $O(n_v \times 2)$ pentru reguli fuzzy (considerând că nu putem avea mai mult de 2 grade de apartenenţă nenule pentru o variabilă de intrare), pentru ultima etapă complexitatea este cea mai redusă.

Deoarece este sistem Sugeno modificat defuzzificarea este o simplă sumă ponderată (unde ponderile erau minimul premiselor regulilor activate). Deci complexitatea sistemului fuzzy este sub cea a unei reţele neuronale (care are etapa de antrenare cu calcul numeric intensiv). SF nu necesită o antrenare, dar alegerea neinspirată a mulţimilor fuzzy asociate intrărilor, respectiv asocierea necorespunzatoare a regulilor fuzzy pentu o posibilă concluzie/clasă poate face sistemul fuzzy neperformant, dar nu este cazul sistemului propus.

Pentru determinarea timpilor de rulare s-au introdus functiile tic/toc în Matlab (acesta este un interpretor şi are oricum timpi mai mari decât s-ar fi obţinut pentru acelaşi algoritm într-un limbaj de nivel înalt). Pentru urmărire contur şi extragere semnătură s-au obţinut timpi de rulare în medie de cca 0.05-0.07 secunde (depinde de dimensiunea obiectului analizat) iar pentru sistemul fuzzy 0.013-0.014 secunde. Timpul a fost obţinut pe un calculator cu performaţe medii, folosind unul din cele două procesoare ale unui sistem Dual-core de 1,8 GHz.

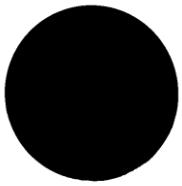
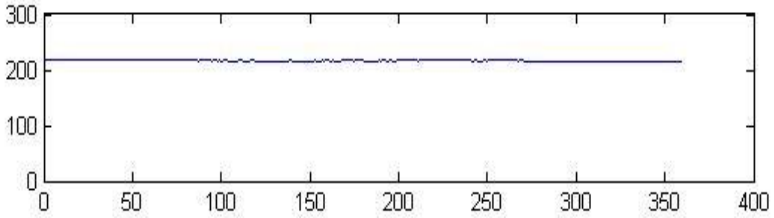
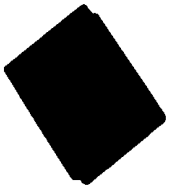
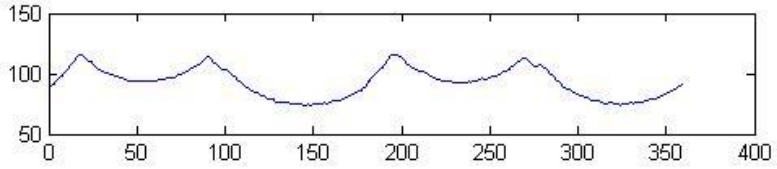
3.2.4 Identificarea formei prin funcţia diferenţă şi corelaţie aplicate pe semnături

Semnătura unei forme este o reprezentare funcţională unidimensională. Pentru sistemul propus s-a ales reprezentarea semnăturii prin distanţa de la centrul formei la fiecare punct de pe contur.

Pentru obiecte de aceeaşi formă dar de dimensiuni diferite, semnăturile vor fi asemănătoare dar de amplitudine diferită (datorită gradului diferit de scalare). Pentru obiecte de orientare diferită semnăturile vor fi asemănătoare dar deplasate spre stânga / dreapta. Aplicând o normalizare asupra semnăturii se elimină efectul scalării. Atât funcţia de corelaţie, cât şi funcţia diferenţă asigură invarianţa la rotaţie.

O altă posibilitate de eliminare a efectului de rotaţie ar fi selectarea punctului de start pentru generarea semnăturii ca fiind punctul cel mai îndepărtat de centru (dacă este punct unic) sau punctul cel mai îndepărtat de pe axe.

Tabelul 3.7 Reprezentarea grafică a semnăturilor

Forma	**Semnătura formei**
	
	

O posibilitate de a obţine invarianţă la mărimea formelor ar fi normalizarea valorilor obţinute: reprezentarea lor în intervalul 0 şi 1. Această modalitate e sensibilă la zgomotele ce afectează valorile maximă şi minimă. Pentru sistemul elaborat mărimea ambalajelor contează, de aceea nu s-a efectuat normalizarea.

Corelaţia este o metodă statistică aplicată la determinarea unei relaţii dintre două sau mai multe variabile. *Coeficientul de corelaţie* (r) este o valoare cantitativă între (-1 şi +1) şi descrie relaţia dintre variabile. Valorile extreme presupun o corelaţie strânsă între variabile în timp ce $r = 0$ reprezintă lipsa totală de relaţie liniară.

Tabelul 3.8 Interpretarea coeficientului de corelaţie Pearson

	Coeficientul, *r*	
Gradul de asociere	Pozitiv	Negativ
Mic	.1 to .3	-0.1 to -0.3
Mediu	.3 to .5	-0.3 to -0.5
Mare	.5 to 1.0	-0.5 to -1.0

Definit pentru două variabile cu repartiţie normală (x,y) coeficientul corelaţiei lineare, numit coeficientul lui Pearson (coeficientul corelaţiei totale), se determină în felul următor [188]:

$$r_{xy} = \frac{\sum_{i=1}^{n}(x_i - m_x)(y_i - m_y)}{\sqrt{\sum_{i=1}^{n}(x_i - m_x)^2 \sum_{i=1}^{n}(y_i - m_y)^2}}$$

unde

$$m_x - media\ valorilor\ variabilei\ x : m_x = \sum_{i=1}^{n} x_i / n$$

$$m_y - media\ valorilor\ variabilei\ y : m_y = \sum_{i=1}^{n} y_i / n$$

O valoare mică a coeficientului Pearson nu semnifică independenţa celor două caracteristici, ci doar indică o necorelare liniară a lor. Acestea pot fi corelate printr-un alt tip de relaţie funcţională – parabolic, logaritmic etc.

Coeficientul cosinus θ este o măsură a distanţei unghiulare, utilizat pentru estimarea similarităţii între forme.

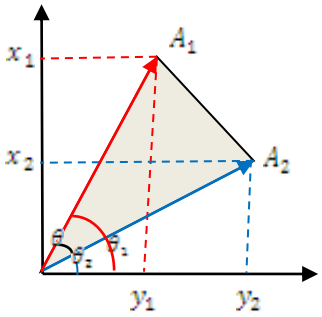


Fig. 3.10 Coeficientul cosinus θ pentru un spaţiu bidimensional [188]

Calculul coeficientului cosinus θ într-un spaţiu bidimensional se bazează pe relaţiile trigonometrice ale unghiurilor şi este exprimat prin următoarea formulă:

$$\cos\theta_{A_1 A_2} = \cos(\theta_1 - \theta_2) = \frac{x_1 y_1 + x_2 y_2}{\sqrt{\left(x_1^2 + y_1^2\right)\left(x_2^2 + y_2^2\right)}}$$

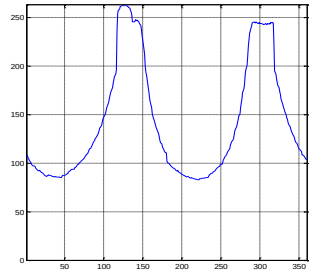
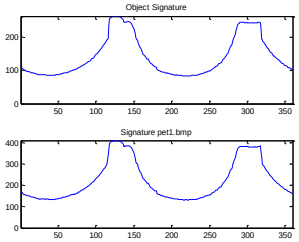
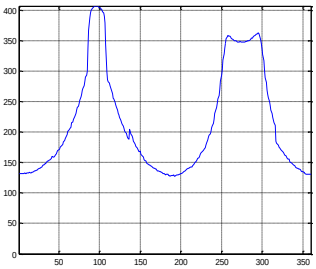
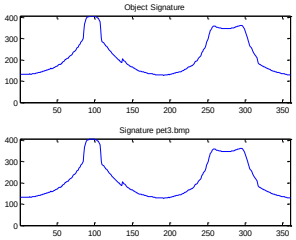
Formula de mai sus poate fi adaptată pentru *n* factori independenţi:

$$\cos\theta_{A_1 A_2} = \frac{\sum_{i=1}^{n} x_i y_i}{\sqrt{\sum_{i=1}^{n} x_i^2 \sum_{i=1}^{n} y_i^2}}$$

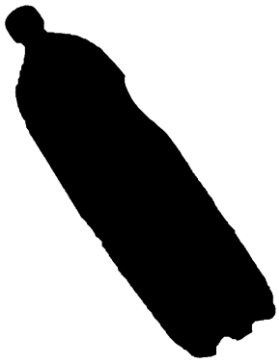
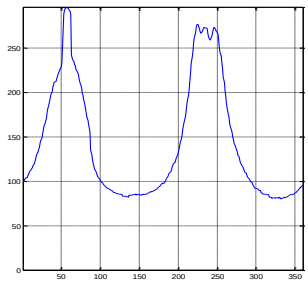
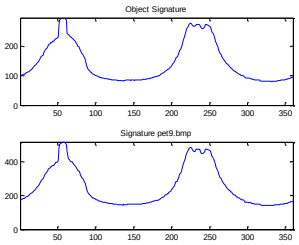
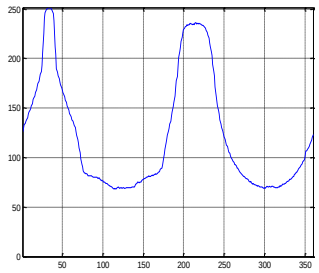
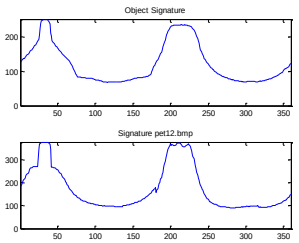
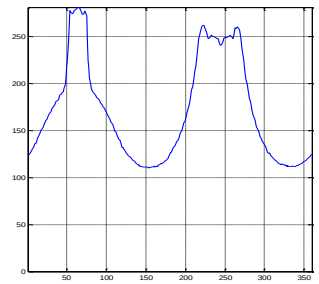
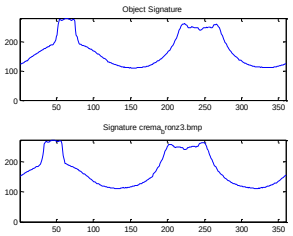
Acest coeficient de corelaţie indică o similaritate completă între două forme A_1 şi A_2 dacă $\cos\theta = 1$ şi o disimilaritate totală dacă $\cos\theta = 0$.

S-au testat ambele formule de calcul şi s-a optat pentru coeficientul corelaţiei totale (coef. lui Pearson). Baza de date conţine semnăturile tuturor formelor (reciclabile, nedeterminate şi periculoase) pentru a putea identifica ambalaje nereciclabile. Au fost testate şi obiecte care nu sunt incluse în BD, rezultatul e satisfăcător, s-a obţinut un randament de recunoaştere de 97%.

Tabelul 3.9 Identificarea semnăturilor prin coeficientul de corelaţie totală

Forma	**Semnătura formei**	**Identificarea semnăturii formei**
1	**2**	**3**
		**Reciclabil (pet1.bmp)** 
		**Reciclabil (pet3.bmp)** 

Continuarea tabelului 3.9

1	2	3
		**Reciclabil (pet9.bmp)** 
 **Notă*** Item reciclabil, ce nu aparţine BD		**Reciclabil (pet12.bmp)** 
 **Notă*** Item nereciclabil, intenţionat inclus		 **Nedeterminat** **(ambalaj_crema.bmp)**

3.3 Compararea performanţelor metodei propuse vs. alte metode

3.3.1 Metoda hibridă bazată pe CCN şi GA vs. metodele SIFT şi ASIFT

Algoritmul *Affine Scale-Invariant Feature Transform* are rezultate bune la rotire, scalare, schimbarea unghiului, iluminare, dar pentru setul dat de imagini, la determinarea similarităţii doar în 40% din cazuri am obţinut rezultate corecte. Pentru adaptarea acestui algoritm la un set concret de imagini ar trebui să se ia în consideraţie numărul de potriviri ale punctelor de interes; imaginile pentru care s-au determinat aproximativ 15 000 de puncte de interes şi se potrivesc doar 5 nu pot fi similare.

Scale-Invariant Feature Transform determină un număr de puncte de interes cu mult mai mic decât ASIFT şi rata lui de succes pentru setul dat de imagini este peste 70% cu imagini de aceleaşi dimensiuni, scară şi iluminare.

În cazul metodelor SIFT şi ASIFT se obţin rezultate diferite la compararea imaginii cu un şablon faţă de compararea şablonului cu aceeaşi imagine, un exemplu este ilustrat în capitolul 1, figura 1.11, ceea ce nu oferă stabilitate rezultatelor.

Algoritmul hibrid bazat pe CCN şi GA are o reuşită de 97%. Tabelul de mai jos arată avantajele acestei metode.

Tabelul 3.10 Compararea algoritmului hibrid propus cu alte metode de referinţă

SIFT	**ASIFT**	**Template Matching bazat pe CCN**	**Algoritmul bazat pe CCN şi GA**
			
"230 keypoints found. 40802 keypoints found. **Found 0 matches."**	"6569 ASIFT keypoints are detected. 31986 ASIFT keypoints are detected. **The two images do not match**."	„**Symbol correctly identified.**"	Fitness:0.747223 Scale:1.27 Rotate:0.00
			
"403 keypoints found. 40802 keypoints found. **Found 0 matches**."	"16089 ASIFT keypoints are detected. 31986 ASIFT keypoints are detected. **The two images do not match**."	„**Symbol correctly identified.**"	Fitness:0.616609 Scale:0.53 Rotate:0.00
			
"269 keypoints found. 5262 keypoints found. **Found 0 matches.**"	"11606 ASIFT keypoints are detected. 18148 ASIFT keypoints are detected. **The two images match!** 38 matchings are identified."	„**Symbol wrongly identified.** **Note:** If we do not put a threshold for the correlation coefficient we get false results."	„The image does not contain the searched symbol! It could be changed the interval conditioning of the fitness."

Analizând exemplele din tabelul de mai sus observăm optimizarea rezultatelor obţinute la aplicarea coeficientului de corelaţie prin generarea a noi seturi de valori cu algoritmul genetic.

