



Universitatea Tehnică din Cluj-Napoca  
Departamentul de Calculatoare

## Procesarea imaginilor

### Digitalizarea grafurilor

*Proiect 2018-2019*

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

Nume: Lăzăreanu Sabina Ștefana  
Email: lazareanusabina@gmail.com

Grupa: 30233

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

# Cuprins

<b>Abstract</b>	<b>3</b>
<b>1 Introducere</b>	<b>4</b>
<b>2 Studiu bibliografic</b>	<b>5</b>
2.1 Metode de rezolvare . . . . .	5
2.2 Avantaje și dezavantaje . . . . .	5
<b>3 Metoda propusa</b>	<b>6</b>
3.1 Schema bloc . . . . .	6
3.2 Descrierea soluției . . . . .	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
<b>4 Rezultate experimentale</b>	<b>11</b>
4.1 Imagini . . . . .	11
4.2 Evaluare numerica . . . . .	13
4.3 Parametrii metodelor . . . . .	13
<b>5 Concluzii și dezvoltări ulterioare</b>	<b>14</b>
<b>Bibliografie</b>	<b>15</b>

# Abstract

Scopul proiectului "Digitalizarea grafurilor" este realizarea unui produs software care își propune detectarea grafurilor și identificarea elementelor componente precum și a informațiilor din interiorul nodurilor. Proiectul are la baza procesarea imaginilor care conțin reprezentări ale grafurilor neorientate.

Propunerea de rezolvare a acestui proiect consta în identificarea fiecărui element ( nod, muchie, informație din nod), apoi maparea la o anumita structura de date.

# Chapter 1

## Introducere

Proiectul "Digitalizarea grafurilor" presupune procesarea imaginilor în care sunt reprezentate vizual (desenate) grafuri și recunoașterea nodurile și muchiile. Rezultatul algoritmului va fi graful reprezentat sub forma de listă de adiacență.

Etapele care au fost urmate în vederea rezolvării proiectului sunt preprocesare imagine, detecție linii (muchii), detecție cercuri (noduri), recunoașterea cifrelor (id-ul fiecărui nod).

În ceea ce privește baza de date: se va crea o bază de date prin scanarea/fotografierea mai multor desene cu grafuri.

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

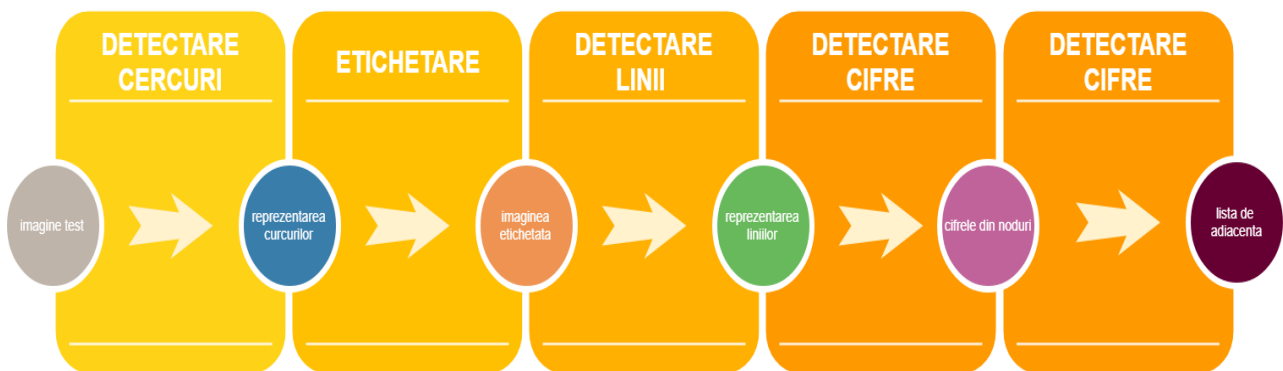
2.2 Avantaje și dezavantaje

# 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 soluției

Soluția propusă este una complexă care trece imaginea prin mai multe etape. O scurtă prezentare a metodei utilizate se poate deduce din următoarea secvență de cod surprinsă din cadrul funcției 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 detecție a cercurilor. Pe lângă aplicarea algoritmului, tot în această etapă se va popula vectorul care păstrează nodurile. Prin această populare a nodurilor se înțelege crearea unei structuri de date care conține un punct. Acest punct este centrul cercului.

Înainte de aplicarea algoritmului Hough de detecție a cercurilor, a fost nevoie de o prelucrare a imaginii prin aplicarea unui filtru Gaussian cu scopul de a blura imaginea.

