Pintescu Dragoș-Mihai

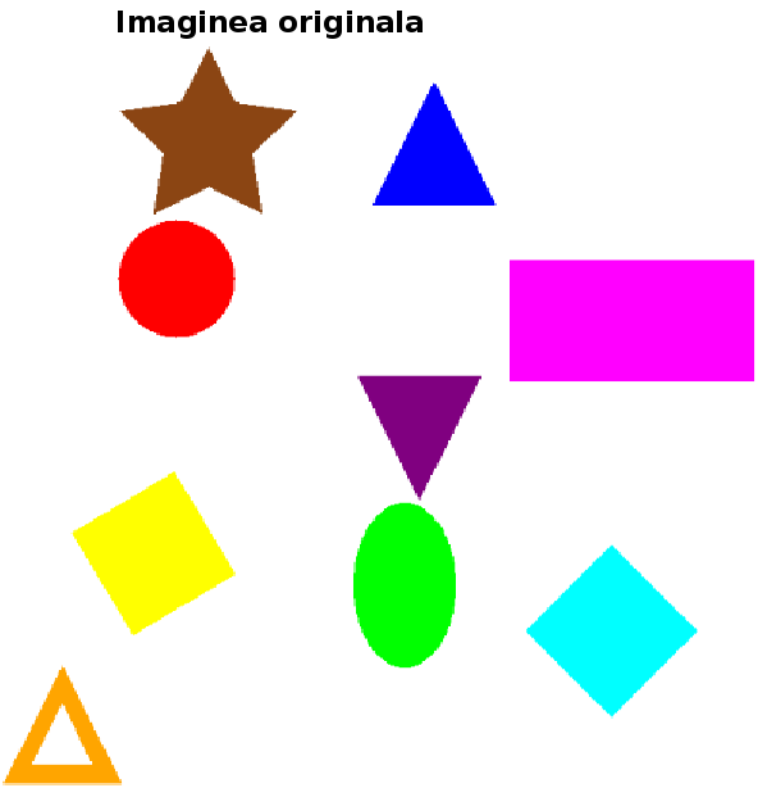
1306A

Tema 2 SVA

Achiziția și formarea imagininor

```
img = imread('1305B_1306A.png');
figure; imshow(img); title('Imaginea originala');
```

În această etapă programul încarcă imaginea sursă de pe disc folosind *imread*, care citește fișierul specificat şi construiește matricea imaginii. Funcția *imshow* apoi afișează imaginea pe ecran.