3.3.2 Sistemul fuzzy vs. metoda de clasificare kNN

Atât sistemul fuzzy, cât şi algoritmul kNN (cu numărul de vecini k=1 şi k=2) au o acurateţe de clasificare de peste 90%. ~1.5% din reciclabile a fost clasificat greşit de ambele metode la nedeterminate, ~3.1% din nedeterminate au fost clasificate – periculoase, iar knn a clasificat ~3% din ambalajele periculoase drept nedeterminate. ~1.5% ambalaje din clasa periculoase algoritmul kNN clasifică ca reciclabile, dacă nu ar urma un pas de verificare, ar fi o eroare cu consecinţe neadmisibile.

Pentru un set de 64 obiecte, Sistemul Fuzzy a avut un F-score de 94%, pe când kNN pentru k=1 şi k=2 a avut un F-score de 87%. Odată cu creşterea valorii k, acurateţea şi F-scorul descreşte.

Tabelul 3.11 Evaluarea performaţei sistemului Fuzzy şi kNN

Metrică	Fuzzy	kNN
Acuratetea	95,24%	90,48%
Rata de eroare	4,76%	9,52%
Precizia	88,89%	90,91%
Recall	100,00%	83,33%
Fscore	94,12%	86,96%

3.4 Concluzii la capitolul 3

În acest capitol au fost implementaţi algoritmi de recunoaştere şi de clasificare automată a obiectelor. Iniţial s-a propus elaborarea unui algoritm de clasificare automată a datelor bazată pe separarea liniară. S-a propus reformularea condiţiilor de optimalitate Kunh-Tucker într-un sistem echivalent de ecuaţii neliniare (cubice) netede. S-a demonstrat că sistemul de ecuaţii poate fi rezolvat eficient cu ajutorul metodei Newton. Metoda propusă este promiţătoare, dar deoarece la clasificare avem mai multe caracteristici (nu numai două) s-a propus crearea unui sistem hibrid de clasificare.

Sistemul de decizie elaborat este bazat pe logica fuzzy şi are ca scop clasificarea automată a deşeurilor (ambalajelor de plastic). Programul este scris în mediul Matlab şi conţine 3 paşi esenţiali: algoritmul hibrid din coeficientul de corelaţie normalizat şi algoritm genetic, algoritmul de clasificare automată bazat pe logica fuzzy şi algoritmul de identificare a semnăturilor.

Metoda hibridă – CCN+GA – permite depistarea simbolurilor de pericol de pe etichetă cu o rată foarte mare de reuşită. Implementarea algoritmului genetic pentru generarea de noi valori pentru rotire şi scalare este o soluţie atunci când imaginea diferă prin unghiul de captare a imaginii şi mărime. GA reduce paşii de calcul ajungând la un rezultat optim într-un interval de timp rezonabil.

Extragerea conturului obiectelor în scopul obţinerii semnăturii se face cu algoritmul de urmărire a conturului. Determinarea corectă a conturului permite de asemenea calcularea eficientă a formei şi dimensiunii – caracteristici utilizate în sistemul fuzzy.

Clasificarea automată a deşeurilor este realizată cu un sistem fuzzy a cărui date de intrare sunt: forma, dimensiunea şi simetria. Alegerea mulţimilor fuzzy este subiectivă şi a fost făcută în urma analizei ambalajelor ce ar trebui să fie reciclate şi cele periculoase care trebuie să fie colectate în containere speciale. Pentru mărirea ponderii deciziei precum că anumite deşeuri reprezintă pericol sau pot fi reciclate se implementează un algoritm de identificare a semnăturii formei (ambalajului).

Sistemul elaborat poate fi adaptat şi la sortarea altor obiecte. S-a ales sortarea deşeurilor pentru a accentua faptul că trierea lor corectă ar fi o soluţie pentru diminuarea poluării, un factor important în zilele noastre.

CONCLUZII GENERALE ŞI RECOMANDĂRI

Lucrarea conţine contribuţii originale la etape importante de procesare a imaginilor: segmentare, recunoaştere şi clasificare automată a formelor. Imaginile test utilizate s-au considerat a fi calitative şi nu au fost supuse unor îmbunătăţiri.

Sintetizând munca din perioada cercetării pot fi prezentate următoarele rezultate ştiinţifice:

1) S-a elaborat un algoritm de segmentare bazat pe G-U-MM.

Au fost analizate mai multe metode de segmentare. S-a elaborat un algoritm de segmentare bazat pe Modelul de Mixturi Gaussiene şi Uniforme (G-U-MM). În conformitate cu faptul că histograma imaginii poate fi utilizată pentru a reprezenta caracterul statistic al funcţiei densitate de probabilitate, G-U-MM este folosit pentru a estima PDF a imaginii cu nivele de gri (poate fi adaptat şi la imagini color). Pragurile optime au fost determinate ca limitele intervalelor normale şi uniforme. A fost prezentată o explicaţie detaliată pentru raţiunea modelelor de mixturi Gaussiene şi uniforme propuse ca un fundament al procesului de segmentare. Rezultatele experimentale arată că metoda propusă poate obţine rezultate mai bune şi este mai robustă.

Majoritatea metodelor cunoscute de segmentare necesită introducerea de către utilizator a unor parametri, de obicei numărul de segmente dorit. În cazul segmentării bazate pe histogramă acestea ar putea fi numărul de maxime gaussiene, în cazul unul algoritm de clusterizare – numărul de centroizi, ceea ce înseamnă de fapt indicarea a câte obiecte sunt reprezentate în imagine. S-a optat pentru o metodă de segmentare mai robustă, adică are loc detectarea automată a numărului de segmente (a pragurilor pe histogramă) utilizând cunoştinţe despre obiecte precum intensitatea pixelilor.

Descrierea algoritmului de segmentare şi a rezultatelor obţinute au fost prezentate la conferinţe şi publicate în [146,147, 149, 150, 155].

2) S-a făcut o analiză comparativă a algoritmilor de segmentare şi a rezultatelor obţinute, demonstrând eficacitatea algoritmului propus. Pentru a putea aprecia calitatea rezultatelor obţinute, au fost calculate unele funcţii propuse pentru evaluarea cantitativă şi au fost comparate rezultatele cu unele din literatură considerate drept referinţă: metoda Otsu şi metoda bazată pe histogramă (se determină vârfurile Gaussienelor – maxime globale, iar pragurile sunt determinate ca minimul între două vârfuri Gaussiene).

Estimarea obiectivă a calităţii segmentării poate fi elaborată doar în cazul când numărul de segmente este acelaşi. Analizând graficele prezentate în capitolul doi se poate confirma că

rezultatele obţinute sunt apropiate celor două metode de referinţă, rezultând astfel eficacitatea segmentării, respectiv cea a algoritmului.

Compararea metodelor de segmentare şi evaluarea obiectivă a rezultatelor obţinute a fost prezentată în [161].

3) S-a făcut o analiză a metodelor de extragere a punctelor de interes şi de corelare a lor: SIFT şi ASIFT, cât şi a algoritmului Template Matching observând avantajele şi dezavantajele fiecărei metode. Primele două metode reprezintă algoritmi complecşi de recunoaştere a imaginilor pe baza potrivirii punctelor de interes. Rezultatele au fost prezentate în [184].

4) A fost îmbunătăţit algoritmul Template Matching prin completarea programului cu un algoritm genetic pentru generarea a noi valori folosite de coeficientul de corelaţie normalizat (NCC) la potrivirea imaginii cu un şablon. A fost implementată această metodă hibiridă pentru a elimina problemele observate la rotire, scalare folosind doar CCN. Coeficientul de corelaţie determină potrivirea sau nu a două imagini, iar algoritmul genetic generează noi dimensiuni ale imaginii căutate de la 1/3 până la 3.0 şi cu valori pentru rotire de la 1 până la 359 grade.

Descrierea algoritmului şi unele rezultate obţinute au fost prezentate în [178].

5) A fost propusă o îmbunătăţire a implementării metodei de clasificare automată SVM. S-au abordat trei modele ale separării liniare. Pentru unul din aceste modele s-a propus o procedură efectivă de rezolvare numerică prin reformularea condiţiilor de optimalitate Karush-Kunh-Tucker şi utilizând metoda Newton. Rezultatele au fost publicate în [166].

6) A fost elaborat un sistem de recunoaştere şi clasificare automată a formelor bazat pe logica fuzzy. Etapele de recunoaştere se completează, dând astfel un randament mare de reuşită. Pe un set de 63 de imagini, s-a obţinut o acurateţe de 95%. S-a propus o etapă de verificare a rezultatelor clasificării prin identificarea semnăturii.

Paşii sistemului şi rezultatele preliminare au fost prezentate la simpozionul *Information in Image and Video Analysis Theory and Applications* (IIVA) 2016, Iasi- Romania.

Sistemul poate fi adaptat uşor şi pentru alte forme decât cele din lucrare.

7) S-au comparat rezultatele obţinute la fiecare etapă a procesării imaginilor cu metode respective de referinţă (segmentare, recunoaştere forme). Rezultatele recunoaşterii imaginii, aplicând algoritmul hibrid – Coeficientul de Corelaţie Normalizat (CCN) + Algoritm Genetic –, au fost comparate cu rezultatele obţinute la recunoaşterea imaginii utilizând doar CCN, demonstrându-se astfel avantajul algoritmului hibrid. În comparaţie cu rezultatele metodelor

SIFT şi ASIFT, algoritmul hibrid dă rezultate satisfăcătoare chiar şi atunci când rezoluţia imaginii este mică şi când imaginile sunt asemănătoare, dar nu identice.

Rezultatele obţinute în urma clasificării au o acurateţe a algoritmului de 95%. Algoritmul de referinţă kNN, pentru k=1 şi k=2 are o acurateţe de 90.5%, care scade odată cu creşterea numărului de vecini (k). Sistemul Fuzzy elaborat are un F-score de 94%, pe când kNN pentru k=1 şi k=2 are un F-score de 87%.

Observaţii generale în urma cercetărilor efectuate:

- metodele de procesare se aleg în dependenţă de setul de imagini şi informaţia ce necesită extrasă;
- este dificil de apreciat rezultatele segmentării (compararea cu un şablon segmentat manual este minuţioasă şi necesită timp);
- clasificarea automată (nesupravegheată) a produselor în timp real (sau chiar mai rapid decât ar face o persoană) e posibilă doar pentru un număr redus de forme sau de acelaşi tip dar cu anumite defecte.

Direcţii de cercetare pentru viitor. Ca viitoare direcţii de cercetare se propune îmbunătăţirea abordărilor prezentate în teză, extinderea evaluărilor algoritmilor elaboraţi şi investigarea sau dezvoltarea altor metode şi modele pentru recunoaşterea formelor.

Direcţiile viitoare de cercetare ar fi următoarele:

- cercetarea posibilităţii de a simplifica paşii algoritmului de recunoaştere a simbolurilor de pericol prin recunoaşterea textului şi ştergerea lui de pe etichetă, rămânând astfel doar simbolurile de atenţionare. Acest algoritm ar putea fi implementat şi la detectarea logourilor de pe un document;
- implementarea unui sistem hibrid GA Fuzzy pentru identificarea altor obiecte decât cele existente în BD;
- aplicarea logicii Fuzzy şi în alte domenii decât procesarea imaginilor.

BIBLIOGRAFIE

1. Mathias M., Timofte R., Benenson R. and Van Gool L., Traffic Sign Recognition - How far are we from the solution?, Proceedings of IEEE International Joint Conference on Neural Networks, USA, 8 p. 2013
2. Timofte R., Zimmermann K., and Van Gool L., Multi-view traffic sign detection, recognition, and 3D localization, Journal of Machine Vision and Applications, 15 p., Springer-Verlag, 2011
3. Chen L., Li Q., Li M., Mao Q., Traffic sign detection and recognition for intelligent vehicle, Intelligent Vehicles Symposium (IV), IEEE, p. 908 – 913, 2011
4. Jain R., Doermann D., "Logo retrieval in document images," in Document Analysis Systems (DAS), 2012 10th IAPR International Workshop on, p. 135–139, IEEE, 2012
5. Augereau O. , Journet N., Domenger J.-Ph., Semi-structured document image matching and recognition, Document Recognition and Retrieval XX, edited by, Proc. of SPIE-IS&T Electronic Imaging, SPIE Vol. 8658, 13 p., 2013
6. Zhu G., Zheng Y., Doermann D., Jaeger S., Signature Detection and Matching for Document Image Retrieval, IEEE Transactions on Pattern Analysis and Machine Intelligence, Vol. 31 (11), p. 2015 – 2031, 2009
7. Iancu I., Constantinescu N., Colhon M., Fingerprints Identification using a Fuzzy Logic System, Int. J. of Computers, Communications & Control, p.1841-9844 Vol. V, No. 4, p. 525-531, 2010
8. Dyre Sh., Sumathi C.P., A survey on various approaches to fingerprint matching for personal verification and identification, International Journal of Computer Science & Engineering Survey (IJCSES), Vol.7, No.4, 17 p. 2016
9. Li H., Lin Z., Brandt J., Shen X., Hua G., Efficient Boosted Exemplar-Based Face Detection, IEEE Conference on Computer Vision and Pattern Recognition, p.1843-1850, 2014
10. Chauhan M., Sakle M., Study & Analysis of Different Face Detection Techniques, International Journal of Computer Science and Information Technologies (IJCSIT), Vol. 5 (2) , p.1615-1618, 2014
11. Raghavachari C., Aparna V, Chithira S, Balasubramanian V., A Comparative Study of Vision based Human Detection Techniques in People Counting Applications, Second International Symposium on Computer Vision and the Internet (VisionNet'15), Elsevier B.V, p. 461 – 469, 2015