### 3.2.2 Etichetarea

Etichetarea este o etapă importantă din flow-ul proiectului deoarece face distincția între elementele grafice(cercuri, linii) și informațiile din fiecare cerc. Am ales această metodă pentru a face un triaj din numărul total de obiecte aflate în imagine.

### 3.2.3 Detectarea liniilor

Precum detecția cercurilor, detecția liniilor se face utilizând algoritmul de detecție a liniilor HoughLineP implementat în OpenCV. O etapă post-mergătoare a fost eliminarea liniilor care erau duplicate. Din varii motive, algoritmul detecta mai multe linii care erau apropiate între ele. Pentru a alege muchia reală am apelat la calcularea centrului fiecărei linii. După compararea centrelor, am șters liniile duplicate, astfel realizându-se detecția și procesarea muchiilor.

### 3.2.4 Detectarea cifrelor

Etapă de detectare a cifrelor a avut o etapă premergătoare de a extrage obiectele din imagine. Am ales această abordare deoarece astfel s-a redus semnificativ regiunea noastră de interes.

După această pre-procesare a imaginii am obținut obiectele asupra cărora vom aplica algoritmul de identificare a cifrelor. Pentru a eficientiza cât mai mult procesul de caracter recognition, am ales să binarizăm imaginea și apoi să numărăm numărul de pixeli negrii. Am aplicat algoritmul KNN doar asupra acelor obiecte care aveau un număr mai mare de pixel decât un prag stabilit de noi.

Detecția caracterelor și implicit a cifrelor are la baza algoritmul K-Nearest Neighbour.

Pe lângă detectarea numerelor, tot la acest pas am decis să facem corelația între cerc și informație. Practic avem o structură de date compusă dintr-un punct și o valoare care reprezintă structura definitorie pentru nod.

### 3.2.5 Digitalizare

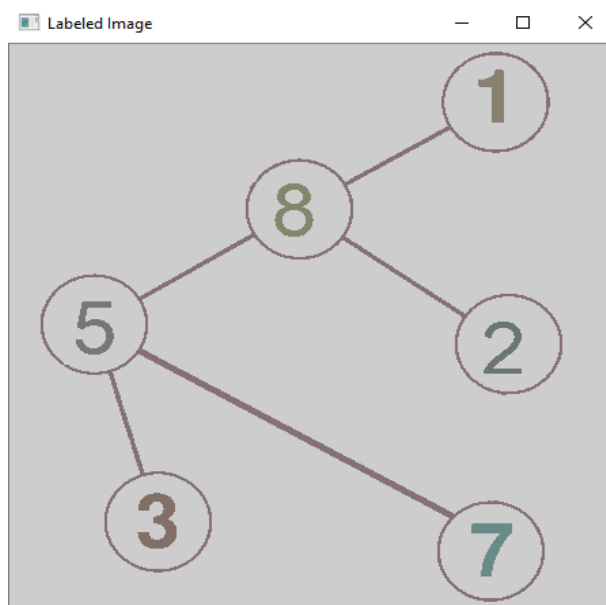
Procesul denumit generic digitalizare are mai multe etape. Însa cea mai importantă este crearea legăturii dintre liniile găsite la pasul anterior și cercurile. Astfel, după această etapă, programul identifică toate elementele și creează legăturile- muchiile între noduri. Etapele ulterioare constau în crearea matricei de adiacență, iar mai apoi crearea listei de adiacență a grafului în discuție.

## 3.3 Imagini intermediare

În dezvoltarea acestui algoritm am fost nevoiți să folosim o mulțime de pași intermediare de testare și verificare. Pentru a reuși să realizăm proiectul în stadiul în care este acum, am folosit diferite metode de detectare a erorilor: debugger, afișarea valorilor în consolă, tool-ul de ImageWatch precum și afișarea imaginilor intermediare.

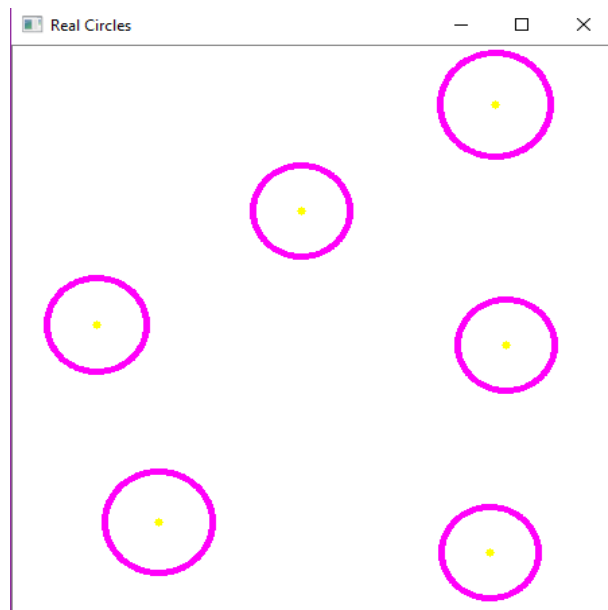
Dintre imaginile afișate în stadiile intermediare fac parte

- Afișarea imaginii etichetate. Etichetarea a făcut distingerea între cifrele din noduri și celelalte elemente: linii, cercuri.

