

Universitatea Tehnica din Cluj-Napoca
Departamentul de Calculatoare

Procesarea imaginilor
Digitalizarea grafurilor
Proiect 2018-2019

Nume: Grigor Sonia Eufrosina Maria
Email: sonia.grigor@gmail.com

Nume: Lazareanu Sabina Stefana
Email: lazareanusabina@gmail.com

Grupa: 30233

As. dr. ing. Diana Borza
Diana.Borza@cs.utcluj.ro

Cuprins

Abstract	3
1 Introducere	4
2 Studiu bibliografic	5
2.1 Metode de rezolvare	5
2.2 Avantaje si dezavantaje	5
3 Metoda propusa	6
3.1 Schema bloc	6
3.2 Descrierea solutiei	6
3.2.1 Detectarea cercurilor	7
3.2.2 Etichetarea	7
3.2.3 Detectarea liniilor	7
3.2.4 Detectarea cifrelor	7
3.2.5 Digitalizare	8
3.3 Imagini intermediare	8
4 Rezultate experimentale	11
4.1 Imagini	11
4.2 Evaluare numerica	13
4.3 Parapetree metodelor	13
5 Concluzii si dezvoltari ulterioare	14
Bibliografie	15

Abstract

Aceasta sectiune contine o prezentare clara si concisa a metodei propuse petru realizarea proiectului "Digitalizarea grfurilor".

Scopul acestui proiect este realizarea unui produs software care isi propune detectarea grafurilor si identificarea elemntelor componente precum si a informatiilor din interiorul nodurilor. Proiectul are la baza procesarea imaginilor care contin reprezentari ale grafuriloae neorientate sau orientate.

Propunerea de rezolvare a acestui proiect consta in identificarea fiecarui element (nod, muchie, informatie din nod), apoi maparea la o anumita structura de date.

Chapter 1

Introducere

Sectiunea dedicata introducerii contine contextul si motivatia proiectului, precum si obiectivul principal al proiectului, dar si scopurile secundare.

O scurta descriere a etapelor de rezolvare a proiectului este urmatoarea:

1. Se vor procesa imagini în care sunt reprezentate vizual (desenate) grafuri și se vor recunoaște nodurile și muchiile. Rezultatul algoritmului va fi graful reprezentat sub forma de listă de adiacență.
2. Preprocesare imagine, detecție linii (muchii), detecție cercuri (noduri), recunoașterea cifrelor (id-ul fiecărui nod)
3. Baza de date: se va crea o bază de date prin scanarea/fotografierea mai multor desene cu grafuri
4. Opțional: se pot reda grafic și interactiv algoritmi simpli (sau complecși) din teoria grafurilor: bread first search, depth first search, Tarjan etc.

Alegerea acestui proiect a avut la baza solide argumente în ceea ce privește acumularea unor noi cunoștințe, aprofundarea cunoștințelor dobândite în anii anteriori, precum și interconectarea lor. Aceasta aplicație este foarte folositoare pentru digitalizarea grafurilor în scopul ușurării manipulării datelor și de asemenea aplicarea unor algoritmi pe rezultatele obținute.