12. Jiang Y., Ma J., Combination Features and Models for Human Detection, IEEE Conference on Computer Vision and Pattern Recognition (CVPR), p.240-248, 2015
13. Garcia-Martin A., Martinez J. M., Robust real time moving people detection in surveillance scenarios, In Advanced Video and Signal Based Surveillance, Seventh IEEE International Conference on, p. 241–247, 2010
14. Hadi R.A., Sulong G., George L.E., Vehicle detection and tracking techniques: a concise review, Signal & Image Processing : An International Journal (SIPIJ) Vol.5, No.1, 13 p. , 2014
15. Arya K.V., Tiwari S., Behwalc S., Real-time vehicle detection and tracking, 13th International Conference on Electrical Engineering/Electronics, Computer, Telecommunications and Information Technology (ECTI-CON), 2016
16. Fujita H., Nogata F., Jiang H. et all, Medical image processing and computer-aided detection/diagnosis (CAD), International Conference on Computerized Healthcare (ICCH), p. 66-71, 2012
17. Katsumata A., Fujita H., Progress of computer-aided detection/diagnosis (CAD) in dentistry, Japanese Dental Science Review, Vol.50 (3), p. 63-68, 2014
18. Sharma P., Malik S., Sehgal S. and Pruthi J., Computer Aided Diagnosis Based on Medical Image Processing and Artificial Intelligence Methods, International Journal of Information and Computation Technology, Vol. 3, No. 9, p. 887-892, 2013
19. Devabalan P., Satellite image processing on a grid based computing environment, International Journal of Computer Science and Mobile Computing, IJCSMC, Vol. 3 (3), p.1039 – 1044, 2014
20. Chiang Y.Y., Leyk S., Knoblock C.A., A Survey of Digital Map Processing Techniques, Journal, ACM Computing Surveys, Vol.47, No.1, 44 p., 2014
21. Sakaguchi H., Kagawa Y., Kazama Y., Satellite Imagery Solution for Natural Resources, Hitachi Review, Vol. 61, No. 12, p.28-34, 2012
22. Vanjare A., Omkar S.N., Senthilnath J., Satellite Image Processing for Land Use and Land Cover Mapping International Journal of Image, Graphics and Signal Processing (IJIGSP), Vol. 6, No. 10, p. 18-28, 2014
23. Sheffield K., Morse-McNabb E., Using satellite imagery to asses trends in soil and crop productivity across landscapes, IOP Conference Series: Earth and Environmental Science, Vol.25, 11p., 2015

24. Jenkinson J., Grigoryan A.M., Hajinoroozi M. et all, Machine Learning and Image Processing in Astronomy with Sparse Data Sets, IEEE International Conference on Systems, Man and Cybernetics (SMC), p.206-209, 2014
25. Kremer J., Stensbo-Smidt K., Gieseke F., Big Universe, Big Data: Machine Learning and Image Analysis for Astronomy, IEEE Intelligent Systems, 15 p., 2017
26. Bloisi D. D., Iocchi L., Nardi D. et all., Ground Traffic Surveillance System for Air Traffic Control, Conference: Proceeding of the12th International Conference on Intelligent Transport Systems Telecommunications (ITST), 5p., 2012
27. Forlenza L., Fasano G., Accardo D., Moccia A., Flight Performance Analysis of an Image Processing Algorithm for Integrated Sense-and-Avoid Systems, International Journal of Aerospace Engineering, 8p., 2012
28. Theuma K., Zammit-Mangion D., An Image Processing Algorithm for Ground Navigation of Aircraft, Advances in Aerospace Guidance, Navigation and Control, p.381-399, 2015
29. Yanushevsky R., Guidance of Unmanned Aerial Vehicles, CRC Press., 353 p., 2011
30. Gray G., Aouf N., Richardson M.A. et all, An intelligent tracking algorithm for an imaging infrared anti-ship missile, Conference: SPIE Security + Defence, 2012
31. Karacor A., Abay R., Torun E., Aircraft classification using image processing techniques and artificial neural networks, Pattern Recognition and Artificial Intelligence, Vol. 25, 2011
32. Roopa K., Vasudeviah R., Raj P., Neural Network Classifier for Fighter Aircraft Model Recognition, Journal of Intelligent Systems, Vol. 27(3), 2017
33. Neagoe V.-E., Carata S.-V., Ciotec A.-D., An advanced neural network-based approach for military ground vehicle recognition in SAR aerial imagery, Scientific research and education in the air force-AFASES, p.41-47, 2016
34. Bekers W., De Meyer R., Strobbe T., Shape recognition for ships: World War I naval camouflage under the magnifying glass, Proceedings of the 3rd International Conference on Defence Sites: Heritage and Future (DSHF), 12p. 2016
35. Shimoyama T., Maritime Infrastructure Security Using Underwater Sonar Systems, Hitachi Review, Vol. 62, No. 3, p. 214-218, 2013
36. Charalambous E. ş.a., Automated motion detection from space in sea surveillance, Proc. SPIE, Third International Conference on Remote Sensing and Geoinformation of the Environment, Vol. 9535, 10 p, 2015
37. Leal N., Leal E., Sanchez G., Marine vessel recognition by acoustic signature, ARPN Journal of Engineering and Applied Sciences, Vol. 10, No 20, p. 9633-9639, 2015

38. Han J., Yang P., Zhang Lu, Object Recognition System of Sonar Image Based on Multiple Invariant Moments and BP Neural Network, International Journal of Signal Processing, Image Processing and Pattern Recognition, Vol.7, No.5, pp.287-298, 2014
39. Yong C.Y., Sudirman R., Chew K. M., Motion Detection and Analysis with Four Different Detectors, Third International Conference on Computational Intelligence, Modelling & Simulation, p.46-50, 2011
40. Pavithra S, Mahanthesh U. M., Stafford M., Shivakumar M., Human Motion Detection and Tracking for Real-Time Security System, nternational Journal of Advanced Research in Computer and Communication Engineering, Vol. 5 (1), p.203-207, 2016
41. Feng G., Li C., Implementation of Real-Time Motion Detecting Algorithm Based on DSP, First International Conference on Multimedia and Image Processing (ICMIP), 2016
42. Jodoin P.-M., Piérard S., Wang Y., Van Droogenbroeck M., Overview and Benchmarking of Motion Detection Methods, chapter 24, Background Modeling and Foreground Detection for Video Surveillance, CRC Press, 26p., 2014
43. Kelly D., Automatic Recognition of Head Movement Gestures in Sign Language Sentences, Proceedings of the 4th China-Ireland Information and Communications Technologies Conference, p.142-145, 2011
44. Lee S.W., Automatic gesture recognition for intelligent human-robot interaction, 7th International Conference on Automatic Face and Gesture Recognition, p. 645 – 650, 2006
45. Loconsole C., Chiaradia D., Bevilacqua V., Frisoli A., Real-Time Emotion Recognition: An Improved Hybrid Approach for Classification Performance, International Conference on Intelligent Computing, p 320-331, 2014
46. Robert Niese ș.a., Emotion Recognition based on 2D-3D Facial Feature Extraction from Color Image Sequences, Journal of Multimedia, Vol. 5, No. 5, p. 488-500, 2010
47. Stolle K. (exhibition director laser world of photonics), Image Processing in the Automotive Industry, 22nd International Trade Fair and Congress for Optical Technologies — Components, Systems and Applications, Germany 16 p., 2015
48. Demant C, Streicher-Abel B., Garnica C., Industrial Image Processing: Visual Quality Control in Manufacturing, Springer-Verlag Berlin Heidelberg, XVII, 369 p., 2013
49. Davies E.R., Machine vision in the food industry, Woodhead Publishing Series in Food Science, Technology and Nutrition, p. 75–110, 2013
50. Vans M., Automatic visual inspection and defect detection on Variable Data Prints, Hewlett-Packard Development Company, 41 p., 2010

51. Omidi-Arjenaki O., Moghaddam P., Motlagh A.M., Online tomato sorting based on shape, maturity, size, and surface defects using machine vision, Turkish Journal of Agriculture and Forestry, Vol. 37(1), p.62-68, 2012
52. Stoliński S., Bieniecki W., Application of OCR systems to processing and digitization of paper documents, WULS Press Warszawa, p. 102-111, 2011
53. Singh A., Bacchuwar K., Bhasin A., A Survey of OCR Applications, International Journal of Machine Learning and Computing, Vol. 2, No. 3, p.314-318, 2012
54. Objelean N., An approach of statistical language model for voice recognition, International Conference MITRE-2011, Chisinau, p.145-146, 2011
55. Ispas I., Modelare şi modele matematice în recunoaşterea obiectelor şi clasificarea automată a imaginilor, Universitatea Petru Maior, Târgu Mureş, 20 p., disponibil pe http://www.upm.ro/facultati_departamente/stiinte_litere/conferinte/situl_integrare_europeana/Lucrari/Ispas.pdf
56. Latharani T.R., Kurian M.Z., Chidananda Murthy M.V., Various object recognition techniques for computer vision, Journal of Analysis and Computation, Vol. 7, No. 1, p. 39-47, 2011
57. Sharma P., Kaur M., Classification in Pattern Recognition: A Review, International Journal of Advanced Research in Computer Science and Software Engineering, Vol. 3, No. 4, p. 298-306, 2013
58. Boiman O., Shechtman E., Irani M., In Defense of Nearest-Neighbor Based Image Classification, IEEE Conference on Computer Vision and Pattern Recognition, 8p. 2008
59. Dasarathy, B. V., Nearest Neighbor (NN) Norms, NN Pattern Classification Techniques, IEEE Computer Society Press, 550 p., 1990
60. Anthony Barnett1 , Jay Santokhi1 , Michael Simpson, Image Classification using non-linear Support Vector Machines on Encrypted Data, IACR Cryptology ePrint Archive, 20 p., 2017
61. Moraru V., Rusu M., „Algorithm for linear pattern separation”, Meridian Ingineresc, No. 2, pp. 26-29, 2013 http://www.utm.md/meridian/2013/MI_nr.2_2013_integral.pdf
62. Park D.-C., Image Classification Using Naïve Bayes Classifier, International Journal of Computer Science and Electronics Engineering (IJCSEE), Vol. 4 (3), p.135-139, 2016
63. Goswami D. , Kalkan S. , Krüger N., Bayesian Classification of Image Structures, Scandinavian Conference on Image Analysis, p. 676-685, 2009

64. Ruiz P., Mateos J., Camps-Valls G. et all, Bayesian Active Remote Sensing Image Classification, IEEE Transactions on Geoscience and Remote Sensing, Vol. 52(4), p. 2186-2196, 2014
65. Samso M., A Bayesian method for classification of images from electron micrographs, Journal of Structural Biology, Vol. 138, p.157–170, 2002
66. Vidal R., Ma Y., Sastry S., Generalized Principal Component Analysis (GPCA), IEEE Transactions on Pattern Analysis and Machine Intelligence, Vol. 27, No. 12, 15p., 2005
67. Sahu M., Verma T., Recognition of Object Using Principle Component Analysis, International Journal of Advance Research in Computer Science and Management Studies, Vol. 2, No. 9, p.180-186, 2014
68. Satonkar S. S. , Kurhe A. B. , Prakash K., Face Recognition Using Principal Component Analysis and Linear Discriminant Analysis on Holistic Approach in Facial Images Database, IOSR Journal of Engineering, Vol. 2, No. 12, p. 15-23, 2012
69. Kramber W. J. ş.a., Principal component analysis of aerial video imagery, International Journal of Remote Sensing, Vol. 9, No. 9, p. 1415-1422, 1988
70. Chi Z., Yan H., Pham T., Fuzzy Algorithms: With Applications to Image Processing and Pattern Recognition, World Scientific, 225 p., 1996
71. Teodorescu H.N., Kandel A., Hall L.O., Report of research activities in fuzzy AI and medicine at USF CSE, Journal Artificial Intelligence in Medicine archive, Vol. 21, p. 177-183, 2001
72. Dubois D., Prade H., Testmale C., Weighted fuzzy pattern matching, Fuzzy Sets and Systems, Vol 28, No.3, p. 313 – 331, 1988
73. Yambal M., Gupta H., Image Segmentation using Fuzzy C Means Clustering: A survey, International Journal of Advanced Research in Computer and Communication Engineering, Vol. 2, No. 7, p.2927-2929, 2013
74. Angelov, P., A fuzzy controller with evolving structure, Information Sciences, Vol. 161, p.21–35, 2004
75. Angelov P.P., Filev D.P., Kasabov N.K., Evolving intelligent systems: methodology and applications, IEEE Press Series in Computational Intelligence, John Wiley and Sons and IEEE Press, New York, 444 p., 2010
76. Lughofer E., Angelov P.P., Detecting and Reacting on Drifts and Shifts in On-Line Data Streams with Evolving Fuzzy Systems, IFSA-EUSFLAT, p. 931-937, 2009.