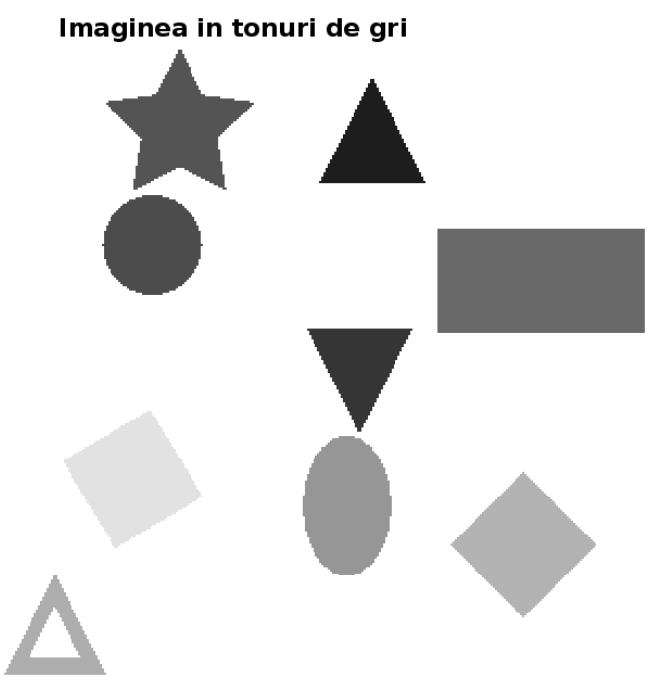
Preprocesarea imaginilor

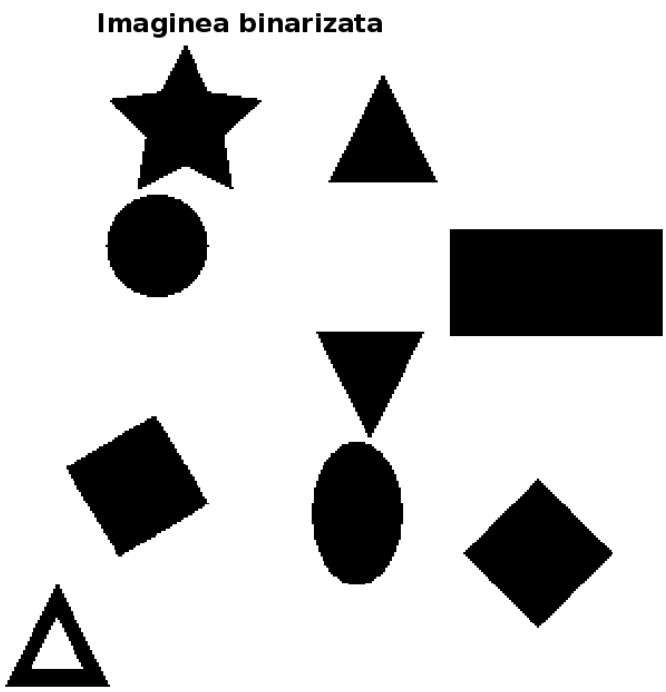
```
img_gray = rgb2gray(img);
figure; imshow(img_gray); title("Imaginea in tonuri de gri")

img_bw = imbinarize(img_gray, 0.89);
figure; imshow(img_bw); title("Imaginea binarizata");

se = strel('square', 5);
```

```
img_bw_op = imopen(img_bw, se);
figure; imshow(img_bw_op); title('Imaginea(imopen)')
img_bw_op = ~img_bw_op;
```

În această etapă imaginea originală color este transformată în tonuri de gri cu rgb2gray. Apoi, cu imbinarize(img_gray,0.89), imaginea gri este transformată în imagine binară pe baza pragului 0.89.. Prin această binarizare se separă obiectele de fundal, transformând imaginea într-o formă în care doar contrastul este relevant pentru segmentare.

Pasul următor aplică o operație morfologică de „deschidere” (imopen) cu un element structurant pătrat de dimensiune 5×5 (creat prin strel('square',5)). Operația de opening este o eroziune urmată de o dilatare folosind același element și servește la eliminarea zgomotului de dimensiuni mici păstrând formele mai mari. Rezultatul acestei operații este apoi inversat prin operatorul ~ pentru a obține imaginea binară finală utilizată în continuare.