Chapter 2

Studiu bibliografic

2.1 Metode de rezolvare

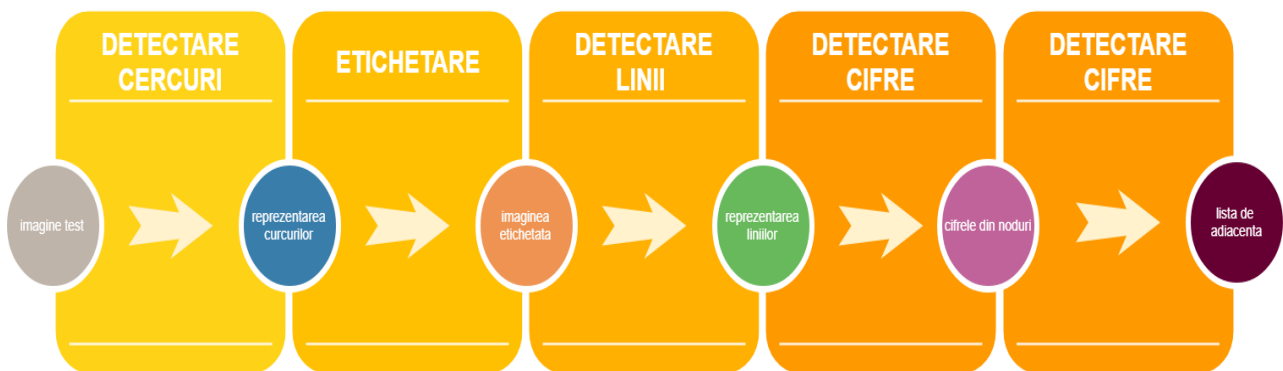
2.2 Avantaje si dezavantaje

Chapter 3

Metoda propusa

3.1 Schema bloc

O schema bloc care sugerează modul în care se face prelucrarea imaginilor este prezentată în figura de mai jos.



3.2 Descrierea solutiei

Solutia propusa este una complexa care trece imaginea prin mai multe etape. O scurta prezentare a metodei utilizate se poate deduce din urmatoare secventa de cod surprinsa din cadrul functiei main:

```
while (openFileDialog(fname))
{
    Mat src, binarizedImg, edgeImg, circleImg, linesImage, imagineEtichetata;
    src = imread(fname, CV_LOAD_IMAGE_GRAYSCALE);
    binarizedImg = binaryImage(src);
    circleImg = houghCirclesFunctionImage(src.clone());
    imshow("Real Circles", circleImg);
    imagineEtichetata = etichetare(binarizedImg.clone());
    imshow("Labeled Image", imagineEtichetata);
    vector<Mat>::iterator it;
    vector<Vec4i> edges = houghLinesFunction(binarizedImg.clone());
    setEdges(edges);
    auto objects = objectsFromImage(src.clone());
```

```

printf("Nr Objects: %d\n", objects.size());
auto digits = detectDigitImages(objects, 50);

bindCircleInfo();
digitalizeGraph(binarizedImg.clone());
printAdjacencyMatrix();
convertAdjacencyMatrixToAdjacencyList();

waitKey();
}

```

3.2.1 Detectarea cercurilor

Etapă de detectare a cercurilor are la baza algoritmul Hough de detectie a cercurilor. Pe langa aplicarea algoritmului, tot in aceasta etapa se va popula vectorul care patrreaza nodurile. Prin aceasta populare a nodurilor se intelege creerea unei structuri de date care contine un punct. Acest punct este centrul cercului.

Inainte de aplicarea algoritmului Hough de detectie a cercurilor, a fost nevoie de o prelucrare a imaginii prin aplicarea unui filtru Gaussian cu scopul de a blur-a imaginea.

3.2.2 Etichetarea

Etichetarea este o etapa importanta din flow-ul proiectului deoarece face distinctia intre elementele grafice(cercuri, linii) si informatiile din fiecare cerc. Am ales aceasta metoda pentru a face un triaj din numarul total de obiecte aflate in imagine.

3.2.3 Detectarea liniilor

Precum dectia cercurilor, detectia liniilor se face utilizand algoritmul de detectie a liniilor HpughLineP implementat in OpenCV. O etapa post-mergatoare a fost eliminarea liniilor care erau duplicate. Din varii motive, algoritmul detecta mai multe linii care erau apropiate intre ele. Pentru a alege muchia reala am apelat la calcularea centrului fiecarei linii. Dupa compararea centrelor, am sters liniile duplicate, astfel realizandu-se detectia si procesarea muchiilor.

3.2.4 Detectarea cifrelor

Etapă de detectare a cifrelor a avut o etapa premergatoare de a extrage obiectele din imagine. Am ales aceasta abordare deoarece astfel s-a redus semnificativ regiunea noastra de interes.