77. Sun C.T., Dynamic Compensatory Pattern Matching in a Fuzzy Rule-Based Control System, American Control Conference, Publisher: IEEE, p. 502 - 505 1991
78. Hartert L., Sayed Mouchaweh M., Billaudel P., Dynamic Fuzzy Pattern Matching for the Monitoring of Non Stationary Processes Fault Detection, Supervision and Safety of Technical Processes, p.516-521, 2009
79. Hartert L., Sayed-Mouchaweh M. , Semisupervised Dynamic Fuzzy K-Nearest Neighbors Learning in Non-Stationary Environments; Methods and Applications, Chapter Learning in Non-Stationary Environments, p 103-124, 2012
80. Hartert L., Mouchaweh M.S., Billaudel P., Dynamic K-Nearest Neighbors for the monitoring of evolving systems, Fuzzy Systems (FUZZ), IEEE International Conference on Pages: 1 - 7, 2010
81. Kumar Jena P., Chattopadhyay S., "Comparative Study of Fuzzy k-Nearest Neighbor and Fuzzy C-means Algorithms," International Journal of Computer Applications, Vol. 57, No. 7, p. 22-32, 2012
82. Broggi A. ş.a., Shape Based Pedestrian Detection, IEEE Inteligent Vehicle Symposium, p.215-220, Dearborn, 2000
83. Mehta S., Dynamic Classification of Defect Structures in Molecular Dynamics Simulation Data, Proceedings of the SIAM International Conference on Data Mining, p.161-172, 2005
84. Chen Ke, On the Dynamic Pattern Analysis, Discovery and Recognition, IEEE Systems, Man & Cybernetics Society E-Newsletter, No. 12, 2005, 10 p.
85. Wu J. ş.a., Mixed Pattern Matching-Based Traffic Abnormal Behavior Recognition, Hindawi Publishing Corporation, The Scientific World Journal, 12 p., 2014
86. Kellokumpu V., Zhao G., Pietikäinen M., Human activity recognition using a dynamic texture based method, British Machine Vision Conference, 10 p., 2008
87. Solmaz B., Moore BE, Shah M., Identifying behaviors in crowd scenes using stability analysis for dynamical systems, IEEE Trans Pattern Anal Mach Intell., Vol.34, No.10, p.64-70, 2012
88. Campedel M., Moulines E., Méthodologie de sélection de caractéristiques pour la classification d'images satellitaires, Conference: Actes de CAP 05, Conférence francophone sur l'apprentissage automatique, 16 p., 2005
89. Forestier G., Wemmert G., Gançarski P., Multisource images analysis using collaborative clustering, EURASIP Journal on Advances in Signal Processing, Vol. 11, p. 374–384, 2008

90. Gançarski P., Wemmert G., Collaborative multi-strategy classification: application to per-pixel analysis of images, Proceedings of the 6th international workshop on Multimedia data mining: mining integrated media and complex data, Vol. 6, p. 595– 608, 2005
91. Wemmert C., Gançarski P., A multi-view voting method to combine unsupervised classifications, Proceedings of the 2nd IASTED International Conference on Artificial Intelligence and Applications, Vol. 2, p. 447–453, 2002
92. Masson M., Denœux T., Clustering interval-valued proximity data using belief functions, Pattern Recognition, Vol. 25, No.2, p. 163–171, 2004
93. Masson M., Denœux T., Ensemble clustering in the belief functions framework, International Journal of Approximate Reasoning, Vol. 52, No. 1, p. 92–109, 2011
94. Xu A., Krzyzak L., Suen C. Y. , Methods of combining multiple classifiers and their applications to handwriting recognition, IEEE Transactions on Systems, Man, and Cybernetics, Vol. 22, No. 3, p. 418–435, 1992
95. Chen Ke, Wang L., Chi H., Methods of Combining Multiple Classifiers with Different Features and Their Applications to Text-Independent Speaker Identification, International Journal of Pattern Recognition and Artificial Intelligence, Vol. 11, No. 3, p. 417-445, 1997
96. Guijarro M., Pajares G., On combining classifiers through a fuzzy multicriteria decision making approach: Applied to natural textured images, Expert Systems with Application, Vol. 39, p. 7262–7269, 2009
97. Urszula M. K., Switek T., Combined unsupervised-supervised classification method, Proceedings of the 13th International Conference on Knowledge Based and Intelligent Information and Engineering Systems: Part II, Vol. 13, p. 861–868, 2009
98. Karem F. , Dhibi M. , Martin A., Combination of Supervised and Unsupervised Classification Using the Theory of Belief Functions, Proc. of the 2nd International Conference on Belief Functions, Vol. 164, p. 85-92, 2012
99. Prudent Y., Ennaji A., Clustering incrémental pour un apprentissage distribué : vers un système évolutif et robuste, Conférence CAP, Vol. 3287, p. 446–453, 2004
100. Goswami D., Kalkan S., Krüger N., Bayesian classification of image structures, Lecture Notes in Computer Science, Image Analysis, Vol. 5575, p. 676-685, 2009
101. K. J. Bharathi, Chand P. P., Mohan Rao S.K., Classification of sports images using naive Bayesian classifier, International Journal of Computer Science, Systems Engineering and Information Technology, Vol. 4, No. 2, p. 151-155, 2011

102. Storvik G., Fjørtoft R., Schistad Solberg A. H., A Bayesian approach to classification of multiresolution remote sensing data, IEEE Transactions on Geoscience and Remote Sensing, Vol. 43, No. 3, p. 539- 547, 2005
103. Mahmud A. R., TahirN. M., Naïve Bayesian classifier for human shape recognition, IEEE 9th International Colloquium on Signal Processing and its Applications (CSPA), p. 219 – 223, 2013
104. Bustamante C., Garrido L., Soto R., Comparing Fuzzy Naive Bayes and Gaussian Naive Bayes for Decision Making in RoboCup 3D, 5th Mexican International Conference on Artificial Intelligence, p 237-247, 2006
105. McCallum A., Nigam K., A comparison of event models for Naive Bayes text classification, Proc. AAAI/ICML-98 Workshop on Learning for Text Categorization, p. 41-48, 1998
106. Vidhya K. A, Aghila G., A Survey of Naïve Bayes Machine Learning approach in Text Document Classification, International Journal of Computer Science and Information Security, Vol. 7, No. 2, 2010
107. Strategia naţională de gestionare a deşeurilor pentru perioada 2013-2027", aprobată prin HG nr. 248 din10.04.2013, disponibilă pe www.mediu.gov.md
108. Capel C., Waste Sorting - A Look At The Separation And Sorting Techniques In Today's European Market, disponibil pe http://www.waste-management-world.com/articles/print/volume-9/issue-4/features/waste-sorting-a-look-at-the-separation-and-sorting-techniques-in-todayrsquos-european-market.html
109. Rusu M., „Computerized visual inspection applied to identification and classification of labeled chemicals", International Conference 7th Edition Electronics, Computers and Artificial Intelligence, Bucharest – Romania, Vol. 7, No. 2, AF11 –AF16, 2015
110. Lowe D. G., "Distinctive Image Features from Scale-Invariant Keypoints", Computer Int. J. Computer Vision 60 (2), pp. 91-110, 2004
111. Lowe D., "Demo Software: SIFT Keypoint Detector", accesat 4.05.2015: http://www.cs.ubc.ca/~lowe/keypoints/
112. Guoshen Yu, Morel J.-M., "A fully affine invariant image comparison method", Proc. IEEE Int. Conf. Acoustics, Speech and Signal Processing (ICASSP), p. 1597–1600, 2009
113. Guoshen Yu, Morel, J.-M., "ASIFT: An Algorithm for Fully Affine Invariant Comparison", accesat 8.05.2015: http://www.ipol.im/pub/art/2011/my-asift/

114. Borne P., Benrejeb M., Haggège J., Les réseaux de neurones: présentation et applications, ed. Technip, Paris, 150 p., 2007
115. Haykin S., Neural Networks: A Comprehensive Foundation, Prentice Hall, 842 p., 1999
116. Souza F., Valle E., Chávez G., Aráujo A. Hue histograms to spatiotemporal local features for action recognition. Computer Vision and Pattern Recognition, Vol. 1, 4 p., 2011
117. Navon E., Miller O., Averbuch A. Color image segmentation based on adaptive local thresholds. Image and Vision Computing, Vol. 23, p. 69-85, 2005
118. Suryanto, Kim D.H., Kim H.K., Ko S.J. Spatial color histogram based center voting method for subsequent object tracking and segmentation, Image and Vision Computing, Vol. 29, p. 850-860, 2011
119. Bong C.W., Rajeswari M. Multi-objective nature-inspired clustering and classification techniques for image segmentation. Applied Soft Computing, Vol.11, 2011, p. 3271-3282
120. Chien B.C., Cheng M.C., A color image segmentation approach based on fuzzy similarity measure. Proceedings of the 2002 IEEE International Conference on Fuzzy Systems, Vol. 1, p. 449- 454, 2002
121. Sowmya B., Rani B.S., Colour image segmentation using fuzzy clustering techniques and competitive neural network. Applied Soft Computing, Vol. 11, p. 3170-3178, 2011
122. Murugesan K. M., Palaniswami S. Efficient colour image segmentation using multi-elitist-exponential particle swarm optimization. Journal of Theoretical and Applied Information Technology, Vol. 18, No. 1, p. 35-41, 2010
123. Deng Y., Manjunath B. S., Shin H. Color image segmentation. IEEE Computer Society Conference on Computer Vision and Pattern Recognition, Vol. 2, p. 446-451, 1999
124. Shih F.Y., Cheng S. Automatic seeded region growing for color image segmentation. Image and Vision Computing, Vol. 23, p. 877-886, 2005
125. Liu L., Sclaroff S. Region segmentation via deformable model-guided split and merge. Proc. Eighth IEEE International Conference on Computer Vision, Vol.1, p. 98- 104, 2001
126. Ma W.Y., Manjunath B. S. EdgeFlow: A technique for boundary detection and image segmentation. IEEE Transactions on Image Processing, Vol. 9, No. 8, p. 1375- 1388, 2000
127. Rao S. R. and al. Natural image segmentation with adaptive texture and boundary encoding. Proc. of the 9th Asian conference on Computer Vision, Vol. 1, p. 135-146, 2009
128. Angulo J., Serra J. Modelling and segmentation of colour images in polar representations. Image and Vision Computing, Vol. 25, p. 475-495, 2007

129. Nikolaev D. P., Nikolayev P.P. Linear color segmentation and its implementation. Computer Vision and Image Understanding, Vol. 94, p. 115-139, 2004
130. Haris K. et al. Hybrid image segmentation using watersheds and fast region merging. IEEE Transactions on Image Processing, Vol. 7, No. 12, p.1684-1699, 1998
131. Kim J. S., Hong K.S. Color–texture segmentation using unsupervised graph cuts. Pattern Recognition, Vol. 42, p. 735- 750, 2009
132. Verikas A., Malmqvist K., Bergman L. Colour image segmentation by modular neural network. Pattern Recognition Letters, Vol. 18, No. 2, p. 173-185, 1997
133. Imaginea test butterfly. Disponibilă: http://www.eecs.berkeley.edu/Research/Projects/CS/vision/bsds/BSDS300/html/dataset/images/gray/35010.html (vizitat 5.03.2013)
134. Imaginea test blood cells. Disponibilă: http://www.pudn.com/downloads58/sourcecode/math/detail206012.html (vizitat 6.03.2012).
135. Papamarkos N. and Gatos B. A new approach for multithreshold selection. Computer Vision, Graphics and Image Processing-Graphical Models and Image Processing, Vol. 56, No. 5, p. 357-370, 1994
136. Imaginea test #260058 (Pyramids). Disponibilă: http://www.eecs.berkeley.edu/Research/Projects/CS/vision/grouping/segbench/BSDS300/html/dataset/images/gray/260058.html (vizitat 5.03.2013)
137. Reynolds D., Rose C. Robust text-independent speaker identification using Gaussian mixture speaker models. IEEE Transactions on Speach and Audio Processing, Vol. 3, No. 1, p.72-83, 1995
138. Huang Z. K., Chau K. W. A new image thresholding method based on Gaussian mixture model. Applied Mathematics and Computation, Vol. 205, p. 899–907, 2008
139. Tang H. et al. A vectorial image soft segmentation method based on neighborhood weighted Gaussian mixture model. Computerized Medical Imaging and Graphics, Vol. 33, p. 644–650, 2009
140. Liu J., Zhang H. Image segmentation using a local GMM in a variational framework. J. Mathematical Imaging and Vision, Vol. 132, No. 46, p. 1992-2013, 2012
141. Reynolds, D.A., Quatieri, T.F., Dunn, R.B., Speaker Verification Using Adapted Gaussian Mixture Models, Digital Signal Processing, Vol. 10(1), pp. 19–41, 2000
142. Imaginea test peppers. Disponibilă:

http://introcs.cs.princeton.edu/java/31datatype/peppers.jpg (vizitat 6.03.2012)

143. Halder A., Pathak N. An evolutionary dynamic clustering based colour image segmentation. International Journal of Image Processing (IJIP), Vol. 4, No. 6, p. 549-556, 2011
144. Canny J. A computational approach to Edge Detection. Pattern Analysis and Machine, Vol. 8, No. 6, p. 679 – 698, 1986
145. Duda R. O., Hart P. E. Use of the Hough transformation to detect lines and curves in pictures. Communications of the ACM, Vol. 15 No. 1, p. 11-15, 1972
146. Rusu M., Teodorescu H.-N. A Method for Image Segmentation based on Histograms – Preliminary Results, 4th International Conference Telecommunications, Electronics and Informatics, p. 351-354, 2012
147. Teodorescu H.-N. Rusu M. Yet Another Method for Image Segmentation based on Histograms and Heuristics. Computer Science Journal of Moldova, vol.20, no.2 (59), 2012, p. 163-177. http://www.math.md/publications/csjm/issues/v20-n2/11087/
148. Berry P.S.M, Bercovitch F.B. Darkening Coat Colour Reveals Life History and Life Expectancy of Male Thornicrofts Giraffes. Journal of Zoology, Vol. 287, No. 3, p.157–160, 2012
149. Teodorescu H.-N., Rusu M. Image Segmentation Based on G-UN-MMs and Heuristics. Theoretical Background and Results. Proceedings of the Romanian Academy, Series A, Vol. 14, No. 1, p. 78–85, 2013, http://www.acad.ro/sectii2002/proceedings/doc2013-1/12-Teodorescu.pdf
150. Teodorescu H.-N., Rusu M. Improved Heterogeneous Gaussian and Uniform Mixed Models (G-U-MM) and Their Use in Image Segmentation, Romanian Journal of Information Science and Technology (ROMJIST), Vol.16, No.1, p.29-51, 2013, http://www.imt.ro/romjist/Volum16/Number16_1/pdf/03-MRusu.pdf
151. Imaginea „Lena”. Disponibilă pe adresa: http://www.ece.rice.edu/wakin/ images/ lena512.bmp
152. D. Duvenaud, H. Nickisch, C. E. Rasmussen, Additive Gaussian processes, 2011, 9 pagini. Disponibil pe adresa: http://arxiv.org/pdf/1112.4394v1.pdf
153. G. Blanchard, M. Kawanabe, et al., In Search of non-Gaussian components of a highdimensional distribution, J. Machine Learning Research, Vol. 7, p. 247-282, 2006
154. Imaginea „rabbit”. Disponibilă pe adresa: http://www.engineering.uiowa.edu/ dip/examples/images/rabbit.jpg

