

# Recunoașterea formelor

Tema 2

Sebastian Lăzărescu

Facultatea de Automatică și Calculatoare  
Universitatea Tehnică „Gheorghe Asachi” din Iași  
Mai 2025

# 1 Rezultate

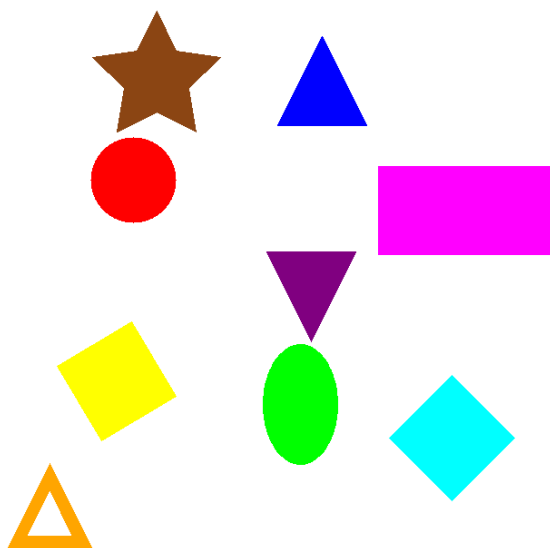


Figure 1: Imaginea pentru testarea algoritmului

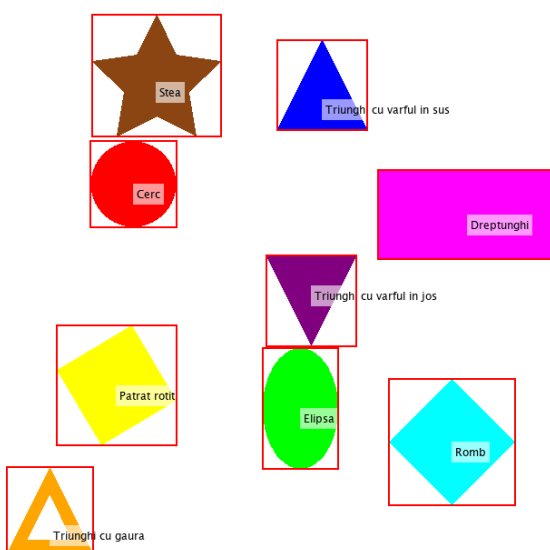


Figure 2: Imaginea finală

## 2 Pasul 1: Preprocesare imagine

La acest pas, imaginea de intrare este prelucrată pentru a fi adusă într-o formă adecvată procesării ulterioare. Etapele sînt următoarele:

**Etapa 1: *Conversia în tonuri de gri.*** Aici s-a folosit funcția MATLAB *rgb2gray*.

**Etapa 2: *Binarizarea imaginii.*** Se aplică binarizarea imaginii în tonuri de gri folosind un prag fix ( $T = 0.89$ ).

**Etapa 3: *Operații morfologice.*** Se definește un element structurant pătrat de 5x5 pixeli. Se aplică operația morfologică de deschidere (*imopen*), care elimină zgomotul de mici dimensiuni și separă obiectele apropiate. Imaginea rezultată este apoi inversată ( $\sim$ )

**Etapa 4: *Etichetarea obiectelor.*** Obiectele conectate din imaginea binară curățată sînt etichetate folosind conectivitatea de 4 pixeli. Fiecărui obiect  $i$  se atribuie o etichetă unică. Imaginea etichetată este apoi convertită într-o imagine color cu funcția *label2rgb*, pentru o vizualizare mai ușoară a obiectelor identificate.

## 3 Pasul 2: Detectarea formelor

Pentru detecția formelor m-am folosit de proprietățile fiecărui obiect detectat. Proprietățile au fost "achiziționate" cu ajutorul funcției MATLAB *regionprops*.

### **Detecția triunghiurilor:**

Din analiza proprietăților se observă că triunghiurile au proprietatea *Eccentricity* aproximativ  $0.5000$ , deci verificăm ca această proprietate să fie cuprinsă între  $0.4890$  și  $0.5050$ .

Pentru a detecta un triunghi cu gaură, ne folosim de proprietatea *Euler-Number*. Dacă aceasta este 0, rezultă că obiectul respectiv are o gaură în el.

Pentru a vedea dacă triunghiul este cu vârful în sus sau în jos, am scos toți

pixelii de pe obiectul detectat, am făcut 2 benzi (banda de sus și banda de jos) cu pixelii cei mai de sus și cei mai de jos, am calculat lungimea acestor benzi și am verificat care este mai mare. Dacă banda de sus  $<$  banda de jos, rezultă triunghi cu vârful în sus, iar dacă banda de sus  $>$  banda de jos, rezultă triunghi cu vârful în jos.

#### **Detecția cercurilor și a elipselor:**

Pentru detectarea cercurilor și a elipselor, m-am folosit de proprietatea *Circularity*. Aceasta arată cât de "cerc" este un obiect. Dacă *Circularity*  $> 0.9$ , sînt șanse foarte mari ca obiectul să fie cerc sau elipsă. Pentru a face distincția dintre cerc și elipsă, m-am folosit de *Eccentricity*. Dacă este 0, înseamnă că obiectul este sută la sută cerc. Dacă este diferit de 0, rezultă că este elipsă (în cazul nostru). *Eccentricity* ne spune cât de alungit este un cerc.

#### **Detecția pătratelor, pătratelor rotite și a romburilor:**

Verificăm ca *MajorAxis* și *MinorAxis* să fie egale și *Circularity*  $< 0.9$  (și cercurile au axele egale). Dacă avem *Orientation* diferit de 0, rezultă că figura este rotită. Dacă *Extent* (care arată cât din figură este încadrată în BoundingBox) este 1, rezultă pătrat, iar dacă *Extent* este 0.5, rezultă că avem un romb.

#### **Detecția dreptunghiurilor:**

Dacă *Extent* este 1 (adică sută la sută încadrat în BoundingBox) și axele nu sînt egale și *emphSolidity* este 1, rezultă că avem un dreptunghi. *Solidity* ne spune cât de "plină" este o figură în comparație cu *convex hull*.

#### **Detecția stelelor:**

Pentru a detecta stelele, m-am folosit de *Solidity*. Steaua are foarte multe coțuri deci nu "umple" foarte mult.