Dupa aceasta pre-procesare a imaginii am obtinut obiectele asupra carora vom aplica algoritmul de identificare a cifrelor. Pentru a eficientiza cat mai mult procesul de character recognision, am ales sa binarizam imaginea si apoi sa numaram numarul de pixeli negrii. Am aplicat algoritmul KNN doar asupra acelor abiecte care aveau un numai mai mare de pizel decat un orag stabilit de noi.

Detectia caracterelor si implicit a cifrelor are la baza algoritmul K-Nearest Neighbour.

Pe langa detectarea numerelor, tot la acest pas am decis sa facem corelatia intre cerc si informatie. Practic avem o structura de date compusa dintr-un punct si o valoare care reprezinta structura definitorie pentru nod.

3.2.5 Digitalizare

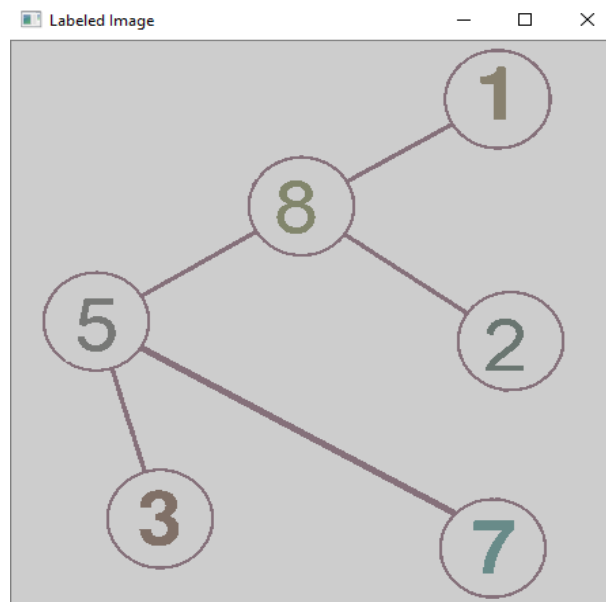
Procesul denumit generic digitalizare are mai multe etape. Insa cea mai importanta este crearea legaturii dintre liniile gasite la pasul anterior si cercurile. Astfel, dupa aceasta etapa, programul identifica toate elementele si creaza legaturile- muchiile intre noduri. Etapele ulterioare constau in crearea matricei de adiacenta, iar mai apoi crearea listei de adiacenta a grafului in discutie.

3.3 Imagini intermediare

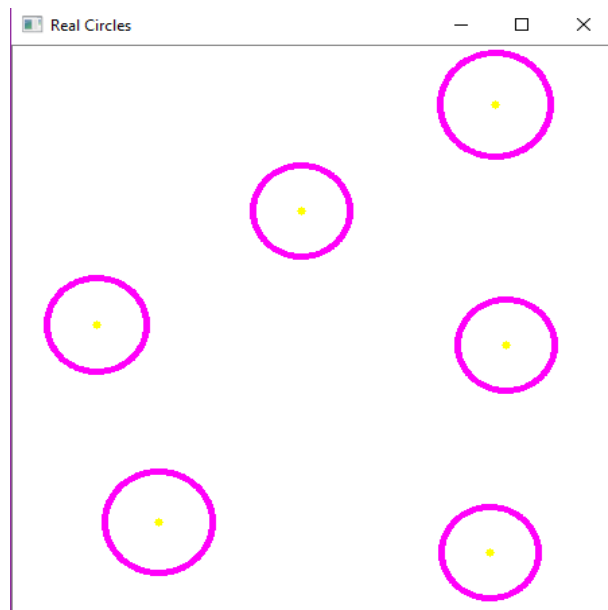
In dezvoltarea acestui algoritm am fost nevoiti sa folosim o multime de pasi intermediare de testare si verificare. Pentru a reusi sa realizam proiectul in stadiul in care este acum, am folosit diferite metode de detectare a erorilor: debugger, afisarea valorilor in consola, tool-ul de ImageWatch precum si afisarea imaginilor intermendiare.