155. Rusu M., Teodorescu H-N., Quality Analysis of Image Segmentation based on G-UN-MMs, 2nd International Conference on Nanotechnologies and Biomedical Engineering, Chisinau, Republic of Moldova, p. 620-624, April 18-20, 2013 http://www.icnbme.sibm.md/bio_imaging.html
156. H. Zhang et al., Image segmentation evaluation: A survey of unsupervised methods, Computer Vision and Image Understanding, Vol. 110, No.2, p. 260-280, 2008
157. K. Udupa et al., A framework for evaluating image segmentation algorithms, Computerized Medical Imaging and Graphics, Vol. 30, p.75-87, 2006
158. Sezgin, M., Sankur, B., Survey over image thresholding techniques and quantitative performance evaluation, Journal of Electronic Imaging, Vol. 13, No. 1, p. 146–165, 2004
159. Ilea, D.E., Whelan P.F., Image segmentation based on the integration of colour – texture descriptors – A review, Pattern Recognition, Vol. 44, p. 2479–2501, 2011
160. Sharma, M., Singh, S., Evaluation of Texture Methods for Image Analysis, The Seventh Australian and New Zealand Intelligent Information Systems Conference, p.117-121, 2001
161. Rusu M., Thresholding methods and quantitative evaluation of results, Meridian Ingineresc, No. 2, p. 18-25, 2013 http://www.utm.md/meridian/2013/MI_nr.2_2013_integral.pdf
162. Baza de Date de imagini USC-SIPI. http://sipi.usc.edu/database/database.php
163. Garcia D., Otsu Image segmentation using Otsu thresholding, 2010, Programul în Matlab este disponibil pe https://www.mathworks.com/matlabcentral/fileexchange/26532-image-segmentation-using-otsu-thresholding
164. Papamarkos N., Gatos B. A new approach for multithreshold selection" Computer Vision, Graphics, and Image Processing-Graphical Models and Image , Processing, Vol. 56, No. 5, pp. 357-370, 1994. Programul în Matlab este disponibil pe https://www.mathworks.com/ matlabcentral/fileexchange/27871-image-multithresholding
165. Rusu, M., Teodorescu, H.N., Improved Heterogeneous Gaussian and Uniform Mixed Models (G-U-MM) and their use in Image Segmentation. Disponibil pe adresa: http://francophonie.utm.md/rusu_mariana/
166. Moraru V., Rusu M., „Algorithm for linear pattern separation”, Meridian Ingineresc, No. 2, pp. 26-29, 2013 http://www.utm.md/meridian/2013/MI_nr.2_2013_integral.pdf
167. Freund, R. M., Pattern Classification, and Quadratic Problems, Massachusetts Institute of Technology, 2004
168. Vapnik, V., The nature of statistical learning theory, Springer-Verlag, New York, 1995

169. Moraru, V., A Smooth Newton Method for Nonlinear Programming Problems with Inequality Constraint, Computer Science Journal of Moldova, Vol. 19, Nr. 3 (57), pp. 333-353, 2011
170. Ypma, T. J. Historical development of the Newton-Raphson method, SIAM Review, Vol. 37, No. 4, pp. 531-551, 1995
171. Byun, H., Lee, S. W., Applications of Support Vector Machines for Pattern Recognition: A Survey, Pattern Recognition with Support Vector Machines, Vol. 2388, pp. 213-236, 2002
172. Bani M.S., Rashid Z.A., Hamid K.H.K., Harbawi M.E., Alias A.B., Aris M.J., "The Development of Decision Support System for Waste Management; a Review", World Academy of Science, Engineering and Technology, International Journal of Chemical, Molecular, Nuclear, Materials and Metallurgical Engineering, Vol. 3, No.1, pp.17-24, 2009
173. Cheng S., Chan C.W., Huang G.H., An integrated multi-criteria decision analysis and inexact mixed integer linear programming approach for solid waste management, Engineering Application of Artificial Intelligence, Vol. 16, pp. 543-554, 2003
174. Musee N., Lorenzen L., Aldrich C., New methodology for hazardous waste classification using fuzzy set theory, Journal of Hazardous Materials, Vol. 154 (1), pp. 1040-1051, 2008
175. Stirling H., The recovery of waste glass cullet for recycling purposes by means of electro-optical sorters, Conservation & Recycling, Vol. 1(2), pp. 209-219, 1977
176. Rahman M.O., Hussain A., Scavino E., Basri H., Hannan M. A., Intelligent computer vision system for segregating recyclable waste papers, Expert Systems with Applications, Vol. 38 (8), pp. 10398–10407, 2011
177. Lewis J. P., "Fast Normalized Cross-Correlation", Vision interface, Vol.10, No.1, p. 120-123, 1995
178. Rusu M., Zbancioc M.-D., Automated identification of objects based on Normalized Cross-Correlation and Genetic Algorithm, IEEE Int. Conf. on E-Health and Bioengineering, 4 p., 2015
179. Poli R. and Cagoni S., "Genetic programming with user-driven selection: Experiments on the evolution of algorithms for image enhancement", Genetic Programming, p.269-277, 1997
180. Paulinas M., Ušinskas A., A survey of genetic algorithms applications for image enhancement and segmentation, Information Technology and Control, Vol.36, No.3, p. 278-284, 2007

181. Suganthan, P.N., Structural pattern recognition using genetic algorithms, Pattern Recognition, Vol. 35, No.9, p. 1883–1893, 2002
182. Tan X., Bhanu B., Fingerprint matching by genetic algorithms, Pattern Recognition, Vol. 39, p.465 – 477, 2006
183. Kobayashi T. and Machida N., Identifying Hand Gesture Images by Using Genetic Algorithms, Conference on Machine Vision Applications, Japan, p. 240-243, 2007
184. M. Rusu, „Computerized visual inspection applied to identification and classification of labeled chemicals", International Conference 7th Edition Electronics, Computers and Artificial Intelligence, Bucharest–Romania, vol. 7, no. 2, AF11 –AF16, 2015
185. Border tracing, sections from Chapter 5 according to the www syllabus of University of Iowa http://user.engineering.uiowa.edu/~dip/LECTURE/Segmentation2.html#tracing
186. Yang M., Kpalma K., Ronsin J., Shape-Based Invariant Feature Extraction for Object Recognition, Chapter Advances in Reasoning-Based Image Processing Intelligent Systems, Vol. 29 of the series Intelligent Systems Reference Library, pp. 255-314, 2012
187. Zhang D., Lu G., Review of shape representation and description techniques, Pattern Recognition, Vol.37 (1), pp. 1–19, 2004
188. Scradeanu D., Modele cantitative statistice, 104 p, available on http://www.unibuc.ro/prof/scradeanu_d/docs/2013/noi/08_21_27_123_Modele_CANTITATIVE_STATISTICE_DSCRD.pdf

ANEXE

Anexa 1: Codul sursă al programului de segmentare bazat pe G-U-MM

```
//-------------------------------------------------------
// Nume fisier: G-U-MM SEG
//-------------------------------------------------------
// autori: drd. Mariana RUSU,
// Prof. dr. ing. HN.Teodorescu, m.c.A.R.
//-------------------------------------------------------
// Versiunea 7 din 2 aprilie 2013
//-------------------------------------------------------
#include <stdio.h>
#include <stdlib.h>
#include <string.h>
#include <ctype.h>
#include <conio.h>
#include <math.h>
#define dim1 170
#define dim2 170

//reprezentarea histogramei
void calcHist(int matr[dim1][dim2],long H[256]){
        int i,j;
        for(i=0;i<256;i++) H[i]=0;

        for(i=0; i<dim1; i++)
        for(j=0; j<dim2; j++) H[matr[i][j]]++;
}

 //medierea histogramei cu fereastra de 20
  void filtru_f20(long H[256]) {
        long HTemp[256];
        long sum;
        int i,j;
        for(i=0;i<256;i++)  {
                sum=0;
                if(i+20<256) for(j=i;j<i+20;j++) sum+=H[j];
                else for(j=i-20;j<256;j++) sum+=H[j];
                HTemp[i]=int(sum/20);
        }
        for(i=0;i<256;i++) H[i]=HTemp[i];
 }

 //medierea histogramei cu fereastra de 11, pixel centrat pe pozitia 6 (5+1+5)
void filtru_f11(long H[256]){
        long HTemp[256];
        long sum;
        int i,j;
        for(i=5;i<256-5;i++)  {
                sum=0;
                for(j=i-5;j<i+5;j++) sum+=H[j];
                HTemp[i]=int(sum/11);
        }
        for(i=5;i<256-5;i++) H[i]=HTemp[i];
}

//filtru median  http://www.librow.com/articles/article-1
void filtru_median(long H[256]){
```

```
        long HTemp[256];
        int i,j;
        int temp,*result;
        //for(i=0;i<2;i++) H[i]=H[i];
        for (i = 2; i < 256 - 2; ++i) {
                //   Pick up window elements
                // window HTemp[5];
                for (j = 0; j < 5; ++j) HTemp[j] = H[i - 2 + j];
                //   Order elements (only half of them)
                for (int j = 0; j < 3; ++j)  {
                        //   Find position of minimum element
                        int min = j;
                        for (int k = j + 1; k < 5; ++k)
                                if (HTemp[k] < HTemp[min])  min = k;
                        //   Put found minimum element in its place
                        //const element temp = window[j];
                        temp=HTemp[j];
                        HTemp[j] = HTemp[min];
                        HTemp[min] = temp;
                }
                //   Get result - the middle element
                //result[i - 2] = HTemp[2];
                H[i] = HTemp[2];
  }
}

//scrierea histogramei in fisier
void HistInFile(long H[256],char *file ){
        FILE *fl;
        char nbr[5];
        fl=fopen(file,"w");
        if((fl=fopen(file,"w"))==NULL) puts("Fisierul nu poate fi deschis");

        for(int i=0;i<256;i++) {
                ltoa(H[i],nbr,10);
                fputs(strcat(nbr,"\n"),fl);
        }
        fclose(fl);
}

// afisarea histogramiei
void affisareHist(long H[256]){
        char ch;
        for(int i=0;i<256;i++) {
                printf("%3d:  %li\n",i, H[i]);
                if((i+1)%25==0) {
                        ch=getch();
                        if(ch=='c') break;
                }
        }
}

//determinarea intervalelor semi-constante
void pragHNT(long H[256],int &pragm1, int &pragm2, long Prags[256]){
        int i, k1,  min, max1, max2, prag1, prag2, prag01, prag02, pprag=0, ppragm=0, cont=0, fl=0, sct=24 ;
        double p1=0.2, p2=1.5;
        long HMax1,HMax2, VMin, VMax, VMinL, VMaxL;
        float res;
        char ch;
```

```
	HMax1=H[0];HMax2=H[0]; max1=0; max2=-1; k1=0; Prags[0]=0;
	for(i=1; i<256; i++) {
		if(HMax1<H[i]) { HMax1=H[i]; max1=i;}
		Prags[i]=0;
	}
	for(i=1; i<256; i++)
		if(HMax2<H[i] && H[i]!=HMax1 && H[i]>=ceil(p1*HMax1)) { HMax2=H[i]; max2=i;}
	//printf("\n\n Max1=%li with gray level %d;\n\n Max2=%li with gray level %d;\n\n " , H[max1],max1,H[max2],max2);
	VMax=H[0]; VMin=H[0];
	i=0; min=0; prag01=0; prag02=0;
	do{
		if(i==(sct*(k1+1)-6*k1) || i==255){
			res=(float)(VMax-VMin)/cont;
			prag1=i-cont; prag2=i-1; //determining the limits of current interval
			if(res<=p2){
				printf("\n Interval %d; prag1=%d, prag2=%d,  VMax= %li, VMin=%li, res=%f
\n",k1,prag1,prag2,VMax,VMin,res);
				if(!fl) prag01=prag1; //initializing the left limit of the quasi-constant interval
				prag02=prag2;	//initializing the right limit of the quasi-constant interval
				pprag++; //counting the number of the sub-intervals in the quasi-constant interval
				fl=1;
			}else fl=0;

			if((!fl && (prag02==prag1+5)) || (fl && i==255)){ //represent the all quasi-constant interval
				if(pprag>=ppragm) {pragm1=prag01; pragm2=prag02; ppragm=pprag;}
				printf("\n Continuous quasi-constant interval %d with the limits (thresholds):  %d si %d
\n",pprag, prag01,prag02);
				Prags[prag01]=1; Prags[prag02]=1;
			}

			if(!fl) {
				pprag=0;
				printf("\n Interval %d;  VMax= %li; VMin=%li , res=%f \n",k1,VMax,VMin,res);
			}

			if(i==255) break;
			i-=6; cont=0;

			k1++;
			VMax=H[i]; VMin=H[i];
			ch=getch();
		}else{
			if(VMin>H[i]) VMin=H[i];		//determining the minimum on the interval
			if(VMax<H[i]) VMax=H[i];	//determining the maximum on the interval
			i++;
			cont++;
		}
	}while(i<256);

	if(ppragm>0) printf("\n Quasi-constant interval with maxim %d sub-intervals with the limits:  %d si %d \n",ppragm,
pragm1,pragm2);
	else printf("\n Not detected any quasi-constant interval \n");
}
```