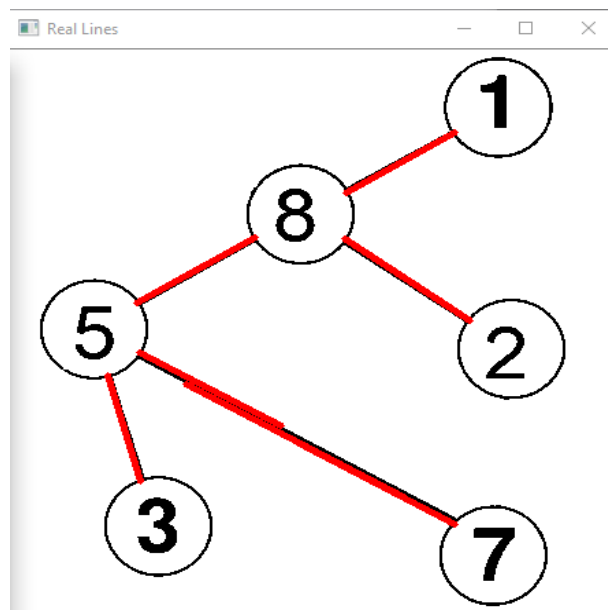


- Afișarea imaginii care reprezintă identificarea cercurilor și centrele lor.

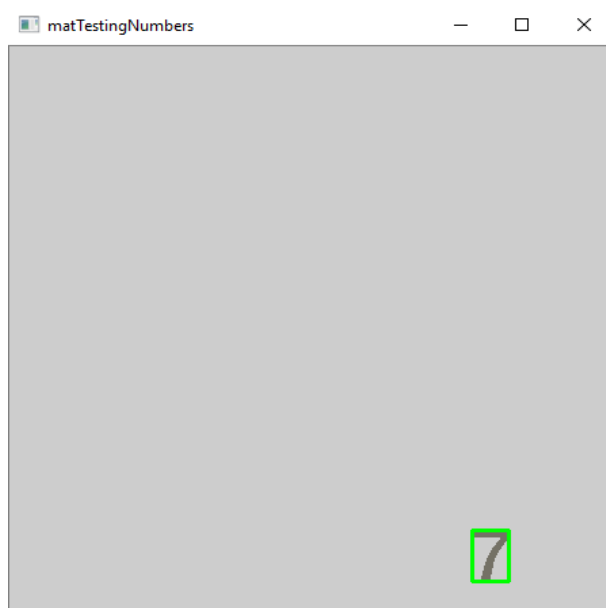




- Afişarea imaginii care reprezintă identificarea liniilor.



- Afişarea imaginii care reprezintă identificarea informațiilor din noduri.