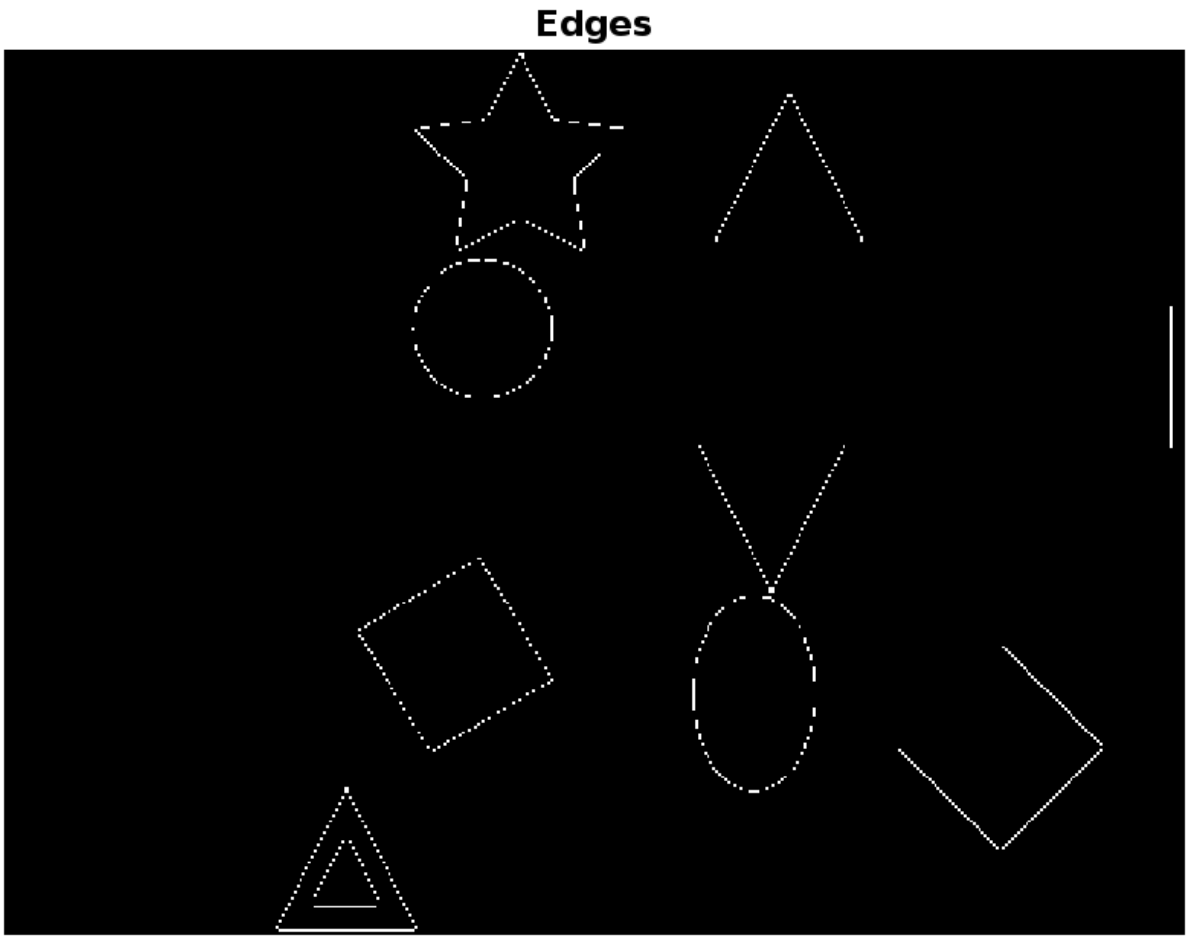
Segmentarea imaginilor

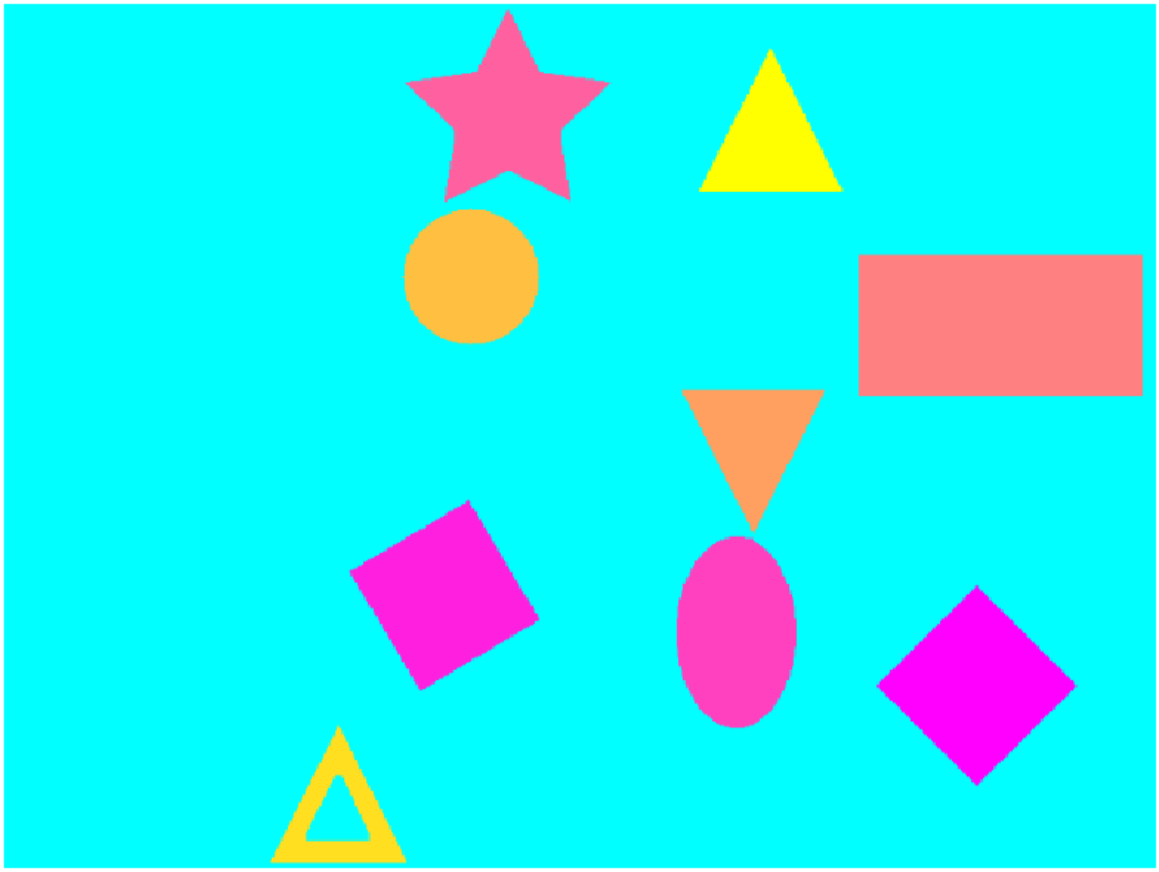
```
img_edges = edge(img_bw_op);
figure; imshow(img_edges); title("Edges");
```

```
 [label, num_obj] = bwlabel(img_bw_op, 4);
label_color = label2rgb(label, @spring, "c", "shuffle");
figure, imshow(label_color)
```

În etapa de segmentare se extrag componentele semnificative ale imaginii pentru a fi analizate separat. Funcția *edge(img_bw_op)* detectează marginile (contururile) obiectelor din imaginea binară de intrare.. Această detecție de margini scoate în evidență limitele obiectelor.