//determinarea intervalelor cu distributie gaussiana

```
void gauss(long H[256], int &pragm1, int &pragm2, long Prags[256], long Hout[256]){
	int  center, i, j, prag01, prag1, prag2, b, b_optim, max1, cc, fl;
	long HMax1,A;
```

```
	float error_q, error_min, error_un;


	cc=0; prag01=0;	prag1=0;  prag2=0; fl=0;
	for(j=0; j<256; j++) Hout[j]=H[j];
	for(j=1; j<256; j++) {
		//if(Prags[j]){				//pentru calcul doar pe intervalele determinate
		if(Prags[j] || j==255){		//pentru calcul pe toate intervalele
			//if(cc==0) prag1=j; //pentru calcul doar pe intervalele determinate
			//else prag2=j;				//pentru calcul doar pe intervalele determinate
			center=(prag2+j)/2;
			if(fl && H[center]<H[prag2] && H[center]>H[j]) {
				Prags[prag1]=0; Prags[prag2]=0;
				prag1=prag01;
				prag2=j;
				//printf("\n\n MERGE: %d - %d ; center is %d with value
%d",prag1,prag2,center,H[center]);
			}else{
				prag01=prag1;
				prag1=prag2;				//pentru calcul pe toate intervalele
				prag2=j;				//pentru calcul pe toate intervalele
			}
			cc++;
			fl=0;
		}
		//if(cc==2){				//pentru calcul doar pe intervalele determinate
		if(cc>0){				//pentru calcul pe toate intervalele
			center=(prag1+prag2)/2; //determine the center of interval
			printf("\n\n Intrval de calcul: %d - %d ; center is %d with value %d",prag1,prag2,center,H[center]);
			if (H[center]>H[prag1]&& H[center]>H[prag2]){
				HMax1=H[prag1];
				//determine the maximum of interval
				for(i=prag1; i<=prag2; i++) if(HMax1<H[i]) { HMax1=H[i]; max1=i;}

				A=H[max1];
				printf("\n maxim: %d in %d", A, max1);
				for(b=3;b<100;b++){
					error_q=0;
					for(i=prag1; i<=max1; i++) {
						error_q+=pow(H[i]-A*exp(-pow((i-max1),2)/b),2);
					}

					if(b==3) {error_min=error_q; b_optim=b;}
					else{
						if(error_min>error_q) {error_min=error_q; b_optim=b;}
					}
				}
				printf("\n\nb_optim=%i ; error_min=%f", b_optim,error_min);
				for(i=prag1; i<=prag2; i++) Hout[i]=A*exp(-pow((i-max1),2)/b_optim);

			}
			else {
				printf("\n\nThe interval represents UN\n\n");
				for(i=prag1; i<=prag2; i++) Hout[i]=H[center];
			}
			cc=0; getch();
		}
	}
```

```
}

void gaussAll(long H[256], long Prags[256], long Pgauss[256]){
	int  center, i, j, k, prag01, prag02, prag1, prag2, b, b_optim, max1, cc, gauss=0;
	long HMax1,A;
	float error_q, error_min, med_l, med_c, med_r;
	printf("este");
	cc=0; prag01=0;	prag1=0;  prag2=0;
	for(j=0; j<256; j++) {Pgauss[j]=0;}
	for(j=1; j<256; j++) {
		//if(Prags[j]){				//pentru calcul doar pe intervalele determinate
		if(Prags[j]>0 || j==255){	//pentru calcul pe toate intervalele

			prag01=prag1;
			prag1=prag2;				//pentru calcul pe toate intervalele
			prag2=j;
			cc++;
		}

		if(cc>0){					//pentru calcul spe toate intervalele
			center=(prag1+prag2)/2; //determine the center of interval
			printf("\n\n Intrval de calcul: %d - %d ; center is %d with value %d",prag1,prag2,center,H[center]);
			if (H[center]>H[prag1]&& H[center]>H[prag2]){
				HMax1=H[prag1];
			 //determine the maximum of interval
				for(i=prag1; i<=prag2; i++) if(HMax1<H[i]) { HMax1=H[i]; max1=i;}

				A=H[max1];

				printf("\n maxim: %d in %d", A, max1);
				for(b=3;b<100;b++){
					error_q=0;
					for(i=prag1; i<=max1; i++) error_q+=pow(H[i]-A*exp(-pow((i-max1),2)/b),2);



					if(b==3) {error_min=error_q; b_optim=b;}
					else{
						if(error_min>error_q) {error_min=error_q; b_optim=b;}
					}

				}
				printf("\n\nb_optim=%i ; error_min=%f", b_optim,error_min);
				Pgauss[prag1]=++gauss; Pgauss[prag2]=gauss;
			}else {
				printf("\n\nThe interval represents UN\n\n");
				for(k=prag1; k<prag2; k+=18){
					prag02=0;
					if(k+24<prag2) prag02=k+24;
					else if(k+18<=prag2) prag02=prag2;

					if(prag02>0){
						center=(k+prag02)/2;med_l=0;med_c=0;med_r=0;
						for(i=0;i<6;i++){
							med_l+=H[k+i]/6;
							med_r+=H[prag02-i]/6;
							med_c+=H[center-3+i]/6;
						}
```

```
                                                        if(med_c>med_l && med_c>med_r) {Pgauss[k]=++gauss;
Pgauss[prag02]=gauss;}
                                                }
                                        }
                                }
                                cc=0; getch();
                        }
                }
                for(i=0; i<=255;i++) if(Pgauss[i]>0)printf("\n Intrval gauss: %d in %d ;",Pgauss[i],i);
}

 void itervaleGauss(long H[256], long Pragsf256]){
                int  center, i, j, k,  prag02, prag1, prag2;
                float med_l, med_c, med_r;

                prag1=0;  prag2=0;
                for(j=1; j<256; j++) {
                        if(Pragsf[j]>0 || j==255){          //pentru calcul pe toate intervalele

                                prag1=prag2;                                //pentru calcul pe toate intervalele
                                prag2=j;

                                for(k=prag1; k<prag2; k+=18){
                                        prag02=0;
                                        if(k+23<prag2) prag02=k+23;
                                        else if(k+17<=prag2) prag02=prag2;
                                        if(prag02>0){
                                                center=(k+prag02)/2;med_l=0;med_c=0;med_r=0;
                                                for(i=0;i<6;i++){
                                                        med_l+=H[k+i]/6;
                                                        med_r+=H[prag02-i]/6;
                                                        med_c+=H[center-3+i]/6;
                                                }
                                                if(med_c>med_l && med_c>med_r) {Pragsf[k]=2; Pragsf[prag02]=2;}
                                                printf("\n\n Intrval de calcul: %d - %d ; medii: %.2f, %.2f, %.2f",k,prag02,med_l,
med_r, med_c);
                                        }
                                }
                        }
                }
}
```

//calcularea erorii aproximării pentru distributia Gaussiana

```
 void gaussCalc1(long H[256], long Pragsf256], float Errorsf20], int Bsf20]){
                int  i, j, prag01, prag1, prag2, b, b1, b2,b_optim,b_optim1,b_optim2, max1, cc, ec,b_un;
                long HMax1,A,center,error_un=0;
                float error_q, error_min, error_q1, error_min1,error_q2, error_min2;

                cc=0; prag01=0;     prag1=0;  prag2=0; ec=0;
                //for(j=0; j<256; j++) Houtf[j]=Hf[j];
                for(j=1; j<256; j++) {
                        if(Pragsf[j]>0 || j==255){          //pentru calcul pe toate intervalele

                                prag01=prag1;
                                prag1=prag2;                                //pentru calcul pe toate intervalele
                                prag2=j;
                                cc++;
                        }
```

```
		if(cc>0){                                                    //pentru calcul spe toate intervalele
			center=(prag1+prag2)/2; //determine the center of interval

			error_un=0;
			for(i=prag1; i<=prag2; i++) error_un+=pow((i-center),2);
			b_un=pow((prag2-prag1),2)/12;

			printf("\n\n Interval de calcul: %d - %d ; center is %d with value %d; limits values: %d ,
%d",prag1,prag2,center,H[center],H[prag1],H[prag2]);
			printf("\n for UN error_un= %li, b_un=%i", error_un, b_un);

			if (H[center]>H[prag1]&& H[center]>H[prag2]){
				HMax1=H[prag1];
			 //determine the maximum of interval
				for(i=prag1; i<=prag2; i++) if(HMax1<H[i]) { HMax1=H[i]; max1=i;}

				A=H[max1];
				printf("\n maxim: %d in %d", A, max1);

				for(b1=3;b1<7200;b1++){
					error_q1=0;
					for(i=prag1; i<=max1; i++)	error_q1+=pow(H[i]-A*exp(-pow((i-
max1),2)/b1),2);

					if(b1==3) {error_min1=error_q1; b_optim1=b1;}
					else{
						if(error_min1>error_q1) {error_min1=error_q1; b_optim1=b1;}
					}
				}

				printf("\n b_optim1=%i ; error_min1=%.2f", b_optim1,error_min1);

				for(b2=3;b2<7200;b2++){
					error_q2=0;
					for(i=max1; i<=prag2; i++) error_q2+=pow(H[i]-A*exp(-pow((i-max1),2)/b2),2);

					if(b2==3) {error_min2=error_q2; b_optim2=b2;}
					else{
						if(error_min2>error_q2) {error_min2=error_q2; b_optim2=b2;}
					}

				}
				printf("\n b_optim2=%i ; error_min2=%f", b_optim2,error_min2);

				b_optim=(b_optim1+b_optim2)/2;
				error_q1=0;
				for(i=prag1; i<=max1; i++) {
					error_q1+=pow(H[i]-A*exp(-pow((i-max1),2)/b_optim),2);
					error_q2=0;
					for(i=max1; i<=prag2; i++) error_q2+=pow(H[i]-A*exp(-pow((i-
max1),2)/b_optim),2);

					error_min=error_q1+error_q2;
				}
				printf("\n b_optim=%i ; error_min=%f", b_optim,error_min);

				Errors[ec]=error_min; Bs[ec++]=b_optim;
				Prags[prag1]=2; Prags[prag2]=2;
			}
```

```
                    else printf("\nThe interval represents UN\n\n");

                    cc=0;
                }
        }
}

//Extindem daca e posibil intervalele cu distributie gaussiana
 void transformGauss(long H[256], long Prags[256], float Errors[20], int Bopt[20]){
        int  i, j, k,  prag1, prag2, prag00, prag01, prag02, b_optim, max1, cc, ec;
        long HMax1,A;
        float e_q, e_q1,e_q2=0, error_min, error_q1, error_q2;

        cc=0; prag1=0;  prag2=0; ec=0;

        for(j=0; j<=255; j++) {
                if(Prags[j]==1) {cc=0; prag00=j;}
                else if((Prags[j]>1 || j==255)&& (j>prag2 ||j==0)){  //pentru calcul pe toate intervalele

                        prag1=prag2;
                //pentru calcul pe toate intervalele
                        prag2=j;
                        cc++;

                }

                if(!(cc<2 || prag1==prag2)){
                        HMax1=H[prag1];
                        prag01=prag1; prag02=prag2;

                         //determine the maximum of interval
                        for(i=prag1; i<=prag2; i++) if(HMax1<H[i]) { HMax1=H[i]; max1=i;}

                        b_optim=Bopt[ec++];
                        printf("\n\n prag1=%d, prag2=%d, b_optim=%i", prag1,prag2,b_optim);
                        for(k=prag1-1; k>=0; k--){
                                e_q1=H[k]-HMax1*exp(-pow((k-max1),2)/b_optim);

                                e_q=k-(prag00+prag01)/2;
                                printf("\n LEFT: e_q=%.2f, e_q1=%.2f, e_q2=%.2f", e_q, e_q1,e_q2);
                                if(fabs(e_q1)<fabs(e_q)) {Prags[prag1]=0; Prags[k]=2; prag1=k;}
                                else break;
                        }
                        error_q1=0;
                        for(i=prag1; i<=max1; i++) error_q1+=pow(H[i]-HMax1*exp(-pow((i-max1),2)/b_optim),2);

                        prag00=255;
                        for(k=prag2+1; k<=255; k++) if(Prags[k]) {prag00=k; break;}
                        for(k=prag2+1; k<=255; k++){
                                e_q1=H[k]-HMax1*exp(-pow((k-max1),2)/b_optim);

                                e_q=k-(prag00+prag02)/2;
                                printf("\n RIGHT: e_q=%.2f, e_q1=%.2f, e_q2=%.2f",  e_q, e_q1,e_q2);
                                if(fabs(e_q1)<fabs(e_q)) {Prags[prag2]=0; Prags[k]=2; prag2=k;}
                                else break;
                        }
                        error_q2=0;
                        for(i=max1; i<=prag2; i++)      error_q2+=pow(H[i]-HMax1*exp(-pow((i-max1),2)/b_optim),2);
```

```
                                error_min=error_q1+error_q2;
                                printf("\n b_optim=%i ; error_min=%f", b_optim,error_min);

                                cc=0; // trecerea intre gausiene: 0 - pentru gausiene separate de uniforme, 1 - pentru gausiene
conesecutive
                                printf("\n\n prag1=%d, prag2=%d, b_optim=%i", prag1,prag2,b_optim);
                        }
                }
}

void gaussCalc2(long H[256], long Prags[256], float Errors[20], int Bs[20]){
        int  center, i, j, prag01, prag1, prag2, b, max1, cc, ec;
        long HMax1,A;
        float error_q, error_min, b1,b2;

        cc=0; prag01=0;     prag1=0;  prag2=0; ec=0;

        for(j=1; j<256; j++) {
                if(Prags[j]>0 || j==255){          //pentru calcul pe toate intervalele

                        prag01=prag1;
                        prag1=prag2;                                    //pentru calcul pe toate intervalele
                        prag2=j;
                        cc++;
                }

                if(cc>0){                                               //pentru calcul spe toate intervalele
                        center=(prag1+prag2)/2; //determine the center of interval
                        printf("\n\n Interval de calcul: %d - %d ; center is %d with value %d; \n limits values: %d ,
%d",prag1,prag2,center,H[center],H[prag1],H[prag2]);
                        if (H[center]>H[prag1]&& H[center]>H[prag2]){
                                HMax1=H[prag1];
                        //determine the maximum of interval
                                for(i=prag1; i<=prag2; i++) if(HMax1<H[i]) { HMax1=H[i]; max1=i;}

                                A=HMax1;
                                printf("\n maxim: %d in %d", A, max1);
                                error_q=0;
                                b1=-pow((prag1-max1),2)/log((float)H[prag1]/A);  //printf("\n H[prag1]=%i, H[prag2]=%i\n",
H[prag1], H[prag2]);
                                b2=-pow((prag2-max1),2)/log((float)H[prag2]/A);  printf("\n log(%f) = %f",
(float)H[prag1]/A,log((float)H[prag1]/A));
                                printf("\n b1=%f, b2=%f\n",b1, b2);
                                b=(b1+b2)/2;

                                for(i=prag1; i<=max1; i++) error_q+=pow(H[i]-A*exp(-pow((i-max1),2)/b),2);

                                printf("\n b_optim=%i ; error_min=%f", b,error_q);
                                Errors[ec]=error_q; Bs[ec++]=b;

                        }
                        else printf("\nThe interval represents UN\n\n");

                        cc=0;
                }
        }
}
```

//segmentarea imaginii

```
void nivelare(long Prags[256],int matr[dim1][dim2], int matr_out[dim1][dim2]) {
        int i,j,k, pp=0, col=0;
        for(k=1; k<256;k++){
                if(Prags[k]||k==255){
                        col+=25;
                        for(i=0;i<dim1;i++)
                                for(j=0;j<dim2;j++)
                                        if(matr[i][j]>pp && matr[i][j]<=k)          matr_out[i][j]=col;


                        pp=k;
                }
        }
}
```