Dintre imaginile afisare in stadiile intermediare fac parte

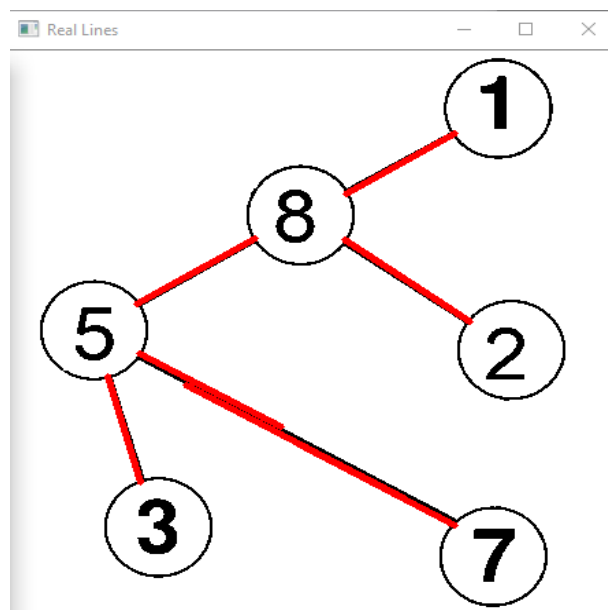
- Afisarea imaginii etichetate. Etichetarea a facut distingerea intre cifrele din noduri si celelalte elemente: linii, cercuri.



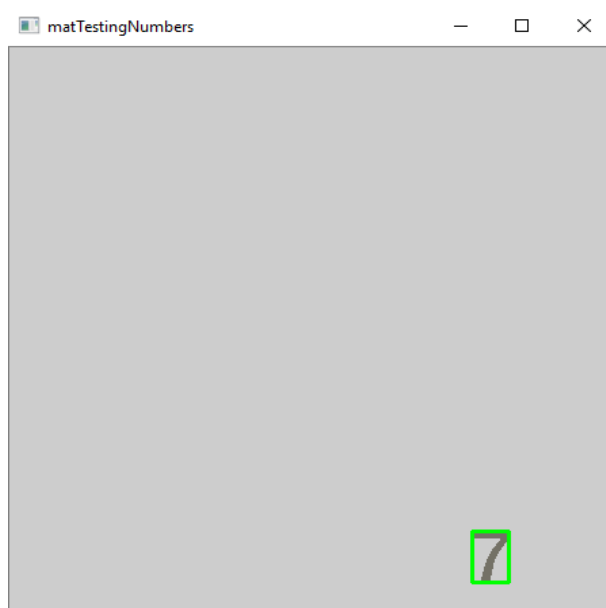
- Afizarea imagii care reprezinta identificarea cercurilor si centrele lor.



- Afizarea imagii care reprezinta identificarea liniilor.



- Afizarea imagii care reprezinta identificarea informatiilor din noduri.