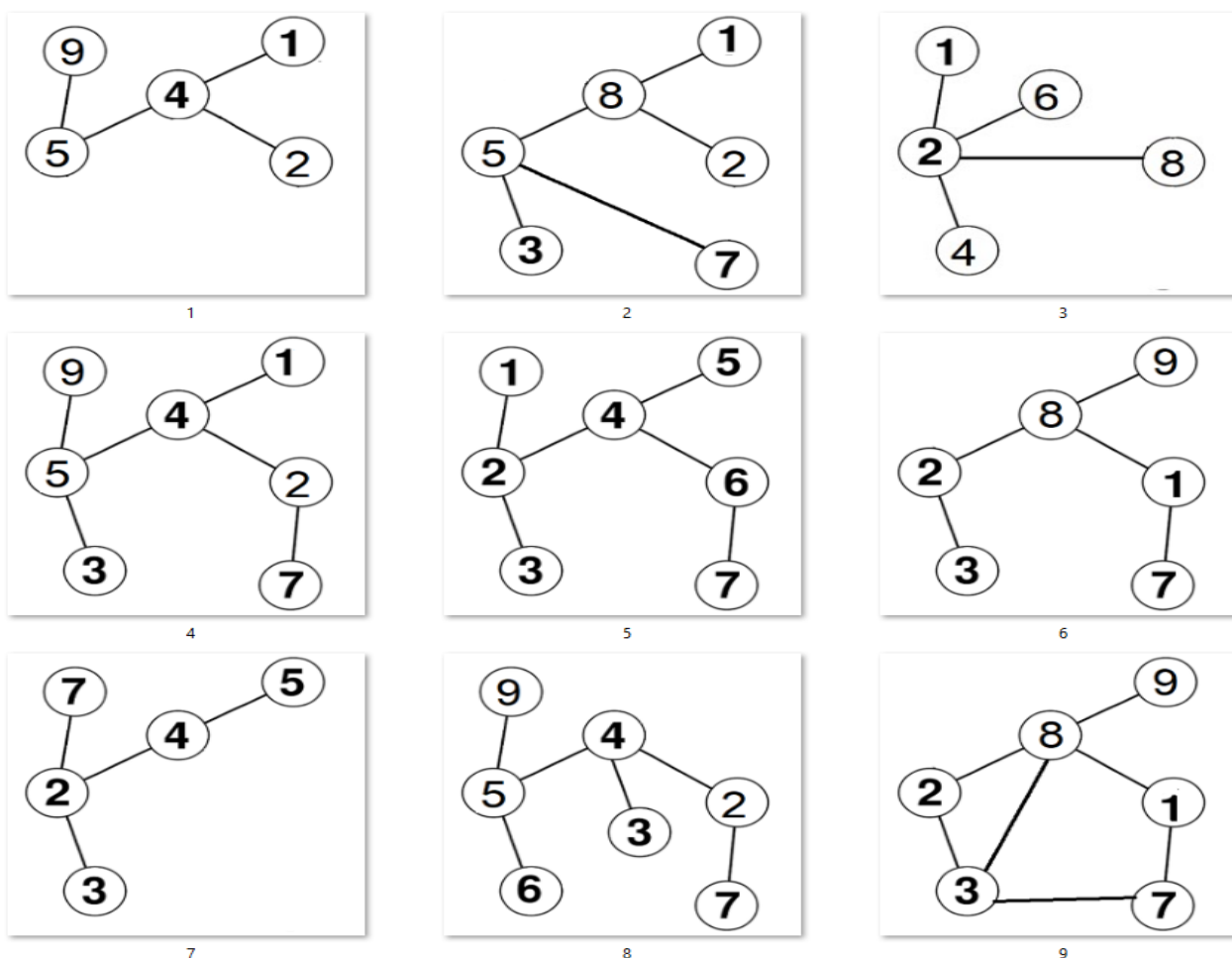
# Chapter 4

## Rezultate experimentale

### 4.1 Imagini

Imaginile pe care le-am folosit ca baza de date pentru test în acest proiect sunt imagini de dimensiuni egale care conțin diferite reprezentări ale unor grafuri conexe. Aceste grafuri conțin un diferit număr de noduri și muchii, unele conțin cicluri sau sunt simple lanțuri. Prin utilizarea acestor imagini am încercat să surprindem cât mai bine eficiența acestui proiect de digitalizare a grafurilor.

Baza de date a fost compusa din 10 imagini ale căror rezultate sunt corecte și verificate. Câteva exemple din cele 10 imagini găsiți în figura de mai jos:

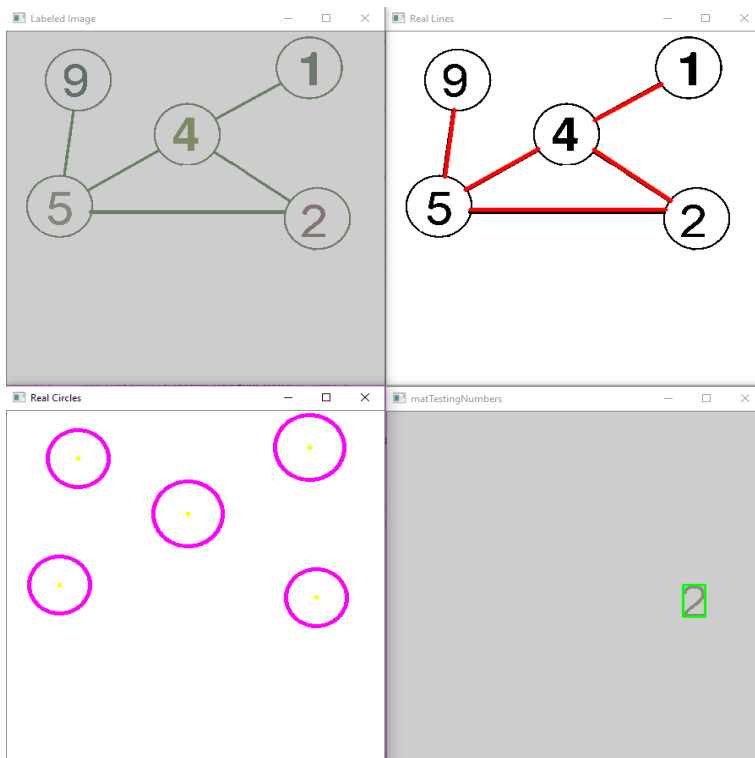


Pentru o buna funcționare a proiectului și obținerea rezultatelor așteptate, este nevoie de respectarea unor parametri ai imaginilor, și anume: imaginile de test sa aibă dimensiunea de 447px x 447px , iar pentru o buna recunoaștere a cifrelor, acestea trebuie sa aibă dimensiunea aproximativa a fontului Arial, dimensiune 45, fără a fi bold sau italic.

Soluția funcționează corect pe baza imaginilor de test puse la dispoziție. Cu toate acestea am observat ca poziția elementelor în imagine este importanta. Calculându-se densitatea de pixeli, algoritmul va fi sensibil în ceea ce privește detecția muchiilor. Estimativ, dintr-o imagine nou introdusa de utilizator, care sa respecte toate cerințele, rata de detecție a muchiilor va fi de 80%. Cifrele sunt identificate corect și nu exista confuzii între ele. Detecția cercurilor are rezultatele așteptate în proporție de 97%.

Rezultatele afișate vor conține rezultate intermediare sub forma imaginilor și rezultate finale sub forma de text afișate în consola

1. imaginea cu etichetarea fiecărui obiect în culori din aceeași gama
2. imaginea cu liniile detectate de către algoritm. Aceasta imagine va avea liniile detectate colorate cu rosu
3. imaginea cu cercurile detectate. Se va prezenta conturul lor împreuna cu centrul
4. imaginea care prezintă gradual concomitent cu afișarea în consola a detecției caracterelor din fiecare cerc.
5. matrice de adiacenta pe baza grafului ales
6. lista de adiacenta a nodurilor din graf



Rezultate intermediare

```
Adjacency Matrix:
0 0 0 0 0 0 0 0 0 0
0 0 0 0 1 0 0 0 0 0
0 0 0 0 1 1 0 0 0 0
0 0 0 0 0 0 0 0 0 0
0 1 1 0 0 1 0 0 0 0
0 0 1 0 1 0 0 0 0 1
0 0 0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0 0 0
0 0 0 0 0 1 0 0 0 0

Adjacency Matrix:
0 :
1 : -> 4
2 : -> 4-> 5
3 :
4 : -> 1-> 2-> 5
5 : -> 2-> 4-> 9
6 :
7 :
8 :
9 : -> 5
```

Rezultate finale

## 4.2 Evaluare numerica

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

## 4.3 Parametrii 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. Acesta parametri se refera de 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 și dezvoltări ulterioare

Proiectul realizat și-a atins scopul, acela de a digitaliza grafurile. Astfel, având la dispoziție o imagine cu un graf neorientat am reușit crearea unui algoritm pe baza căruia se poate digitaliza graful din imaginea inițială. Digitalizarea unei astfel de imagini înseamnă crearea și popularea unei structuri de date cu datele din imagine. Structura pe care am ales să o folosim este o listă de adiacență, aceasta fiind specificarea proiectului. Totuși, în crearea unei liste de adiacență, am creat mai întâi o matrice de adiacență. Dezavantajul matricii este faptul că trebuie alocată memoria în prealabil, pe când, în cazul listei se va alocă dinamic.

În ceea ce privește posibilele dezvoltări ulterioare acestea se pot materializa pe mai multe direcții:

- extinderea proiectului actual pentru a fi capabil să proceseze o gamă mai largă a informațiilor din noduri. În momentul de față, se face procesare doar pentru cifre din intervalul 1-9.
- realizarea posibilității de digitalizarea a grafurilor orientate.
- implementarea proiectului în vederea procesării imaginilor de dimensiuni diferite, fonturi diferite. Practic posibilitatea de a putea procesa orice tip de imagine.
- atașarea unui modul ce permite digitalizarea automata prin conectarea unei camere și obținerea rezultatului în 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](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](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](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](https://sci-hub.tw/10.1007/s001380050054?fbclid=IwAR0G_F-heHfJkyycsU-JDAzGHZbr_tDAYjsiXrU1EGh149mWTww0QhhXBaM)