// eliminarea spatiilor la citirea imaginii-matrice din fisier

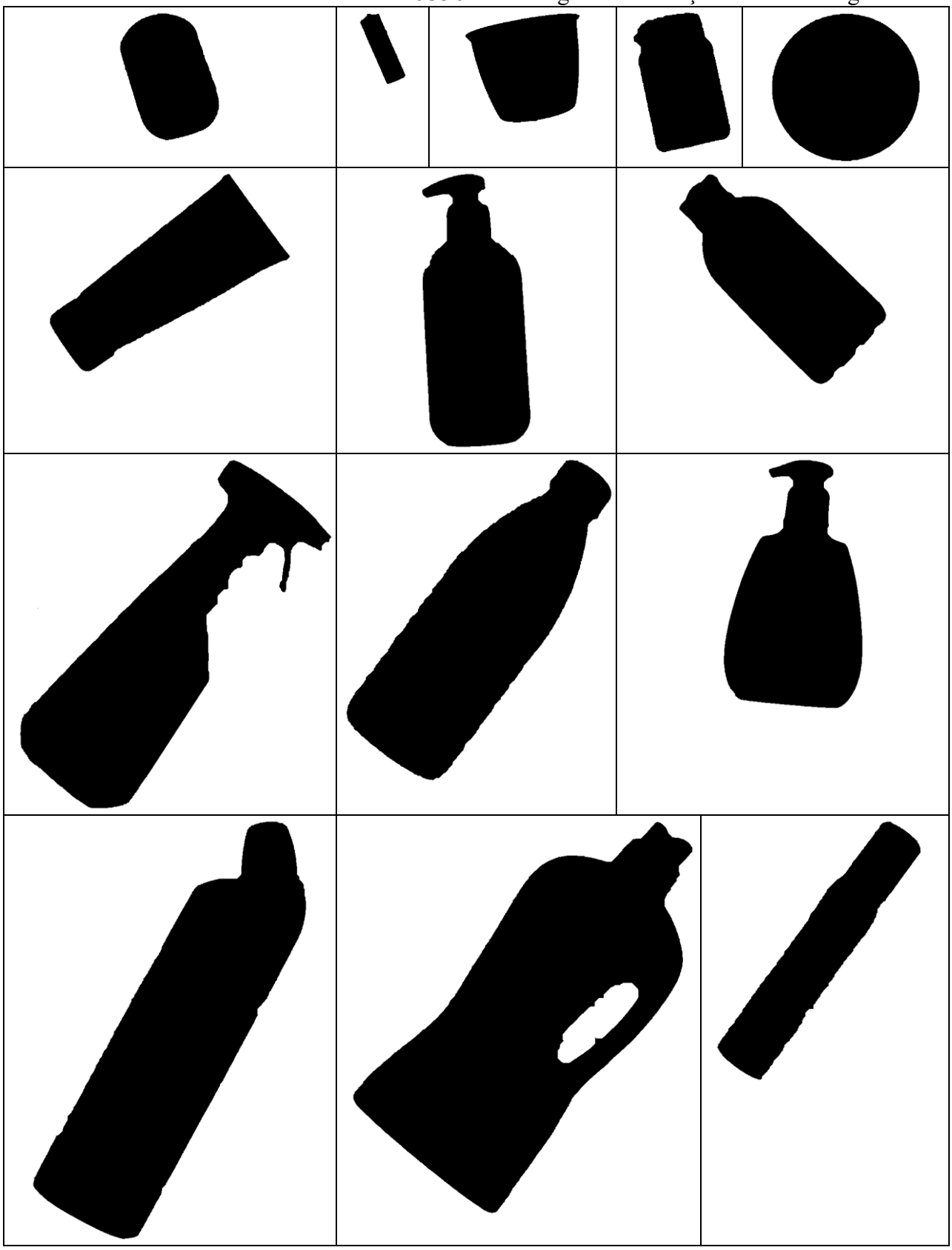
```
char *trim(char *str)
// http://www8.cs.umu.se/~isak/snippets/trim.c
{
        char *ibuf = str, *obuf = str;
        int i = 0, cnt = 0;

        // Trap NULL
        if (str)
        {               //  Remove leading spaces
                        for (ibuf = str; *ibuf && isspace(*ibuf); ++ibuf);
                        if (str != ibuf)  memmove(str, ibuf, ibuf - str);

                        // Collapse embedded spaces
                        while (*ibuf)
                        {
                                if (isspace(*ibuf) && cnt)      ibuf++;
                                else
                                {   if (!isspace(*ibuf))  cnt = 0;
                                                else
                                                {     *ibuf = ' ';
                                                                cnt = 1;
                                                }
                                                obuf[i++] = *ibuf++;
                                }
                        }
                        obuf[i] = 0;

                        // Remove trailing space
                        while (--i >= 0)
                        {
                                if (!isspace(obuf[i])) break;
                        }
                        obuf[++i] = 0;
        }
        return str;
}

void main(void){
        FILE *r;
        char name[6], ch;
        int matrice[dim1][dim2],matr_out[dim1][dim2],M1[dim1][dim2],M2[dim1][dim2],M3[dim1][dim2];
```

```
        long Hist[256], Prags[256], Hout[256];
        int c, i, j, pr, prag1, prag2,a;
        int cont, b_min[20], b_q[20];
        long Sum,Sum1;
        //long sum11,sum12, sum21,sum22, sum3, sum4, sum5, sum6, sum7;
        float media,s, errors_min[20], errors_q[20];
        //clrscr();

        r=fopen("D:/C-Free Standard/samples/work/blood_cells.txt","r"); //citirea imaginii matrice din fisier
        printf("Matricea initiala a imaginii: \n");
        c=0; i=0; j=0;
        while(fgets(name, 5, r)!=NULL) if(strlen(trim(name))>0){
                matrice[i][j]=atoi(name);
                printf("%d  -  %d\t",c,matrice[i][j]);
                c++;

                if(c%dim2==0) { //citeste pe linii
                        j=0; i++;
                //              ch=getch();
                //              if(ch=='c') break;
                }else j++;
        }

        fclose(r);
        printf("\n");
        printf("Calcularea histogramei initiale:\n");
        calcHist(matrice,Hist);
        // printf("histograma initiala:\n");
        // affisareHist(Hist);
        printf("\n copiem in file:\n");
        HistInFile(Hist,"D:/C-Free Standard/samples/work/out.csv");

        printf("\n filtram:\n");
        filtru_f11(Hist);
        filtru_f11(Hist);

        filtru_median(Hist);
        //  printf("\n histograma filtrata:\n");
        // affisareHist(Hist);

HistInFile(Hist,"D:/C-Free Standard/samples/work/out1.csv");

        pragHNT(Hist, prag1, prag2, Prags);

        ch=getch();
        printf("\n calcul gauss:\n");
        itervaleGauss(Hist,  Prags);
        for(i=0; i<=255;i++) if(Prags[i]>0)printf("\n Interval gauss: %d in %d ;",Prags[i],i);
        ch=getch();
        gaussCalc1(Hist,  Prags, errors_min, b_min);
        ch=getch();
        for(i=0; i<=255;i++) if(Prags[i]>0)printf("\n Interval gauss: %d in %d ;",Prags[i],i);
        transformGauss(Hist,  Prags, errors_min, b_min);
        ch=getch();
        for(i=0; i<=255;i++) if(Prags[i]>0)printf("\n Interval gauss: %d in %d ;",Prags[i],i);
        //gaussCalc2(Hist,  Prags, errors_q, b_q);
        //for(i=0;i<5;i++) printf("\nb_min=%d ; b_q=%d ",b_min[i],b_q[i]);

        //gaussAll(Hist,  Prags, Hout);
```

```
        ch=getch();
    //printf("\n\n Interval continuu maxim cu pragurile:  %d si %d \n\n", prag1, prag2);
    nivelare(Prags, matrice,matr_out);
    HistInFile(Hout,"D:/C-Free Standard/samples/work/out2.csv");
    FILE *fp; //fp este un pointer catre un fisier
        //se deschide fisierul in mod scriere (write)
        if((fp=fopen("D:/C-Free Standard/samples/work/blood_cells1.txt","w"))==NULL)
        puts("Fisierul nu poate fi deschis");
        for (i=0;i<dim1;i++){ //se scriu succesiv elementele
                for(j=0;j<dim2;j++)  fprintf (fp, "%d\t", matr_out[i][j]);
                fprintf(fp, "\n");
         }

        fclose (fp); //inchidere fisier
}
```

Anexa 2. Exemple de imagini binare obţinute în urma segmentării

Tabelul A1. Imagini binare obţinute în urma segmentării



Anexa 3. Simbolurile de pericol şi rezultatele obţinute la identificarea lor aplicând diverse metode

Tabelul A3.1 Simbolurile de pericol

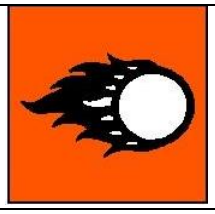
			
Danger **10**	Danger **11**	Danger**12**	Danger **13**
			
Danger**14**	Danger **15**	Danger **16**	

Simbolurile de pericol sunt preluate de pe site-ul: http://www.ilo.org/legacy/english/protection/safework/cis/products/safetytm/clasann1.htm

Tabelul A3.2 Numărul de puncte de interes determinate

Image	**ASIFT**	**SIFT**
Danger10	37 396	1 242
Danger11	11 606	269
Danger12	9 995	232
Danger13	24 181	948
Danger14	6 569	230
Danger15	14 531	414
Danger16	16 089	403

Tabelul A3.3 Modificări aplicate şablonului

		
imrotate(I,90,'bilinear');	imrotate(I,180,'bilinear');	imresize(I, 0.9);
		
imresize(I, 0.6);	imadjust(I,[],[],0.5);	imadjust(I,[],[],1.5);

Tabelul A3.4 Potrivirea imaginilor cu şablonul nemodificat cu metodele SIFT şi ASIFT

Nr. of tests	**Image and template**	**SIFT**		**ASIFT**	
		Nr. of matches	Result	Nr. of matches	Result
1	Danger10 & Danger11	0	F	369	T
2	Danger11 & Danger10	0	F	0	F
3	Danger10 & Danger12	0	F	434	T
4	Danger12 & Danger10	1	F	0	F
5	Danger10 & Danger13	0	F	0	F
6	Danger13 & Danger10	0	F	0	F
7	Danger10 & Danger14	2	F	64	T
8	Danger14 & Danger10	2	F	0	F
9	Danger10 & Danger15	0	F	0	F
10	Danger15 & Danger10	0	F	0	F
11	Danger10 & Danger16	2	F	135	T
12	Danger16 & Danger10	0	F	0	F
13	Danger11 & Danger12	4	T	395	T
14	Danger12 & Danger11	4	T	185	T
15	Danger11 & Danger13	9	T	0	F
16	Danger13 & Danger11	2	F	328	T
17	Danger11 & Danger14	2	F	44	T
18	Danger14 & Danger11	1	F	0	F
19	Danger11 & Danger15	0	F	0	F
20	Danger15 & Danger11	4	T	0	F
21	Danger11 & Danger16	0	F	79	T
22	Danger16 & Danger11	1	F	325	T
23	Danger12 & Danger13	0	F	187	T
24	Danger13 & Danger12	0	F	257	T
25	Danger12 & Danger14	1	F	56	T
26	Danger14 & Danger12	0	F	0	F
27	Danger12 & Danger15	2	F	68	T
28	Danger15 & Danger12	0	F	0	F
29	Danger12 & Danger16	8	T	90	T
30	Danger16 & Danger12	0	F	297	T
31	Danger13 & Danger14	10	T	41	T
32	Danger14 & Danger13	5	T	18	T
33	Danger13 & Danger15	0	F	0	F
34	Danger15 & Danger13	2	F	54	T
35	Danger13 & Danger16	8	T	61	T
36	Danger16 & Danger13	2	F	0	F
37	Danger14 & Danger15	1	F	52	T

38	Danger15 & Danger14	6	T	37	T
39	Danger14 & Danger16	8	T	76	T
40	Danger16 & Danger14	5	T	40	T
41	Danger15 & Danger16	5	T	54	T
42	Danger16 & Danger15	1	F	0	F

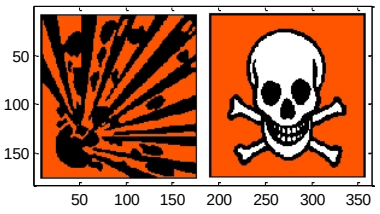
Note: T-True, F-False.

Algoritmul SIFT este elaborat de Lowe D., "Demo Software: SIFT Keypoint Detector", disponibil pe: http://www.cs.ubc.ca/~lowe/keypoints/

Algoritmul ASIFT este elaborate de Guoshen Yu, Morel, J.-M., "ASIFT: An Algorithm for Fully Affine Invariant Comparison", disponibil pe: http://www.ipol.im/pub/art/2011/my-asift/

1. Exemple de test de comparare a şabloanelor cu metoda SIFT:

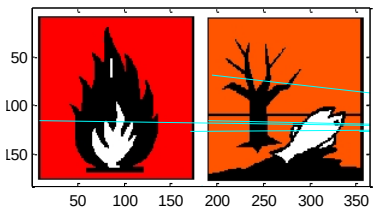
Test 1. Compararea *Danger10* cu *Danger13* şi viceversa :



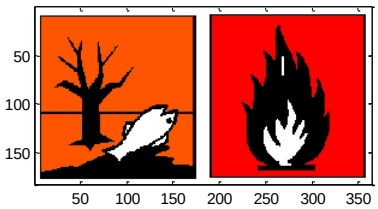

```
Finding keypoints...
1242 keypoints found.
Finding keypoints...
269 keypoints found.
Found 0 matches.
Elapsed time is 2.516349 seconds.
```


```
Finding keypoints...
269 keypoints found.
Finding keypoints...
1242 keypoints found.
Found 0 matches.
Elapsed time is 2.504944 seconds.
```

Test 2. Compararea *Danger12* cu *Danger16* şi viceversa :




```
Finding keypoints...
232 keypoints found.
Finding keypoints...
403 keypoints found.
Found 8 matches.
Elapsed time is 2.139371 seconds.
```




```
Finding keypoints...
403 keypoints found.
Finding keypoints...
232 keypoints found.
Found 0 matches.
Elapsed time is 1.315992 seconds.
```

Test 3. Compararea Danger13 cu Danger14 şi viceversa:

Finding keypoints...
948 keypoints found.
Finding keypoints...
230 keypoints found.
Found 10 matches.
Elapsed time is 1.738212 seconds.