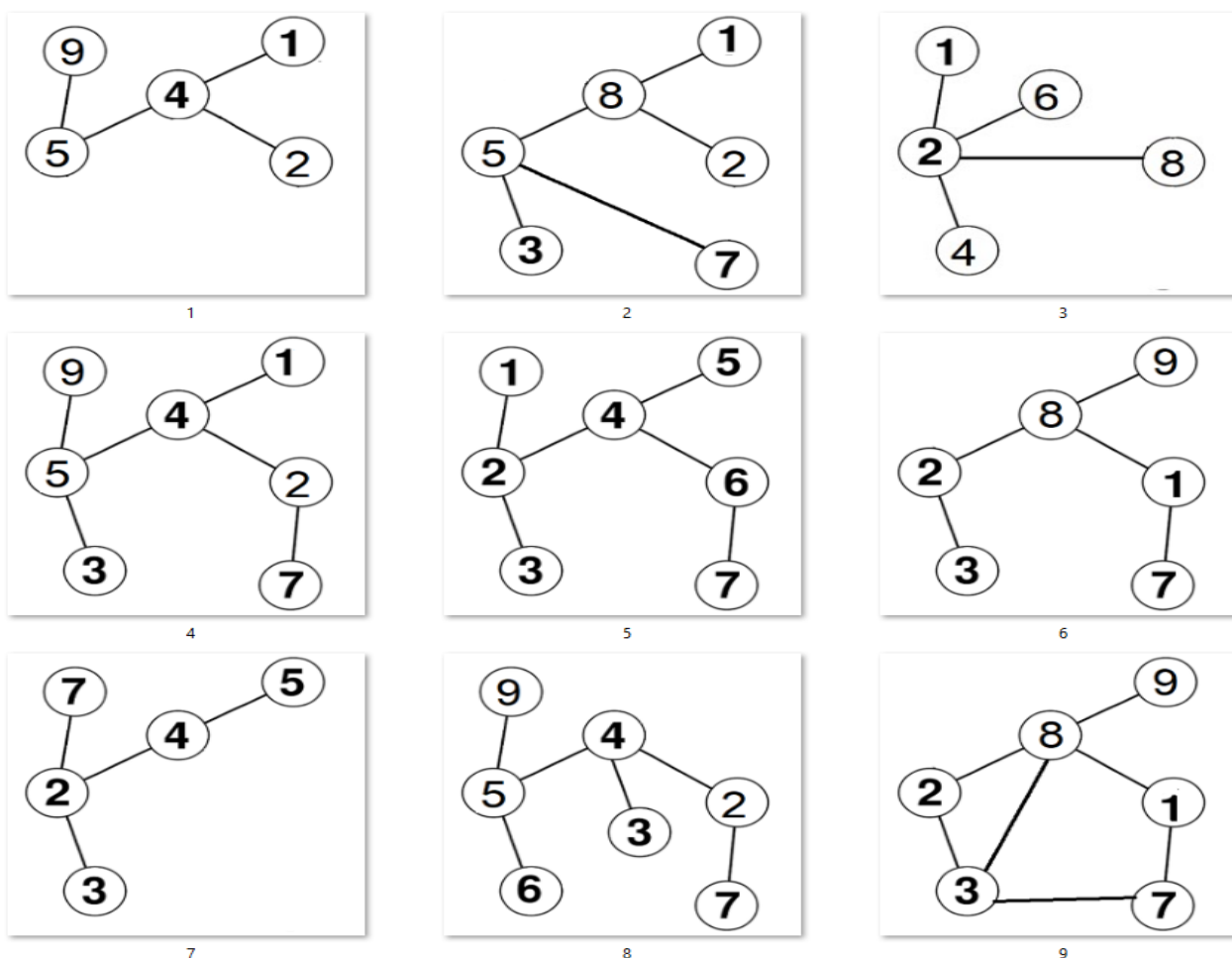
Chapter 4

Rezultate experimentale

4.1 Imagini

Imaginile pe care le-am folosit ca baza de date pentru test in acest proiect sunt imagini de dimensiuni egale care contin diferite reprezentari ale unor grafuri conexe. Aceste grafuri contin un diferit numat de noduri si muchii, unele contin cicluri sau sunt simple lanturi. Prin utilizarea acestor imagini am incercat sa surprindem cat mai bine eficienta acestui proiect de digitalizare a grafurilor.

Baza de date a fost compusa din 10 imagini ale caror rezultate sunt corecte si verificate. Cateva exemple din cele 10 imagini gasiti in figura de mai jos:



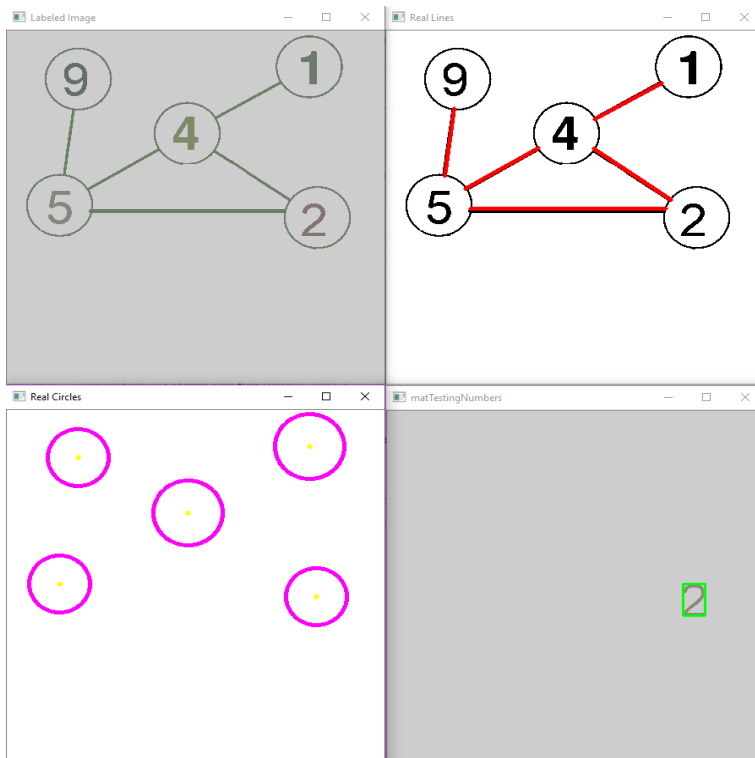
Pentru o buna functionare a proiectului si obtinerea rezultatelor asteptate, este nevoie de respectarea unor parametri ai imaginilor, si anume: imaginile de test sa aiba dimensiunea de 447px x 447px , iar pentru o buna recunoastere a cifrelor, acestea trebuie sa aiba dimensiunea aproximativa a fontului Arial, dimensiune 45, fara a fi bold sau italic.

Vom descrie acum rezultatele proiectului realizand un scurt tutorial prezentand pasii care trebuie urmati:

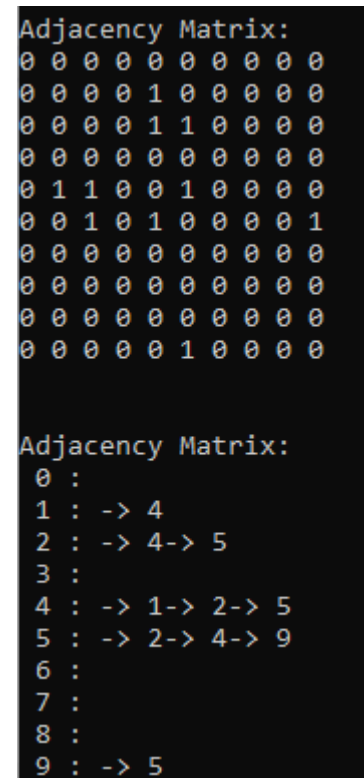
1. Lansare programului in executie apasand butonul "Run" sau "Local Windows Debugger"
2. Navigarea in folderul proiectului care contine imaginile de test `\OpenCVApplication-VS2017_Project\Images\Project`
3. Alegerea unei imagini asupra careia sa se aplice algoritmul de digitalizare a grafurilor
4. Vizualizarea rezultatelor intermediare sub forma de imagini sau in consola
5. Vizualizarea rezultatelor finale in consola.

Rezultatele afisate vor contine rezultate intermediare sub forma imaginilor si rezultate finale sub forma de text afisate in consola

1. imaginea cu etichetarea fiecarui obiect in culori din aceeasi gama
2. imaginea cu liniile detectate de catre algoritm. Aceasta imagine va avea liniile detectate colorate cu rosu
3. imaginea cu cercurile detectate. Se va prezenta conturul lor impreuna cu centrul
4. imaginea care prezinta gradual concomitent cu afisarea in consola a detectiei caracterelor din fiecare cerc.
5. matrice de adiacenta pe baza grafului ales
6. lista de adiacenta a nodurilor din graf



Rezultate intermediare



Rezultate finale

4.2 Evaluare numerica

În ceea ce privește evaluarea numerică, metoda propusă de noi funcționează conform așteptărilor utilizând imaginile de test puse la dispoziție în folderul Images. În mod normal, dacă se respectă indicațiile cu privire la dimensiunea imaginii, grosimea liniilor, forma cercurilor și fontul cifrelor, proiectul ar trebui să aibă un rezultat favorabil, lista de adiacență rezultată fiind în concordanță cu imaginea gradului inițial.