Următoarea linie folosește *bwlabel(img_bw_op,4),* care etichetează fiecare obiect conectat din imaginea binară. Cu conectivitatea 4, obiectele care au pixeli adiacenți pe lățime și înălțime (nu pe diagonală) sunt considerate conexe. Funcția *bwlabel* întoarce o matrice de etichetă în care fiecare obiect are un număr diferit şi pe baza căreia se numără obiectele (num_obj). Matricea label cu etichete poate fi apoi colorată cu *label2rgb* pentru vizualizare, dar scopul principal aici este identificarea și izolarea obiectelor (formelor) din imagine.





4. Descrierea

```
% region props
props = regionprops(logical(label), 'Area', 'Perimeter', 'Eccentricity', 'Extent',
'BoundingBox', 'Centroid', 'Solidity','Orientation');
img_annotated = img;
```

În această fază se extrag caracteristicile fiecărui obiect segmentat, folosind regionprops. Această funcție măsoară proprietăți ale regiuniilor conectate din imaginea binară etichetată. În cod, regionprops

calculează pentru fiecare obiect valori precum aria (Area), perimetrul (Perimeter), excentricitatea (Eccentricity), cât de mult umple obiectul chenarul său (Extent), caseta delimitatoare (BoundingBox), centrul de greutate (Centroid), cât de solid e obiectul (Solidity) şi orientarea acestuia (Orientation). Aceste caracteristici numerice permit descrierea formei obiectului în termeni de dimensiune, compactitate și formă generală. Ele vor fi folosite ulterior pentru recunoaşterea formei (ex: cerc, triunghi, pătrat etc.).