Finding keypoints...
230 keypoints found.
Finding keypoints...
948 keypoints found.
Found 5 matches.
Elapsed time is 1.666036 seconds.

2. Exemple de test de comparare a şabloanelor cu metoda ASIFT:

Test 1. Compararea *Danger10* cu *Danger13* şi viceversa :

Computing keypoints on the two images...
37396 ASIFT keypoints are detected.
24181 ASIFT keypoints are detected.
Keypoints computation accomplished in 5 seconds.
Matching the keypoints...
The two images do not match. The matching is not significant: log(nfa)=0.605484.
Keypoints matching accomplished in 3 seconds.
ASIFT 0 matches.

Computing keypoints on the two images...
24181 ASIFT keypoints are detected.
37396 ASIFT keypoints are detected.
Keypoints computation accomplished in 6 seconds.
Matching the keypoints...
The two images do not match. The matching is not significant: log(nfa)=8.7987.
Keypoints matching accomplished in 3 seconds.
ASIFT 0 matches

Test 2. Compararea *Danger15* cu *Danger16* şi viceversa. Observăm rezultate diferite la multiple rulări ale programului.

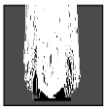
Computing keypoints on the two images...
14531 ASIFT keypoints are detected.
16089 ASIFT keypoints are detected.
Keypoints computation accomplished in 5 seconds.
Matching the keypoints...
The two images match! **54 matchings** are identified. log(nfa)=-11.0315.
Keypoints matching accomplished in 1 seconds.


Computing keypoints on the two images...
14531 ASIFT keypoints are detected.
16089 ASIFT keypoints are detected.
Keypoints computation accomplished in 4 seconds.
Matching the keypoints...
The two images match! **63 matchings** are identified. log(nfa)=-11.2553.
Keypoints matching accomplished in 1 seconds.


Computing keypoints on the two images...
16089 ASIFT keypoints are detected.
14531 ASIFT keypoints are detected.
Keypoints computation accomplished in 5 seconds.
Matching the keypoints...
The two images do not match. The matching is not significant: log(nfa)=4.65373.
Keypoints matching accomplished in 2 seconds.
ASIFT 0 matches.

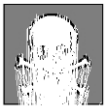
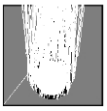
Test 3. Compararea *Danger12* cu *Danger11* şi viceversa.


Computing keypoints on the two images...
9995 ASIFT keypoints are detected.
11606 ASIFT keypoints are detected.
Keypoints computation accomplished in 5 seconds.
Matching the keypoints...
The two images match! **185 matchings** are identified. log(nfa)=-64.8258.
Keypoints matching accomplished in 1 seconds.

Computing keypoints on the two images...
11606 ASIFT keypoints are detected.
9995 ASIFT keypoints are detected.
Keypoints computation accomplished in 4 seconds.
Matching the keypoints...
The two images match! **395 matchings are identified**. log(nfa)=-14.7821.
Keypoints matching accomplished in 1 seconds.

Test 4. Compararea *Danger13* cu *Danger11* şi viceversa.

Computing keypoints on the two images...
24181 ASIFT keypoints are detected.
11606 ASIFT keypoints are detected.
Keypoints computation accomplished in 4 seconds.
Matching the keypoints...
The two images match! **324 matchings are identified**. log(nfa)=-6.05321.
Keypoints matching accomplished in 2 seconds.


Computing keypoints on the two images...
24181 ASIFT keypoints are detected.
11606 ASIFT keypoints are detected.
Keypoints computation accomplished in 5 seconds.
Matching the keypoints...
The two images match! **328 matchings are identified**. log(nfa)=-17.7008.
Keypoints matching accomplished in 1 seconds.

Computing keypoints on the two images...
11606 ASIFT keypoints are detected.
24181 ASIFT keypoints are detected.
Keypoints computation accomplished in 5 seconds.
Matching the keypoints...
The two images do not match. The matching is not significant: log(nfa)=4.43016.
Keypoints matching accomplished in 2 seconds.
ASIFT 0 matches.

Test 5. Compararea *Danger15* cu *Danger13* şi viceversa.

Computing keypoints on the two images...
14531 ASIFT keypoints are detected.
24181 ASIFT keypoints are detected.
Keypoints computation accomplished in 5 seconds.
Matching the keypoints...
The two images match! 54 matchings are identified. log(nfa)=-3.65579.
Keypoints matching accomplished in 1 seconds.

Computing keypoints on the two images...
24181 ASIFT keypoints are detected.
14531 ASIFT keypoints are detected.
Keypoints computation accomplished in 5 seconds.
Matching the keypoints...
The two images do not match. The matching is not significant: log(nfa)=2.07578.
Keypoints matching accomplished in 2 seconds.
ASIFT 0 matches.

Anexa 4. Algoritmul fuzzy pentru clasificarea obiectelor

[System]
Name='ShapesClassification'
Type='mamdani'
Version=2.0
NumInputs=3
NumOutputs=1
NumRules=13
AndMethod='min'
OrMethod='max'
ImpMethod='min'
AggMethod='max'
DefuzzMethod='mom'

[Input1]
Name='Shape'
Range=[0 1]
NumMFs=4
MF1='Elongated':'trapmf',[0 0 0.45 0.55]
MF2='Irregular':'trapmf',[0.45 0.55 0.7 0.75]
MF3='Oval':'trapmf',[0.7 0.75 0.85 0.885]
MF4='Round':'trapmf',[0.85 0.885 1 1]

[Input2]
Name='Size'
Range=[0 0.5]
NumMFs=4
MF1='Small':'trapmf',[0 0 0.025 0.04]
MF2='Medium':'trapmf',[0.025 0.04 0.06 0.1]
MF3='Large':'trapmf',[0.06 0.1 0.15 0.2]
MF4='ExtraLarge':'trapmf',[0.15 0.2 0.5 0.5]

[Input3]
Name='Symmetry'
Range=[0 1]
NumMFs=3
MF1='Small':'trapmf',[0 0 0.2 0.3]
MF2='Medium':'trapmf',[0.2 0.3 0.55 0.6]
MF3='Big':'trapmf',[0.55 0.6 1 1]

[Output1]
Name='ObjectType'
Range=[-0.1 1.1]
NumMFs=3
MF1='Dangerous':'trimf',[-0.05 0 0.05]
MF2='Undetermined':'trimf',[0.45 0.5 0.55]
MF3='Recyclable':'trimf',[0.95 1 1.05]

[Rules]
1 1 3, 1 (1) : 1
1 2 3, 1 (1) : 1
1 3 3, 1 (1) : 1
1 4 3, 2 (1) : 1
1 1 2, 1 (0.7) : 1
1 2 2, 1 (0.7) : 1
1 3 2, 1 (0.7) : 1
1 4 2, 2 (1) : 1
1 1 1, 2 (1) : 1
1 2 1, 2 (1) : 1
1 3 1, 2 (1) : 1
1 4 1, 2 (1) : 1
1 3 2, 3 (1) : 1

DECLARAŢIA PRIVIND ASUMAREA RĂSPUNDERII

Subsemnata, declar pe proprie răspundere că materialele prezentate în teza de doctorat sunt rezultatul propriilor cercetări şi realizări ştiinţifice, în caz contrar urmând să suport consecinţele, în conformitate cu legislaţia în vigoare.

RUSU Mariana

Semnătura 

Data 19.11.2018

CURRICULUM VITAE

INFORMAŢII PERSONALE

Nume, prenume: Rusu Mariana

Data naşteri: 23/08/1981

Locul naşterii: Călăraşi

Cetăţenia: Republica Moldova

rusu.maryana@gmail.com

EXPERIENŢA PROFESIONALĂ

06/2008–09/2011 Inginer şi Responsabil de Spaţiu Francofon la Filiera Francofonă Informatica, Universitatea Tehnică a Moldovei, Chişinău (Republica Moldova)

01/09/2008–Prezent Lector la Filiera Francofonă Informatica, Universitatea Tehnică a Moldovei, Chişinău (Republica Moldova)

EDUCAŢIE ŞI FORMARE

2003–2007 Licenţiat în Informatică, Universitatea Tehnică a Moldovei

2008–2010 Master în Ştiinţe Exacte, Universitatea Tehnică a Moldovei

2010–2013 doctorandă la specialitatea „Modelare matematică, metode matematice, produse program", Universitatea Tehnică a Moldovei

COMPETENŢE PERSONALE

Limba(i) maternă(e) română

Alte limbi străine cunoscute : franceză şi rusă experimentat, engleză independent, germană elementar.

INFORMAŢII SUPLIMENTARE

Conferinţe, distincţii, şcoli doctorale

- 17 mai 2012– participarea la a 4-a Conferinţă Internaţională de Telecomunicaţii, Electronică şi Informatică organizată de Universitatea Tehnică a Moldovei, ICTEI 2012
- 14-18 iunie – Summer School on Human-Machine Systems, Cyborgs, and Enhancing Devices, organizată de Universitatea Tehnică «Gheorghe Asachi» din Iaşi

- 18 -23 iunie – Doctoral Summer School on Evolutionary Computing in Optimisation and Data Mining (ECODAM), organizată de Universitatea "Al.I.Cuza" din Iasi,
- 18-20 aprilie 2013 ICNBME – 2nd International Conference on Nanotechnologies and Biomedical Engineering, Chisinau
- Bursa "Eugen Ionescu" oferită de Guvernul României, sub egida Agenţiei Universitare a Francofoniei (AUF), anul de studii 2012-2013
- Bursa de excelenţă a Guvernului pentru doctoranzi pe anul 2013, HOTĂRÎRE Nr. 994 din 26.12.2012, publicat : 28.12.2012 în Monitorul Oficial Nr. 273-279, art Nr : 1070, http://lex.justice.md/viewdoc.php?action=view&view=doc&id=346113&lang=1
- Membru al Consiliului doctoranzilor de pe lângă Consiliul Naţional pentru Acreditare şi Atestare pentru doctoranzi pe anul 2013
- Premiul senatului UTM – Cel mai bun doctorand al anului 2011-2012 (gr. II)
- Premiul senatului UTM – Cel mai bun doctorand al anului 2012-2013 (gr. III)
- Bursa "Eugen Ionescu" oferită de Guvernul României, sub egida Agenţiei Universitare a Francofoniei (AUF), anul de studii 2013-2014
- 26-27 iunie 2015 - participarea la International Conference 7th Edition Electronics, Computers and Artificial Intelligence, Bucharest – Romania
- 19-21 noiembrie 2015 - participarea la IEEE Int. Conf. on E-Health and Bioengineering (EHB 2015), Iasi- Romania.
- 24 martie 2016 - participarea la simpozionul IIVA 2016 (Information in Image and Video Analysis Theory and Applications) Simpozion Aniversar Dedicat aniversării a 150 de ani de la înfiinţarea Academiei Române & 25 de ani de la înfiinţarea Secţiei de Ştiinţa şi Tehnologia Informaţiei, Iasi-Romania.

Publicaţii

- Luchianov L., Istrati D., Popescu M., Zubco S., Sisteme de operare "Introducere în Linux" Partea I, Îndrumar de laborator, 50 pagini, UTM, Chişinău, 2009
- Rusu M., Horia-Nicolai L. Teodorescu, “A Method for Image Segmentation based on Histograms – Preliminary Results”, 4th International Conference Telecommunications, Electronics and Informatics, pp.351-354, UTM, 2012.
- Teodorescu HN., Rusu M., “Yet Another Method for Image Segmentation based on Histograms and Heuristics”, Computer Science Journal of Moldova, vol.20, no.2(59), pp. 163-177, 2012, http://www.math.md/publications/csjm/issues/v20-n2/11087/.

- Teodorescu HN., Rusu M., “Image Segmentation Based on G-UN-MMs and Heuristics - Theoretical Background and Results –” Proceedings of the Romanian Academy, Series A, Vol. 14, No. 1/2013, pp. 78–85, 2013 http://www.acad.ro/sectii2002/proceedings/doc2013-1/12-Teodorescu.pdf
- Rusu M., Teodorescu HN., „Quality Analysis of Image Segmentation based on G-UN-MMs”, 2nd International Conference on Nanotechnologies and Biomedical Engineering, Chisinau, Republic of Moldova, pp. 620-624, April 18-20, 2013 http://www.icnbme.sibm.md/bio_imaging.html
- Moraru V., Rusu M., „Algorithm for linear pattern separation”, Meridian Ingineresc, No. 2, pp. 26-29, 2013 http://www.utm.md/meridian/2013/0_Meridian_Ingineresc_nr2_2013.pdf
- Rusu M., „Thresholding methods and quantitative evaluation of results”, Meridian Ingineresc, No. 2, pp. 18-25, 2013 http://www.utm.md/meridian/2013/0_Meridian_Ingineresc_nr2_2013.pdf
- Teodorescu HN., Rusu M., Improved Heterogeneous Gaussian and Uniform Mixed Models (G-U-MM) and Their Use in Image Segmentation, Romanian Journal of Information Science and Technology, Volume 16, Number 1, 2013, pp. 29-51 http://www.imt.ro/romjist/.
- Rusu M., „Computerized visual inspection applied to identification and classification of labeled chemicals”, International Conference 7th Edition Electronics, Computers and Artificial Intelligence, IEEE Conference Publications, Bucharest – Romania, Vol. 7, No. 2, 6p., 2015 http://ieeexplore.ieee.org/xpl/articleDetails.jsp?reload=true&arnumber=7301214
- Rusu M., Zbancioc M.-D., Automated identification of objects based on Normalized Cross-Correlation and Genetic Algorithm, IEEE Int. Conf. on E-Health and Bioengineering, 4 p., 2015 http://ieeexplore.ieee.org/xpl/articleDetails.jsp?arnumber=7391457
- Rusu M., Zbancioc M.-D., Fuzzy Rule-based System for Pattern Recognition and Automated Classiffcation, Computer Science Journal of Moldova, Vol.25, No.1(73), 19 p., 2017