4.3 Parapetreei metodelor

Referitor la parametrii metodelor, aceștia nu ar trebui modificați deoarece sunt aleși în așa fel încât să funcționeze corect. A fost nevoie de ajustarea lor pentru a ajunge la rezultatele așteptate. Acești parametri se referă la parametrii metodelor predefinite din OpenCV cum ar fi

void HoughLinesP(InputArray image, OutputArray lines, double rho, double theta, int threshold, double minLineLength=0, double maxLineGap=0)

sau

void Canny(InputArray image, OutputArray edges, double threshold1, double threshold2, int apertureSize=3, bool L2gradient=false).

Dacă vorbim despre parametrii algoritmului sub forma unor date de intrare, nu este nevoie de așa ceva. Singura informație pe care trebuie să o furnizeze utilizatorul este o imagine de test.

Chapter 5

Concluzii si dezvoltari ulterioare

Proiectul realizat si-a atins scopul, acela de a digitaliza grafurile. Astfel, avand la dispozitie o imagine cu un graf neorientat am reusit crearea unui algoritm pe baza caruia se poate digitaliza graful din imaginea initiala. Deigitalizarea unei astfel de imagini inseamna crearea si popularea unei structuri de date cu datele din imagine. Structura pe care am ales sa o folosim este o lista de adiacenta, aceasta fiind specificarea proiectului. Totusi, in crearea unei liste de adiacenta, am creat mai intai o matrice de adiacenta. Dezavantajul matricii este faptul ca trebuie alocata memoria in preaabil, pe cand, in vazul listei se va aloca dinamic.

In ceea ce priveste posibilele dezvoltari ulterioare acestea se pot materializa pe mai multe directii:

- extinderea proiectului actual pentru a fi capabil sa proceseze o gama mai larga a informatiilor din noduri. In momentul de fata, se face procesare doar pentru cifre din intervalul 1-9.
- realizarea posibilitatii de digitalizarea a grafurilor orientate.
- implementarea proiectului in vederea procesarii imaginilor de dimensiuni diferite, fonturi diferite. Practic posibilitatea de a putea procesa orice tip de imagine.
- atasarea unui modul ce permite digitalizarea automata prin conectarea unei camere si obtinerea rezultatului in timp real.

Bibliografie

1. P. Hough, "Method and means for recognizing complex patterns", US patent 3,069,654, 1962.
2. R. O. Duda and P. E. Hart, "Use of the Hough Transformation to Detect Lines and Curves in Pictures," Comm. ACM, Vol. 15, pp. 11–15, 1972.
3. https://docs.opencv.org/2.4/doc/tutorials/imgproc/imgtrans/canny_detector/canny_detector.html?highlight=edge
4. http://users.utcluj.ro/~rdanescu/srf/lab_11r.pdf
5. https://docs.opencv.org/2.4/doc/tutorials/imgproc/imgtrans/hough_circle/hough_circle.html
6. <https://www.infosun.fim.uni-passau.de/~chris/down/OpticalGraphRecognition.pdf?fbclid=IwAR31jJ6oNmsoeoQEoP6PgT66z8TdGZ5AJ8bV10IA148yHp2NmUE9MUNQqWo>
7. https://sci-hub.tw/10.1007/s001380050054?fbclid=IwAR0G_F-heHfJkyycsU-JDAzGHZbr_tDAYjsiXrU1EGh149mWTww0QhhXBaM