5. Recunoașterea formelor și interpretarea

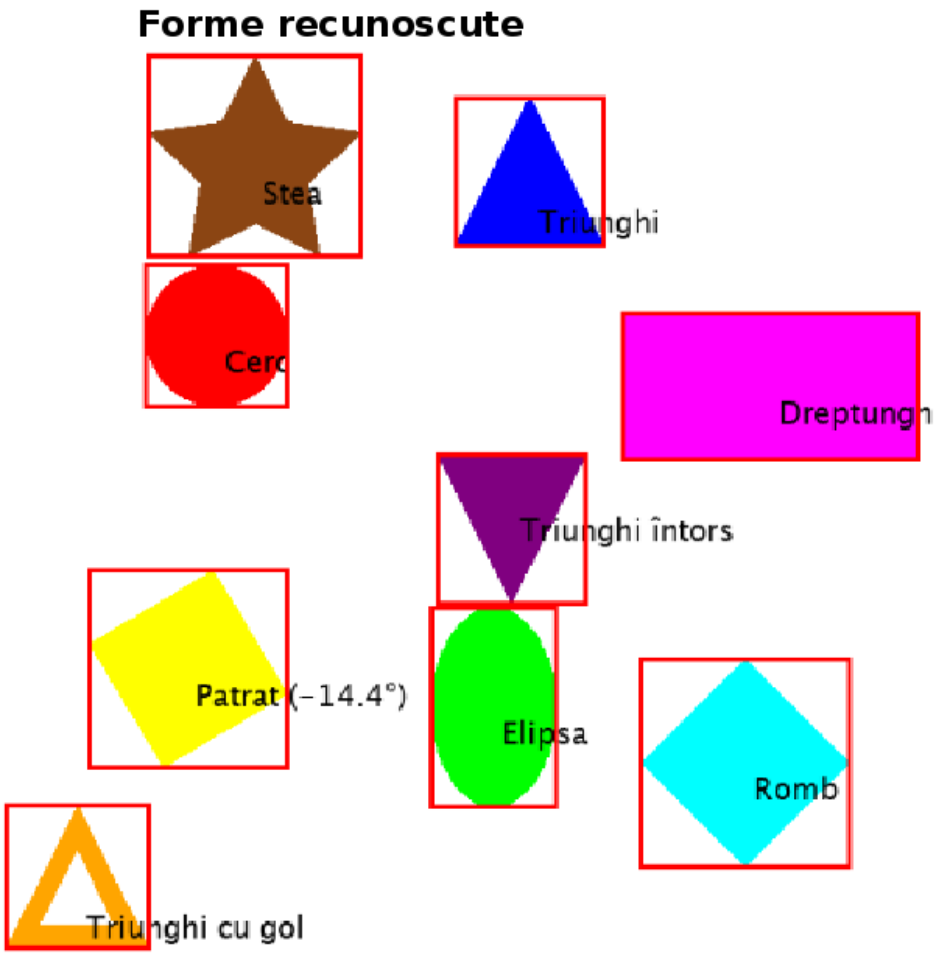
```
for i = 1:num_obj
    centroid = props(i).Centroid;
    area = props(i).Area;
    perimeter = props(i).Perimeter;
    raport = perimeter^2 / area;
    extent = props(i).Extent;
    ecc = props(i).Eccentricity;
    solidity = props(i).Solidity;
    bbox = props(i).BoundingBox;
    aspect_ratio = bbox(3) / bbox(4);
    orientation = props(i).Orientation;
    circular  = 4 * pi * area / (perimeter^2);

    object_img = imcrop(img_bw_op, bbox);
    corners = corner(object_img);
    num_corners = size(corners, 1);

    if ecc < 0.2 && circular > 0.85
        shape = 'Cerc';
    elseif raport > 19 && raport <=21
        if orientation < 0
            shape = 'Triunghi întors';
        else
            shape = 'Triunghi';
        end
    elseif extent > 0.95 && aspect_ratio > 1
        shape = 'Dreptunghi';
    elseif raport > 15 && raport <= 16
        if aspect_ratio >= 0.9 && aspect_ratio <= 1.1
            if orientation == 0
            shape = 'Romb';
            else
            shape = sprintf('Patrat (%.1f°)', orientation);
            end
        end
    elseif num_corners >= 15 && solidity < 0.8
        shape = 'Stea';
    elseif solidity < 0.9 && ecc < 0.8
        shape = 'Triunghi gol';
    elseif extent > 0.7 && extent < 0.95 && solidity > 0.85
        shape = 'Elipsa';
    end

    label_text = sprintf('%s' , shape);
```

```
    img_annotated = insertText(img_annotated, centroid, label_text, 'BoxColor',
'white', 'TextColor', 'black', 'BoxOpacity', 0, 'FontSize', 20);
    img_annotated = insertShape(img_annotated, 'Rectangle', bbox, 'Color', 'red',
'LineWidth', 3);
end

figure, imshow(img_annotated), title('Forme recunoscute');
```

În bucla principală se parcurg obiectele identificate pentru a le recunoaște forma. Se preiau proprietățile calculate: centrul de greutate, aria, perimetrul, excentricitatea, etc. Se calculează câțiva indicatori suplimentari precum raportul *raport = perimeter^2/area* și circularitatea *circular = 4π·area/perimeter^2*. Funcția imcrop obține imaginea fiecărui obiect (după BoundingBox), iar corner detectează colțurile acelui obiect. Numărul colțurilor ajută la identificarea unor forme complexe (de ex. o stea poate avea multe colțuri).

Regulile din secțiunea if... elseif definesc recunoașterea pe baza acestor caracteristici. De pildă, un obiect cu excentricitate foarte mică (<0.2) și circularitate apropiată de 1 este etichetat ca „Cerc”. Valorile prag cum sunt raport între 19 şi 21 sunt folosite pentru a detecta un triunghi, iar semnul orientării poate determina dacă e un „Triunghi întors”. Întinderea (Extent), raportul dintre laturi (aspect_ratio) și orientarea servesc la diferențierea dreptunghiului, pătratului și rombului. Numărul mare de colțuri și solitatea mică sunt condiții pentru a recunoaște o stea. Astfel, combinații de atribute numerice duc la atribuirea unei etichete textuale fiecărui obiect în shape.

În această ultimă etapă, rezultatul recunoașterii este prezentat sub forma unei interpretări vizuale. Cu insertText, codul scrie pe imagine numele formei recunoscute (coordonatele Centroid). Totodată, insertShape desenează dreptunghiuri roșii ('Rectangle') în jurul fiecărui obiect, folosind BoundingBox pentru poziție și dimensiune. Rezultatul final afișat (imshow(img_annotated)) este imaginea originală căreia îi sunt atașate etichete și contururi, arătând ce forme au fost identificate.


Forme recunoscute
Stea
Triunghi
Cerc
Dreptungh
Triunghi întors
Patrat (-14.4°)
Elipsa
Romb
Triunghi cu gol